

INFORME MANTENIMIENTO Y DESARROLLO ESPECIALIZADO SISTEMA DE INFORMACIÓN DE GESTIÓN DE PROYECTOS – SIGP

TERCER PAGO CONTRATO 237-2024

TABLA DE CONTENIDO

INFORME MANTENIMIENTO Y DESARROLLO ESPECIALIZADO SISTEMA DE INFORMACIÓN DE GESTIÓN DE PROYECTOS - SIGP	1
1. MÓDULO SIGP VIEJO - ORACLE ADF, (No. Obligación Contrato: 2.2.1, 2.2.5, 2,2,6)	1
Pantalla Histórico de Proyectos y Programas generados en años anteriores a 2017 extrayendo la información del SIGP Antiguo	3
Pantalla Histórico de Proyectos de Beneficios Tributarios generados en años anteriores a 2017 extrayendo la información del SIGP Antiguo	158
2. INFORME DE ACTIVIDADES DE SOPORTE EN EL USO DEL SIGP EN LA CREACION DE CONVOCATORIAS Y TRANSFERIR EL CONOCIMIENTO DEL SIGP A LOS FUNCIONARIOS DE MINCIENCIAS, (No. Obligación Contrato: 2.2.3, 2.2.4)	337
Introducción	337
Resumen de Actividades	338
Detalle de Casos Atendidos.....	338
Parametrización de Convocatorias	338
Parametrización de Rúbricas de Evaluación	357
Generación de Reportes.....	361
Solución de Errores	413
Actualización de Datos	416
Requerimientos.....	425
Conclusiones.....	427

INFORME MANTENIMIENTO Y DESARROLLO ESPECIALIZADO SISTEMA DE INFORMACIÓN DE GESTIÓN DE PROYECTOS - SIGP

1. MÓDULO SIGP VIEJO - ORACLE ADF, (No. Obligación Contrato: 2.2.1, 2.2.5, 2,2,6)

Solicitud de requerimiento No. 157812

Descripción y alcance del Requerimiento

Se requiere la creación de un nuevo menú de consulta dentro de la plataforma actual, que permita visualizar la información histórica de proyectos y beneficios tributarios registrados antes del año 2017, cuya fuente de datos corresponde al SIGP antiguo.

El objetivo es ofrecer una herramienta de consulta accesible y organizada que permita recuperar, visualizar y descargar información clave de estos registros históricos.

La funcionalidad para desarrollar debe permitir:

1. Consulta de información general del proyecto o beneficio tributario, incluyendo los siguientes campos:
 - Convocatoria
 - Programa Nacional de CTel
 - Estado
 - Código de Registro
 - Código del Proyecto
 - Tipo (Proyecto / Beneficio Tributario)
 - Entidad o Persona Beneficiaria
 - Tipo de Financiación
 - Duración
 - Ejecución del Cronograma
 - Lugar de Ejecución
 - Fecha de Creación
 - Fecha de Radicación

- Fecha de Revisión
- Fecha de Última Modificación
- 2. Visualización de pestañas complementarias:
 - Pestaña de Requisitos del proyecto
- 3. Opciones de descarga:
 - Ficha técnica del proyecto
 - Reporte de calificación
 - Reporte de evaluación, incluyendo los datos de las personas que evaluaron el proyecto
- 4. Fuente de datos:
 - Toda la información debe extraerse desde las bases de datos del SIGP antiguo, asegurando integridad y trazabilidad de los datos históricos.

Desarrollo

Para el desarrollo de este requerimiento, se realizaron los siguientes ajustes:

- Creación de un menú que permita la consulta de proyectos y programas generados en años anteriores a 2017 extrayendo la información del SIGP Antiguo
 - Implementar funcionalidad de ver detalles de los proyectos y programas
 - Implementar funcionalidad de descarga de la ficha de los proyectos antiguos
 - Implementar funcionalidad de ver requisitos de los proyectos y programas, tabla con requisitos, opciones de descargar de documentos.
 - Implementar funcionalidad de ver la evaluación de los proyectos y programas. Descarga del reporte de evaluación
- Creación de un menú que permita la consulta de proyectos de beneficios tributarios generados en años anteriores a 2017 extrayendo la información del SIGP Antiguo.
 - Crear pantalla con tabla de visualización de registros de proyectos antiguos. Búsqueda global y avanzada de los proyectos de beneficios tributarios
 - Implementar funcionalidad de ver detalles de los proyectos de beneficios tributarios
 - Implementar funcionalidad de descarga de la ficha de los proyectos antiguos

- Implementar funcionalidad de ver requisitos de los proyectos de beneficios tributarios, tabla con requisitos, opciones de descargar de documentos.
- Implementar funcionalidad de ver la evaluación de los proyectos de beneficios tributarios. Descarga del reporte de evaluación

Pantalla Histórico de Proyectos y Programas generados en años anteriores a 2017 extrayendo la información del SIGP Antiguo

Función:

La pantalla tiene como función principal facilitar la consulta, visualización y descarga de información histórica relacionada con los proyectos y programas generados antes del año 2017, cuya fuente de datos corresponde al sistema SIGP antiguo.

Esta funcionalidad permitirá a los usuarios acceder de manera estructurada y centralizada a registros previos a la migración del sistema actual, conservando el detalle técnico y documental de cada proyecto o beneficio tributario.

Funcionalidades clave:

- Consulta detallada de proyectos/programas antiguos mediante filtros y listado.
- Visualización de información general del proyecto: convocatoria, estado, tipo, entidad, fechas clave, lugar de ejecución, entre otros campos relevantes.
- Acceso a pestañas relacionadas, como:
 - Requisitos del proyecto
 - Detalles del proyecto
 - Descarga de documentos asociados, tales como:
 - Ficha técnica del proyecto
 - Reporte de calificación
 - Reporte de evaluación con información de los evaluadores

Con esta pantalla, se asegura la preservación, trazabilidad y disponibilidad de la información histórica de CTel, facilitando procesos de consulta, auditoría y análisis institucional.

Creación nuevo menú de Proyectos Antiguos

```
-- funciones (menús)
INSERT INTO HERCULES.SB_FUNCIONES (funcion_id, aplicacion, funcion, nombre_funcion,
tipo_funcion, menu, opcion, url, url_ayuda, funcion_padre, icono, usuario_creador,
fecha_creacion, usuario_modificador, fecha_modificacion)
VALUES (null, 'SIGP_WEB', 'GEST5400', 'Proyectos Antiguos', 'F', null, null,
'/app/proyectos/proyectosAntiguos.xhtml', null, 'GEST5000', null, 'SIGP_WEB',
to_date('25-09-2023 08:06:40', 'dd-mm-yyyy hh24:mi:ss'), null, null);
```



```
INSERT INTO HERCULES.SB_FUNCIONES (funcion_id, aplicacion, funcion, nombre_funcion,
tipo_funcion, menu, opcion, url, url_ayuda, funcion_padre, icono, usuario_creador,
fecha_creacion, usuario_modificador, fecha_modificacion)
VALUES (null, 'SIGP_WEB', 'GEST5410', 'Requisitos de Proyectos Antiguos', 'F', null,
null, '/app/proyectos/requisitosAntiguos.xhtml', null, 'GEST5400', null, 'SIGP_WEB',
to_date('25-09-2023 08:06:40', 'dd-mm-yyyy hh24:mi:ss'), null, null);

--Derechos funciones
INSERT INTO HERCULES.SB_DERECHOS_FUNCIONES (derecho_funcion_id, aplicacion, funcion,
grupo, estado, usuario_creador, fecha_creacion, usuario_modificador, fecha_modificacion)
values (null, 'SIGP_WEB', 'GEST5400', 'SGPS_ADMINISTRADOR', 'H', 'SIGP_WEB',
to_date('22-02-2023 07:25:36', 'dd-mm-yyyy hh24:mi:ss'), null, null);

INSERT INTO HERCULES.SB_DERECHOS_FUNCIONES (derecho_funcion_id, aplicacion, funcion,
grupo, estado, usuario_creador, fecha_creacion, usuario_modificador, fecha_modificacion)
values (null, 'SIGP_WEB', 'GEST5410', 'SGPS_ADMINISTRADOR', 'H', 'SIGP_WEB',
to_date('22-02-2023 07:25:36', 'dd-mm-yyyy hh24:mi:ss'), null, null);
```

*Crear nuevo campo de entidad antigua en la tabla **SGP_PROYECTOS** en la base de datos sigp*

```
@JoinColumn(name = "ENTIDAD_ID", referencedColumnName = "ENTIDAD_ID")
@ManyToOne
private EntidadAntigua entidadAntiguaId;
```

Este campo establece una relación **ManyToOne** con la entidad EntidadAntigua, permitiendo acceder a los datos históricos de la entidad que presentó el proyecto en convocatorias pasadas.

También se incluyeron sus respectivos **getters y setters**:

```
public EntidadAntigua getEntidadAntiguaId() {
    return entidadAntiguaId;
}

public void setEntidadAntiguaId(EntidadAntigua entidadAntiguaId) {
    this.entidadAntiguaId = entidadAntiguaId;
}
```

Nota: Se usa una clase distinta (EntidadAntigua) para evitar conflictos con la entidad actual (Entidad) y separar la lógica de datos históricos.

Actualización del DTO: ProyectoDTO.java

Para reflejar esta información en el frontend (JSF), se adicionó también el campo correspondiente en el DTO:

```
@Bind(name = "entidadAntiguaId", returnType = EntidadAntiguaDTO.class, type =
EntidadAntigua.class)
private EntidadAntiguaDTO entidadAntiguaId;
```

Con sus métodos de acceso:

```
public EntidadAntiguaDTO getEntidadAntiguaId() {  
    return entidadAntiguaId;  
}  
  
public void setEntidadAntiguaId(EntidadAntiguaDTO entidadAntiguaId) {  
    this.entidadAntiguaId = entidadAntiguaId;  
}
```

- Adición del menú **“Proyectos Antiguos”** en la plantilla del menú de Proyectos/Programas – Sistema del aplicativo de gestión

El siguiente código define un botón en el menú de navegación de la aplicación que permite acceder a la pantalla de proyectos Antiguos del sistema.

```
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST5400')}">  
    <li class="#{selectedPage eq 'proyectosAntiguos' ? 'active':''}">  
        <h:commandLink  
            actionListener="#{proyectoAntiguoViewController.restablecerLista()}" immediate="true">  
            <i class="ui-link ui-widget" style="font-style:  
normal;">#{msg['sb.menu.gestion.proyectosProgramas.proyectosAntiguos']}</i>  
        </h:commandLink>  
    </li>  
</ui:fragment>
```

El acceso al módulo de histórico de proyectos fue agregado dentro del **template principal** del sistema, en el bloque de opciones de menú, utilizando JSF (<ui:fragment>, h:commandLink) y control de seguridad a través del SeguridadSessController.

Código del Menú

```
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST5400')}">  
    <li class="#{selectedPage eq 'proyectosAntiguos' ? 'active':''}">  
        <h:commandLink actionListener="#{proyectoAntiguoViewController.restablecerLista()}"  
            immediate="true">  
            <i class="ui-link ui-widget" style="font-style: normal;">  
                #{msg['sb.menu.gestion.proyectosProgramas.proyectosAntiguos']}  
            </i>  
        </h:commandLink>  
    </li>  
</ui:fragment>
```

Explicación Técnica

- **Control de acceso:** El ítem de menú solo se renderiza si el usuario tiene el permiso 'GEST5400', verificado a través de SeguridadSessController.mapaMenu.

- **Selección activa:** Se aplica la clase CSS active si la página actual es proyectosAntiguos, permitiendo resaltar visualmente el ítem seleccionado en el menú.
- **Acción de navegación:** El menú ejecuta el método restablecerLista() del controlador proyectoAntiguoViewController para inicializar o refrescar la lista de proyectos antiguos.
- **Internacionalización:** El nombre del ítem se obtiene desde un archivo de mensajes ({msg[...]}), permitiendo compatibilidad multilingüaje.

Controlador Referenciado

El método llamado al hacer clic:

```
public void restablecerLista() {  
    buscarVisible = true;  
    verProyecto = false;  
    init();  
    this.redirectPage("app/proyectos/proyectosAntiguos.xhtml");  
}
```

Este método se asegura de cargar la información actualizada de los proyectos antiguos antes de mostrar la vista asociada.

Observaciones

- Es importante usar immediate="true" en el h:commandLink para evitar validaciones JSF innecesarias antes de la carga.
- Esta opción de menú da acceso a la vista proyectosAntiguos.xhtml, ubicada en el módulo gestion/proyectosProgramas.

Controlador de Vista ProyectoAntiguoViewController

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package com.sb.colciencias.gestion.web.controller.proyectos.view;  
  
import com.sb.colciencias.gestion.web.controller.proyectos.session.*;  
import com.sb.colciencias.gestion.business.domain.scienti.ScientiPersonaFormacion;  
import com.sb.colciencias.gestion.business.facade.configuracion.ConfiguracionServiceFacade;  
import com.sb.colciencias.gestion.business.facade.proyectos.ProyectosServiceFacade;
```

```
import com.sb.colciencias.gestion.business.transfer.ConvocatoriaCampoDTO;
import com.sb.colciencias.gestion.business.transfer.ConvocatoriaDTO;
import com.sb.colciencias.gestion.business.transfer.ConvocatoriaMenuDTO;
import com.sb.colciencias.gestion.business.transfer.DescripcionDTO;
import com.sb.colciencias.gestion.business.transfer.DocumentoDTO;
import com.sb.colciencias.gestion.business.transfer.FocoDTO;
import com.sb.colciencias.gestion.business.transfer.MarcoObjGeneralDTO;
import com.sb.colciencias.gestion.business.transfer.ObservacionProyectoDTO;
import com.sb.colciencias.gestion.business.transfer.OtroResultadoDTO;
import com.sb.colciencias.gestion.business.transfer.PermisoDTO;
import com.sb.colciencias.gestion.business.transfer.PersonaProgramaDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoEntidadDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoEvaluacionDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoPersonaDTO;
import com.sb.colciencias.gestion.business.transfer.RegionImpactoDTO;
import com.sb.colciencias.gestion.business.transfer.RubroDTO;
import com.sb.colciencias.gestion.business.transfer.TipoPersonalDTO;
import com.sb.colciencias.gestion.business.transfer.TramiteEstimuloDTO;
import com.sb.colciencias.gestion.business.transfer.UsuarioDTO;
import com.sb.colciencias.gestion.business.transfer.ValorDefinicionDTO;
import com.sb.colciencias.gestion.business.valueObjects.DetalleProyectoVO;
import com.sb.colciencias.gestion.business.valueObjects.ValoresRubroVO;
import com.sb.colciencias.gestion.web.controller.application.ConfigApplController;
import com.sb.colciencias.gestion.web.util.security.SeguridadConvController;
import com.sb.standard.web.controller.session.GenericSessionController;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collection;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.annotation.PostConstruct;
import javax.ejb.EJB;
import javax.enterprise.context.SessionScoped;
import javax.faces.event.ValueChangeEvent;
import javax.inject.Inject;
import javax.inject.Named;
import org.primefaces.PrimeFaces;
import org.primefaces.model.LazyDataModel;
import org.primefaces.model.SortOrder;
import org.primefaces.model.FilterMeta;
import java.util.TreeMap;
import org.primefaces.model.StreamedContent;

/**
 *
 * @author lsuarez
 */
@Named(value = "proyectoAntiguoViewController")
@SessionScoped
public class ProyectoAntiguoViewController extends GenericSessionController<ProyectoDTO>
implements Serializable {

    @EJB
    private ProyectosServiceFacade proyectosServiceFacade;
    @EJB
```

```
private ConfiguracionServiceFacade configuracionServiceFacade;
@Inject
private EvaluarProyectoSessController evaluarProyectoSessController;
@Inject
private SeguridadConvController seguridadConvController;
@Inject
private ConfigApplController configApplController;
//
private LazyDataModel<ProyectoDTO> model;
private ValoresRubroVO valoresRubroVO;
private Map<String, ValorDefinicionDTO> listaEstados;
private Map<String, ValorDefinicionDTO> listaTramites;
private Map<String, ValorDefinicionDTO> listaTramitesExencion;
private Map<String, ValorDefinicionDTO> listaTramitesDeducccion;
private List<ObservacionProyectoDTO> listaObservacionProyecto;
private String urlReportes;
private List<ConvocatoriaDTO> listaConvocatorias;
private ProyectoDTO proyectoFilter;
private ValorDefinicionDTO tipoTramite;
private DetalleProyectoVO detalleProyectoVO;
private List<Object[]> listRubros;
private List<RubroDTO> rubrosDetalleProyectoList;
private String descripcionRubro;
private MarcoObjGeneralDTO marcoObjGeneralDTO;
private List<PermisoDTO> listaPermisos;
private Map<String, ValorDefinicionDTO> listaPermisosContrato;
private ProyectoDTO programaSelected;

///
private List<ProyectoDTO> listaProyectoTramites;
private boolean tramitesSecundarios;
private boolean buscarVisible;
private String tipoExencion;

private String nombreDependenciaConvocatoria;
private String nombreFichaTecnica;
private ValorDefinicionDTO componente;
private List<DescripcionDTO> listaDescripciones;
private DescripcionDTO descripcion;
private RubroDTO rubroSelected;
private List<OtroResultadoDTO> listResultadoDesarrolloTec;
private List<OtroResultadoDTO> listResultadoGeneracion;
private List<RegionImpactoDTO> listCoberturas;
private List<TipoPersonalDTO> listaTipoPersonal;
private List<PersonaProgramaDTO> listaPersonaPrograma;
private HashMap<String, HashMap<String, String>> mapaCampos;
private Map<String, ConvocatoriaMenuDTO> listaMenu;
private ProyectoPersonaDTO proyectoPersonaDTO;
private PersonaProgramaDTO personaProgramaDTO;
private List<UsuarioDTO> listaUsuarios;
private List<ValorDefinicionDTO> listRolPersonas;
private String descRolPersona;
private boolean proyectoAntiguo;
private boolean verProyecto;
private String parameterUrlCvLAC;
private List<ScientiPersonaFormacion> listaScientiPersonaFormacions;
private List<ProyectoEvaluacionDTO> listaEvaluaciones;
private List<DocumentoDTO> listaDocumentos;
private List<ProyectoEntidadDTO> listaProyectoEntidades;
private ProyectoEntidadDTO proyectoEntidadSelected;
```

```
// Evaluacion
private Integer countEvaluaciones;
private ProyectoEvaluacionDTO proyectoEvaluacionSelected;
private DocumentoDTO documentoEvaluacion;
private StreamedContent archivoEvaluacion;
private ValorDefinicionDTO documentoEvaluacionTipo;

//Documentos
private ValorDefinicionDTO documentoTipoSgpProyectos;
private ValorDefinicionDTO documentoTipoSgpProyectosXml;
private ValorDefinicionDTO documentoTipoSgpRegistroProyectos;
private StreamedContent archivoDocumento;

// Documento Hoja de Vida Personal
private ValorDefinicionDTO tipoDocHojaVida;
private StreamedContent archivoHojaVida;

private List<FocoDTO> listaFocos;
private List<TramiteEstimuloDTO> listaTramiteEstimulos;
private List<Object[]> listProyectosPrograma;
//
private ArrayList<Integer> listAnios;
// Totales
private double totalMonInstitucionSol;
private double totalConEspecieSol;
private double totalConEfectivoSol;
private double totalSol;

private static final org.apache.log4j.Logger LOG =
org.apache.log4j.Logger.getLogger(ProyectoAntiguoViewController.class);

private boolean filtroActivo;

public ProyectoAntiguoViewController() {
    super(ProyectoDTO.class);
    this.proyectoFilter = new ProyectoDTO();
    listaMenu = new TreeMap<>();
}

@PostConstruct
public void init() {

    this.model = new LazyDataModel<ProyectoDTO>() {

        @Override
        public List<ProyectoDTO> load(int first, int pageSize, String sortField,
SortOrder sortOrder, Map<String, FilterMeta> filterBy) {
            Map<String, Object> filters = new TreeMap<>();
            filterBy.forEach((k, v) -> {
                if (k.equals("globalFilter")) {
                    if (v.getFilterValue() != null &&
!v.getFilterValue().toString().isEmpty()) {
                        filters.put(k, v.getFilterValue());
                    }
                }
            });
            List<ProyectoDTO> proyectoList = new ArrayList<>();
            try {
                //filtro por busqueda avanzada
                if (proyectoFilter != null) {
                    filters.put("proyectoFiltro", proyectoFilter);
                }
            } catch (Exception e) {
                LOG.error(e.getMessage());
            }
            return proyectoList;
        }
    };
}
```

```

        }
        filters.put("proyectosAntiguos", true); //filtro por proyectos antiguos
        filters.put("claveNull", true); //filtro por clave nula (ya que los
proyectos antiguos no la registraban)
        int count = proyectosServiceFacade.proyectosListCount(filters);
        if (count > 0) {
            proyectoList = proyectosServiceFacade.proyectosList(first, pageSize,
sortField, sortOrder.equals(SortOrder.ASCENDING) ? "A" : "D", filters);
        }
        model.setRowCount(count);
    } catch (Exception ex) {
        LOG.error("Error al cargar el listado de proyectos antiguos ", ex);
    }
    return proyectoList;
}
}

@Inject
public void cargarListas() {
    try {
        listaEstados = configApplController.getMapValorDefinicion().get("ESTADO");
        // System.out.println("listaEstados SIZE: "+ listaEstados.size());
        listaConvocatorias =
proyectosServiceFacade.obtenerConvocatoriasProyectosAntiguos();
        // System.out.println("listaConvocatorias Antiguas SIZE: " +
listaConvocatorias.size());
        listaPermisosContrato =
configApplController.getMapValorDefinicion().get("CONTPERM");
        urlReportes =
proyectosServiceFacade.obtenerParametroSistema("URLREPORTE").getCadena();
        parameterUrlCvLAC =
proyectosServiceFacade.obtenerParametroSistema("URLCVLAC2").getCadena();
        documentoEvaluacionTipo =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("PRY_EVALUACION");
        documentoTipoSgpProyectos =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_PROYECTOS");
        documentoTipoSgpProyectosXml =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_PROYECTOS_XML");
        documentoTipoSgpRegistroProyectos =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_REGISTRO_PROYECTOS");

        // cargamos el listado de años de los proyectos
        Date fechaFinProy =
proyectosServiceFacade.obtenerParametroSistema("PRYFECHAIN").getFecha();
        Calendar fechaFinal = Calendar.getInstance();
        fechaFinal.setTime(fechaFinProy);
        int anioInicio = 2000;
        //
        int añoFinal = fechaFinal.get(Calendar.YEAR);
        listAnios = new ArrayList<>();
        for (int i = anioInicio; i <= añoFinal; i++) {
            listAnios.add(i);
        }

    } catch (Exception ex) {

}

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

```

    public void setTipoProyecto(ValueChangeEvent event) {
        String dato = String.valueOf(event.getNewValue());
        switch (dato) {
            case "F":
                this.proyectoFilter.setAnteproyecto("S");
                this.proyectoFilter.setExencion("R");
                break;
            case "R":
                this.proyectoFilter.setExencion("R");
                this.proyectoFilter.setAnteproyecto("N");
                break;
            case "P":
                this.proyectoFilter.setExencion("P");
                this.proyectoFilter.setAnteproyecto(null);
                break;
            default:
                break;
        }
    }

    public void cargarDetallesProyecto(ProyectoDTO proyectoDTO) {
        try {
            this.selected = proyectoDTO;

            this.selected.setFormularioProyectoId(projectosServiceFacade.obtenerFormularioProyectoId(selected.getProjectoId()));
            valoresRubroVO =
            proyectosServiceFacade.montosRubroProyecto(proyectoDTO.getProjectoId(), null);
            listaObservacionProyecto =
            proyectosServiceFacade.obtenerObservaciones(proyectoDTO.getProjectoId());

            selected.setPalabraClaveList(projectosServiceFacade.obtenerPalabrasClave(selected.getProjectoId()));
            nombreFichaTecnica =
            proyectosServiceFacade.obtenerFichaTecnica(selected.getAnteproyectoId());
            cargarCamposConvocatoria();
            cargarPersonasProyecto();
            cargarListaMenusConvocatoria();
            cargarEntidadesProyecto();
            // cargarPresupuestoProyecto();
            Map<String, Object> filters = new HashMap<>();
            filters.put("proyectoId", this.selected.getProjectoId());
            countEvaluaciones =
            proyectosServiceFacade.proyectoEvaluacionesListCount(filters);
        } catch (Exception ex) {

            Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
        }

        public void cargarInformacionBasicaProyecto() {
            try {

                selected.setPermisosList(projectosServiceFacade.obtenerPermisosProyecto(selected.getProjectoId()));

                selected.setComponenteList(projectosServiceFacade.obtenerComponentes(selected.getProjectoId()));
            }
        }
    }

```



```
selected.setSectoresTecnologicos(proyectosServiceFacade.obtenerSectoresTecnologicos(selected
.getProyectoId()));

selected.setClasesInvencionList(proyectosServiceFacade.obtenerClasesInvencion(selected.getPr
oyectoId()));

selected.setSublineaSecundarialist(proyectosServiceFacade.obtenerSublineasSecundarias(select
ed.getProyectoId()));

selected.setRolEntidadList(proyectosServiceFacade.obtenerRolesEntidad(selected.getProyectoId
()));

selected.setFocosTematicosList(proyectosServiceFacade.obtenerFocosTematicos(selected.getProy
ectoId()));
    nombreDependenciaConvocatoria =
proyectosServiceFacade.obtenerDependenciaConvocatoria(selected.getConvocatoriaId().getConvoc
atoriaId());
    componente =
proyectosServiceFacade.obtenerCategoria(selected.getProyectoId()).getTipoComponenteId();

selected.setTematicasOrientadorasList(proyectosServiceFacade.obtenerTematicasOrientadorasFoc
os(selected.getProyectoId()));
    listaFocos =
proyectosServiceFacade.obtenerFocosProyecto(selected.getProyectoId());
    cargarTramitesEstimulos();
    if (selected.getExencion().equals("R")) {
        cargarProyectosAsociados();
    }
} catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
}

    public void cargarEntidadesProyecto() {
        try {
            listaProyectoEntidades = new ArrayList<>();
            listaProyectoEntidades =
proyectosServiceFacade.obtenerProyectoEntidades(selected.getProyectoId());
        } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
        }

        public void cargarRolEntidadesProyecto() {
            try {

selected.setRolEntidadList(proyectosServiceFacade.obtenerRolesEntidad(selected.getProyectoId
()));
            } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
        }

        public void cargarPresupuestoProyecto() {
            try {
                listRubros = new ArrayList<>();
```

```

        listRubros =
projectosServiceFacade.obtenerRubrosProyAntiguos(selected.getProjectoId());
        calcularTotales();
    } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
    }

    // Método para calcular los totales
    public void calcularTotales() {
        totalMonInstitucionSol = 0;
        totalConEspecieSol = 0;
        totalConEfectivoSol = 0;
        totalSol = 0;

        if (listRubros != null && !listRubros.isEmpty()) {
            for (Object[] rubro : listRubros) {
                totalMonInstitucionSol += ((Number) rubro[1]).doubleValue();
                totalConEspecieSol += ((Number) rubro[2]).doubleValue();
                totalConEfectivoSol += ((Number) rubro[3]).doubleValue();
                totalSol += ((Number) rubro[4]).doubleValue();
            }
        }
    }

    public void cargarResultadosProyecto() {
        try {
            detalleProyectoVO =
projectosServiceFacade.cargarResultadosProyecto(selected.getProjectoId());
            listResultadoGeneracion =
projectosServiceFacade.obtenerOtrosResultados(selected.getProjectoId(), "G");
            listResultadoDesarrolloTec =
projectosServiceFacade.obtenerOtrosResultados(selected.getProjectoId(), "D");
        } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
        }

        public void cargarImpactosProyecto() {
            try {
                detalleProyectoVO =
projectosServiceFacade.obtenerImpactosProyecto(selected.getProjectoId());
            } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
            }

            public void cargarCoberturasProyecto() {
                try {
                    listCoberturas =
projectosServiceFacade.obtenerCoberturas(selected.getProjectoId());
                } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
                }
            }

            public void cargarMarcoLogico() {

```

```

        try {
            detalleProyectoVO =
projectosServiceFacade.cargarMarcoLogicoProyecto(selected.getProjectoId());
            marcoObjGeneralDTO = detalleProyectoVO.getListaObjetivosGeneral() != null &&
!detalleProyectoVO.getListaObjetivosGeneral().isEmpty() ?
detalleProyectoVO.getListaObjetivosGeneral().get(0) : null;
        } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
        }

        public void cargarTipoPersonal() {
            try {
                listaTipoPersonal =
projectosServiceFacade.obtenerTipoPersonal(selected.getProjectoId());

            } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
            }

        }

        public void cargarPersonasProyecto() {
            try {
                //Tipo Documento para Hoja de Vida
                tipoDocHojaVida =
configApplController.getMapValorDefinicion().get("TIPDOCPRY").get("Documento Hoja de Vida
Personal No CvLAC");

                HashMap<String, Object> parametros = new HashMap<>();
                parametros.put("fechaCreacionProyecto", selected.getFechaCreacion());
                parametros.put("tipoProyecto", selected.getExencion());
                ConvocatoriaDTO convocatoriaDTO = selected.getConvocatoriaId();
                String convocatoriaPrograma = convocatoriaDTO.getPrograma() != null ?
convocatoriaDTO.getPrograma() : "N";
                parametros.put("convocatoriaPrograma", convocatoriaPrograma);
                parametros.put("proyectoId", selected.getProjectoId());
                proyectoAntiguo = proyectosServiceFacade.verificarProyectoPrograma(parametros);
                if (proyectoAntiguo) {
                    detalleProyectoVO =
projectosServiceFacade.cargarPersonasProyecto(selected.getProjectoId());
                } else {
                    listaPersonaPrograma = proyectosServiceFacade.obtenerPryPersonasPrograma(0,
projectosServiceFacade.countPryPersonasPrograma(parametros, new PersonaProgramaDTO()), null,
null, parametros, new PersonaProgramaDTO());
                }
            } catch (Exception ex) {
                LOG.error("Error al cargar las personas del proyecto", ex);
            }
        }

        public void cargarCronogramaProyecto() {
            try {
                detalleProyectoVO =
projectosServiceFacade.cargarCronogramaProyecto(selected.getProjectoId());
            } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }

```

```
public void cargarMontosProyecto(ProyectoDTO proyectoDTO) {
    try {
        this.selected = proyectoDTO;

        cargarListaProyectosTramites(proyectoDTO.getProyectoId());
        valoresRubroVO =
proyectosServiceFacade.montosRubroProyecto(proyectoDTO.getProyectoId(), tipoExencion);
    } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
    }
}

public boolean renderCampo(String campo, String pantalla) {
    HashMap<String, String> camposValor;
    String render = "";
    if (mapaCampos != null) {
        if (mapaCampos.containsKey(pantalla)) {
            camposValor = mapaCampos.get(pantalla);
            if (camposValor.containsKey(campo)) {
                render = camposValor.get(campo);
            } else {
                render = "S";
            }
        }
    }
    return render.equals("S");
}

private void cargarCamposConvocatoria() {
    try {
        List<ConvocatoriaCampoDTO> listConvocatoriaCampos =
proyectosServiceFacade.buscarCamposConvocatoria(selected.getConvocatoriaId().getConvocatoria
Id());

        HashMap<String, String> camposValor;
        mapaCampos = new HashMap<>();
        if (listConvocatoriaCampos != null && !listConvocatoriaCampos.isEmpty()) {
            for (ConvocatoriaCampoDTO campo : listConvocatoriaCampos) {
                if
(!mapaCampos.containsKey(campo.getPantallaId().getTipoValidacion().getTipoDefinicion())) {
                    camposValor = new HashMap<>();
                    if (campo.getCampoId() != null) {
                        camposValor.put(campo.getCampoId().getValorDefinicion(),
campo.getVisible());
mapaCampos.put(campo.getPantallaId().getTipoValidacion().getTipoDefinicion(), camposValor);
                    }
                } else if (campo.getCampoId() != null) {
                    camposValor =
mapaCampos.get(campo.getPantallaId().getTipoValidacion().getTipoDefinicion());
                    camposValor.put(campo.getCampoId().getValorDefinicion(),
campo.getVisible());
mapaCampos.put(campo.getPantallaId().getTipoValidacion().getTipoDefinicion(), camposValor);
                }
            }
        }
    } catch (Exception ex) {
        LOG.error("Error al consultar los campos de la convocatoria", ex);
    }
}
```

```
}

public boolean existeMenu(String funcion) {
    return listaMenu.containsKey(funcion);
}

private void cargarListaMenusConvocatoria() {
    try {
        listaMenu = new HashMap<>();
        List<ConvocatoriaMenuDTO> listaMenuDTO =
projectosServiceFacade.obtenerConvocatoriaMenu(this.selected.getConvocatoriaId());
        if (listaMenuDTO != null && !listaMenuDTO.isEmpty()) {
            for (ConvocatoriaMenuDTO convocatoriaMenuDTO : listaMenuDTO) {
                if (convocatoriaMenuDTO.getMenuId() != null) {
                    String funcion = convocatoriaMenuDTO.getMenuId().getComentario();
                    if (funcion != null && !funcion.equals("")) {
                        listaMenu.put(funcion, convocatoriaMenuDTO);
                    }
                }
            }
        }
    } catch (Exception ex) {
        LOG.error("Error al cargar la lista de menus");
    }
}

public String cargarRolesPersona(Long personaProyectoId) {
    try {
        if (proyectoPersonaDTO != null && personaProyectoId != null) {
            listRolPersonas =
projectosServiceFacade.obtenerRolesPersona(proyectoPersonaDTO.getPersonaProyectoId());
            descRolPersona = new String();
            if (listRolPersonas != null && !listRolPersonas.isEmpty()) {
                listRolPersonas.forEach((valorDefinicionDTO) -> {
                    descRolPersona += valorDefinicionDTO.getDescripcionValorDefinicion()
+ ", ";
                });
            }
            return descRolPersona;
        }
    } catch (Exception ex) {
        LOG.error("Error al cargar los roles de la persona", ex);
    }
    return descRolPersona;
}

public void cargarListaProyectosTramites(Long proyectoId) {
    try {
        listaProyectoTramites =
projectosServiceFacade.obtenerProyectosTramites(proyectoId);
        if (listaProyectoTramites != null && !listaProyectoTramites.isEmpty()) {
            tramitesSecundarios = true;
        }
    } catch (Exception ex) {
        Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

```

    }
}

public void actualizarLista(Long proyectoId) {
    proyectoFilter.setProyectoPadreId(proyectoId);
    buscarVisible = false;
    init();
}

public void restablecerLista() {
    buscarVisible = true;
    verProyecto = false;
    init();
    this.redirectPage("app/proyectos/proyectosAntiguos.xhtml");
}

public void cargarPermisos() {
    listaPermisos = this.selected.getPermisoList() != null ? new
ArrayList<>(this.selected.getPermisoList()) : null;
    if (listaPermisosContrato != null && !listaPermisosContrato.isEmpty()) {
        if (listaPermisos == null) {
            listaPermisos = new ArrayList<>();
        }
        for (Map.Entry<String, ValorDefinicionDTO> entry :
listaPermisosContrato.entrySet()) {
            PermisoDTO permisoDTO = new PermisoDTO();
            permisoDTO.setPermisoId(entry.getValue());
            if (!listaPermisos.contains(permisoDTO)) {
                listaPermisos.add(permisoDTO);
            }
        }
    }
}

public String consultarProgramaPadre(Long proyectoId) throws Exception {
    ProyectoDTO proyectoPadre =
proyectosServiceFacade.consultarProyectoPadre(proyectoId);
    return proyectoPadre.getDependenciaId() != null ?
proyectoPadre.getDependenciaId().getNombreDependencia() : "";
}

public Collection<UsuarioDTO> getListUsuariosSigp(String query) {
    try {
        listaUsuarios =
proyectosServiceFacade.obtenerUsuariosSigp(query.replaceAll("^\\s*", ""));
    } catch (Exception ex) {
        LOG.error("Error al obtener el listado de usuarios ", ex);
    }
    return listaUsuarios;
}

public void generarFicha(ProyectoDTO proyecto) {
    String pryId =
seguridadConvController.encriptarCodigo(proyecto.getProyectoId().toString());
    PrimeFaces.current().executeScript("descargarFicha('" + pryId + "')");
}

private void cargarTramitesEstimulos() {

```

```

        try {
            listaTramiteEstimulos = new ArrayList<>();

            Map<String, Object> filters = new HashMap<>();
            //Filtro por Proyecto ID
            filters.put("proyectoId", selected.getProyectoId());
            int countTramites = proyectosServiceFacade.tramiteEstimulosListCount(filters);
            if (countTramites > 0) {
                listaTramiteEstimulos = proyectosServiceFacade.tramiteEstimulosList(-1, -1,
"tramiteId.descripcionValorDefinicion", "A", filters);
            }
        } catch (Exception ex) {
            LOG.error("Error al obtener el listado de tramites de beneficios tributarios del
proyecto ", ex);
        }
    }

    private void cargarProyectosAsociados() {
        try {
            listProyectosPrograma = new ArrayList<>();
            listProyectosPrograma =
proyectosServiceFacade.obtenerProyectosPorPrograma(selected.getProyectoId());

        } catch (Exception ex) {
            LOG.error("Error al obtener el listado de tematicas del proyecto ", ex);
        }
    }

    /**
     * Metodo para consultar la información de un proyecto
     *
     * @param proyectoId
     * @return
     */
    public ProyectoDTO consultarInfoProyecto(Long proyectoId) {
        try {
            return proyectosServiceFacade.obtenerProyecto(proyectoId);
        } catch (Exception ex) {
            LOG.error("Error al consultar la información del proyecto ", ex);
        }
        return new ProyectoDTO();
    }

    /**
     * Metodo para cargar las evaluaciones del proyecto
     */
    public void cargarEvaluacionesProyecto() {
        try {
            listaEvaluaciones = new ArrayList();

            Map<String, Object> filters = new HashMap<>();
            filters.put("proyectoId", this.selected.getProyectoId());
            int countEva = proyectosServiceFacade.proyectoEvaluacionesListCount(filters);
            // System.out.println("countEvaluaciones: " + countEva);
            if (countEva > 0) {
                listaEvaluaciones = proyectosServiceFacade.proyectoEvaluacionesList(-1, -1,
"evaluacionProyectoId", "D", filters);
            }
        }
    }

```

```

    }

    } catch (Exception ex) {
        LOG.error("Error al cargar las evaluaciones de proyectos antiguos ", ex);
    }
}

public void cargarDatosProyectoEval(ProyectoEvaluacionDTO proyectoEvaluacionDTO) {
    try {
        evaluarProyectoSessController.setSelected(proyectoEvaluacionDTO);
        cargarMontosProyecto(proyectoEvaluacionDTO.getProyectoId());
        evaluarProyectoSessController.obtenerDocumentoEvaluacion(proyectoEvaluacionDTO);
        evaluarProyectoSessController.obtenerDocumentoPanel(proyectoEvaluacionDTO);
    } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
    }
}

/**
 * Metodo para cargar los documentos del proyecto
 */
public void cargarDocumentosProyecto() {
    try {
        listaDocumentos = new ArrayList();
        List<DocumentoDTO> documentosSgpProyectos = new ArrayList();
        List<DocumentoDTO> documentosSgpProyectosXml = new ArrayList();
        List<DocumentoDTO> documentosSgpRegistroProyectos = new ArrayList();

        //--Documentos asociados al tipo de origen SGP_PROYECTOS
        Map<String, Object> filtersSgpProyectos = new HashMap<>();
        filtersSgpProyectos.put("origenId", this.selected.getProyectoId());
        filtersSgpProyectos.put("tipoOrigenId",
documentoTipoSgpProyectos.getValorDefinicionId());
        int countDocumentosSgpProyectos =
configuracionServiceFacade.documentosListCount(filtersSgpProyectos);
        //      System.out.println("countDocumentosSgpProyectos: " +
countDocumentosSgpProyectos);
        if (countDocumentosSgpProyectos > 0) {
            documentosSgpProyectos = configuracionServiceFacade.documentosList(-1, -1,
"documentoId", "A", filtersSgpProyectos);
        }

        //--Documentos asociados al tipo de origen SGP_PROYECTOS_XML
        //      System.out.println("formularioProyectoId: " +
selected.getFormularioProyectoId());
        if (selected.getFormularioProyectoId() != null) {
            Map<String, Object> filtersSgpProyectosXML = new HashMap<>();
            filtersSgpProyectosXML.put("origenId",
this.selected.getFormularioProyectoId());
            filtersSgpProyectosXML.put("tipoOrigenId",
documentoTipoSgpProyectosXml.getValorDefinicionId());
            int countDocumentosProyectosXML =
configuracionServiceFacade.documentosListCount(filtersSgpProyectosXML);
            //      System.out.println("countDocumentosProyectosXML: " +
countDocumentosProyectosXML);
            if (countDocumentosProyectosXML > 0) {
                documentosSgpProyectosXml = configuracionServiceFacade.documentosList(-
1, -1, "documentoId", "A", filtersSgpProyectosXML);
            }
        }
    }
}

```



```

    }

    //--Documentos asociados al tipo de origen SGP_REGISTRO_PROYECTOS
    Map<String, Object> filtersSgpRegistroProyectos = new HashMap<>();
    filtersSgpRegistroProyectos.put("origenId", this.selected.getProyectoId());
    filtersSgpRegistroProyectos.put("tipoOrigenId",
    documentoTipoSgpRegistroProyectos.getValorDefinicionId());
    int countDocumentosRegistro =
    configuracionServiceFacade.documentosListCount(filtersSgpRegistroProyectos);
    //      System.out.println("countDocumentosRegistro: " + countDocumentosRegistro);
    if (countDocumentosRegistro > 0) {
        documentosSgpRegistroProyectos = configuracionServiceFacade.documentosList(-
1, -1, "documentoId", "A", filtersSgpRegistroProyectos);
    }

    listaDocumentos.addAll(documentosSgpProyectos);
    listaDocumentos.addAll(documentosSgpProyectosXml);
    listaDocumentos.addAll(documentosSgpRegistroProyectos);

    } catch (Exception ex) {
        LOG.error("Error al cargar los documentos de proyectos antiguos ", ex);
    }
}

/**
 * Metodo para cargar las observaciones de los proyectos
 */
public void cargarObservacionesProyecto() {
    try {
        listaObservacionProyecto =
proyectosServiceFacade.obtenerObservaciones(this.selected.getProyectoId());
    } catch (Exception ex) {
        LOG.error("Error al cargar las observaciones de proyectos antiguos ", ex);
    }
}
}
}

```

El controlador `ProyectoAntiguoViewController` es un *managed bean* con alcance de sesión (`@SessionScoped`) que gestiona la lógica de presentación y navegación para la vista de proyectos antiguos en el sistema. Utiliza tecnologías JSF, PrimeFaces, EJB y se integra con distintos servicios y controladores del sistema.

```

@Named(value = "proyectoAntiguoViewController")
@SessionScoped
public class ProyectoAntiguoViewController extends GenericSessionController<ProyectoDTO>
implements Serializable

```

2. Inyección de Dependencias

El controlador inyecta varios EJBs y beans de control de sesión:

- `proyectosServiceFacade`: Acceso a servicios de negocio relacionados con proyectos.
- `configuracionServiceFacade`: Acceso a configuraciones globales.

- evaluarProyectoSessController, seguridadConvController, configApplController: Beans auxiliares del contexto de sesión.

3. Atributos Principales

El controlador define un conjunto amplio de atributos:

- **Modelo de datos lazy (LazyDataModel<ProyectoDTO>)** para la tabla de proyectos antiguos con paginación y filtros.
- **Filtros y DTOs auxiliares** para mantener el estado de búsqueda, configuraciones y selección.
- **Listas y mapas** de configuraciones, evaluaciones, rubros, entidades, etc.

Esto permite mantener un estado completo de navegación y gestión del proyecto antiguo seleccionado.

4. Método init()

```
@PostConstruct
public void init() {

    this.model = new LazyDataModel<ProyectoDTO>() {

        @Override
        public List<ProyectoDTO> load(int first, int pageSize, String sortField,
SortOrder sortOrder, Map<String, FilterMeta> filterBy) {
            Map<String, Object> filters = new TreeMap<>();
            filterBy.forEach((k, v) -> {
                if (k.equals("globalFilter")) {
                    if (v.getFilterValue() != null &&
!v.getFilterValue().toString().isEmpty()) {
                        filters.put(k, v.getFilterValue());
                    }
                }
            });
            List<ProyectoDTO> proyectoList = new ArrayList<>();
            try {
                //filtro por busqueda avanzada
                if (proyectoFilter != null) {
                    filters.put("proyectoFiltro", proyectoFilter);
                }
                filters.put("proyectosAntiguos", true); //filtro por proyectos antiguos
                filters.put("claveNull", true); //filtro por clave nula (ya que los
proyectos antiguos no la registraban)
                int count = proyectosServiceFacade.proyectosListCount(filters);
                if (count > 0) {
                    proyectoList = proyectosServiceFacade.proyectosList(first, pageSize,
sortField, sortOrder.equals(SortOrder.ASCENDING) ? "A" : "D", filters);
                }
            } catch (Exception e) {
                //log error
            }
            return proyectoList;
        }
    };
}
```

```
        }  
        model.setRowCount(count);  
    } catch (Exception ex) {  
        LOG.error("Error al cargar el listado de proyectos antiguos ", ex);  
    }  
    return proyectoList;  
    }  
};  
}
```

Este método se ejecuta automáticamente después de construir el bean. Dentro de él:

- Se inicializa el modelo LazyDataModel, necesario para paginar los resultados del DataTable en la vista.
- Se definen filtros automáticos para solo cargar proyectos antiguos (clave nula y flag proyectosAntiguos = true).
- Se llama el servicio para contar y listar registros con paginación.

5. Método cargarListas()

```
@Inject  
public void cargarListas() {  
    try {  
        listaEstados = configApplController.getMapValorDefinicion().get("ESTADO");  
        // System.out.println("listaEstados SIZE: "+ listaEstados.size());  
        listaConvocatorias =  
projectosServiceFacade.obtenerConvocatoriasProyectosAntiguos();  
        // System.out.println("listaConvocatorias Antiguas SIZE: " +  
listaConvocatorias.size());  
        listaPermisosContrato =  
configApplController.getMapValorDefinicion().get("CONTPERM");  
        urlReportes =  
projectosServiceFacade.obtenerParametroSistema("URLREPORTE").getCadena();  
        parameterUrlCvLAC =  
projectosServiceFacade.obtenerParametroSistema("URLCVLAC2").getCadena();  
        documentoEvaluacionTipo =  
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("PRY_EVALUACION");  
        documentoTipoSgpProyectos =  
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_PROYECTOS");  
        documentoTipoSgpProyectosXml =  
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_PROYECTOS_XML");  
        documentoTipoSgpRegistroProyectos =  
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_REGISTRO_PROYECTOS");  
  
        // cargamos el listado de años de los proyectos  
        Date fechaFinProy =  
projectosServiceFacade.obtenerParametroSistema("PRYFECHAIN").getFecha();  
        Calendar fechaFinal = Calendar.getInstance();  
        fechaFinal.setTime(fechaFinProy);  
        int anioInicio = 2000;  
        //
```

```
        int añoFinal = fechaFinal.get(Calendar.YEAR);
        listAnios = new ArrayList<>();
        for (int i = anioInicio; i <= añoFinal; i++) {
            listAnios.add(i);
        }

    } catch (Exception ex) {

        Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Este método inicializa listas necesarias para la vista:

- Estados (listaEstados), convocatorias antiguas (listaConvocatorias), permisos de contrato, URLs de reportes y CVLAC.
- Tipos de documentos utilizados en el módulo.
- Años desde el 2000 hasta el último año con proyectos antiguos.

6. Método setTipoProyecto()

```
public void setTipoProyecto(ValueChangeEvent event) {
    String dato = String.valueOf(event.getNewValue());
    switch (dato) {
        case "F":
            this.proyectoFilter.setAnteproyecto("S");
            this.proyectoFilter.setExencion("R");
            break;
        case "R":
            this.proyectoFilter.setExencion("R");
            this.proyectoFilter.setAnteproyecto("N");
            break;
        case "P":
            this.proyectoFilter.setExencion("P");
            this.proyectoFilter.setAnteproyecto(null);
            break;
        default:
            break;
    }
}
```

Permite actualizar el filtro de proyecto (proyectoFilter) basado en el tipo de trámite seleccionado por el usuario.

7. Método cargarDetallesProyecto()

```
public void cargarDetallesProyecto(ProyectoDTO proyectoDTO) {
    try {
        this.selected = proyectoDTO;

        this.selected.setFormularioProyectoId(projectosServiceFacade.obtenerFormularioProyectoId(selected.getProjectoId()));
        valoresRubroVO =
        proyectosServiceFacade.montosRubroProyecto(proyectoDTO.getProjectoId(), null);
        listaObservacionProyecto =
        proyectosServiceFacade.obtenerObservaciones(proyectoDTO.getProjectoId());

        selected.setPalabraClaveList(projectosServiceFacade.obtenerPalabrasClave(selected.getProjectoId()));
        nombreFichaTecnica =
        proyectosServiceFacade.obtenerFichaTecnica(selected.getAnteproyectoId());
        cargarCamposConvocatoria();
        cargarPersonasProyecto();
        cargarListaMenusConvocatoria();
        cargarEntidadesProyecto();
        // cargarPresupuestoProyecto();
        Map<String, Object> filters = new HashMap<>();
        filters.put("proyectoId", this.selected.getProjectoId());
        countEvaluaciones =
        proyectosServiceFacade.proyectoEvaluacionesListCount(filters);
    } catch (Exception ex) {

        Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Carga todos los datos relevantes del proyecto seleccionado:

- Formulario asociado, valores de rubros, observaciones, palabras clave, ficha técnica, evaluaciones.
- También llama métodos auxiliares para cargar personas, menú de convocatoria y entidades.

8. Método cargarInformacionBasicaProyecto()

```
public void cargarInformacionBasicaProyecto() {
    try {

        selected.setPermisosList(projectosServiceFacade.obtenerPermisosProyecto(selected.getProjectoId()));

        selected.setComponentesList(projectosServiceFacade.obtenerComponentes(selected.getProjectoId()));

        selected.setSectoresTecnologicos(projectosServiceFacade.obtenerSectoresTecnologicos(selected.getProjectoId()));

        selected.setClasesInvencionList(projectosServiceFacade.obtenerClasesInvencion(selected.getProjectoId()));
    }
}
```

```
selected.setSublineaSecundariaList(projectosServiceFacade.obtenerSublineasSecundarias(selected.getProjectoId()));

selected.setRolEntidadList(projectosServiceFacade.obtenerRolesEntidad(selected.getProjectoId()));

selected.setFocosTematicosList(projectosServiceFacade.obtenerFocosTematicos(selected.getProjectoId()));
    nombreDependenciaConvocatoria =
projectosServiceFacade.obtenerDependenciaConvocatoria(selected.getConvocatoriaId().getConvocatoriaId());
    componente =
projectosServiceFacade.obtenerCategoria(selected.getProjectoId().getTipoComponenteId());

selected.setTematicasOrientadorasList(projectosServiceFacade.obtenerTematicasOrientadorasFocos(selected.getProjectoId()));
    listaFocos =
projectosServiceFacade.obtenerFocosProyecto(selected.getProjectoId());
    cargarTramitesEstimulos();
    if (selected.getExencion().equals("R")) {
        cargarProyectosAsociados();
    }
} catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
}
}
```

Este método profundiza en la información del proyecto seleccionado:

- Roles, componentes, clases de invención, sublíneas, sectores tecnológicos, focos, temáticas, trámites, etc.
- Carga el nombre de la dependencia de la convocatoria y el componente relacionado.
- Si el proyecto es de tipo "R" (exención), también se cargan proyectos asociados.

Notas Importantes

- Toda la lógica está encapsulada adecuadamente, siguiendo el patrón MVC.
- El uso de LazyDataModel es clave para evitar problemas de rendimiento con grandes volúmenes de datos.
- Se implementan buenas prácticas como @PostConstruct, @EJB, @Inject, manejo de excepciones y separación de lógica por métodos.

♦ **public void cargarEntidadesProyecto()**

¿Qué hace?

Carga la lista de **entidades asociadas** al proyecto seleccionado (selected) desde el servicio proyectosServiceFacade.

Detalle:

```
listaProyectoEntidades =  
proyectosServiceFacade.obtenerProyectoEntidades(selected.getProyectoId());
```

Esto llena listaProyectoEntidades con los datos retornados por el servicio, usando el ID del proyecto seleccionado.

◆ public void cargarRolEntidadesProyecto()

¿Qué hace?

Carga los **roles de cada entidad** en el proyecto y los guarda directamente en el objeto selected.

```
selected.setRolEntidadList(proyectosServiceFacade.obtenerRolesEntidad(selected.getProyectoId()  
()));
```

◆ public void cargarPresupuestoProyecto()

¿Qué hace?

Carga los **rubros presupuestales** del proyecto antiguo y calcula los totales por tipo de aporte (institución, especie, efectivo).

```
listRubros = proyectosServiceFacade.obtenerRubrosProyAntiguos(selected.getProyectoId());  
calcularTotales();
```

◆ public void calcularTotales()

¿Qué hace?

Suma los valores presupuestales de los rubros cargados en listRubros, dividiendo por tipo:

- totalMonInstitucionSol: aporte institucional
- totalConEspecieSol: aporte en especie

- totalConEfectivoSol: aporte en efectivo
- totalSol: total solicitado

```
for (Object[] rubro : listRubros) {  
    totalMonInstitucionSol += ((Number) rubro[1]).doubleValue();  
    ...  
}
```

◆ **public void cargarResultadosProyecto()**

¿Qué hace?

Carga los **resultados del proyecto** (científicos y de desarrollo tecnológico) en los siguientes atributos:

- detalleProyectoVO (resultado principal)
- listResultadoGeneracion (otros resultados tipo "G")
- listResultadoDesarrolloTec (otros resultados tipo "D")

```
detalleProyectoVO =  
projectosServiceFacade.cargarResultadosProyecto(selected.getProjectoId());
```

◆ **public void cargarImpactosProyecto()**

¿Qué hace?

Carga los **impactos del proyecto** y los guarda en detalleProyectoVO.

```
detalleProyectoVO =  
projectosServiceFacade.obtenerImpactosProyecto(selected.getProjectoId());
```

◆ **public void cargarCoberturasProyecto()**

¿Qué hace?

Carga las **coberturas geográficas o poblacionales** del proyecto.

```
listCoberturas = proyectosServiceFacade.obtenerCoberturas(selected.getProjectoId());
```

◆ **public void cargarMarcoLogico()**

¿Qué hace?

Carga la información del **marco lógico** del proyecto, incluyendo objetivos generales.

```
detalleProyectoVO =  
proyectosServiceFacade.cargarMarcoLogicoProyecto(selected.getProyectoId());  
marcoObjGeneralDTO = detalleProyectoVO.getListaObjetivosGeneral().get(0);
```

◆ **public void cargarTipoPersonal()**

¿Qué hace?

Carga la lista de tipos de personal que participaron en el proyecto (investigadores, administrativos, etc.).

```
listaTipoPersonal = proyectosServiceFacade.obtenerTipoPersonal(selected.getProyectoId());
```

◆ **public void cargarPersonasProyecto()**

¿Qué hace?

Carga las **personas que participaron** en el proyecto.

1. Primero obtiene el tipo de documento definido para hoja de vida.
2. Verifica si el proyecto es antiguo usando verificarProyectoPrograma(...).
3. Si es antiguo, carga desde detalleProyectoVO.
4. Si **no** es antiguo, carga desde obtenerPryPersonasPrograma(...) con paginación.

```
proyectoAntiguo = proyectosServiceFacade.verificarProyectoPrograma(parametros);  
if (proyectoAntiguo) {  
    detalleProyectoVO =  
proyectosServiceFacade.cargarPersonasProyecto(selected.getProyectoId());  
} else {  
    listaPersonaPrograma = proyectosServiceFacade.obtenerPryPersonasPrograma(...);  
}
```

◆ **public void cargarCronogramaProyecto()**

¿Qué hace?

Carga el **cronograma de actividades del proyecto** y lo asigna a detalleProyectoVO.

```
detalleProyectoVO =  
proyectosServiceFacade.cargarCronogramaProyecto(selected.getProyectoId());
```

◆ **public void cargarMontosProyecto(ProyectoDTO proyectoDTO)**

¿Qué hace?

Carga los **montos por rubro** del proyecto (para exención tributaria) y también carga trámites asociados.

```
cargarListaProyectosTramites(proyectoDTO.getProyectoId());  
valoresRubroVO = proyectosServiceFacade.montosRubroProyecto(proyectoDTO.getProyectoId(),  
tipoExencion);
```

◆ **public boolean renderCampo(String campo, String pantalla)**

¿Qué hace?

Verifica si un campo específico debe renderizarse en una pantalla según un mapa de configuración (mapaCampos).

- Si no encuentra la configuración, por defecto retorna true ("S").

```
return render.equals("S");
```

◆ **private void cargarCamposConvocatoria()**

¿Qué hace?

Carga la configuración de **campos visibles** para una convocatoria específica y los organiza en un mapa (mapaCampos) para controlar qué se muestra por pantalla.

- Usa ConvocatoriaCampoDTO para consultar visibilidad por pantalla y campo.
- Llena mapaCampos<pantalla, Map<campo, visible>>.

◆ **public boolean existeMenu(String funcion)**

¿Qué hace?

Verifica si una **función o menú específico existe** en el mapa listaMenu (cargado previamente).

◆ **private void cargarListaMenusConvocatoria()**

¿Qué hace?

Carga los **menús configurados para la convocatoria actual**, y los guarda en listaMenu.

- Se usa para decidir qué pestañas o secciones mostrar en la UI.
-

◆ **public String cargarRolesPersona(Long personaProyectold)**

¿Qué hace?

Carga los **roles de una persona** asociada al proyecto (por ejemplo: investigador, coordinador).

- Retorna un String con la concatenación de descripciones de roles.
-

◆ **public void cargarListaProyectosTramites(Long proyectold)**

¿Qué hace?

Consulta los **trámites asociados al proyecto**, y si existen, activa la bandera tramitesSecundarios.

◆ **public void actualizarLista(Long proyectold)**

¿Qué hace?

Actualiza el filtro proyectoFilter para listar los proyectos hijos de un proyecto padre. Llama a init() para recargar la vista.

◆ **public void restablecerLista()**

¿Qué hace?

Restaura la vista a su estado inicial, con la lista de proyectos. También hace una redirección a la página principal de proyectos antiguos.

◆ **public void cargarPermisos()**

¿Qué hace?

Carga los permisos actuales del proyecto (selected.getPermisoList()), y además **agrega permisos adicionales** si vienen desde listaPermisosContrato.

◆ **public String consultarProgramaPadre(Long proyectold)**

¿Qué hace?

Consulta el **proyecto padre** y devuelve el nombre de la dependencia asociada.

◆ **public Collection<UsuarioDTO> getListaUsuariosSigp(String query)**

¿Qué hace?

Busca usuarios del SIGP según un texto de búsqueda (query) ingresado, y los retorna.

◆ **public void generarFicha(ProyectoDTO proyecto)**

¿Qué hace?

Genera una ficha del proyecto usando su ID encriptado. Ejecuta un JavaScript (descargarFicha(...)) en el navegador.

◆ **private void cargarTramitesEstimulos()**

¿Qué hace?

Carga una lista de **trámites de estímulos tributarios** asociados al proyecto, si existen.

◆ **private void cargarProyectosAsociados()**

¿Qué hace?

Consulta y carga todos los **proyectos asociados** a un programa (proyecto principal).

◆ **public ProyectoDTO consultarInfoProyecto(Long proyectoid)**

¿Qué hace?

Consulta toda la información del proyecto por su ID y la retorna como ProyectoDTO.

◆ **public void cargarEvaluacionesProyecto()**

¿Qué hace?

Consulta las **evaluaciones del proyecto** (si existen), y las guarda en listaEvaluaciones.

◆ **public void cargarDatosProyectoEval(ProyectoEvaluacionDTO proyectoEvaluacionDTO)**

¿Qué hace?

Carga todos los datos necesarios para mostrar la evaluación seleccionada:

1. Montos del proyecto.
 2. Documento de evaluación.
 3. Documento del panel evaluador.
-

◆ **public void cargarDocumentosProyecto()**

¿Qué hace?

Carga todos los **documentos asociados** al proyecto desde tres fuentes distintas:

- SGP_PROYECTOS
 - SGP_PROYECTOS_XML (si tiene formularioProyectoid)
-

- SGP_REGISTRO_PROYECTOS

Agrupar todo en listaDocumentos.

◆ **public void cargarObservacionesProyecto()**

¿Qué hace?

Consulta y carga las **observaciones** que se han hecho al proyecto.

Controlador de Conversación ProyectoAntiguoConvController.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.sb.colciencias.gestion.web.controller.proyectos.conversation;

import com.sb.colciencias.gestion.business.domain.scienti.ScientiPersona;
import com.sb.colciencias.gestion.business.facade.convocatoria.ConvocatoriaServiceFacade;
import com.sb.colciencias.gestion.business.facade.proyectos.ProyectosServiceFacade;
import com.sb.colciencias.gestion.business.facade.scienti.ScientiServiceFacade;
import com.sb.colciencias.gestion.business.transfer.ConvocatoriaDTO;
import com.sb.colciencias.gestion.business.transfer.DocumentoDTO;
import com.sb.colciencias.gestion.business.transfer.EntidadDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoEvaluacionDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoPersonaDTO;
import com.sb.colciencias.gestion.business.transfer.RubroDTO;
import com.sb.colciencias.gestion.business.transfer.ValorDefinicionDTO;
import com.sb.colciencias.gestion.business.util.SystemException;
import com.sb.colciencias.gestion.web.controller.application.ConfigAppController;
import com.sb.colciencias.gestion.web.controller.proyectos.session.ProyectoAllSessController;
import com.sb.colciencias.gestion.web.controller.proyectos.view.ProyectoAntiguoViewController;
import com.sb.colciencias.gestion.web.util.component.ReporteUtil;
import com.sb.colciencias.gestion.web.util.poi.ReporteExcelMarcoLogico;
import com.sb.colciencias.gestion.web.util.security.SeguridadSessController;
import com.sb.standard.web.controller.conversation.GenericConversationController;
import java.io.ByteArrayInputStream;
import java.io.Serializable;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.logging.Level;
import javax.ejb.EJB;
```

```
import javax.enterprise.context.ConversationScoped;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.event.ActionEvent;
import javax.faces.model.SelectItem;
import javax.inject.Inject;
import javax.inject.Named;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletResponse;
import org.apache.log4j.Logger;
import org.primefaces.component.datatable.DataTable;
import org.primefaces.model.DefaultStreamedContent;

/**
 *
 * @author lsuarez
 */
@Named(value = "proyectoAntiguoConvController")
@ConversationScoped
public class ProyectoAntiguoConvController extends GenericConversationController<EntidadDTO>
implements Serializable {

    @EJB
    private ProyectosServiceFacade proyectoServiceFacade;
    @EJB
    private ScientiServiceFacade scientiServiceFacade;
    @EJB
    private ConvocatoriaServiceFacade convocatoriaServiceFacade;
    @Inject
    private ProyectoAntiguoViewController proyectoAntiguoViewController;
    @Inject
    private ProyectoAllSessController proyectoAllSessController;
    @Inject
    private SeguridadSessController seguridadSessController;
    @Inject
    private ReporteExcelMarcoLogico reporteExcelMarcoLogico;
    @Inject
    private ConfigApplController configApplController;

    private Boolean cambioEstado;
    private String estadoAnterior;
    private String observacionJustificacion;
    private String estadoRetiro;

    private final String programa = "GEST5400";

    /**
     * Define el log del sistema
     */
    private static final Logger LOG = Logger.getLogger(ProyectoAntiguoConvController.class);

    public ProyectoAntiguoConvController() {
        this.selected = new EntidadDTO();
        cambioEstado = false;
    }

    @Override
    public void guardar(ActionEvent ae) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    }
}
```

```
@Override
public void buscar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void eliminar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void editar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void mostrar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void goNuevo(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void goEditar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void goBuscar(int i) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void goEliminar() {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

public void cambiarEstado(String estado) throws SystemException {
    ProyectoDTO proyectoDTO = proyectoAllSessController.getSelected();
    switch (estado) {
        case "REVISADO":
            String codigoEntidad = proyectoDTO.getEntidadNId().getCodigoEntidad();
            if (codigoEntidad != null) {
                if (codigoEntidad.length() < 4) {
                    codigoEntidad = "000" + codigoEntidad;
                    codigoEntidad = codigoEntidad.substring(codigoEntidad.length(),
codigoEntidad.length() - 4);
                }
            }
        }
    }
```



```

        proyectoAllSessController.getSelected().setCodigoProyecto(codigoEntidad
+ proyectoDTO.getConvocatoriaId().getCodigoConvocatoria() + proyectoDTO.getProyectoId());
        cambioEstado = true;
    } else {
        cambioEstado = false;
        throw new
SystemException("#{msg['gestion.beneficiosTributarios.evaluadores.error.codigoEntidad']}");
    }
    break;
    case "DILIGENCIANDO":
        if (!convocatoriaActiva()) {
            throw new
SystemException("#{msg['gestion.beneficiosTributarios.evaluadores.error.diligenciado']}");
        } else {
            proyectoDTO.setFechaEnvio(null);
            cambioEstado = true;
        }
        break;
    default:
        cambioEstado = true;
        break;
    }
    if (cambioEstado) {
        estadoAnterior =
proyectoAllSessController.getSelected().getEstadoProyectoId().getDescripcionValorDefinicion(
);
proyectoDTO.setEstadoProyectoId(proyectoAllSessController.getListaEstados().get(estado));
        this.editar(null);
    }
}

/**
 * Este metodo devuelve true en caso de que se deba manejar el formulario
 * como se maneja antes de acuerdo al parametro de fecha
 *
 * @return
 */
public boolean verificarProyectoPrograma() {
    try {
        HashMap<String, Object> parametros = new HashMap<>();
        parametros.put("fechaCreacionProyecto",
proyectoAllSessController.getSelected().getFechaCreacion());
        parametros.put("tipoProyecto",
proyectoAllSessController.getSelected().getExencion());
        ConvocatoriaDTO convocatoriaDTO =
proyectoAllSessController.getSelected().getConvocatoriaId();
        String convocatoriaPrograma = convocatoriaDTO.getPrograma() != null ?
convocatoriaDTO.getPrograma() : "N";
        parametros.put("convocatoriaPrograma", convocatoriaPrograma);
        return proyectoServiceFacade.verificarProyectoPrograma(parametros);
    } catch (Exception ex) {
        LOG.error("Error al verificar metodo Proyecto programa ", ex);
    }
    return false;
}

public String isBeforeToday(Date d) {
    Calendar cal = Calendar.getInstance();
    cal.set(Calendar.HOUR_OF_DAY, 0);
    cal.set(Calendar.MINUTE, 0);

```

```

        cal.set(Calendar.SECOND, 0);
        cal.set(Calendar.MILLISECOND, 0);
        Date today = cal.getTime();
        if (d.before(today)) {
            cal.setTime(today);
            cal.add(Calendar.DATE, -1);
            Date ayer = cal.getTime();
            if (d.before(ayer)) {
                long startTime = d.getTime();
                long endTime = today.getTime();
                long diffTime = endTime - startTime;
                long diffDays = diffTime / (1000 * 60 * 60 * 24);
                return diffDays + " " +
this.resolveExpression("#{msg['gestion.beneficiosTributarios.proyectos.diasAtras']}");
            } else {
                return
this.resolveExpression("#{msg['gestion.beneficiosTributarios.proyectos.ayer']}");
            }
        } else {
            return
this.resolveExpression("#{msg['gestion.beneficiosTributarios.proyectos.hoy']}");
        }
    }

    public Boolean convocatoriaActiva() {
        ConvocatoriaDTO convocatoriaDTO =
projectoAllSessController.getSelected().getConvocatoriaId();
        Date dt = new Date();
        Calendar c = Calendar.getInstance();
        c.setTime(dt);
        c.add(Calendar.DATE, 1);
        dt = c.getTime();
        return !(convocatoriaDTO.getFechaCierre().before(dt)
            || convocatoriaDTO.getActivo().equals("N")
            || convocatoriaDTO.getPublicar().equals("N"));
    }

    public void lazyLoad() {
        proyectoAntiguoViewController.setFiltroActivo(true);
        DataTable dataTable = (DataTable)
FacesContext.getCurrentInstance().getViewRoot().findComponent("proyectodtoForm:Table");
        dataTable.setFilterBy(null);
        dataTable.loadLazyData();
    }

    public void lazyLoadReg() {
        proyectoAllSessController.setFiltroActivoReg(true);
        DataTable dataTable = (DataTable)
FacesContext.getCurrentInstance().getViewRoot().findComponent("proyectodtoForm:Table");
        dataTable.setFilterBy(null);
        dataTable.loadLazyData();
    }

    public void limpiarFiltros() {
        proyectoAntiguoViewController.setProyectoFilter(new ProyectoDTO());
        proyectoAntiguoViewController.setTipoTramite(new ValorDefinicionDTO());
        proyectoAntiguoViewController.setProgramaSelected(null);
        proyectoAntiguoViewController.setFiltroActivo(false);
        System.out.println("limpiando");
    }
}

```

```

public void limpiarFiltrosReg() {
    proyectoAllSessController.setProyectoFilterReg(new ProyectoDTO());
    proyectoAllSessController.setTipoTramite(new ValorDefinicionDTO());
    proyectoAllSessController.setFiltroActivoReg(false);
    proyectoAllSessController.setProgramaSelected(null);
    System.out.println("limpiando");
}

public void filtrarHijos(ProyectoDTO proyectoDTO) {
    if (proyectoDTO != null && proyectoDTO.getProyectoId() != null) {
        ProyectoDTO filtro = new ProyectoDTO(proyectoDTO.getProyectoId());
        filtro.setExencion("H");
        proyectoAntiguoViewController.setProgramaSelected(proyectoDTO);
        proyectoAntiguoViewController.setProyectoFilter(filtro);
        proyectoAntiguoViewController.setFiltroActivo(false);
        this.lazyLoadSinFiltroActivo();
    }
}

public void lazyLoadSinFiltroActivo() {
    DataTable dataTable = (DataTable)
FacesContext.getCurrentInstance().getViewRoot().findComponent("proyectodtoForm:Table");
    dataTable.setFilterBy(null);
    dataTable.loadLazyData();
}

public void filtrarHijosReg(ProyectoDTO proyectoDTO) {
    if (proyectoDTO != null && proyectoDTO.getProyectoId() != null) {
        ProyectoDTO filtro = new ProyectoDTO(proyectoDTO.getProyectoId());
        filtro.setExencion("H");
        proyectoAllSessController.setProgramaSelected(proyectoDTO);
        proyectoAllSessController.setProyectoFilterReg(filtro);
        proyectoAllSessController.setFiltroActivoReg(false);
        this.lazyLoadSinFiltroActivo();
    }
}

public void cargarProgramas() {
    proyectoAllSessController.setProgramaSelected(null);
    ProyectoDTO filtro = new ProyectoDTO();
    filtro.setExencion("R");
    filtro.setAnteproyecto("N");
    proyectoAllSessController.setProyectoFilterReg(filtro);
    this.lazyLoadReg();
}

public String getObservacionJustificacion() {
    return observacionJustificacion;
}

public void setObservacionJustificacion(String observacionJustificacion) {
    this.observacionJustificacion = observacionJustificacion;
}

public void cargarDescripcionesProyecto() {
    try {
        proyectoAntiguoViewController.setListaDescripciones(proyectoServiceFacade.obtenerDescripcion
es(proyectoAntiguoViewController.getSelected().getProyectoId()));
    } catch (Exception ex) {

```

```

java.util.logging.Logger.getLogger(ProyectoAntiguoConvController.class.getName()).log(Level.
SEVERE, null, ex);
    }
}

public void cargarDetallesRubrosProyecto(Long rubroId) {
    try {

        Long proyectoId = proyectoAntiguoViewController.getSelected().getProyectoId();
        String tipoExencion = proyectoAntiguoViewController.getTipoExencion();

        List<RubroDTO> listaRubros = null;
        Map<String, Object> filters = new HashMap<>();
        filters.put("proyectoId", proyectoId);
        filters.put("rubroId", rubroId);

        if (proyectoAntiguoViewController.getTipoExencion() != null) {
            filters.put("tipoExencion", tipoExencion);
        }
        int count = proyectoServiceFacade.countRubros(filters);
        if (count > 0) {
            listaRubros = proyectoServiceFacade.obtenerRubros(-1, -1,
"presupuestoRubroId", "A", filters);
        }
        proyectoAntiguoViewController.setRubrosDetalleProyectoList(listaRubros);
    } catch (Exception ex) {

java.util.logging.Logger.getLogger(ProyectoAntiguoConvController.class.getName()).log(Level.
SEVERE, null, ex);
    }
}

public Long calcularTotalRubroConvocatoria(Long cuadroPresupuestoId) {
    Long total = null;
    try {
        total =
proyectoServiceFacade.obtenerTotalRubroConvocatoriaAntiguas(proyectoAntiguoViewController.ge
tSelected().getProyectoId(), cuadroPresupuestoId,
proyectoAntiguoViewController.getTipoExencion());
    } catch (Exception e) {
        LOG.error("Error al calcular total rubro cuadro " + cuadroPresupuestoId, e);
    }
    return total == null ? 0L : total;
}

public void crearReporteMarcoLogico(ProyectoDTO proyectoSelected) {
    try {
        byte reporte[] = reporteExcelMarcoLogico.generarReporte(proyectoSelected);
        HttpServletResponse response = (HttpServletResponse)
FacesContext.getCurrentInstance().getExternalContext().getResponse();
        response.setContentType(reporteExcelMarcoLogico.isFormatoXlsx() ?
"application/x-xls" : "application/xls");
        response.setHeader("Content-Disposition", "attachment;
filename=\"reporteMarcoLogico_" + proyectoSelected.getProyectoId() + "." +
(reporteExcelMarcoLogico.isFormatoXlsx() ? "xlsx" : "xls") + "\"");
        response.setContentLength(reporte.length);
        response.getOutputStream().write(reporte, 0, reporte.length);
        response.getOutputStream().flush();
        response.getOutputStream().close();
        FacesContext.getCurrentInstance().responseComplete();
    }
}

```

```

    } catch (Exception ex) {
        LOG.error("Error al generar el reporte de marco lógico ", ex);
    }
}

public void cargarInfoPersonal(ProyectoPersonaDTO proyectoPersonaDTO) {
    try {
        if (proyectoPersonaDTO.getRegistroCvllac() != null &&
proyectoPersonaDTO.getRegistroCvllac().equals("S")) {
            //Consultamos la informacion de la persona en CvLAC
            proyectoPersonaDTO = consultarInformacionCvllac(proyectoPersonaDTO);
        }
        //Agregamos el documento de Hoja de vida (No CvLAC)
        if (proyectoPersonaDTO.getRegistroCvllac() == null ||
proyectoPersonaDTO.getRegistroCvllac().equals("N")) {
            proyectoPersonaDTO.setDocumentoHojaDeVida(proyectoServiceFacade.obtenerDocumento(proyectoAnt
iguoViewController.getTipoDocHojaVida(), proyectoPersonaDTO.getPersonaProyectoId()));
        }
        proyectoAntiguoViewController.setProyectoPersonaDTO(proyectoPersonaDTO);
    } catch (Exception ex) {
        LOG.error("Error al cargar la información de la persona", ex);
    }
}

/**
 * Metodo para consultar la información de la persona en CvLAC
 *
 * @throws SystemException
 */
private ProyectoPersonaDTO consultarInformacionCvllac(ProyectoPersonaDTO
proyectoPersonaDTO) throws SystemException {
    try {
        if (proyectoPersonaDTO.getRegistroCvllac().equals("S")) {
            String tipoDoc = "";
            /* 010 = Cedula de Ciudadania
            020 = Tarjeta de Identidad
            050 = Cedula de Extranjeria
            */
            if
(proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("050")) {
                tipoDoc = "E";
            } else if
(proyectoPersonaDTO.getTipoIdentificacionId().getDescripcionValorDefinicion().contains("PASA
PORTE")) {
                tipoDoc = "P";
            } else if
(proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("010")) {
                tipoDoc = "C";
            } else if
(proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("020")) {
                tipoDoc = "T";
            }
            ScientiPersona scientiPersona =
scientiServiceFacade.obtenerScientiPersona(tipoDoc,
String.valueOf(proyectoPersonaDTO.getNumeroIdentificacion()) == null ? "" :
proyectoPersonaDTO.getNumeroIdentificacion());
            if (scientiPersona != null) {
                //Enfoque Diferencial
                if (scientiPersona.getStaEnfDiferencial().equals("T")) {

```

```

        if (scientiPersona.getTxtDscCategoria() != null &&
!scientiPersona.getTxtDscCategoria().isEmpty()) {
projectoPersonaDTO.setCategoria scientiPersona.getTxtDscCategoria());
        }
        if (scientiPersona.getStaVictimaConflicto() != null &&
!scientiPersona.getStaVictimaConflicto().isEmpty()) {
projectoPersonaDTO.setPoblacionVictimaConflicto( scientiPersona.getStaVictimaConflicto().equa
ls("T") ? "S" : "N");
        }
        if (scientiPersona.getTxtPoblacionDisc() != null &&
!scientiPersona.getTxtPoblacionDisc().isEmpty()) {
projectoPersonaDTO.setPoblacionCondicionDisc( scientiPersona.getDescPoblacionDisc());
        }
        if (scientiPersona.getTxtGrupoEtnico() != null &&
!scientiPersona.getTxtGrupoEtnico().isEmpty()) {
projectoPersonaDTO.setGrupoEtnico( scientiPersona.getDescGrupoEtnico());
        }
        if (scientiPersona.getDescGrupoEtario() != null &&
!scientiPersona.getDescGrupoEtario().isEmpty()) {
projectoPersonaDTO.setGrupoEtario( scientiPersona.getDescGrupoEtario());
        }
    }
    }
    return proyectoPersonaDTO;
}
} catch (Exception ex) {
    LOG.error("Error al consultar la información de la persona en CvLAC ", ex);
}
return proyectoPersonaDTO;
}

/**
 * Metodo para obtener a hoja de vida del personal de CvLAC
 *
 * @param personaDTO
 * @return
 */
public String obtenerUrlHojaVidaCvLAC(ProyectoPersonaDTO personaDTO) {
    String url = "";
    if (personaDTO != null) {
        if (personaDTO.getTipoIdentificacionId() != null &&
personaDTO.getNumeroIdentificacion() != null
        && personaDTO.getRegistroCvLac() != null &&
personaDTO.getRegistroCvLac().equals("S")) {
            try {
                ProyectoPersonaDTO proyectoPersonaDTO =
proyectoAntiguoViewController.getProyectoPersonaDTO();
                String tipoDoc = "";
                /* 010 = Cedula de Ciudadania
                020 = Tarjeta de Identidad
                050 = Cedula de Extranjeria
                */
                if
                (proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("050")) {
                    tipoDoc = "E";
                }
            }
        }
    }
}

```

```

        } else if
        (proyectoPersonaDTO.getTipoIdentificacionId().getDescripcionValorDefinicion().contains("PASA
        PORTE")) {
            tipoDoc = "P";
        } else if
        (proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("010")) {
            tipoDoc = "C";
        } else if
        (proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("020")) {
            tipoDoc = "T";
        }
        String numeroIdentificacion =
        proyectoPersonaDTO.getNumeroIdentificacion().trim();
        ScientiPersona scientiPersona =
        scientiServiceFacade.obtenerScientiPersona(tipoDoc, numeroIdentificacion != null ?
        numeroIdentificacion : "");
        if (scientiPersona != null) {
            url = proyectoAntiguoViewController.getParameterUrlCvLAC() +
        scientiPersona.getCodRh();
        }
        } catch (Exception ex) {
            LOG.error("Error al obtener la url de la hoja de vida de CvLAC ", ex);
        }
    }
}
return url;
}

public void descargarArchivoHojaVida(DocumentoDTO documentoTO) {
    ByteArrayInputStream bis = new ByteArrayInputStream(documentoTO.getArchivo());
    proyectoAntiguoViewController.setArchivoHojaVida(new DefaultStreamedContent(bis,
    documentoTO.getMime(), documentoTO.getNombre()));
}

public String getEstadoRetiro() {
    return estadoRetiro;
}

public void setEstadoRetiro(String estadoRetiro) {
    this.estadoRetiro = estadoRetiro;
}

public String consultarProgramaPadre(Long proyectoId) throws Exception {
    ProyectoDTO proyectoPadre =
    proyectoServiceFacade.consultarProyectoPadre(proyectoId);
    return proyectoPadre.getDependenciaId() != null ?
    proyectoPadre.getDependenciaId().getNombreDependencia() : "";
}

public String consultarValorTipoEvaluacion(String valor) {
    List<SelectItem> lista = configAppController.getListaTipoEvaluacion();
    if (lista != null && !lista.isEmpty()) {
        for (SelectItem selectItem : lista) {
            if (selectItem.getValue().equals(valor)) {
                return selectItem.getLabel();
            }
        }
    }
}
return "";
}
}

```

```

public String nombreCompletoPersona(ProyectoEvaluacionDTO proyectoEvaluacionDTO) {
    String nombre = "";
    if (proyectoEvaluacionDTO != null) {
        if (proyectoEvaluacionDTO.getPersonaId() != null) {
            nombre = proyectoEvaluacionDTO.getPersonaId().getNombreCompletoPersona();
        } else if (proyectoEvaluacionDTO.getEvaluadorScientiId() != null) {
            nombre = proyectoEvaluacionDTO.getEvaluadorScientiId().getNombreCompleto();
        }
    }
    return nombre;
}

public void prepararEditar(ProyectoEvaluacionDTO proyectoEvaluacionSeleccionado) {
    try {
        //Se consulta la evaluacion por si se actualizaron los datos
        ProyectoEvaluacionDTO proyectoEvaluacionDTO =
        proyectoServiceFacade.obtenerEvaluacion(proyectoEvaluacionSeleccionado.getEvaluacionProyecto
        Id(), proyectoEvaluacionSeleccionado.getProyectoId().getProyectoId());
        obtenerDocumentoEvaluacion(proyectoEvaluacionDTO);

        proyectoAntiguoViewController.setProyectoEvaluacionSelected(proyectoEvaluacionDTO);
    } catch (Exception ex) {
        LOG.error("Error al preparaEditar la Evaluacion ", ex);
    }
}

/**
 * Metodo para obtener el documento de evaluacion
 */
@param evaluacionSeleccionada
*/
public void obtenerDocumentoEvaluacion(ProyectoEvaluacionDTO evaluacionSeleccionada) {
    try {
        DocumentoDTO documentoEvaluacion =
        proyectoServiceFacade.obtenerDocumento(proyectoAntiguoViewController.getDocumentoEvaluacionT
        ipo(), evaluacionSeleccionada.getEvaluacionProyectoId());
        proyectoAntiguoViewController.setDocumentoEvaluacion(documentoEvaluacion);
    } catch (Exception ex) {
        LOG.error("Error al obtener el documento de la Evaluacion ", ex);
    }
}

/**
 * Metodo que descarga el documento de evaluacion
 */
public void descargarArchivoEvaluacion() {
    ByteArrayInputStream bis = new
    ByteArrayInputStream(proyectoAntiguoViewController.getDocumentoEvaluacion().getArchivo());
    proyectoAntiguoViewController.setArchivoEvaluacion(
        new DefaultStreamedContent(
            bis,
            proyectoAntiguoViewController.getDocumentoEvaluacion().getMime(),
            proyectoAntiguoViewController.getDocumentoEvaluacion().getNombre());
}

/**
 * Metodo para descargar el reporte de evaluacion calificacion
 */
@param tipoReporte

```



```

    * @param evaluacionProyectoId
    */
    public void downloadReporteEvaluacion(String tipoReporte, Integer evaluacionProyectoId)
    {
        FacesContext context = FacesContext.getCurrentInstance();
        HttpServletResponse response = (HttpServletResponse)
context.getExternalContext().getResponse();
        String proyectoId =
proyectoAntiguoViewController.getSelected().getProyectoId().toString();
        try {
            ReporteUtil reporteUtil = new ReporteUtil();
            HashMap hashMap = new HashMap();
            hashMap.put("PROYECTO_ID", proyectoId);
            if (evaluacionProyectoId != null) {
                hashMap.put("evaluacion_proyecto_id", String.valueOf(evaluacionProyectoId));
            }

            byte reporte[];

            //carpeta por defecto
            reporteUtil.setNomCarpeta("ReporteEvaluacionHistorico");
            String nombreReporte;
            String headerReporte;
            //Reporte de Evaluacion
            if (tipoReporte.equals("evaluacion")) {
                nombreReporte = "ReporteEvaluacionHistorico";
                headerReporte = nombreReporte + "_" + proyectoId;
            } else {
                //Reporte de Evaluacion Calificacion
                nombreReporte = "ReporteEvaluacionCalificacionHistorico";
                headerReporte = nombreReporte + "_" + proyectoId + "_" +
(evaluacionProyectoId != null ? evaluacionProyectoId.toString() : "");
            }

            //Se setea la imagen del logo parametrizada, guardada en la base de datos
            String imagenLogoBase64 =
proyectoServiceFacade.obtenerParametroSistema("LOGIN_LOGO").getDetalle();
            String[] parts = imagenLogoBase64.split(",");
            String part1 = parts[0]; // data:image/jpeg;base64
            String imagen = parts[1]; // IMAGEN
            hashMap.put("IMG_BASE64", imagen);
            //
            reporte = reporteUtil.generarReporte(hashMap, nombreReporte, "PDF", context);
            response.setContentType("application/pdf");
            response.setHeader("Content-Disposition", "attachment; filename=\"" +
headerReporte + ".pdf\"");
            response.setContentLength(reporte.length);
            ServletOutputStream outputStream = response.getOutputStream();
            outputStream.write(reporte, 0, reporte.length);
            outputStream.flush();
            outputStream.close();
        } catch (Exception ex) {
            FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
            LOG.error("Error al generar el reporte de evaluacion calificación", ex);

            FacesContext.getCurrentInstance().addMessage(null, message);
        }
    }
}

```

```

/**
 * Metodo para descargar documentos adjuntos
 *
 * @param documentoDTO
 */
public void descargarArchivoDocumentos(DocumentoDTO documentoDTO) {
    try {
        //Para estar seguros antes, vuelvo y consulto el documento
        DocumentoDTO documentoConsultado = documentoDTO;
        ByteArrayInputStream bis = new
        ByteArrayInputStream(documentoConsultado.getArchivo());
        proyectoAntiguoViewController.setArchivoDocumento(new
        DefaultStreamedContent(bis, documentoConsultado.getMime(),
        documentoConsultado.getNombre()));
    } catch (Exception ex) {
        LOG.error("Error al descargar el documento " + documentoDTO.getNombre() + " con
        tipo origen ID " + documentoDTO.getTipoOrigenId().getValorDefinicionId() + " - " +
        documentoDTO.getTipoOrigenId().getDescripcionValorDefinicion() + " en los documentos del
        proyecto ID " + proyectoAntiguoViewController.getSelected().getProyectoId() + " ", ex);
    }
}

/**
 * Metodo para consultar y obtener la dependencia asociada
 *
 * @param dependenciaId
 * @return
 */
public String consultarDependencia(Long dependenciaId) {
    if (dependenciaId != null) {
        try {
            return
            convocatoriaServiceFacade.consultarProgramaConvocatoria(dependenciaId);
        } catch (Exception ex) {
            LOG.error("Error al consultar la dependencia ", ex);
        }
    }
    return null;
}
}

```

ProyectoAntiguoConvController es un controlador de conversación encargado de manejar la lógica de negocio relacionada con la visualización, edición, filtrado y reporte de proyectos antiguos vinculados a convocatorias en el sistema SIGP. Este controlador trabaja de forma coordinada con otros controladores y servicios para garantizar la gestión eficiente de los proyectos según su estado, tipo de exención y reglas de negocio del sistema.

Inyecciones de Dependencia

```

@EJB
private ProyectosServiceFacade proyectoServiceFacade;
@EJB
private ScientiServiceFacade scientiServiceFacade;

```

```
@EJB
private ConvocatoriaServiceFacade convocatoriaServiceFacade;
@Inject
private ProyectoAntiguoViewController proyectoAntiguoViewController;
@Inject
private ProyectoAllSessController proyectoAllSessController;
@Inject
private SeguridadSessController seguridadSessController;
@Inject
private ReporteExcelMarcoLogico reporteExcelMarcoLogico;
@Inject
private ConfigApplController configApplController;
```

Descripción:

Proveen acceso a la lógica de negocio (EJBs) y otros controladores de sesión o visualización para operaciones sobre proyectos y convocatorias.

Atributos Principales

- **cambioEstado (Boolean):** Bandera para determinar si se realizó un cambio de estado del proyecto.
- **estadoAnterior (String):** Almacena el estado anterior del proyecto antes de realizar el cambio.
- **observacionJustificacion (String):** Campo utilizado para registrar observaciones justificativas del cambio de estado.
- **estadoRetiro (String):** Valor de estado de retiro (pendiente por uso en métodos posteriores).
- **programa (String):** Código de programa utilizado en filtros y validaciones (GEST5400).

Métodos Clave

cambiarEstado(String estado)

Modifica el estado del proyecto seleccionado dependiendo de la lógica definida para cada estado:

- En estado **REVISADO**: genera el código del proyecto basado en el código de la entidad y la convocatoria.

- En estado **DILIGENCIANDO**: valida que la convocatoria esté activa y limpia la fecha de envío.
- Otros estados: permite el cambio directo.

verificarProyectoPrograma()

Evalúa si un proyecto debe seguir tratándose como un formulario antiguo, según condiciones como:

- Fecha de creación
- Tipo de exención
- Programa asociado a la convocatoria

isBeforeToday(Date d)

Devuelve un texto indicando si la fecha dada corresponde a:

- Hoy
- Ayer
- N días atrás

convocatoriaActiva()

Verifica si una convocatoria está activa, evaluando:

- Fecha de cierre
- Estado de publicación
- Estado de activación

Carga y Filtros

- lazyLoad(), lazyLoadReg()
Recargan los datos de proyectos aplicando el filtro correspondiente (activo o registro).

- `limpiarFiltros()`, `limpiarFiltrosReg()`
Limpian los filtros de búsqueda de los proyectos y tipo de trámite.
- `filtrarHijos(...)`, `filtrarHijosReg(...)`
Filtran los proyectos hijos asociados al proyecto padre seleccionado, utilizando exención tipo H.
- `lazyLoadSinFiltroActivo()`
Carga datos sin aplicar filtros activos.
- `cargarProgramas()`
Inicializa los proyectos con exención tipo R (registro), sin anteproyecto, para filtrado.

Operaciones Adicionales

`cargarDescripcionesProyecto()`

Consulta y asigna las descripciones del proyecto seleccionado al ViewController.

`cargarDetallesRubrosProyecto(Long rubroId)`

Obtiene los detalles del rubro presupuestal de un proyecto específico, filtrando por `projectId`, `rubroId` y `tipoExencion`.

`calcularTotalRubroConvocatoria(Long cuadroPresupuestId)`

Calcula el total asignado a un rubro específico en convocatorias antiguas.

`crearReporteMarcoLogico(ProyectoDTO proyectoSelected)`

Genera y descarga un reporte en Excel (XLS/XLSX) con la estructura del marco lógico del proyecto seleccionado.

Observaciones de Diseño

- Se hace uso intensivo del patrón de inyección de dependencias (`@Inject`, `@EJB`) para mantener el controlador limpio y desacoplado.

- El controlador está diseñado para manejar tanto proyectos antiguos como registros activos, diferenciados por banderas (filtroActivo, filtroActivoReg).
- Se integra con componentes JSF (DataTable) para recarga de datos con lazy loading.

Este controlador es responsable de gestionar la información relacionada con la evaluación de proyectos antiguos, incluyendo datos personales, documentos y reportes.

1. Carga de información personal del investigador

- El método **cargarInfoPersonal** determina si se debe consultar la información desde CvLAC (registroCvLac == 'S') o desde la base de datos local.
- Si no se usa CvLAC, se carga el documento de hoja de vida almacenado.
- **consultarInformacionCvLac** obtiene datos poblacionales y de enfoque diferencial desde SCIENTI.

2. Hoja de vida

- **obtenerUrlHojaVidaCvLAC**: construye la URL del CvLAC de la persona, si es evaluador registrado.
- **descargarArchivoHojaVida**: descarga el archivo cargado como hoja de vida (no-CvLAC).

3. Evaluaciones

- **prepararEditar**: vuelve a cargar los datos actualizados de una evaluación seleccionada.
- **obtenerDocumentoEvaluacion**: recupera el archivo asociado a una evaluación.
- **descargarArchivoEvaluacion**: permite descargar el archivo PDF de la evaluación.

4. Reporte de Evaluación

- **downloadReporteEvaluacion**: genera y descarga un reporte PDF de evaluación o de calificación usando ReporteUtil.
 - Incluye logo parametrizado (LOGIN_LOGO).
 - Usa parámetros como PROYECTO_ID y evaluacion_proyecto_id.

5. Descarga de documentos

- **descargarArchivoDocumentos:** permite descargar cualquier documento adjunto al proyecto, como soportes.

6. Dependencia y evaluador

- **consultarProgramaPadre:** obtiene el nombre del programa padre del proyecto.
- **consultarValorTipoEvaluacion:** devuelve la descripción asociada a un tipo de evaluación.
- **nombreCompletoPersona:** arma el nombre completo del evaluador según la fuente (local o SCIENTI).
- **consultarDependencia:** obtiene el nombre de la dependencia académica asociada a una convocatoria.

Observaciones Técnicas

- Usa EJBs como `proyectoServiceFacade`, `scientiServiceFacade`, y `convocatoriaServiceFacade` para acceder a la lógica de negocio.
- Controla errores con logs (`LOG.error(...)`) y muestra mensajes con `FacesMessage` en el contexto JSF.
- Emplea clases DTO (`ProyectoPersonaDTO`, `ScientiPersona`, `DocumentoDTO`, `ProyectoEvaluacionDTO`) para el transporte de datos.

*Se definen métodos de consulta en Facade EJB (**ProyectosServiceFacade.java**) para funcionalidad de pantalla*

```
//  
  
//PROYECTOS  
/**  
 * Consulta el listado de proyectos y programas registrados en la base de  
 * datos  
 *  
 * @param first indica desde donde empieza la consulta  
 * @param pageSize indica el final de la consulta (first + pageSize)  
 * @param sortField indica el campo por el que se realiza el ordenamiento  
 * @param sortOrder indica el orden (ascendente o descendente)
```

```

    * @param filters contiene los filtros al momento de solicitar el listado de
    * proyectos y programas
    * @return lista de proyectos y programas encontrados en el rango enviado y
    * basado en los filtros enviados
    * @throws Exception
    */
    List<ProyectoDTO> proyectosList(int first, int pageSize, String sortField, String
sortOrder, Map<String, Object> filters) throws Exception;

    /**
    * Cuenta la cantidad de proyectos y programas actualmente registrados en la
    * base de datos
    *
    * @param filters
    * @return
    * @throws Exception
    */
    int proyectosListCount(Map<String, Object> filters) throws Exception;

```

*Se implementan métodos consulta en facade EJB (**ProyectosFacadeServiceBean.java**) para funcionalidad de pantalla*

```

//----- Proyectos y programas -----//
@Override
public List<ProyectoDTO> proyectosList(int first, int pageSize, String sortField, String
sortOrder, Map<String, Object> filters) throws Exception {
    ObjectConverter<ProyectoDTO, Proyecto> converter =
objectConverterFactory.checkOut();
    List<ProyectoDTO> listProyectoDTOS =
converter.toTransferList(proyectosService.proyectosList(first, pageSize, sortField,
sortOrder, filters), ProyectoDTO.class);
    objectConverterFactory.checkIn(converter);
    return listProyectoDTOS;
}

@Override
public int proyectosListCount(Map<String, Object> filters) throws Exception {
    ProyectoDTO proyectoFilterDTO = (ProyectoDTO) filters.get("proyectoFiltro");
    ObjectConverter<ProyectoDTO, Proyecto> converter =
objectConverterFactory.checkOut();
    Proyecto proyectoFiltro = converter.toEntity(proyectoFilterDTO, Proyecto.class);
    filters.put("proyectoFiltro", proyectoFiltro);
    return proyectosService.proyectosListCount(filters);
}

```

*Se ajusta interfaz **ProyectoService.java** que define métodos de consulta en Service EJB para funcionalidad de pantalla*

```

//----- Proyectos y Programas -----//
List<Proyecto> proyectosList(int first, int pageSize, String sortField, String
sortOrder, Map<String, Object> filters) throws SystemException;

int proyectosListCount(Map<String, Object> filters) throws SystemException;

```


*Se ajusta **ProyectoServiceBean.java** donde implementan métodos de consulta en Service EJB para funcionalidad de pantalla*

```
//----- Proyectos y Programas -----//
@Override
public List<Proyecto> proyectosList(int first, int pageSize, String sortField, String
sortOrder, Map<String, Object> filters) throws SystemException {
    return this.getStoreManager(ProyectosStoreManager.class).proyectosList(first,
pageSize, sortField, sortOrder, filters);
}

@Override
public int proyectosListCount(Map<String, Object> filters) throws SystemException {
    return
this.getStoreManager(ProyectosStoreManager.class).proyectosListCount(filters);
}
```

Internacionalización en español e inglés

```
#ES
##### Proyectos y Programas ANTIGUOS
#####

gestion.proyectosProgramas.proyectosAntiguos.header =
Proyectos y Programas Antiguos
gestion.proyectosProgramas.proyectosAntiguos.breadcrum =
Proyectos Antiguos
gestion.proyectosProgramas.proyectosAntiguos.title =
Proyectos Antiguos
gestion.proyectosProgramas.proyectosAntiguos.formularioProyectoId =
Formulario Proyecto ID
gestion.proyectosProgramas.proyectosAntiguos.codigoProyecto =
Código Registro
gestion.proyectosProgramas.proyectosAntiguos.codigoProyecto.placeholder =
Ingrese el Código del Registro
gestion.proyectosProgramas.proyectosAntiguos.codigoProyectoId =
Código Proyecto
gestion.proyectosProgramas.proyectosAntiguos.codigoProyectoId.placeholder =
Ingrese el Código del Proyecto
gestion.proyectosProgramas.proyectosAntiguos.titulo =
Titulo Proyecto
gestion.proyectosProgramas.proyectosAntiguos.titulo.placeholder =
Ingrese el Titulo del Proyecto
gestion.proyectosProgramas.proyectosAntiguos.tipoProyecto = Tipo
Proyecto
gestion.proyectosProgramas.proyectosAntiguos.anioRegistro = Año
Registro
gestion.proyectosProgramas.proyectosAntiguos.fechaRegistro = Fecha
de Registro
gestion.proyectosProgramas.proyectosAntiguos.fechaEnvio = Fecha
de Envío
gestion.proyectosProgramas.proyectosAntiguos.fechaRevision = Fecha
Revisión
```

gestion.proyectosProgramas.proyectosAntiguos.tipoTramite	=
Trámite	
gestion.proyectosProgramas.proyectosAntiguos.convocatoria	=
Convocatoria	
gestion.proyectosProgramas.proyectosAntiguos.programa	= Prog
Nacional de CTeI	
gestion.proyectosProgramas.proyectosAntiguos.programa1	=
Programa Nacional 1	
gestion.proyectosProgramas.proyectosAntiguos.programa2	=
Programa Nacional 2	
gestion.proyectosProgramas.proyectosAntiguos.estado	=
Estado	
gestion.proyectosProgramas.proyectosAntiguos.estado.proy	=
Estado Proyecto	
gestion.proyectosProgramas.proyectosAntiguos.entidad	=
Entidad Ejecutora	
gestion.proyectosProgramas.proyectosAntiguos.lugar	= Lugar
de Ejecución	
gestion.proyectosProgramas.proyectosAntiguos.fechaModificacion	=
Ultima Modificación	
gestion.proyectosProgramas.proyectosAntiguos.fechaCreacion	= Fecha
Creación	
gestion.proyectosProgramas.proyectosAntiguos.financiacion	= Tipo
Financiación	
gestion.proyectosProgramas.proyectosAntiguos.duracion	=
Duración	
gestion.proyectosProgramas.proyectosAntiguos.ejecucion	=
Ejecución	
gestion.proyectosProgramas.proyectosAntiguos.meses	= Meses
gestion.proyectosProgramas.proyectosAntiguos.palabras	=
Palabras Claves	
gestion.proyectosProgramas.proyectosAntiguos.montos	=
Montos del proyecto	
gestion.proyectosProgramas.proyectosAntiguos.montosProg	=
Montos del programa	
gestion.proyectosProgramas.proyectosAntiguos.montosFicTec	=
Montos de la ficha Técnica	
gestion.proyectosProgramas.proyectosAntiguos.requisitos	=
Requisitos Adjuntos	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.btn	=
Requisitos	
gestion.proyectosProgramas.proyectosAntiguos.especie	=
Especie	
gestion.proyectosProgramas.proyectosAntiguos.dinero	=
Dinero	
gestion.proyectosProgramas.proyectosAntiguos.solicitado	=
Solicitado	
gestion.proyectosProgramas.proyectosAntiguos.total	= Total
gestion.proyectosProgramas.proyectosAntiguos.radicadoBtn	=
Cambiar a Radicado	
gestion.proyectosProgramas.proyectosAntiguos.rechazarBtn	=
Rechazar	
gestion.proyectosProgramas.proyectosAntiguos.evaluacionBtn	=
Enviar a Evaluación	
gestion.proyectosProgramas.proyectosAntiguos.registro.requisitos.error	= Debe
revisar los requisitos antes de pasar el proyecto a la etapa de evaluación	
gestion.proyectosProgramas.proyectosAntiguos.registro.requisitos.error2	= No se
puede pasar el proyecto a la etapa de evaluación porque no cumple uno o más requisitos	
obligatorios	
gestion.proyectosProgramas.proyectosAntiguos.pasarDiligenciado	= Pasar
a Diligenciado	

gestion.proyectosProgramas.proyectosAntiguos.pasarDecision	=
Enviar a Decisión	
gestion.proyectosProgramas.proyectosAntiguos.devolverEvaluacion	=
Devolver a Evaluación	
gestion.proyectosProgramas.proyectosAntiguos.devolverSecretaria	=
Devolver a la Secretaría Técnica CNBT	
gestion.proyectosProgramas.proyectosAntiguos.devolverRegistro	=
Devolver a Registro	
gestion.proyectosProgramas.proyectosAntiguos.subsanable	=
Subsanable	
gestion.proyectosProgramas.proyectosAntiguos.agregarObservacion	=
Agregar Observación	
gestion.proyectosProgramas.proyectosAntiguos.editarObservacion	=
Editar Observación	
gestion.proyectosProgramas.proyectosAntiguos.hoy	= Hoy
gestion.proyectosProgramas.proyectosAntiguos.ayer	= Ayer
gestion.proyectosProgramas.proyectosAntiguos.diasAtras	= Días
atrás a las	
gestion.proyectosProgramas.proyectosAntiguos.fechaObservacion	= Fecha
Observación	
gestion.proyectosProgramas.proyectosAntiguos.fechaObservacion.required	= Debe
Seleccionar Fecha Observación	
#Requisitos	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.header	=
Requisitos	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.obligatorio.msg	=
Recuerde que los requisitos que tengan asterisco (*) son requeridos.	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.requisito.label	=
Requisito	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.nombre.archivo.label	=
Nombre Archivo	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.tamano.label	=
Tamaño	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.progreso.label	=
Progreso	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.progreso.ok	=
DOCUMENTO CARGADO	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.cumple.label	=
Verificación Requisito	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.cumple.required	=
Debe ingresar el tipo de cumplimiento del requisito	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.observaciones.label	=
Observaciones	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.certificado.error	=
Debe adjuntar todos los requisitos requeridos	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.vacio	= No
hay Adjunto	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.vacio.mensaje	=
Debe adjuntar un archivo para poder agregarle una observación	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.concepto.label	=
Concepto panel de Evaluación	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.puntaje.label	=
Puntaje	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.elegible.label	=
Elegible	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.boton.generar	=
Generar Certificado	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.fechaCumplimiento.label	=
Fecha Cumplimiento	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.modal.label	=
Cumplimiento del requisito	

gestion.proyectosProgramas.proyectosAntiguos.requisitos.tooltip.cumplimiento Cumplimiento del requisito	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.irDetallesPry.btn a Detalles	= Ir
gestion.proyectosProgramas.proyectosAntiguos.observaciones.requisitos.title Observaciones Generales de los Requisitos	=
gestion.proyectosProgramas.proyectosAntiguos.observaciones.requisitos.label Observaciones	=
#pestañas	
gestion.proyectosProgramas.proyectosAntiguos.tab.infoBasica Básica	= Info
gestion.proyectosProgramas.proyectosAntiguos.tab.observaciones Observaciones	=
gestion.proyectosProgramas.proyectosAntiguos.tab.revisionEvaluacion Evaluación	=
gestion.proyectosProgramas.proyectosAntiguos.tab.entidades Entidades	=
gestion.proyectosProgramas.proyectosAntiguos.tab.rolEntidades Entidades	= Roles
gestion.proyectosProgramas.proyectosAntiguos.tab.presupuesto Presupuesto	=
gestion.proyectosProgramas.proyectosAntiguos.tab.descripciones Descripciones	=
gestion.proyectosProgramas.proyectosAntiguos.tab.impactos Impactos	=
gestion.proyectosProgramas.proyectosAntiguos.tab.coberturas Coberturas	=
gestion.proyectosProgramas.proyectosAntiguos.coberturas.label Coberturas	=
gestion.proyectosProgramas.proyectosAntiguos.tab.productos Productos	=
gestion.proyectosProgramas.proyectosAntiguos.tab.cronograma Actividades Cronograma	=
gestion.proyectosProgramas.proyectosAntiguos.tab.tipoPersonal de Personal	= Tipo
gestion.proyectosProgramas.proyectosAntiguos.tab.personal Personal	=
gestion.proyectosProgramas.proyectosAntiguos.tab.marcoLogico Lógico	= Marco
gestion.proyectosProgramas.proyectosAntiguos.tab.documentos Documentos	=
#	
gestion.proyectosProgramas.proyectosAntiguos.entidades.empty encontraron Entidades	= No se
gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.empty encontraron Rol Entidades	= No se
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.empty encontraron Presupuesto	= No se
gestion.proyectosProgramas.proyectosAntiguos.impactos.empty encontraron Impactos	= No se
gestion.proyectosProgramas.proyectosAntiguos.coberturas.empty encontraron Coberturas	= No se
gestion.proyectosProgramas.proyectosAntiguos.productos.empty encontraron Productos	= No se
gestion.proyectosProgramas.proyectosAntiguos.cronograma.empty encontraron Actividades Cronograma	= No se
gestion.proyectosProgramas.proyectosAntiguos.personal.empty encontraron Personal	= No se
gestion.proyectosProgramas.proyectosAntiguos.marcoLogico.empty encontraron Marco Lógico	= No se

gestion.proyectosProgramas.proyectosAntiguos.descripcion.empty	= No se
encontraron Descripciones	
#	
gestion.proyectosProgramas.proyectosAntiguos.column.convocatoria	=
Convocatoria	
gestion.proyectosProgramas.proyectosAntiguos.column.programa	=
Programa	
gestion.proyectosProgramas.proyectosAntiguos.column.tramite	=
Trámite	
gestion.proyectosProgramas.proyectosAntiguos.column.fichaProyecto	= Ficha
Proyecto	
gestion.proyectosProgramas.proyectosAntiguos.column.fichaPrograma	= Ficha
Programa	
gestion.proyectosProgramas.proyectosAntiguos.column.búsquedaAvanzada	=
Búsqueda Avanzada	
gestion.proyectosProgramas.proyectosAntiguos.reporteRubrosPry	=
Reporte Rubros y Proyectos	
gestion.proyectosProgramas.proyectosAntiguos.reporteMarcoLogico	=
Reporte Marco Lógico	
gestion.proyectosProgramas.proyectosAntiguos.reporteEvaluacion	=
Reporte Evaluación	
gestion.proyectosProgramas.proyectosAntiguos.nivelMadurezTecnologica	= Nivel
de madurez tecnológica	
gestion.proyectosProgramas.proyectosAntiguos.tipoPersona	= Tipo
persona	
gestion.proyectosProgramas.proyectosAntiguos.CiudadFormacion	=
Ciudad de formación	
gestion.proyectosProgramas.proyectosAntiguos.valorInscripcion	= Valor
de la inscripción	
gestion.proyectosProgramas.proyectosAntiguos.viaPresentacion	= Vía
de Presentación	
gestion.proyectosProgramas.proyectosAntiguos.origenInvencion	=
Información de Titularidad	
gestion.proyectosProgramas.proyectosAntiguos.vinculoLaboral	=
Vínculo Laboral	
#Rol Entidades	
gestion.proyectosProgramas.proyectosAntiguos.entidades.entidad	=
Entidad	
gestion.proyectosProgramas.proyectosAntiguos.entidades.nit	= NIT
#Rol Entidades	
gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.entidad	=
Entidad	
gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.rol	= Rol
#Descripciones	
gestion.proyectosProgramas.proyectosAntiguos.tabla.orden	= Orden
gestion.proyectosProgramas.proyectosAntiguos.tabla.descripcion	=
Descripción	
gestion.proyectosProgramas.proyectosAntiguos.cargando	=
CARGANDO DESCRIPCIONES...	
gestion.proyectosProgramas.proyectosAntiguos.btn.ver	= Ver
descripción	
gestion.proyectosProgramas.proyectosAntiguos.descripcion.header	=
Descripción	
#Impactos	
gestion.proyectosProgramas.proyectosAntiguos.impacto.label	=
= Impactos	
gestion.proyectosProgramas.proyectosAntiguos.impacto.tipo	=
= Tipo Impacto	
gestion.proyectosProgramas.proyectosAntiguos.impacto.descripcion	=
= Descripción	

```

gestion.proyectosProgramas.proyectosAntiguos.impacto.año
= Año
gestion.proyectosProgramas.proyectosAntiguos.impacto.empty
= No se encontraron Impactos para este proyecto
#Cronograma
gestion.proyectosProgramas.proyectosAntiguos.cronograma.label
= Actividades Cronograma
gestion.proyectosProgramas.proyectosAntiguos.cronograma.empty
= No se encontraron Actividades
gestion.proyectosProgramas.proyectosAntiguos.cronograma.nActividad
= N° Actividad
gestion.proyectosProgramas.proyectosAntiguos.cronograma.actividad
= Actividad
gestion.proyectosProgramas.proyectosAntiguos.cronograma.inicio
= Inicio
gestion.proyectosProgramas.proyectosAntiguos.cronograma.fin
= Fin
gestion.proyectosProgramas.proyectosAntiguos.cronograma.ejecucion
= Ejecución En
#Tipo de Personal
gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.label
= Tipo de Personal
gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.empty
= No se encontraron tipos de personal
gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.tipoPersonal
= Rol en el Proyecto
gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.numeroPersonas
= Número Personas
#Personal
gestion.proyectosProgramas.proyectosAntiguos.personal.label
= Personal
gestion.proyectosProgramas.proyectosAntiguos.personal.empty
= No se encontraron Personas Registradas
gestion.proyectosProgramas.proyectosAntiguos.personal.persona
= Persona
gestion.proyectosProgramas.proyectosAntiguos.personal.tipo
= Vinculación Proyecto
gestion.proyectosProgramas.proyectosAntiguos.personal.identificacion
= N° Identificación
gestion.proyectosProgramas.proyectosAntiguos.personal.entidad
= Entidad
gestion.proyectosProgramas.proyectosAntiguos.personal.vinculo
= Vinculo
gestion.proyectosProgramas.proyectosAntiguos.personal.fechaNacimiento
= Fecha Nacimiento
gestion.proyectosProgramas.proyectosAntiguos.personal.pais
= País
gestion.proyectosProgramas.proyectosAntiguos.personal.email
= Email
gestion.proyectosProgramas.proyectosAntiguos.personal.tipoIdentificación
= Tipo Identificación
gestion.proyectosProgramas.proyectosAntiguos.personal.numeroIdentificación
= Número Identificación
gestion.proyectosProgramas.proyectosAntiguos.personal.genero
= Género
gestion.proyectosProgramas.proyectosAntiguos.personal.dedicacion
= Dedicación (h/semana)
gestion.proyectosProgramas.proyectosAntiguos.personal.nMeses
= Número Meses
gestion.proyectosProgramas.proyectosAntiguos.personal.funcion
= Función en el proyecto

```

```

gestion.proyectosProgramas.proyectosAntiguos.personal.registradoCvLAC
= ¿Registrado en CvLAC?
gestion.proyectosProgramas.proyectosAntiguos.personal.documentoHojaVida
= Documento Adjunto Hoja de Vida
gestion.proyectosProgramas.proyectosAntiguos.personal.poblacionVictimaConflicto
= Población Víctima del Conflicto Armado
gestion.proyectosProgramas.proyectosAntiguos.personal.poblacionCondicionDisc
= Población en Situación de Discapacidad
gestion.proyectosProgramas.proyectosAntiguos.personal.grupoEtnico
= Grupo Étnico
gestion.proyectosProgramas.proyectosAntiguos.personal.grupoEtario
= Grupo Etario
gestion.proyectosProgramas.proyectosAntiguos.personal.categoria
= Categoría
#Marco Logico
gestion.proyectosProgramas.proyectosAntiguos.marco.label
= Marco Lógico
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoGral
= Objetivo General
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoGeneral.empty
= No se encontró Objetivo General
gestion.proyectosProgramas.proyectosAntiguos.marco.meta
= Meta
gestion.proyectosProgramas.proyectosAntiguos.marco.fuenteVerificacion
= Fuente Verificación
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivosEspecificos.label
= Objetivos Específicos
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivosEspecificos.empty
= No se encontraron Objetivos Específicos
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.label
= Objetivo Específico N°
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.objetivo.label
= Objetivo Específico
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.meta.label
= Meta
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.fuente.label
= Fuente de verificación
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.indicador.label
= Indicador
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.actividades.empty
= Este Objetivo no tiene Actividades
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.resultados.empty
= Esta Actividad no tiene Resultados
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.indicadores.empty
= Este Resultado no tiene Indicadores
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.periodoInicial.label
= Periodo Inicial
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.periodoFinal.label
= Periodo Final
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.duracion.label
= Duración
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.medioVerif.label
= Medio Verificación
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.mesCumpli.label
= Mes de Cumplimiento
gestion.proyectosProgramas.proyectosAntiguos.marco.resultados.label
= Actividades, Resultados e Indicadores
#Presupuesto
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tab
= Presupuesto

```



```

gestion.proyectosProgramas.proyectosAntiguos.presupuesto.loading
= CARGANDO RUBROS
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.rubro.header
= Rubro
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.monInstitucionSol.header
= Monto financiado solicitado
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEspecieSol.header
= Contrapartida especie solicitado
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEfectivoSol.header
= Contrapartida efectivo solicitado
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.totalSol.header
= Total solicitado
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.monInstitucionAprob.header
= Monto financiado aprobado
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEspecieAprob.header
= Contrapartida especie aprobado
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEfectivoAprob.header
= Contrapartida efectivo aprobado
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.totalAprob.header
= Total aprobado

gestion.proyectosProgramas.proyectosAntiguos.presupuesto.entidad
= Entidad
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.anio
= Año
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tipoPresupuesto
= Tipo Presupuesto
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.modalidad
= Modalidad
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.financiado
= Financiado
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.contrapartida
= Contrapartida
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.especie
= Especie
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.efectivo
= Efectivo
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tipoFinanciacion
= Tipo Financiación
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.propuesto
= Monto Propuesto
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.aprobado
= Monto Aprobado
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.rubro
= Rubro
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.nDias
= # Días
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.nPersonas
= # Personas
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.cantidad
= Cantidad
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.costoPersona
= Costo por Persona
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.valorUnitario
= Valor Unitario
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tema
= Tema
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.descripcion
= Descripción
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.lugarViaje
= Lugar de viaje

```



```

gestion.proyectosProgramas.proyectosAntiguos.presupuesto.lugarJustificacion
= Lugar / Justificación
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.justificacion
= Justificación
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.nombres
= Nombres y Apellidos
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.funcion
= Función
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.dedicacionMeses
= Dedicación Meses
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.dedicacionSemanas
= Dedicación Semanas
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.proveedor
= Proveedor
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.institucion
= Institución Financiadora
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tipoFinanciacion
= Tipo Financiación
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.total
= Total
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.modificar
= Modificar Rubros
gestion.proyectosProgramas.proyectosAntiguos.ficha.loading
= Cargando Ficha, espere por favor...
#Evaluacion
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.emptyTable
= No se encontraron Evaluadores Asignados
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.infoGeneral
= Información General
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.tipoEvaluacion
= Tipo Evaluación
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.estadoEvaluador
= Estado Evaluador
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.panel
= Panel de Evaluación
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.panel.nombrePanel
= Nombre Panel
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.panel.actaPanel
= Acta panel
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.infoEvaluador
= Información Evaluador
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.tipoEvaluador
= Tipo Evaluador
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.dependencia
= Dependencia
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.funcionario
= Funcionario
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.nombres
= Nombres
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.usuarioSigp
= Usuario SIGP
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.evaluacion
= Evaluación
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.fechaAsignacionEvaluador
= Fecha Asignación Evaluador
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.fechaEvaluacion
= Fecha Evaluación
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.duracionEvaluacion
= Duración Evaluación (días)
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.conceptoFinal
= Concepto Final

```

```
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.recomenFinal
= Recomendación Final
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.concepto
= Concepto
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.recomendacionEvaluador
= Recomendación Evaluador
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.puntaje
= Puntaje
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.documentoEvaluacion
= Documento Evaluación
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.pago
= Pago
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.valorEvaluacion
= Valor Evaluación
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.fechaPago
= Fecha Pago
#Documentos
gestion.proyectosProgramas.proyectosAntiguos.documentos.emptyTable
= No se encontraron Documentos asociados
#Observaciones
gestion.proyectosProgramas.proyectosAntiguos.observaciones.emptyTable
= No se encontraron Observaciones
```

```
#EN
##### Proyectos y Programas ANTIGUOS
#####

gestion.proyectosProgramas.proyectosAntiguos.header = Old
Projects and Programs
gestion.proyectosProgramas.proyectosAntiguos.breadcrum = Old
Projects
gestion.proyectosProgramas.proyectosAntiguos.title = Old
Projects
gestion.proyectosProgramas.proyectosAntiguos.formularioProyectoId =
Project ID Form
gestion.proyectosProgramas.proyectosAntiguos.codigoProyecto =
Registration Code
gestion.proyectosProgramas.proyectosAntiguos.codigoProyecto.placeholder = Enter
the Registration Code
gestion.proyectosProgramas.proyectosAntiguos.codigoProyectoId =
Project Code
gestion.proyectosProgramas.proyectosAntiguos.codigoProyectoId.placeholder = Enter
the Project Code
gestion.proyectosProgramas.proyectosAntiguos.titulo = Title
Project
gestion.proyectosProgramas.proyectosAntiguos.titulo.placeholder = Enter
the Project Title
gestion.proyectosProgramas.proyectosAntiguos.tipoProyecto = Type
Project
gestion.proyectosProgramas.proyectosAntiguos.anioRegistro = Year
of Registration
gestion.proyectosProgramas.proyectosAntiguos.fechaRegistro =
Registration Date
gestion.proyectosProgramas.proyectosAntiguos.fechaEnvio =
Shipping Date
gestion.proyectosProgramas.proyectosAntiguos.fechaRevision = Date
Review
```

gestion.proyectosProgramas.proyectosAntiguos.tipoTramite Procedure	=
gestion.proyectosProgramas.proyectosAntiguos.convocatoria	= Call
gestion.proyectosProgramas.proyectosAntiguos.programa National Prog of CTeI	=
gestion.proyectosProgramas.proyectosAntiguos.programa1 National Program 1	=
gestion.proyectosProgramas.proyectosAntiguos.programa2 National Program 2	=
gestion.proyectosProgramas.proyectosAntiguos.estado	= State
gestion.proyectosProgramas.proyectosAntiguos.estado.proy Project Status	=
gestion.proyectosProgramas.proyectosAntiguos.entidad Executing Entity	=
gestion.proyectosProgramas.proyectosAntiguos.lugar of Execution	= Place
gestion.proyectosProgramas.proyectosAntiguos.fechaModificacion Modification	= Last
gestion.proyectosProgramas.proyectosAntiguos.fechaCreacion Creation Date	=
gestion.proyectosProgramas.proyectosAntiguos.financiacion of financing	= Type
gestion.proyectosProgramas.proyectosAntiguos.duración Duration	=
gestion.proyectosProgramas.proyectosAntiguos.ejecucion Execution	=
gestion.proyectosProgramas.proyectosAntiguos.meses Months	=
gestion.proyectosProgramas.proyectosAntiguos.palabras Keywords	=
gestion.proyectosProgramas.proyectosAntiguos.montos Project amounts	=
gestion.proyectosProgramas.proyectosAntiguos.montosProg Amounts of the program	=
gestion.proyectosProgramas.proyectosAntiguos.montosFicTec Amounts of the technical file	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos Attached Requirements	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.btn Requirements	=
gestion.proyectosProgramas.proyectosAntiguos.especie Species	=
gestion.proyectosProgramas.proyectosAntiguos.dinero	= Money
gestion.proyectosProgramas.proyectosAntiguos.solicitado Requested	=
gestion.proyectosProgramas.proyectosAntiguos.total	= Total
gestion.proyectosProgramas.proyectosAntiguos.radicadoBtn Switch to filed	=
gestion.proyectosProgramas.proyectosAntiguos.rechazarBtn Reject	=
gestion.proyectosProgramas.proyectosAntiguos.evaluacionBtn to Evaluation	= Send
gestion.proyectosProgramas.proyectosAntiguos.registro.requisitos.error must review the requirements before moving the project to the evaluation stage	= You
gestion.proyectosProgramas.proyectosAntiguos.registro.requisitos.error2 project can not be passed to the evaluation stage because it does not meet one or more mandatory requirements	= The
gestion.proyectosProgramas.proyectosAntiguos.pasarDiligenciado Diligence	= Go to
gestion.proyectosProgramas.proyectosAntiguos.pasarDecision to Decision	= Send

gestion.proyectosProgramas.proyectosAntiguos.devolverEvaluacion Return to Evaluation	=
gestion.proyectosProgramas.proyectosAntiguos.devolverSecretaria Return to the Technical Secretariat CNBT	=
gestion.proyectosProgramas.proyectosAntiguos.devolverRegistro Return to Register	=
gestion.proyectosProgramas.proyectosAntiguos.subsanable Subsanable	=
gestion.proyectosProgramas.proyectosAntiguos.agregarObservacion Observation	= Add
gestion.proyectosProgramas.proyectosAntiguos.editarObservacion Observation	= Edit
gestion.proyectosProgramas.proyectosAntiguos.hoy	= Today
gestion.proyectosProgramas.proyectosAntiguos.ayer Yesterday	=
gestion.proyectosProgramas.proyectosAntiguos.diasAtras ago at	= Days
gestion.proyectosProgramas.proyectosAntiguos.fechaObservacion Observation	= Date
gestion.proyectosProgramas.proyectosAntiguos.fechaObservacion.required Must Select Date Observation	= You
#Requisitos	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.header Requirements	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.obligatorio.msg Remember that requirements with an asterisk (*) are required.	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.requisito.label Requirement	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.nombre.archivo.label File Name	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.tamano.label Size	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.progreso.label Progress	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.progreso.ok UPLOADED DOCUMENT	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.cumple.label Verification Requirement	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.cumple.required must enter the type of compliance with the requirement	= You
gestion.proyectosProgramas.proyectosAntiguos.requisitos.observaciones.label Observations	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.certificado.error must attach all the required requirements	= You
gestion.proyectosProgramas.proyectosAntiguos.requisitos.vacio Attachment	= No
gestion.proyectosProgramas.proyectosAntiguos.requisitos.vacio.mensaje must attach a file to be able to add an observation	= You
gestion.proyectosProgramas.proyectosAntiguos.requisitos.concepto.label Evaluation panel concept	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.puntaje.label Score	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.elegible.label Eligible	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.boton.generar Generate Certificate	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.fechaCumplimiento.label Date Compliance	=
gestion.proyectosProgramas.proyectosAntiguos.requisitos.modal.label Compliance with the requirement	=

gestion.proyectosProgramas.proyectosAntiguos.requisitos.tooltip.cumplimiento	=
Compliance with the requirement	
gestion.proyectosProgramas.proyectosAntiguos.requisitos.irDetallesPry.btn	= Go
to Details	
gestion.proyectosProgramas.proyectosAntiguos.observaciones.requisitos.title	=
General Comments of the Requirements	
gestion.proyectosProgramas.proyectosAntiguos.observaciones.requisitos.label	=
Observations	
#pestañas	
gestion.proyectosProgramas.proyectosAntiguos.tab.infoBasica	= Basic
Info	
gestion.proyectosProgramas.proyectosAntiguos.tab.observaciones	=
Observations	
gestion.proyectosProgramas.proyectosAntiguos.tab.revisionEvaluacion	=
Evaluation	
gestion.proyectosProgramas.proyectosAntiguos.tab.entidades	=
Entities	
gestion.proyectosProgramas.proyectosAntiguos.tab.rolEntidades	=
Entities Roles	
gestion.proyectosProgramas.proyectosAntiguos.tab.presupuesto	=
Budget	
gestion.proyectosProgramas.proyectosAntiguos.tab.descripciones	=
Descriptions	
gestion.proyectosProgramas.proyectosAntiguos.tab.impactos	=
Impacts	
gestion.proyectosProgramas.proyectosAntiguos.tab.coberturas	=
Coverages	
gestion.proyectosProgramas.proyectosAntiguos.coberturas.label	=
Coverages	
gestion.proyectosProgramas.proyectosAntiguos.tab.productos	=
Products	
gestion.proyectosProgramas.proyectosAntiguos.tab.cronograma	=
Activities Schedule	
gestion.proyectosProgramas.proyectosAntiguos.tab.tipoPersonal	= Type
of Staff	
gestion.proyectosProgramas.proyectosAntiguos.tab.personal	=
Personal	
gestion.proyectosProgramas.proyectosAntiguos.tab.marcoLogico	=
Logical Framework	
gestion.proyectosProgramas.proyectosAntiguos.tab.documentos	=
Documents	
#	
gestion.proyectosProgramas.proyectosAntiguos.entidades.empty	= No
Entities found	
gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.empty	= No
Entities Roles found	
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.empty	= No
Budget found	
gestion.proyectosProgramas.proyectosAntiguos.impactos.empty	= No
Impacts found	
gestion.proyectosProgramas.proyectosAntiguos.coberturas.empty	= No
Coverages found	
gestion.proyectosProgramas.proyectosAntiguos.productos.empty	= No
Products found	
gestion.proyectosProgramas.proyectosAntiguos.cronograma.empty	= No
Activities Schedule found	
gestion.proyectosProgramas.proyectosAntiguos.personal.empty	= No
Personnel found	
gestion.proyectosProgramas.proyectosAntiguos.marcoLogico.empty	= No
Logical Framework found	

gestion.proyectosProgramas.proyectosAntiguos.descripcion.empty	= No
descriptions found	
#	
gestion.proyectosProgramas.proyectosAntiguos.column.convocatoria	= Call
gestion.proyectosProgramas.proyectosAntiguos.column.programa	=
Program	
gestion.proyectosProgramas.proyectosAntiguos.column.tramite	=
Procedure	
gestion.proyectosProgramas.proyectosAntiguos.column.fichaProyecto	=
Project tab	
gestion.proyectosProgramas.proyectosAntiguos.column.fichaPrograma	=
Program tab	
gestion.proyectosProgramas.proyectosAntiguos.column.busquedaAvanzada	=
Advanced Search	
gestion.proyectosProgramas.proyectosAntiguos.reporteRubrosPry	= Items
and Projects Report	
gestion.proyectosProgramas.proyectosAntiguos.reporteMarcoLogico	=
Logical Framework Report	
gestion.proyectosProgramas.proyectosAntiguos.reporteEvaluacion	=
Evaluation Report	
gestion.proyectosProgramas.proyectosAntiguos.nivelMadurezTecnologica	= Level
of technological maturity	
gestion.proyectosProgramas.proyectosAntiguos.tipoPersona	= Type
person	
gestion.proyectosProgramas.proyectosAntiguos.CiudadFormacion	=
Formation city	
gestion.proyectosProgramas.proyectosAntiguos.valorInscripcion	=
Registration value	
gestion.proyectosProgramas.proyectosAntiguos.viaPresentacion	=
Presentation Route	
gestion.proyectosProgramas.proyectosAntiguos.origenInvencion	=
Ownership Information	
gestion.proyectosProgramas.proyectosAntiguos.vinculoLaboral	=
Employment relationship	
#Entidades	
gestion.proyectosProgramas.proyectosAntiguos.entidades.entidad	=
Entity	
gestion.proyectosProgramas.proyectosAntiguos.entidades.nit	= NIT
#Rol Entidades	
gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.entidad	=
Entity	
gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.rol	= Role
#Descripciones	
gestion.proyectosProgramas.proyectosAntiguos.tabla.orden	= Order
gestion.proyectosProgramas.proyectosAntiguos.tabla.descripcion	=
Description	
gestion.proyectosProgramas.proyectosAntiguos.cargando	=
LOADING DESCRIPTIONS...	
gestion.proyectosProgramas.proyectosAntiguos.btn.ver	= See
description	
gestion.proyectosProgramas.proyectosAntiguos.descripcion.header	=
Description	
#Impactos	
gestion.proyectosProgramas.proyectosAntiguos.impacto.label	= Impacts
= Impacts	
gestion.proyectosProgramas.proyectosAntiguos.impacto.tipo	= Type Impact
= Type Impact	
gestion.proyectosProgramas.proyectosAntiguos.impacto.descripcion	= Description
= Description	
gestion.proyectosProgramas.proyectosAntiguos.impacto.año	= Year
= Year	

```

gestion.proyectosProgramas.proyectosAntiguos.impacto.empty
= No impacts found for this project
#Cronograma
gestion.proyectosProgramas.proyectosAntiguos.cronograma.label
= Activities Schedule
gestion.proyectosProgramas.proyectosAntiguos.cronograma.empty
= No activities found
gestion.proyectosProgramas.proyectosAntiguos.cronograma.nActividad
= N° Activity
gestion.proyectosProgramas.proyectosAntiguos.cronograma.actividad
= Activity
gestion.proyectosProgramas.proyectosAntiguos.cronograma.inicio
= Start
gestion.proyectosProgramas.proyectosAntiguos.cronograma.fin
= End
gestion.proyectosProgramas.proyectosAntiguos.cronograma.ejecucion
= Execution In
#Tipo de Personal
gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.label
= Type of Staff
gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.empty
= No staff types found
gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.tipoPersonal
= Role in the Project
gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.numeroPersonas
= Number of People
#Personal
gestion.proyectosProgramas.proyectosAntiguos.personal.label
= Personal
gestion.proyectosProgramas.proyectosAntiguos.personal.empty
= No Registered Persons were found
gestion.proyectosProgramas.proyectosAntiguos.personal.persona
= Person
gestion.proyectosProgramas.proyectosAntiguos.personal.tipo
= Project Linkage
gestion.proyectosProgramas.proyectosAntiguos.personal.identificacion
= N° Identification
gestion.proyectosProgramas.proyectosAntiguos.personal.entidad
= Entity
gestion.proyectosProgramas.proyectosAntiguos.personal.vinculo
= Link
gestion.proyectosProgramas.proyectosAntiguos.personal.fechaNacimiento
= Date Birth
gestion.proyectosProgramas.proyectosAntiguos.personal.pais
= Country
gestion.proyectosProgramas.proyectosAntiguos.personal.email
= Email
gestion.proyectosProgramas.proyectosAntiguos.personal.tipoIdentificación
= Type Identification
gestion.proyectosProgramas.proyectosAntiguos.personal.numeroIdentificación
= Identification number
gestion.proyectosProgramas.proyectosAntiguos.personal.genero
= Gender
gestion.proyectosProgramas.proyectosAntiguos.personal.dedicacion
= Dedication (h / week)
gestion.proyectosProgramas.proyectosAntiguos.personal.nMeses
= Number Months
gestion.proyectosProgramas.proyectosAntiguos.personal.funcion
= Function in the project
gestion.proyectosProgramas.proyectosAntiguos.personal.registradoCvLAC
= Registered in CvLAC?

```



```

gestion.proyectosProgramas.proyectosAntiguos.personal.documentoHojaVida
= Attached Document Curriculum vitae
gestion.proyectosProgramas.proyectosAntiguos.personal.poblacionVictimaConflicto
= Population Victim of the Armed Conflict
gestion.proyectosProgramas.proyectosAntiguos.personal.poblacionCondicionDisc
= Population in a Situation of Disability
gestion.proyectosProgramas.proyectosAntiguos.personal.grupoEtnico
= Ethnic Group
gestion.proyectosProgramas.proyectosAntiguos.personal.grupoEtario
= Age Group
gestion.proyectosProgramas.proyectosAntiguos.personal.categoria
= Category
#Marco Logico
gestion.proyectosProgramas.proyectosAntiguos.marco.label
= Logical Framework
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoGral
= General Objective
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoGeneral.empty
= Not found General Objective
gestion.proyectosProgramas.proyectosAntiguos.marco.meta
= Goal
gestion.proyectosProgramas.proyectosAntiguos.marco.fuenteVerificacion
= Source Verification
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivosEspecificos.label
= Specific Objectives
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivosEspecificos.empty
= No Specific Objectives Found
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.label
= Specific Objective N°
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.objetivo.label
= Specific Objective
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.meta.label
= Goal
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.fuente.label
= Source Verification
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.indicador.label
= Indicator
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.actividades.empty
= This Objective has no Activities
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.resultados.empty
= This Activity has no Results
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.indicadores.empty
= This Result does not have Indicators
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.periodoInicial.label
= Initial Period
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.periodoFinal.label
= Final Period
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.duracion.label
= Duration
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.medioVerif.label
= Medium Verification
gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.mesCumpli.label
= Month of Compliance
gestion.proyectosProgramas.proyectosAntiguos.marco.resultados.label
= Activities, Results and Indicators
#Presupuesto
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tab
= Budget
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.loading
= LOADING ITEMS

```



```

gestion.proyectosProgramas.proyectosAntiguos.presupuesto.rubro.header
= Item
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.monInstitucionSol.header
= Requested financed amount
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEspecieSol.header
= Species counterpart requested
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEfectivoSol.header
= Cash counterpart requested
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.totalSol.header
= Total requested
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.monInstitucionAprob.header
= Approved financed amount
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEspecieAprob.header
= Species counterpart approved
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEfectivoAprob.header
= Approved cash counterpart
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.totalAprob.header
= Approved total

gestion.proyectosProgramas.proyectosAntiguos.presupuesto.entidad
= Entity
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.anio
= Year
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tipoPresupuesto
= Budget Type
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.modalidad
= Mode
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.financiado
= Funded
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.contrapartida
= Counterpart
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.especie
= Species
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.efectivo
= Cash
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tipoFinanciacion
= Financing Type
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.propuesto
= Proposed Amount
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.aprobado
= Approved Amount
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.rubro
= Item
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.nDias
= # Days
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.nPersonas
= # People
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.cantidad
= Quantity
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.costoPersona
= Cost per Person
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.valorUnitario
= Unit Value
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tema
= Theme
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.descripcion
= Description
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.lugarViaje
= Travel place
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.lugarJustificacion
= Place / Justification

```

```

gestion.proyectosProgramas.proyectosAntiguos.presupuesto.justificacion
= Justification
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.nombres
= First and Last Names
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.funcion
= Function
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.dedicacionMeses
= Dedication Months
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.dedicacionSemanas
= Dedication Weeks
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.proovedor
= Provider
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.institucion
= Financing Institution
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.tipoFinanciacion
= Financing Type
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.total
= Total
gestion.proyectosProgramas.proyectosAntiguos.presupuesto.modificar
= Modify Items
gestion.proyectosProgramas.proyectosAntiguos.ficha.loading
= Loading File, please wait...
#Evaluacion
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.emptyTable
= No Assigned Evaluators Found
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.infoGeneral
= General Information
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.tipoEvaluacion
= Evaluation Type
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.estadoEvaluador
= Evaluator Status
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.panel
= Evaluation Panel
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.panel.nombrePanel
= Panel Name
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.panel.actaPanel
= Panel Minutes
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.infoEvaluador
= Evaluator Information
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.tipoEvaluador
= Evaluator Type
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.dependencia
= Dependency
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.funcionario
= Official
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.nombres
= Names
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.usuarioSigp
= SIGP User
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.evaluacion
= Evaluation
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.fechaAsignacionEvaluador
= Evaluator Assignment Date
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.fechaEvaluacion
= Evaluation Date
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.duracionEvaluacion
= Evaluation Duration (days)
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.conceptoFinal
= Final Concept
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.recomenFinal
= Final Recommendation

```

```
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.concepto
= Concept
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.recomendacionEvaluador
= Evaluator Recommendation
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.puntaje
= Score
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.documentoEvaluacion
= Evaluation Document
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.pago
= Payment
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.valorEvaluacion
= Evaluation Value
gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.fechaPago
= Payment Date
#Documentos
gestion.proyectosProgramas.proyectosAntiguos.documentos.emptyTable
= No associated documents found
#Observaciones
gestion.proyectosProgramas.proyectosAntiguos.observaciones.emptyTable
= No Comments Found
```

Tabla de visualización de la información en la pantalla de Proyectos Antiguos

El siguiente representa una tabla de visualización de datos utilizando **PrimeFaces** dentro de una aplicación basada en **JSF (JavaServer Faces)**. La tabla está diseñada para mostrar información sobre **auditorías** dentro del sistema y cuenta con funcionalidades avanzadas como **lazy loading**, **filtros**, **ordenamiento**, **selección de filas**, **expansión de filas**, y **acciones personalizadas**.

```
<!-- TABLA PRINCIPAL -->
<div class="row">
  <div class="col-lg-12" style="overflow-x: scroll;margin-bottom: 40px;">
    <p:dataTable id="Table"
      value="#{proyectoAntiguoViewController.modelo}"
      var="proyecto"
      lazy="true"
      sortBy="#{proyecto.proyectoId}"
      sortOrder="DESCENDING"
      currentPageReportTemplate="Total de registros:
{totalRecords}"
      styleClass="table table-striped table-bordered table-
hover dataTables-entidades"
      reflow="true"
      rowKey="#{proyecto.proyectoId}"
      paginator="true"
      paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
      paginatorPosition="bottom"
      emptyMessage="#{msg['sb.datatable.registro.empty']}"
      widgetVar="items#{proyectoAntiguoViewController.modulo}"
      rows="10">
      <f:facet name="header">
        <p:outputPanel style="text-align: right;"
rendered="#{proyectoAntiguoViewController.buscarVisible}">
```

```

                                <h:outputText
value="#{msg['sb.component.tableHeader.buscar']}" />
                                <p:inputText id="globalFilter"
placeholder="#{msg['sb.component.tableHeader.buscar.placeholder']}"
onkeyup="PF('items#{proyectoAntiguoViewController.modulo}').filter()" style="width:200px"
disabled="#{proyectoAntiguoViewController.programaSelected != null or
proyectoAntiguoViewController.filtroActivo eq 'true'}"/>
                                </p:outputPanel>
                                </f:facet>
                                <p:column style="text-align: center; padding: 5px;"
width="3%">
                                    <p:rowToggler/>
                                </p:column>
                                <p:column filterBy="#{solicitud.fechaRegistro}"
filterMatchMode="contains"
                                filterStyle="display:none;"
                                width="5%"
                                style="text-align: center"

headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.anioRegistro']}">
                                <h:outputText value="#{proyecto.fechaRegistro != null ?
proyecto.fechaRegistro : ''}" >
                                    <f:convertDateTime type="date" pattern="yyyy"/>
                                </h:outputText>
                                </p:column>
                                <p:column filterBy="#{proyecto.proyectoId}" filterMatchMode="contains"
rendered="false"
                                filterStyle="display:none;"
                                width="8%"
                                style="text-align: center"

headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.codigoProyecto']}">
                                <h:outputText value="#{proyecto.proyectoId}" />
                                </p:column>
                                <p:column filterBy="#{proyecto.codigoProyecto}"
filterMatchMode="contains"
                                filterStyle="display:none;"
                                width="12%"
                                style="text-align: center"

headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.codigoProyectoId']}">
                                <h:outputText value="#{proyecto.codigoProyecto}" />
                                </p:column>
                                <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.titulo']}"
                                filterBy="#{proyecto.titulo}"
filterMatchMode="contains"
                                filterStyle="display:none;"
                                >
                                    <h:outputText id="tituloProyecto" value="#{proyecto.titulo}"/>
                                    <p:tooltip id="tituloProyectoTooltip" for="tituloProyecto"
position="right">

                                <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.column.convocatoria']} : </b>
                                #{proyecto.convocatoriaId.descripcionConvocatoria}<br/>

                                <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.column.programa']} : </b>
                                #{proyecto.proyectoPadreId eq null ? proyecto.dependenciaId.nombreDependencia :
                                proyectoAntiguoConvSessController.consultarProgramaPadre(proyecto.proyectoPadreId)}<br/>
                                </p:tooltip>
                                </p:column>

```

```

        <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.entidad']}"
        filterBy="#{proyecto.entidadAntiguaId.nombreEntidad}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        >
            <h:outputText value="#{proyecto.entidadAntiguaId ne null ?
proyecto.entidadAntiguaId.nombreEntidad : '' }"/>
        </p:column>
        <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.estado']}"
filterBy="#{proyecto.estadoProyectoId.descripcionValorDefinicion}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        style="text-align: center;"
        width="10%"
        >
            <small style="white-space: normal;"
                class="label label-
#{proyecto.estadoProyectoId.descripcionValorDefinicion eq 'DILIGENCIANDO' ? 'primary'
:
proyecto.estadoProyectoId.descripcionValorDefinicion eq 'RADICADO' ? 'warning'
:
(proyecto.estadoProyectoId.descripcionValorDefinicion eq 'RECHAZADO'
or
proyecto.estadoProyectoId.descripcionValorDefinicion.contains('RETIRADO')) ? 'danger' :
''}">

#{proyecto.estadoProyectoId.descripcionValorDefinicion}
            </small>
        </p:column>
        <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.tipoProyecto']}"
        filterBy="#{proyecto.exencion}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        width="8%"
        style="text-align: center"
        >
            <h:outputText value="#{proyecto.exencion eq 'P' ? 'Proyecto' :
'Programa'}" />
        </p:column>
        <p:column headerText="#{msg['sb.component.boton.opciones']}"
            width="8%"
            >
            <p:commandLink
update=":#{proyectoAntiguoViewController.modulo}FormDetalle"
                style="color: #fff"
                process="@this"
                id="goDetalles"

                onStart="PF('blockTable').show();"

                actionListener="#{proyectoAntiguoViewController.cargarDetallesProyecto(proyecto)}"

                title="#{msg['sb.component.boton.detalles']}"

                styleClass=" btn btn-xs btn-
primary sbTooltip"
                onComplete="cambiarTab();
scrolllear();PF('blockTable').hide();">

```

```

        <i class="fa fa-list"></i>
    </p:commandLink>
    <p:spacer></p:spacer>
    <p:commandLink
title="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.btn']}"
styleClass=" btn btn-xs btn-
primary sbTooltip"

    action="/app/proyectos/requisitosAntiguos"

    ajax="false"

    actionListener="#{requisitoProyectoAntiguoViewController.cargarRequisitosProyecto(proye
cto)}"

    process="@this"
    immediate="true"

    rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST5410')}"
    >
        <f:setPropertyActionListener
target="#{requisitoProyectoAntiguoViewController.proyectoSelected}" value="#{proyecto}"/>
        <i class="fa fa-cubes"></i>
    </p:commandLink>
    <p:spacer></p:spacer>
    <p:commandLink title="#{msg['sb.component.boton.download']}" -
#{proyecto.exencion eq 'P'
?
msg['gestion.proyectosProgramas.proyectosAntiguos.column.fichaProyecto']
:
msg['gestion.proyectosProgramas.proyectosAntiguos.column.fichaPrograma']}"
styleClass=" btn btn-xs btn-primary
sbTooltip"

onstart="descargarFicha('#{proyectoAntiguoViewController.urlReportes}GenerarReporteProyectos
Antiguos?proyecto_id=#{proyecto.proyectoId}&formato=#{SeguridadSessController.formatoRep
orte}')"

    >
        <i class="fa fa-download"></i>
        <f:setPropertyActionListener
value="#{proyecto.proyectoId}" target="#{SeguridadSessController.proyectoId}"/>
    </p:commandLink>
    <p:spacer></p:spacer>
</p:column>
<p:rowExpansion>
    <div class="mt-element-ribbon bg-grey-steel">
        <div class="row">
            <div class="col-lg-8 col-lg-offset-2">
                <p:panelGrid columns="2"

columnClasses="label,value"

                >

                    <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.codigoProyecto']}" />
                    <h:outputText
value="#{proyecto.proyectoId}" />

                    <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.fechaRegistro']}" />
                    <h:outputText
value="#{proyecto.fechaRegistro != null ? proyecto.fechaRegistro : ''}" >
                    <f:convertDateTime
type="date" pattern="dd/MM/yyyy"/>

```

```

</h:outputText>
<h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.column.convocatoria']}" />
<h:outputText
value="#{proyecto.convocatoriaId.descripcionConvocatoria}" />
<h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.column.programa']}" />
<h:outputText
value="#{proyecto.proyectoPadreId eq null
? proyecto.dependenciaId.nombreDependencia
:
proyectoAntiguoConvSessController.consultarProgramaPadre(proyecto.proyectoPadreId)}" />

<h:outputText
value="#{msg['sb.menu.gestion.usuarioCreador']}" />
<h:outputText
value="#{proyecto.usuarioCreador}" />

<h:outputText
value="#{msg['sb.menu.gestion.fechaCreacion']}" />
<h:outputText
value="#{proyecto.fechaCreacion}">
<f:convertDateTime
locale="es" pattern="dd/MM/yyyy"/>
</h:outputText>

</p:panelGrid>
</div>
</div>
<p:spacer/>
</div>
</p:rowExpansion>
</p:dataTable>
<p:blockUI block="Table" widgetVar="blockTable">
#{msg['sb.mensajes.cargando.label']}<br />
<div class="progress progress-striped active m-b-sm">
<div style="width: 100%;" class="progress-bar"></div>
</div>
</p:blockUI>
<p:blockUI block="Table" widgetVar="blockFicha">
#{msg['gestion.proyectosProgramas.proyectosAntiguos.ficha.loading']}<br />
<div class="progress progress-striped active m-b-sm">
<div style="width: 100%;" class="progress-bar"></div>
</div>
</p:blockUI>
<p:blockUI block="contenidoPagina" widgetVar="blockContent">
#{msg['sb.mensajes.descargando.espera.label']}<br />
<div class="progress progress-striped active m-b-sm">
<div style="width: 100%;" class="progress-bar"></div>
</div>
</p:blockUI>
</div>
</div>

```

◆ **Sección: Contenedor Principal**

```
<div class="row">
```

```
<div class="col-lg-12" style="overflow-x: scroll;margin-bottom: 40px;">
```

- Se utiliza Bootstrap para definir el layout.
- overflow-x: scroll permite visualizar toda la tabla en pantallas pequeñas, forzando scroll horizontal si hay muchas columnas.

◆ p:dataTable – Definición de Tabla

```
<p:dataTable id="Table"
    value="#{proyectoAntiguoViewController.model}"
    var="proyecto"
    lazy="true"
    ...
    rows="10">
```

- id="Table": identificador del componente.
- value="#{proyectoAntiguoViewController.model}": binding a la lista de datos (LazyDataModel), lo cual activa la carga bajo demanda (paginada).
- var="proyecto": nombre de la variable de iteración.
- lazy="true": activa la carga perezosa, optimizando el rendimiento en listas grandes.
- rows="10": muestra 10 registros por página.

◆ Orden y Paginación

```
sortBy="#{proyecto.proyectoId}"
sortOrder="DESCENDING"
paginator="true"
paginatorTemplate="..."
```

- Ordena inicialmente por ID del proyecto, de forma descendente.
- Se habilita paginación con una plantilla personalizada ({FirstPageLink} ... {RowsPerPageDropdown}).

◆ Filtro Global


```
<f:facet name="header">
  <p:outputPanel rendered="#{proyectoAntiguoViewController.buscarVisible}">
    ...
    <p:inputText id="globalFilter"
onkeyup="PF('items#{proyectoAntiguoViewController.modulo}').filter()"
disabled="#{proyectoAntiguoViewController.programaSelected != null or
proyectoAntiguoViewController.filtroActivo eq 'true'}"/>
  </p:outputPanel>
</f:facet>
```

- Campo de texto para búsqueda global (no por columna).
- Se desactiva cuando hay un programa seleccionado o un filtro activo, evitando sobrecargar los filtros.

◆ Columnas

◆ Columna: Expandir Fila

```
<p:column style="text-align: center;" width="3%">
  <p:rowToggler/>
</p:column>
```

- Agrega ícono de expandir para mostrar información detallada del proyecto (ver p:rowExpansion abajo).

◆ Columna: Año de Registro

```
<p:column filterBy="#{solicitud.fechaRegistro}" ...>
  <h:outputText value="#{proyecto.fechaRegistro}">
    <f:convertDateTime pattern="yyyy"/>
  </h:outputText>
</p:column>
```

- Muestra solo el año de la fecha de registro.
- **Nota:** el filtro se realiza sobre solicitud.fechaRegistro, pero el valor mostrado es proyecto.fechaRegistro → esto podría ser un bug o error de referencia.

◆ Columna: Código del Proyecto

```
<p:column filterBy="#{proyecto.codigoProyecto}" ...>
  <h:outputText value="#{proyecto.codigoProyecto}" />
</p:column>
```

- Muestra el código asignado al proyecto (posiblemente diferente del ID).

◆ Columna: Título con Tooltip

```
<p:column filterBy="#{proyecto.titulo}" ...>
  <h:outputText id="tituloProyecto" value="#{proyecto.titulo}" />
  <p:tooltip for="tituloProyecto">
    <b>Convocatoria: </b> #{proyecto.convocatoriaId.descripcionConvocatoria}<br/>
    <b>Programa: </b>
    #{proyectoAntiguoConvSessController.consultarProgramaPadre(...)}
  </p:tooltip>
</p:column>
```

- Muestra el título del proyecto con un tooltip contextual, enriquecido con datos relacionados (convocatoria y programa).

◆ Columna: Estado del Proyecto con Estilo

```
<small class="label label-#{...}">
  #{proyecto.estadoProyectoId.descripcionValorDefinicion}
</small>
```

- Se asigna un color al estado usando clases de Bootstrap:
 - primary: DILIGENCIANDO
 - warning: RADICADO
 - danger: RECHAZADO o RETIRADO
- Ayuda visualmente a identificar el estado.

◆ Columna: Tipo de Proyecto

```
<h:outputText value="#{proyecto.exencion eq 'P' ? 'Proyecto' : 'Programa'}" />
```

- Se traduce el campo exención de 'P' a "Proyecto", o se muestra "Programa" si es otro valor.

◆ Columna de Opciones

```
<p:column headerText="Opciones">
  <p:commandLink
    actionListener="#{proyectoAntiguoViewController.cargarDetallesProyecto(proyecto)}" ...>
    <i class="fa fa-list"></i>
  </p:commandLink>
  ...
  <p:commandLink action="/app/proyectos/requisitosAntiguos"
    actionListener="#{requisitoProyectoAntiguoViewController.cargarRequisitosProyecto(proyecto)}"
  >
    rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST5410')}">
      <i class="fa fa-cubes"></i>
    </p:commandLink>
    ...
    <p:commandLink onstart="descargarFicha(...)">
      <i class="fa fa-download"></i>
    </p:commandLink>
  </p:column>
```

- **Detalle (fa-list):** Carga detalle y cambia de pestaña.
- **Requisitos (fa-cubes):** Redirige a requisitos (solo si el usuario tiene permiso).
- **Descarga (fa-download):** Inicia descarga del informe PDF o RTF.

◆ Expansión de Fila (p:rowExpansion)

```
<p:rowExpansion>
  <p:panelGrid columns="2">
    <h:outputText value="Código Proyecto"/>
    <h:outputText value="#{proyecto.proyectoId}"/>
    ...
  </p:panelGrid>
</p:rowExpansion>
```

- Se muestra un resumen expandido del proyecto: ID, fecha registro, convocatoria, dependencia, usuario creador y fecha creación.

◆ p:blockUI – Indicadores de Carga

```
<p:blockUI block="Table" widgetVar="blockTable">...</p:blockUI>
```

- Bloquea la tabla durante procesos largos (carga de detalles, descarga).
- Mejora experiencia de usuario evitando múltiples clics y errores.

- Botón de Filtro por Búsqueda Avanzada

```
<!-- FILTRO AVANZADO -->
<h:panelGroup layout="block" id="formaFiltro" class="row">
    <div class="col-lg-12">
        <ui:fragment rendered="#{projectoAntiguoViewController.buscarVisible and
projectoAntiguoViewController.programaSelected == null}">
            <button type="button" class="btn btn-white btn-bitbucket" data-
toggle="collapse" data-target="#demo"><i class="fa fa-filter"></i>
            </button>
            </ui:fragment>
            <div id="demo" class="collapse #{projectoAntiguoViewController.filtroActivo ?
'in' : ''} well form-horizontal" style="padding: 20px">
                <div class="row">
                    <div class="col-lg-6" jsf:rendered="false">
                        <div class="form-group">
                            <p:outputLabel
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.codigoProyecto']}"
styleClass="col-lg-4
control-label"
for="codigoRegistroBusq"/>
                            <div class="col-lg-8">
                                <p:inputText
value="#{projectoAntiguoViewController.proyectoFilter.proyectoId}"
styleClass="form-control"
id="codigoRegistroBusq"
placeholder="#{msg['gestion.proyectosProgramas.proyectosAntiguos.codigoProyecto.placeholder'
]}"/>
                                </div>
                            </div>
                            <div class="col-lg-6">
                                <div class="form-group">
                                    <p:outputLabel
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.codigoProyectoId']}"
styleClass="col-lg-4
control-label"
for="codigoProyectoBusq"/>
                                    <div class="col-lg-8">
                                        <p:inputText
value="#{projectoAntiguoViewController.proyectoFilter.codigoProyecto}"
```

```

styleClass="form-control"
id="codigoProyectoBusq"
placeholder="#{msg['gestion.proyectosProgramas.proyectosAntiguos.codigoProyectoId.placeholde
r']}"
/>
</div>
</div>
<div class="col-lg-6">
  <div class="form-group">
    <p:outputLabel
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.anioRegistro']}"
control-label" styleClass="col-lg-4
for="anioRegistroBusq"/>
    <div class="col-lg-8">
      <p:selectOneMenu
value="#{proyectoAntiguoViewController.proyectoFilter.anioRegistro}"
id="anioRegistroBusq"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains"
      <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
      <f:selectItems
value="#{proyectoAntiguoViewController.listAnios}" />
    </p:selectOneMenu>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-lg-6">
    <div class="form-group">
      <p:outputLabel
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.estado']}"
control-label" styleClass="col-lg-4
for="estadoBusq-
#{msg['sb.convert.selectOne.estadoProyecto.ID']}" />
      <div class="col-lg-8">
        <p:selectOneMenu
value="#{proyectoAntiguoViewController.proyectoFilter.estadoProyectoId}"
id="estadoBusq-#{msg['sb.convert.selectOne.estadoProyecto.ID']}"

```

```

converter="valorDefinicionConverter"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains"

itemLabel="#{msg['sb.selectOneMenu.registro']}" />
value="#{proyectoAntiguoViewController.listaEstados}"
var="valorDef"
itemLabel="#{valorDef.descripcionValorDefinicion}"
itemValue="#{valorDef}"

</p:selectOneMenu>
</div>
</div>
<div class="col-lg-6">
  <div class="form-group">
    <p:outputLabel
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.tipoProyecto']}"
control-label"
styleClass="col-lg-4
for="tipoProyectoBusq"/>
    <div class="col-lg-8">
      <p:selectOneMenu
value="#{proyectoAntiguoViewController.proyectoFilter.exencion}"
id="tipoProyectoBusq"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains">
        <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
        <f:selectItem
itemLabel="Programa" itemValue="R"/>
        <f:selectItem
itemLabel="Proyecto" itemValue="P"/>

```

```

                                </p:selectOneMenu>
                                </div>
                                </div>
                                </div>
                                </div>
                                <div class="row">
                                    <div class="col-lg-12">
                                        <div class="form-group">
                                            <p:outputLabel
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.programa']}"
control-label"
                                styleClass="col-lg-2
                                for="programaBusq"/>
                                <div class="col-lg-10">
                                    <p:selectOneMenu
value="#{proyectoAntiguoViewController.proyectoFilter.dependenciaId}"
id="programaBusq"
converter="dependenciaConverter"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains"
                                >
                                <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
                                <f:selectItems
value="#{proyectoBeneficiosEvaluadoresSessController.listaDependencias}"
var="dependencia"
itemDescription="#{dependencia.nombreDependencia}"
itemLabel="#{dependencia.nombreDependencia}"
itemValue="#{dependencia}"
                                />
                                </p:selectOneMenu>
                                </div>
                                </div>
                                </div>
                                </div>
                                <div class="row">
                                    <div class="col-lg-12">
                                        <div class="form-group">
                                            <p:outputLabel
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.convocatoria']}"
control-label"
                                styleClass="col-lg-2
                                for="convocatoriaBusq-Antiguos"/>
                                <div class="col-lg-10">

```

```

                                <p:selectOneMenu
value="#{proyectoAntiguoViewController.proyectoFilter.convocatoriaId}"
id="convocatoriaBusq-Antiguos"
converter="convocatoriaProyectosConverter"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains"
                                >
                                <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
                                <f:selectItems
value="#{proyectoAntiguoViewController.listaConvocatorias}"
var="convocatoria"
itemLabel="#{convocatoria.descripcionConvocatoria}"
itemValue="#{convocatoria}"
                                />
                                </p:selectOneMenu>
                                </div>
                                </div>
                                </div>
                                </div>
                                <div class="row">
                                <div class="col-lg-12">
                                <div class="form-group">
                                <p:outputLabel
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.titulo']}"
                                styleClass="col-lg-2
control-label"
for="tituloProyectoBusq"/>
                                <div class="col-lg-10">
                                <p:inputText
value="#{proyectoAntiguoViewController.proyectoFilter.titulo}"
styleClass="form-control"
id="tituloProyectoBusq"
placeholder="#{msg['gestion.proyectosProgramas.proyectosAntiguos.titulo.placeholder']}"
                                />
                                </div>
                                </div>
                                </div>
                                <div class="hr-line-dashed"></div>
                                <div class="row">
                                <div class="col-lg-6">

```



```

        <div class="form-group">
            <div class="col-lg-8 col-lg-offset-6">
                <p:commandLink id="buscar"

process="formaFiltro"
update="Table"

actionListener="#{proyectoAntiguoConvController.lazyLoad}"
styleClass="btn btn btn-primary pull-left"

                >
                    <i class="fa fa-search"></i>

                #{msg['sb.component.boton.filtrar'}}
            </p:commandLink>
        </div>
    </div>
    <div class="col-lg-6">
        <div class="form-group">
            <div class="col-lg-8 col-lg-offset-4">
                <p:commandLink id="limpiar"

update="formaFiltro,Table"
process="@this"
oncomplete="initChoosen();"
actionListener="#{proyectoAntiguoConvController.limpiarFiltros}"
styleClass="btn btn btn-primary pull-right"

                >
                    <i class="fa fa-times"></i>

                #{msg['sb.component.boton.cancelar']}
            </p:commandLink>
        </div>
    </div>
</div>
</h:panelGroup>

```

Esta sección se encuentra dentro del h:panelGroup con id="formaFiltro", y corresponde a una **interfaz colapsable** que permite aplicar filtros avanzados sobre la tabla p:dataTable de proyectos.

♦ Estructura General del Filtro

```
<h:panelGroup layout="block" id="formaFiltro" class="row">
```

- Contenedor visual con layout en bloque que agrupa todos los filtros.
- Usado como objetivo para process y update desde los botones de acción ("Filtrar", "Limpiar").

◆ Botón para Expandir/Contraer

```
<ui:fragment rendered="#{proyectoAntiguoViewController.buscarVisible and
proyectoAntiguoViewController.programaSeleccionado == null}">
  <button type="button" class="btn btn-white btn-bitbucket" data-toggle="collapse" data-
target="#demo">
    <i class="fa fa-filter"></i>
  </button>
  #{msg['gestion.proyectosProgramas.proyectosAntiguos.column.busquedaAvanzada']}
</ui:fragment>
```

- Este botón permite al usuario expandir o colapsar los filtros.
- Solo se muestra si buscarVisible es true y no hay programa seleccionado.

◆ Panel de Filtros (id="demo")

```
<div id="demo" class="collapse #{proyectoAntiguoViewController.filtroActivo ? 'in' : ''} well
form-horizontal" style="padding: 20px">
```

- collapse in: hace que el panel esté expandido si filtroActivo es true.
- Se aplica el estilo form-horizontal para distribuir etiquetas y campos.

Campos de Filtro

A continuación se explican los filtros implementados:

1. ID del Proyecto (Técnico)

```
<p:inputText value="#{proyectoAntiguoViewController.proyectoFilter.proyectoId}" .../>
```

- Permite buscar directamente por el ID técnico del proyecto (clave primaria en BD).

2. 🔍 Código de Proyecto (Funcional)

```
<p:inputText value="#{proyectoAntiguoViewController.proyectoFilter.codigoProyecto}" .../>
```

- Busca por el código asignado al proyecto, distinto del ID interno.

3. 📅 Año de Registro

```
<p:selectOneMenu value="#{proyectoAntiguoViewController.proyectoFilter.anioRegistro}" ...>  
  <f:selectItems value="#{proyectoAntiguoViewController.listaAnios}"/>  
</p:selectOneMenu>
```

- Selector desplegable con los años disponibles cargados desde el backend.

4. 📌 Estado del Proyecto

```
<p:selectOneMenu value="#{proyectoAntiguoViewController.proyectoFilter.estadoProyectoId}"  
  converter="valorDefinicionConverter" ...>  
  <f:selectItems value="#{proyectoAntiguoViewController.listaEstados}" ... />  
</p:selectOneMenu>
```

- Lista desplegable para seleccionar el estado del proyecto.
- Usa un converter para mapear entre la entidad y el valor en la vista.

5. 🏷️ Tipo de Proyecto

```
<p:selectOneMenu value="#{proyectoAntiguoViewController.proyectoFilter.exencion}">  
  <f:selectItem itemLabel="Programa" itemValue="R"/>  
  <f:selectItem itemLabel="Proyecto" itemValue="P"/>  
</p:selectOneMenu>
```

- Determina si el registro es un **programa (R)** o un **proyecto (P)**.

6. 🏠 Programa / Dependencia

```
<p:selectOneMenu value="#{proyectoAntiguoViewController.proyectoFilter.dependenciaId}"
converter="dependenciaConverter">
    <f:selectItems value="#{proyectoBeneficiosEvaluadoresSessController.listaDependencias}"
    ... />
</p:selectOneMenu>
```

- Filtro por la unidad académica o administrativa asociada.
- Usa un converter específico para dependencias.

7. Convocatoria

```
<p:selectOneMenu value="#{proyectoAntiguoViewController.proyectoFilter.convocatoriaId}"
converter="convocatoriaProyectosConverter">
    <f:selectItems value="#{proyectoAntiguoViewController.listaConvocatorias}" ... />
</p:selectOneMenu>
```

- Filtro por convocatoria asociada al proyecto.
- Permite identificar la “ronda” o periodo de participación.

8. Título del Proyecto

```
<p:inputText value="#{proyectoAntiguoViewController.proyectoFilter.titulo}" ... />
```

- Permite realizar búsqueda por coincidencia del texto del título.

Acciones del Filtro

Botón Filtrar

```
<p:commandLink id="buscar"
    process="formaFiltro"
    update="Table"
    actionListener="#{proyectoAntiguoConvController.lazyLoad}"
    styleClass="btn btn-primary">
    <i class="fa fa-search"></i> #{msg['sb.component.boton.filtrar']}
</p:commandLink>
```

- Procesa el formulario formaFiltro.

- Llama al método lazyLoad que probablemente recarga el LazyDataModel según los filtros actuales.
- Actualiza la tabla principal Table.

✗ Botón Limpiar Filtros

```
<p:commandLink id="limpiar"
                process="@this"
                update="formaFiltro,Table"
                oncomplete="initChoosen();"
                actionListener="#{proyectoAntiguoConvController.limpiarFiltros}"
                styleClass="btn btn-primary">
    <i class="fa fa-times"></i> #{msg['sb.component.boton.cancelar']}
</p:commandLink>
```

- Limpia todos los campos del filtro (se asume que limpiarFiltros reinicia el objeto proyectoFilter).
- Recarga el formulario y la tabla.
- Ejecuta JavaScript initChoosen() para volver a aplicar estilos o plugins (como Select2 o Chosen).

La pantalla se ve de la siguiente forma:

MINCIENCIAS

Gestión Minciencias

Usuario: SIGP_WEB

Inicio

Administración

Configuración

Reconocimiento

Beneficios Tributarios

Proyectos/Programas

Registro

Proyectos

Proyectos Antiguos

Evaluación

Decisión

Consultas Reportes

Invencción Patentes

Ayuda

Proyectos y Programas Antiguos

Inicio / Proyectos/Programas / Proyectos Antiguos

Proyectos Antiguos

Búsqueda Avanzada

BUSCAR TODOS:

	Año Registro	Código Proyecto	Título Proyecto	Entidad Ejecutora	Estado	Tipo Proyecto	Opciones
	2016	53640472	Proyecto prueba 03	ABONO ORGANICO MAGANGUE	INDICADO	Proyecto	
	2016	416053640470	Proyecto prueba 02	ABONOS COLOMBIANOS S.A. - ABOCOL S.A.	INDICADO	Proyecto	
	2016	481553640468	Proyecto prueba 01	A SERVIR Y A CONSTRUIR "ASECO"	INDICADO	Proyecto	
	2016	416053640466	Proyecto prueba	ABONOS COLOMBIANOS S.A. - ABOCOL S.A.	INDICADO	Proyecto	
	2015	49840425	prueba2 con sgo_registro_proyectos 3	software builders	REVISADO	Programa	
	2015	49840423	prueba2 con sgo_registro_proyectos 2	software builders	REVISADO	Programa	
	2015	49840419	prueba2 con sgo_registro_proyectos	software builders	REVISADO	Programa	
	2015	49840417	prueba2	software builders	REVISADO	Programa	
	2015	49840414	prueba	software builders	REVISADO	Programa	
	2015	481553640408	Proyecto pruebas Luigui y Natalia 123456789	A SERVIR Y A CONSTRUIR "ASECO"	REVISADO	Proyecto	

TOTAL DE REGISTROS: 21390

1

2

3

4

5

6

7

8

9

10

»

Detalles de Proyectos Antiguos

◆ **Funcionalidad General**

El panel de detalles se muestra cuando el usuario hace clic en el botón “Detalles” desde la tabla principal de proyectos antiguos. Se despliega un formulario dividido en secciones que presentan información clave del proyecto y enlaces a acciones adicionales, como descarga de reportes y requisitos asociados.

◆ **Estructura del Panel**

Layout Principal

- Utiliza un div.row con dos columnas:
 - **Columna Izquierda (col-lg-10):** Contiene la información detallada del proyecto.
 - **Columna Derecha (col-lg-2):** Lista de palabras clave asociadas al proyecto.

Encabezado del Panel

- Título del proyecto (`{proyectoAntiguoViewController.selected.titulo}`).
- Botón “Volver” que ejecuta la función `cambiarTabInicial()` para regresar a la tabla principal.

Información General

Usa elementos `<dl class="dl-horizontal">` para mostrar pares de datos (dt / dd) como:

- Código de proyecto
- Convocatoria
- Dependencia
- Estado del proyecto (con estilos condicionales para resaltar estado)
- Tipo de proyecto (Ficha técnica, Proyecto, Programa)

- Entidad ejecutora
- Duración, lugar de ejecución, ejecución del cronograma, tipo de financiación (si aplica)

Fechas Importantes

- Registro
- Revisión
- Modificación (formateadas con f:convertDateTime)

Subsanabilidad

- Representada con un h:selectBooleanCheckbox deshabilitado.

Acciones Adicionales

Requisitos del Proyecto

- Si el usuario tiene permisos (GEST5410 en SeguridadSessController.mapaMenu), se presenta un botón que redirige a /app/proyectos/requisitosAntiguos con carga de requisitos mediante:

```
actionListener="#{requisitoProyectoAntiguoViewController.cargarRequisitosProyecto(...)}"
```

Ficha del Proyecto/Programa

- Se permite descarga en formato PDF o Word vía descargarFicha(...), con parámetros dinámicos (proyecto_id y formato).

Reporte Marco Lógico

- Se muestra si el menú FOR1060 está disponible.
- Acción delegada a proyectoAntiguoConvController.crearReporteMarcoLogico(...).

Reporte de Evaluación

- Disponible si countEvaluaciones > 0.

- Acción delegada a proyectoAntiguoConvController.downloadReporteEvaluacion(...).

Palabras Clave

En la columna derecha, se listan las palabras clave del proyecto:

```
<ui:repeat value="#{proyectoAntiguoViewController.selected.palabraClaveList}" var="palabra">
  <li><a><i class="fa fa-tag"></i> #{palabra.palabra}</a></li>
</ui:repeat>
```

Consideraciones Técnicas

- La visibilidad de secciones depende de múltiples condiciones (rendered, valores nulos, tipo de proyecto).
- Uso de utilidades JS (cambiarTabInicial(), descargarFicha(...)).
- La vista depende de un backing bean llamado proyectoAntiguoViewController y también interactúa con otros controladores como requisitoProyectoAntiguoViewController, proyectoAntiguoConvController y SeguridadSessController.

1. Pestaña de Información Básica

Esta pestaña contiene información estructurada del proyecto y está organizada en subtabs o secciones internas, cada una con funcionalidades específicas:

- **Entidades**
- **Roles Entidades**
- **Presupuesto**
- **Descripciones**
- **Impactos**
- **Coberturas**
- **Actividades Cronograma**
- **Tipo de Personal**

- **Personal**

Cada una de estas secciones probablemente se carga de forma dinámica o condicional con su propia lógica en el controlador.

Las pestañas principales se gestionan con Bootstrap (<ul class="nav nav-tabs">) y los eventos de cambio de pestaña llaman funciones JavaScript:

```
<a href="#tab-basica" data-toggle="tab" onclick="cargarInformacionBasica();" >
```

Las demás pestañas también siguen esta lógica.

Subpestaña: Entidades

◆ **Propósito**

Esta subpestaña muestra las **entidades vinculadas al proyecto** antiguo, incluyendo su nombre, NIT y dígito de verificación, y permite consultar sus detalles.

```
<!-- Pestaña Entidades -->
<div id="tab-entidades" class="tab-pane active" >
  <div class="panel-body">
    <p:remoteCommand id="cargarEntidadesRC" name="cargarEntidades"
update="entidadesProyecto"
actionListener="#{proyectoAntiguoViewController.cargarEntidadesProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarEntidadesRC"
widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.entidades']}</b>
    </h2>
    <br/>
    <h:panelGroup id="entidadesProyecto">
      <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoAntiguoViewController.listaProyectoEntidades.size() eq 0 or
proyectoAntiguoViewController.listaProyectoEntidades == null}">

      #{msg['gestion.proyectosProgramas.proyectosAntiguos.entidades.empty']}
      </h:panelGroup>
      <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoAntiguoViewController.listaProyectoEntidades.size() gt 0}">
        <p:dataTable
id="#{proyectoAntiguoViewController.modulo}tableEntidades"
          var="entidad"

value="#{proyectoAntiguoViewController.listaProyectoEntidades}"
          paginator="true"
          selectionMode="single"
```

```

reflow="true"
rowKey="#{entidad.entidadProyectoId}"
currentPageReportTemplate="Total de

registros: {totalRecords}"

paginatorTemplate="{CurrentPageReport}"
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"

paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"

widgetVar="items#{proyectoAntiguoViewController.modulo}"
rows="10"
>
    <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.entidades.entidad']}"
filterStyle="display:none;"
>
    <h:outputText
value="#{entidad.nombreEntidad}"/>
    </p:column>
    <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.entidades.nit']}"
filterStyle="display:none;"
width="30%"
style="text-align: center;"
>
    <h:outputText value="#{entidad.nit} -
#{entidad.digitoVerificacion}"/>
    </p:column>
    <p:column
headerText="#{msg['sb.component.boton.detalles']}"
width="15%"
style="text-align: center;"
>
    <!-- row record -->
    <p:commandLink
title="#{msg['sb.component.boton.detalles']}"
oncomplete="$('#dlgDetallesEntidad').modal();"
update="formDetallesEntidad"
styleClass="btn
btn-sm btn-primary"
>
    <f:setPropertyActionListener
target="#{proyectoAntiguoViewController.proyectoEntidadSeleccionado}" value="#{entidad}"/>
    <i class="fa fa-list"></i>
    </p:commandLink>
    </p:column>
</p:dataTable>
</h:panelGroup>
</h:panelGroup>
</div>
</div>

```

◆ **Estructura**

- **Activación de contenido:** Al hacer clic en la pestaña Entidades, se ejecuta la función cargarEntidades() a través de un <p:remoteCommand> con update="entidadesProyecto" y acción en proyectoAntiguoViewController.cargarEntidadesProyecto().

```
<p:remoteCommand id="cargarEntidadesRC" name="cargarEntidades"
  update="entidadesProyecto"
  actionListener="#{proyectoAntiguoViewController.cargarEntidadesProyecto()}" />
```

- **Carga visual (bloqueo):** Se emplea un p:blockUI con efecto de carga mientras se ejecuta la llamada remota.
 - **Vista sin resultados:**
Si listaProyectoEntidades está vacía o null, se muestra un mensaje indicando que no hay entidades registradas.
 - **Vista con resultados:**
Si hay entidades, se presenta una tabla (p:dataTable) con paginación y filtrado oculto.
- ◆ **Componentes de la tabla (p:dataTable)**
- value="#{proyectoAntiguoViewController.listaProyectoEntidades}"
 - selectionMode="single" – permite seleccionar una fila.
 - widgetVar="items#{proyectoAntiguoViewController.modulo}" – permite interactuar con JavaScript.
 - rows="10" – se muestran 10 registros por página.

Columnas:

1. **Entidad**
Muestra el nombre de la entidad ({entidad.nombreEntidad})
2. **NIT**
Muestra el NIT con dígito de verificación ({entidad.nit} - {entidad.digitoVerificacion})
3. **Botón de Detalles**
 - p:commandLink que abre el modal #dlgDetallesEntidad
 - Actualiza formDetallesEntidad

- Usa `f:setPropertyActionListener` para pasar el objeto entidad seleccionado a `proyectoAntiguoViewController.proyectoEntidadSelected`

◆ Interfaz visual

- La pestaña está identificada por el ícono `fa fa-building` y el label dinámico:

```
#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.entidades']}
```

◆ Modal relacionado

- Se abre `#dlgDetallesEntidad` (no incluido en el código, pero referenciado desde el botón de detalles).

Subpestaña: Rol Entidades

◆ Propósito

Esta sección muestra el **rol desempeñado por cada entidad** participante dentro del proyecto antiguo, especificando tanto la entidad como su rol asociado.

```
<!-- Pestaña Rol Entidades -->
<div id="tab-rolEntidades" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarRolEntidadesRC" name="cargarRolEntidades"
update="rolEntidadesProyecto"
actionListener="#{proyectoAntiguoViewController.cargarRolEntidadesProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarRolEntidadesRC"
widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.rolEntidades']}</b>
    </h2>
    <br/>
    <h:panelGroup id="rolEntidadesProyecto">
      <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoAntiguoViewController.selected.rolEntidadList.size() eq 0 or
proyectoAntiguoViewController.selected.rolEntidadList == null}">

        #{msg['gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.empty']}
        </h:panelGroup>
        <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoAntiguoViewController.selected.rolEntidadList.size() gt 0}">
          <p:dataTable
id="#{proyectoAntiguoViewController.modulo}tableRolEntidades"
var="rolEntidad"
```

```
value="#{proyectoAntiguoViewController.selected.rolEntidadList}"
                                paginator="true"
                                selectionMode="single"
                                reflow="true"
                                rowKey="#{rolEntidad.rolEntidadId}"
                                currentPageReportTemplate="Total de
registros: {totalRecords}"
                                paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
                                paginatorPosition="bottom"
emptyMessage="#{msg['sb.datatable.registro.empty']}"
widgetVar="items#{proyectoAntiguoViewController.modulo}"
                                rows="10"
                                >
                                <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.entidad']}"
                                filterStyle="display:none;"
                                >
                                <h:outputText
value="#{rolEntidad.entidadProyectoId.nombreEntidad}"/>
                                </p:column>
                                <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.rol']}"
                                filterStyle="display:none;"
                                width="30%"
                                style="text-align: center;"
                                >
                                <h:outputText
value="#{rolEntidad.rolId.descripcionValorDefinicion}"/>
                                </p:column>
                                </p:dataTable>
                                </h:panelGroup>
                                </h:panelGroup>
                                </div>
                                </div>
```

◆ Estructura y funcionamiento

- **Activación de contenido:** Al seleccionar la pestaña, se invoca el método cargarRolEntidades() definido como un <p:remoteCommand> que actualiza la sección rolEntidadesProyecto y ejecuta el método del controlador:

```
<p:remoteCommand id="cargarRolEntidadesRC" name="cargarRolEntidades"
update="rolEntidadesProyecto"
```

actionListener="#{proyectoAntiguoViewController.cargarRolEntidadesProyecto()}" />

- **Carga visual:** Se utiliza p:blockUI con animación de carga para el contenedor projecChildrens.

- **Mensajes vacíos:**

Si la lista rolEntidadList asociada al proyecto está vacía o nula, se muestra el mensaje:

```
{msg['gestion.proyectosProgramas.proyectosAntiguos.rolEntidades.empty']}
```

- **Tabla de resultados:**

Cuando la lista rolEntidadList contiene registros, se presenta un p:dataTable con paginación y visualización responsive.

- ◆ **Componentes principales (p:dataTable)**

- value="#{proyectoAntiguoViewController.selected.rolEntidadList}"
- selectionMode="single" – permite seleccionar una única fila.
- widgetVar="items#{proyectoAntiguoViewController.modulo}" – identificador dinámico.
- rows="10" – número de registros por página.

Columnas:

1. **Entidad**

Muestra el nombre de la entidad asociada:

```
{rolEntidad.entidadProyectoId.nombreEntidad}
```

2. **Rol**

Muestra el rol asignado a esa entidad:

```
{rolEntidad.rolId.descripcionValorDefinicion}
```

- ◆ **Interfaz visual**

- Ícono: fa fa-building
- Etiqueta dinámica:

```
{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.rolEntidades']}
```

Subpestaña: Presupuesto

- ◆ **Propósito**

Mostrar los rubros presupuestales solicitados y aprobados por el proyecto antiguo, discriminando los valores por fuente de financiación y en formato de tabla, junto con un acceso a los detalles de cada rubro.

```
<!-- Pestaña Presupuesto-->
<div id="tab-presupuesto" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarPresupuestoRC" name="cargarPresupuesto"
update="presupuestoProyecto"
actionListener="#{proyectoAntiguoViewController.cargarPresupuestoProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarPresupuestoRC"
widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.presupuesto']}</b>
    </h2>
    <br/>
    <h:panelGroup id="presupuestoProyecto">
      <div class="panel-group" id="accordionPresupuesto">
        <h:panelGroup id="presupuesto" styleClass="animated
fadeInLeft">
          <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoAntiguoViewController.listRubros.size() lt 1 or
proyectoAntiguoViewController.listRubros == null}">

            #{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.empty']}
            </h:panelGroup>
            <h:panelGroup layout="block"
rendered="#{proyectoAntiguoViewController.listRubros.size() gt 0}">
              <p:dataTable id="Table"

value="#{proyectoAntiguoViewController.listRubros}"

              var="rubro"
              sortOrder="descending"
              styleClass="table

              reflow="true"
              rowKey="#{rubro[6]}"
              paginator="true"

              currentPageReportTemplate="Total de registros: {totalRecords}"

              paginatorTemplate="{CurrentPageReport} {FirstPageLink} {PreviousPageLink} {PageLinks}
{NextPageLink} {LastPageLink} {RowsPerPageDropdown}"

              paginatorPosition="bottom"

              emptyMessage="#{msg['sb.datatable.registro.empty']}"

              widgetVar="itemsRubros"

              <f:facet name="header">
                <p:outputPanel style="text-
align: right;">
```

```

value="#{msg['sb.component.tableHeader.buscar']}" />
id="globalFilter" placeholder="#{msg['sb.component.tableHeader.buscar.placeholder']}"
onkeyup="PF('itemsRubros').filter()" style="width:200px"/>
filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
center"
ID">
value="#{rubro[9]}" />
filterMatchMode="contains"
filterStyle="display:none;"
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.rubro.header']}"
>
value="#{rubro[0]}" />
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.total']}"
style="text-align: right; text-transform: uppercase;" />
filterMatchMode="contains"
filterStyle="display:none;"
right;"
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.monInstitucionSo
l.header']}">
value="#{rubro[1]}">
currencySymbol="$" type="currency" locale="es"/>
value="#{proyectoAntiguoViewController.totalMonInstitucionSol}">
<f:convertNumber currencySymbol="$" type="currency" locale="es"/>
filterMatchMode="contains"
filterStyle="display:none;"

```



```

right;"
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEspecieSol.header']}">
value="#{rubro[2]}">
currencySymbol="$" type="currency" locale="es"/>
value="#{proyectoAntiguoViewController.totalConEspecieSol}">
    <f:convertNumber currencySymbol="$" type="currency" locale="es"/>
    </h:outputText>
    </f:facet>
    </p:column>
    <p:column filterBy="#{rubro[3]}"
filterMatchMode="contains"
filterStyle="display:none;"
right;"
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEfectivoSol.header']}">
value="#{rubro[3]}">
currencySymbol="$" type="currency" locale="es"/>
value="#{proyectoAntiguoViewController.totalConEfectivoSol}">
    <f:convertNumber currencySymbol="$" type="currency" locale="es"/>
    </h:outputText>
    </f:facet>
    </p:column>
    <p:column filterBy="#{rubro[4]}"
filterMatchMode="contains"
filterStyle="display:none;"
right;"
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.totalSol.header']}">
value="#{rubro[4]}">
currencySymbol="$" type="currency" locale="es"/>
value="#{proyectoAntiguoViewController.totalSol}">
    <f:convertNumber currencySymbol="$" type="currency" locale="es"/>
    </h:outputText>
    </f:facet>

```

```

filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
right;"
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.monInstitucionAprob.header']}">
value="#{rubro[5]}">
currencySymbol="$" type="currency" locale="es"/>

filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
right;"
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEspecieAprob.header']}">
value="#{rubro[6]}">
currencySymbol="$" type="currency" locale="es"/>

filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
right;"
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.conEfectivoAprob.header']}">
value="#{rubro[7]}">
currencySymbol="$" type="currency" locale="es"/>

filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
right;"
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.totalAprob.header']}">
value="#{rubro[8]}">
currencySymbol="$" type="currency" locale="es"/>

```

```

<p:column width="6%">
    <p:commandLink styleClass=" btn
btn-xs btn-primary sbTooltip"
style="color: #fff;"
title="#{msg['sb.component.boton.detalles']}"
actionListener="#{proyectoAntiguoConvController.cargarDetallesRubrosProyecto(rubro[9])}"
oncomplete="openDialogRubros();"
update="formDialogRubros"
    <i class="fa fa-
money"></i>
    <f:setPropertyActionListener target="#{proyectoAntiguoViewController.descripcionRubro}"
value="#{rubro[0]}" />
    </p:commandLink>
</p:column>
</p:dataTable>
</h:panelGroup>
</h:panelGroup>
</div>
</h:panelGroup>
</div>
</div>

```

◆ Funcionamiento general

- **Carga del contenido:**

Al activarse la pestaña, se ejecuta un `<p:remoteCommand>` que:

- Llama al método `cargarPresupuestoProyecto()` del controlador `proyectoAntiguoViewController`.
- Actualiza el panel `presupuestoProyecto`.

```

<p:remoteCommand id="cargarPresupuestoRC" name="cargarPresupuesto"
update="presupuestoProyecto"

```

```

actionListener="#{proyectoAntiguoViewController.cargarPresupuestoProyecto()}" />

```

- **Bloqueo de interfaz mientras carga:**

Se utiliza `p:blockUI` para bloquear visualmente `projecChildrens` con una barra de progreso mientras se carga la información.

♦ **Vista de resultados**

- **Condición vacía:**

Si listRubros está vacía o nula, se muestra un mensaje:

```
{msg['gestion.proyectosProgramas.proyectosAntiguos.presupuesto.empty']}
```

- **Tabla principal:**

Si listRubros contiene elementos, se muestra un <p:dataTable> con paginación, ordenamiento y filtro global. Cada rubro es una fila representada por un arreglo de objetos (rubro[n]).

♦ **Columnas mostradas (visibles)**

Índice	Campo	Descripción
rubro[0]	Rubro	Nombre o descripción del rubro.
rubro[1]	Monto institución solicitante	Valor en efectivo solicitado por la institución.
rubro[2]	Contribución en especie solicitada	Aportes no monetarios solicitados.
rubro[3]	Contribución en efectivo solicitada	Aportes monetarios solicitados por terceros.
rubro[4]	Total solicitado	Suma total solicitada del rubro.

Todas estas columnas están configuradas con formato de moneda (f:convertNumber con locale="es" y currencySymbol="\$").

Además, cada una incluye su respectivo **footer** que muestra totales provenientes del controlador, por ejemplo:

```
<f:facet name="footer">
  <h:outputText value="#{proyectoAntiguoViewController.totalSol}">
    <f:convertNumber currencySymbol="$" type="currency" locale="es"/>
  </h:outputText>
</f:facet>
```

◆ Columnas ocultas (rendered="false")

Estas columnas están presentes en el modelo pero no se muestran directamente:

- rubro[5] – Monto institución aprobado
- rubro[6] – Contribución especie aprobada
- rubro[7] – Contribución efectivo aprobada
- rubro[8] – Total aprobado
- rubro[9] – Rubro ID

◆ Detalles por rubro

En la última columna, cada fila incluye un botón tipo commandLink que:

- Llama al método cargarDetallesRubrosProyecto(rubro[9]) del controlador proyectoAntiguoConvController.
- Abre un diálogo (openDialogRubros()) para mostrar el detalle del rubro seleccionado.
- Actualiza el formulario formDialogRubros.
- Establece el nombre del rubro en descripcionRubro.

```
<p:commandLink ...  
  actionListener="#{proyectoAntiguoConvController.cargarDetallesRubrosProyecto(rubro[9])}"  
  oncomplete="openDialogRubros();"   
  update="formDialogRubros">  
  <i class="fa fa-money"></i>  
  <f:setPropertyActionListener target="#{proyectoAntiguoViewController.descripcionRubro}"  
  value="#{rubro[0]}" />  
</p:commandLink>
```

◆ Filtro global

El encabezado de la tabla tiene un buscador global que filtra todas las columnas simultáneamente:

```
<p:inputText id="globalFilter" onkeyup="PF('itemsRubros').filter()" />
```

Subpestaña: Descripciones

◆ Propósito

Presentar las descripciones textuales configuradas para el proyecto antiguo, en el orden y clasificación establecidos, permitiendo visualizarlas a través de un diálogo.

```
<!-- Pestaña Descripciones -->
<div id="tab-descripciones" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarDescripcionesRC" name="cargarDescripciones"
update="descripciones"
actionListener="#{proyectoAntiguoConvController.cargarDescripcionesProyecto}" />
    <p:blockUI block="projecChildrens" trigger="cargarDescripcionesRC"
widgetVar="demoBlock">
      #{msg['gestion.proyectosProgramas.proyectosAntiguos.cargando']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.descripciones']}</b>
    </h2>
    <br/>
    <h:panelGroup id="descripciones" styleClass="animated fadeInLeft">
      <h:panelGroup layout="block" id="empty" styleClass="well"
rendered="#{proyectoAntiguoViewController.listaDescripciones.size() lt 1 or
proyectoAntiguoViewController.listaDescripciones == null}">

        #{msg['gestion.proyectosProgramas.proyectosAntiguos.descripcion.empty']}
      </h:panelGroup>
      <h:panelGroup layout="block"
rendered="#{proyectoAntiguoViewController.listaDescripciones.size() gt 0}">
        <h:dataTable
value="#{proyectoAntiguoViewController.listaDescripciones}"
var="descripcion"

id="#{proyectoAntiguoViewController.modulo}Table"
styleClass="table table-striped table-
bordered table-hover"

          <h:column>
            <!-- column header -->
            <f:facet
name="header">#{msg['gestion.proyectosProgramas.proyectosAntiguos.tabla.orden']}</f:facet>
            <!-- row record -->

            #{descripcion.configuracionDescripcionId.ordenPresentacion}
          </h:column>
          <h:column>
            <!-- column header -->
            <f:facet
name="header">#{msg['gestion.proyectosProgramas.proyectosAntiguos.tabla.descripcion']}</f:fa
cet>
            <!-- row record -->

            #{descripcion.configuracionDescripcionId.tipoDescripcionId.descripcionValorDefinicion}
```

```

        </h:column>

        <h:column>
            <!-- column header -->
            <f:facet
name="header">#{msg['sb.component.boton.opciones']}

```

◆ Funcionamiento general

- **Carga de datos:**

Se ejecuta el siguiente <p:remoteCommand> al activarse la pestaña, que:

- Llama al método cargarDescripcionesProyecto() del proyectoAntiguoConvController.
- Actualiza el panel descripciones.

```

<p:remoteCommand id="cargarDescripcionesRC" name="cargarDescripciones"
                update="descripciones"

actionListener="#{proyectoAntiguoConvController.cargarDescripcionesProyecto}" />

```

- **Bloqueo de interfaz:**

El componente p:blockUI bloquea la sección projecChildrens mientras se ejecuta la carga de las descripciones.

◆ **Visualización de contenido**

- **Panel sin datos:**

Si listaDescripciones está vacía o es null, se muestra un mensaje indicando que no hay descripciones registradas:

```
# {msg[ 'gestion.proyectosProgramas.proyectosAntiguos.descripcion.empty' ]}
```

- **Tabla de descripciones:**

Si existen registros en listaDescripciones, se despliega una tabla HTML (<h:dataTable>) con las siguientes columnas:

◆ **Columnas de la tabla**

Columna	Contenido
Orden	descripcion.configuracionDescripcionId.ordenPresentacion Número que indica la posición que debe ocupar la descripción.
Descripción	descripcion.configuracionDescripcionId.tipoDescripcionId.descripcionValorDefinicion Texto o tipo de contenido configurado para la descripción del proyecto.
Opciones	Botón "Ver", que abre un diálogo con el contenido completo de la descripción.

◆ **Acceso al detalle**

En la columna **Opciones**, cada fila tiene un botón commandLink que:

- Asigna la entidad descripcion seleccionada a la propiedad descripcion del proyectoAntiguoViewController.

- Ejecuta oncomplete="openDialogDescripcion();" para abrir el diálogo modal con el detalle.
- Actualiza el formulario formDialogDescripcion.

```
<p:commandLink value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.btn.ver']}"
oncomplete="openDialogDescripcion();"
update="formDialogDescripcion"
styleClass="btn btn-sm btn-primary">
    <f:setPropertyActionListener value="#{descripcion}"
target="#{proyectoAntiguoViewController.descripcion}"/>
</p:commandLink>
```

◆ Componente adicional: inicialización JS

Se carga dinámicamente el idioma para DataTables usando jQuery (aunque la tabla no es un <p:dataTable> de PrimeFaces, se espera que se le haya aplicado una clase dataTable-descripciones):

```
$(document).ready(function () {
    initTablaNobar("#{request.contextPath}/resources/js/plugins/datatables-
plugins/i18n/#{i18nBean.currentLocale}.lang", ".dataTables-descripciones");
});
```

Este script una función personalizada initTablaNobar(...), que aplica funcionalidades de búsqueda, ordenamiento y traducción.

Subpestaña: Impactos

◆ Propósito

Mostrar la lista de **impactos esperados o generados** del proyecto antiguo, agrupados por tipo y año. Permite conocer los efectos previstos según la planificación del proyecto.

```
<!-- Pestaña Impactos -->
<div id="tab-impactos" class="tab-pane" >
    <div class="panel-body">
        <p:remoteCommand id="cargarImpactosRC" name="cargarImpactos"
update="impactosProyecto"
actionListener="#{proyectoAntiguoViewController.cargarImpactosProyecto()}" />
        <p:blockUI block="projecChildrens" trigger="cargarImpactosRC"
widgetVar="demoBlock">
            #{msg['sb.mensajes.cargando.label']}<br/>
            <div class="progress progress-striped active m-b-sm">
                <div style="width: 100%;" class="progress-bar"></div>
            </div>
        </p:blockUI>
```

```

<h2 style="text-align: center">

<b>#{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.tab.impactos' ]}</b>
</h2>
<br/>
<h:panelGroup id="impactosProyecto">
    <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoAntiguoViewController.detalleProyectoV0.listaImpactos.size() eq 0 or
proyectoAntiguoViewController.detalleProyectoV0.listaImpactos == null}">

    #{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.impactos.empty' ]}
    </h:panelGroup>
    <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoAntiguoViewController.detalleProyectoV0.listaImpactos.size() gt 0}">
        <h:dataTable
value="#{proyectoAntiguoViewController.detalleProyectoV0.listaImpactos}"
var="impacto"

id="#{proyectoAntiguoViewController.modulo}tableImpactos"
styleClass="table table-striped table-
bordered table-hover dataTables"
        >
            <h:column>
                <!-- column header -->
                <f:facet
name="header">#{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.impacto.tipo' ]}</f:facet>
                <!-- row record -->
                #{impacto.configuracionImpactoId.descripcion}
            </h:column>
            <h:column>
                <!-- column header -->
                <f:facet
name="header">#{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.impacto.descripcion' ]}</f:
facet>
                <!-- row record -->
                #{impacto.descripcion != null ?
impacto.descripcion : impacto.impactoEsperado}
            </h:column>
            <h:column>
                <!-- column header -->
                <f:facet
name="header">#{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.impacto.año' ]}</f:facet>
                <!-- row record -->
                #{impacto.anoEsperadoImpacto}
            </h:column>

        </h:dataTable>
        <script>
            $(document).ready(function () {

                initTablaNobar("#{request.contextPath}/resources/js/plugins/datatables-
plugins/i18n/#{i18nBean.currentLocale}.lang", ".dataTables-impactos");
            });

        </script>
    </h:panelGroup>
</h:panelGroup>
</div>
</div>

```

◆ **Funcionamiento general**

- **Carga de datos:**

Utiliza un `<p:remoteCommand>` que se ejecuta para inicializar la pestaña y cargar los impactos desde el backend:

```
<p:remoteCommand id="cargarImpactosRC" name="cargarImpactos"
```

```
update="impactosProyecto"
```

```
actionListener="#{proyectoAntiguoViewController.cargarImpactosProyecto()}" />
```

- **Bloqueo visual durante la carga:**

Se emplea un `p:blockUI` que bloquea visualmente el componente `projecChildrens` mientras se obtienen los datos:

```
<p:blockUI block="projecChildrens" trigger="cargarImpactosRC" widgetVar="demoBlock">
```

◆ **Visualización del contenido**

- **Panel sin impactos:**

Si `listaImpactos` es null o está vacía (`size() eq 0`), se muestra un mensaje tipo well que indica que no hay impactos disponibles:

```
{msg['gestion.proyectosProgramas.proyectosAntiguos.impactos.empty']}
```

- **Tabla de impactos:**

Si existen registros, se muestra una tabla `<h:dataTable>` con clase `dataTables`, lo cual sugiere que luego es transformada en tabla dinámica vía jQuery.

◆ **Columnas de la tabla**

Columna	Contenido
Tipo de Impacto	impacto.configuracionImpactoId.descripcion Define la categoría o naturaleza del impacto (económico, ambiental, social, etc.).
Descripción	Muestra impacto.descripcion, y si este campo está nulo, muestra impacto.impactoEsperado. Esto permite cubrir proyectos antiguos que solo registraban el campo impactoEsperado.
Año Esperado	impacto.anoEsperadoImpacto Corresponde al año en el cual se proyecta que se manifieste el impacto.

◆ **Componente adicional: inicialización DataTables**

Al igual que otras pestañas, el siguiente script inicializa el DataTable usando configuración de idioma localizada:

```
$(document).ready(function () {
    initTablaNobar("#{request.contextPath}/resources/js/plugins/datatables-
    plugins/i18n/#{i18nBean.currentLocale}.lang", ".dataTables-impactos");
});
```

Se infiere que:

- La clase dataTables-impactos debe aplicarse manualmente al componente.
- La tabla se convierte dinámicamente en una tabla interactiva con búsqueda, orden y paginación.

Subpestaña: Coberturas

◆ **Propósito**

Mostrar las **coberturas geográficas** del proyecto antiguo, indicando las **regiones** beneficiadas y su respectivo **porcentaje de impacto** sobre el total del proyecto.

```
<!-- Pestaña Coberturas -->
<div id="tab-coberturas" class="tab-pane" >
    <div class="panel-body">
        <p:remoteCommand id="cargarCoberturasRC" name="cargarCoberturas"
        update="coberturasProyecto"
        actionListener="#{proyectoAntiguoViewController.cargarCoberturasProyecto()}" />
```

```

        <p:blockUI block="projecChildrens" trigger="cargarCoberturasRC"
widgetVar="demoBlock">
            #{msg['sb.mensajes.cargando.label']}<br/>
            <div class="progress progress-striped active m-b-sm">
                <div style="width: 100%;" class="progress-bar"></div>
            </div>
        </p:blockUI>
        <h2 style="text-align: center;">

        <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.coberturas']}</b>
        </h2>
        <br/>
        <h:panelGroup id="coberturasProyecto">
            <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoAntiguoViewController.listCoberturas.size() eq 0 or
proyectoAntiguoViewController.listCoberturas == null}">

                #{msg['gestion.proyectosProgramas.proyectosAntiguos.coberturas.empty']}
                </h:panelGroup>
                <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoAntiguoViewController.listCoberturas.size() gt 0}">
                    <ul class="sortable-list connectList agile-list" id="todo">
                        <ui:repeat var="cobertura"

value="#{proyectoAntiguoViewController.listCoberturas}">
                            <li class="success-element"
id="cobertura#{cobertura.regionImpactoId}">
                                #{cobertura.regionId.nombreRegion}
                                <b>({cobertura.porcentaje}%)</b>
                                <div class="agile-detail">
                                    <i class="fa fa-clock-o"></i>
                                    <h:outputText

value="#{cobertura.fechaCreacion}">
                                        <f:convertDateTime
locale="es" pattern="dd.MM.yyyy"/>
                                    </h:outputText>
                                </div>
                            </li>
                        </ui:repeat>
                    </ul>
                    <script>
                        $(function () {
                            if ($('#todo').height() > 400) {
                                $('#todo').slimScroll({
                                    height: '400px'
                                });
                            }
                        });
                    </script>
                </h:panelGroup>
            </h:panelGroup>
        </div>
    </div>

```

◆ Carga de datos

- Se utiliza un `<p:remoteCommand>` para invocar la acción en el servidor:

```
<p:remoteCommand id="cargarCoberturasRC"
    name="cargarCoberturas"
    update="coberturasProyecto"

    actionListener="#{proyectoAntiguoViewController.cargarCoberturasProyecto()}" />
```

- Mientras se ejecuta la acción, se activa un `<p:blockUI>` sobre el contenedor `projecChildrens`, que muestra una barra de carga:

```
<p:blockUI block="projecChildrens" trigger="cargarCoberturasRC" widgetVar="demoBlock">
```

◆ Visualización del contenido

- El contenido principal se ubica en el componente `<h:panelGroup id="coberturasProyecto">`.
- Si `listCoberturas` está vacío o nulo, se muestra un mensaje de tipo `well` que indica que no existen coberturas registradas:

```
rendered="#{proyectoAntiguoViewController.listCoberturas.size() eq 0 or
proyectoAntiguoViewController.listCoberturas == null}"
```

- Si hay coberturas, se renderiza una lista `<ul>` de elementos `<li>`:

```
<ui:repeat var="cobertura" value="#{proyectoAntiguoViewController.listCoberturas}">
```

◆ Estructura del elemento de cobertura ()

Cada entrada representa una región de impacto con la siguiente información:

Elemento visual	Contenido
 Nombre de la región	<code>#{cobertura.regionId.nombreRegion}</code>
 Porcentaje de cobertura	<code>#{cobertura.porcentaje}%</code>
 Fecha de creación	<code>#{cobertura.fechaCreacion}</code> formateada con dd.MM.yyyy

```
<li class="success-element" id="cobertura#{cobertura.regionImpactoId}">
  #{cobertura.regionId.nombreRegion} <b>({#{cobertura.porcentaje}%}</b>
  <div class="agile-detail">
    <i class="fa fa-clock-o"></i>
    <h:outputText value="#{cobertura.fechaCreacion}">
      <f:convertDateTime locale="es" pattern="dd.MM.yyyy"/>
    </h:outputText>
  </div>
</li>
```

◆ Comportamiento dinámico (scroll)

- Si la altura del listado excede los **400px**, se activa un scroll vertical mediante la librería **jQuery SlimScroll**:

```
$(function () {
  if ($('#todo').height() > 400) {
    $('#todo').slimScroll({
      height: '400px'
    });
  }
});
```

Esto garantiza una buena experiencia visual sin alargar innecesariamente el panel.

Subpestaña: Cronograma de Actividades

◆ Propósito

Presentar de forma tabular el **cronograma de actividades** asociadas al proyecto antiguo, detallando el número de la actividad, su descripción, fechas de inicio y fin, y el tipo de ejecución.

```
<!-- Pestaña Cronograma de Actividades -->
<div id="tab-actividades" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarCronogramaRC" name="cargarCronograma"
    update="cronogramaProyecto"
    actionListener="#{proyectoAntiguoViewController.cargarCronogramaProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarCronogramaRC"
    widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">
      <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.cronograma']}</b>
    </h2>
    <br/>
    <h:panelGroup id="cronogramaProyecto">
```

```

<h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoAntiguoViewController.detalleProyectoV0.listaCronograma.size() eq 0 or
proyectoAntiguoViewController.detalleProyectoV0.listaCronograma == null}">

    #{msg['gestion.proyectosProgramas.proyectosAntiguos.cronograma.empty']}
</h:panelGroup>
<h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoAntiguoViewController.detalleProyectoV0.listaCronograma.size() gt 0}">
    <p:dataTable
id="#{proyectoAntiguoViewController.modulo}tableCronogramas"
var="actividadCronograma"

value="#{proyectoAntiguoViewController.detalleProyectoV0.listaCronograma}"
paginator="true"
selectionMode="single"
reflow="true"

rowKey="#{actividadCronograma.cronogramaProyectoId}"
currentPageReportTemplate="Total de
registros: {totalRecords}"
paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"
rows="10"
>
        <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.cronograma.nActividad']}"
filterStyle="display:none;"
width="10%"
style="text-align: center;"
>
            <h:outputText
value="#{actividadCronograma.numeroActividad}"/>
        </p:column>
        <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.cronograma.actividad']}"
filterStyle="display:none;"
>
            <h:outputText
value="#{actividadCronograma.actividad}"/>
        </p:column>
        <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.cronograma.inicio']}"
filterStyle="display:none;"
width="10%"
style="text-align: center;"
>
            <h:outputText
rendered="#{actividadCronograma.fechaInicial != null}"
value="#{actividadCronograma.fechaInicial}"
            <f:convertDateTime pattern="dd/MM/yyyy"
/>
        </h:outputText>
        <h:outputText
rendered="#{actividadCronograma.periodoInicial != null}"

```



```

value="#{actividadCronograma.periodoInicial}"
                                >
                                </h:outputText>
                                </p:column>
                                <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.cronograma.fin']}"
                                filterStyle="display:none;"
                                width="10%"
                                style="text-align: center;"
                                >
                                <h:outputText
rendered="#{actividadCronograma.fechaFinal != null}"
value="#{actividadCronograma.fechaFinal}">
                                <f:convertDateTime pattern="dd/MM/yyyy"
/>
                                </h:outputText>
                                <h:outputText
rendered="#{actividadCronograma.periodoFinal != null}"
value="#{actividadCronograma.periodoFinal}"
                                >
                                </h:outputText>
                                </p:column>
                                <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.cronograma.ejecucion']}"
                                filterStyle="display:none;"
                                width="10%"
                                style="text-align: center;"
                                >
                                <h:outputText
value="#{actividadCronograma.cronograma == 'D' ? msg['listaDuracion.D'] :
actividadCronograma.cronograma == 'S' ? msg['listaDuracion.S'] : msg['listaDuracion.M']}" />
                                </p:column>
                                </p:dataTable>
                                </h:panelGroup>
                                </h:panelGroup>
                                </div>
</div>

```

◆ Carga de datos

Se realiza mediante un `<p:remoteCommand>` que invoca un método del ViewController:

```

<p:remoteCommand id="cargarCronogramaRC"
                name="cargarCronograma"
                update="cronogramaProyecto"

actionListener="#{proyectoAntiguoViewController.cargarCronogramaProyecto()}" />

```

Este método se acompaña de un `<p:blockUI>` que bloquea la sección `projecChildrens` mientras se cargan los datos:

```
<p:blockUI block="projecChildrens" trigger="cargarCronogramaRC" widgetVar="demoBlock">
```

◆ Visualización

El contenedor principal es:

```
<h:panelGroup id="cronogramaProyecto">
```

- Si listaCronograma está vacío o nulo, se muestra un mensaje en un well:

```
rendered="#{proyectoAntiguoViewController.detalleProyectoV0.listaCronograma.size() eq 0 or  
proyectoAntiguoViewController.detalleProyectoV0.listaCronograma == null}"
```

- Si existen actividades, se renderiza un <p:dataTable> con paginación y filas dinámicas.

◆ Estructura de la tabla

Cada fila representa una actividad cronológica del proyecto. Los campos visualizados son:

Campo	Descripción	Fuente
 N. Actividad	Número secuencial de la actividad	actividadCronograma.numeroActividad
 Actividad	Nombre o descripción de la actividad	actividadCronograma.actividad
 Inicio	Fecha de inicio o período inicial	fechaInicial o periodoInicial
 Fin	Fecha de finalización o período final	fechaFinal o periodoFinal
 Ejecución	Tipo de duración: diaria, semanal o mensual	actividadCronograma.cronograma

Formato de fechas

Las fechas (fechaInicial y fechaFinal) se muestran en formato dd/MM/yyyy:

```
<f:convertDateTime pattern="dd/MM/yyyy" />
```

Tipo de ejecución

El tipo de cronograma se interpreta según el código:

- D → Diario (listaDuracion.D)
- S → Semanal (listaDuracion.S)
- M → Mensual (listaDuracion.M)

◆ Funcionalidades adicionales de la tabla

-  Paginación (paginator="true")
-  Reflujo para pantallas pequeñas (reflow="true")
-  Desactivación de filtros por columna (filterStyle="display:none;")
-  Selección de fila única (selectionMode="single")
-  Navegación entre páginas (paginatorTemplate)

Subpestaña: Tipo Personal

◆ Propósito

Visualizar de forma tabular la distribución del **personal asignado** al proyecto antiguo, detallando el tipo de personal y el número de personas involucradas en el proyecto.

```
<!-- Pestaña Tipo Personal -->
<div id="tab-tipoPersonal" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarTipoPersonalRC" name="cargarTipoPersonal"
update="tipoPersonalProyecto"
actionListener="#{proyectoAntiguoViewController.cargarTipoPersonal()}" />
    <p:blockUI block="projecChildrens" trigger="cargarTipoPersonalRC"
widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">
```

```

<b>#{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.tab.tipoPersonal' ]}</b>
</h2>
<br/>
<h:panelGroup id ="tipoPersonalProyecto"
rendered="#{proyectoAntiguoViewController.proyectoAntiguo}">
    <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoAntiguoViewController.listaTipoPersonal.size() eq 0 or
proyectoAntiguoViewController.listaTipoPersonal == null}">

        #{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.empty' ]}
    </h:panelGroup>
    <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoAntiguoViewController.listaTipoPersonal.size() gt 0}">

        <p:dataTable
id="#{proyectoAntiguoViewController.modulo}tableTipoPersonal"
var="tipoPersonal"

value="#{proyectoAntiguoViewController.listaTipoPersonal}"
paginador="true"
selectionMode="single"
reflow="true"

rowKey="#{tipoPersonal.personalProyectoId}"
currentPageReportTemplate="Total de
registros: {totalRecords}"
paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
paginatorPosition="bottom"

emptyMessage="#{msg[ 'sb.datatable.registro.empty' ]}"
rows="10"
styleClass="table table-striped table-
bordered table-hover"
>
            <p:column>
                <!-- column header -->
                <f:facet
name="header">#{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.tipoPersonal'
]}</f:facet>
                <!-- row record -->

                #{tipoPersonal.tipoPersonalId.descripcionValorDefinicion}
            </p:column>
            <p:column width="30%" style="text-align: center;">
                <!-- column header -->
                <f:facet
name="header">#{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.tipoPersonal.numeroPersona
s' ]}</f:facet>
                <!-- row record -->
                #{tipoPersonal.numeroPersonas}
            </p:column>
        </p:dataTable>
    </h:panelGroup>
</div>
</div>

```

◆ Carga de datos

La carga se ejecuta mediante un `<p:remoteCommand>` que invoca un método del ViewController:

```
<p:remoteCommand id="cargarTipoPersonalRC"
    name="cargarTipoPersonal"
    update="tipoPersonalProyecto"
    actionListener="#{proyectoAntiguoViewController.cargarTipoPersonal()}" />
```

Durante esta operación se bloquea visualmente el contenedor `projecChildrens` mediante un `<p:blockUI>`:

```
<p:blockUI block="projecChildrens" trigger="cargarTipoPersonalRC" widgetVar="demoBlock">
```

◆ Visualización

El contenido se agrupa dentro de:

```
<h:panelGroup id="tipoPersonalProyecto"
    rendered="#{proyectoAntiguoViewController.proyectoAntiguo}">
```

- Si la lista está vacía o nula (`listaTipoPersonal`), se muestra un mensaje de vacío con estilo de resaltado:

```
<h:panelGroup layout="block" styleClass="well"
    rendered="#{proyectoAntiguoViewController.listaTipoPersonal.size() eq 0 or
    proyectoAntiguoViewController.listaTipoPersonal == null}">
```

- Si contiene datos, se renderiza un `<p:dataTable>` con paginación y formato responsivo.

◆ Estructura de la tabla

La tabla muestra dos columnas por registro de personal:

Campo	Descripción	Fuente
 Tipo de Personal	Rol o categoría del personal asignado	tipoPersonal.tipoPersonalId.descripcionValorDefinicion
 Número de Personas	Cantidad de personas asociadas al tipo de personal	tipoPersonal.numeroPersonas

♦ **Características adicionales de la tabla**

-  **Paginación** activada (paginator="true")
-  **Selección de fila única** (selectionMode="single")
-  **Diseño responsivo** (reflow="true")
-  **Mensaje personalizado** para tabla vacía (emptyMessage)
-  **Estilos de tabla Bootstrap** (table-striped, table-bordered, table-hover)

Subpestaña: Personal

♦ **Propósito**

Mostrar el listado de personas asignadas a un proyecto antiguo, incluyendo su rol, nombre completo, número de identificación y un acceso a los detalles individuales de cada participante.

```
<!-- Pestaña Personal -->
<div id="tab-personal" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarPersonasRC" name="cargarPersonas"
update="personasProyecto"
actionListener="#{proyectoAntiguoViewController.cargarPersonasProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarPersonasRC"
widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center;">

    <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.personal']}</b>
```

```

        </h2>
        <br/>
        <h:panelGroup id ="personasProyecto"
rendered="#{proyectoAntiguoViewController.proyectoAntiguo}">
            <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoAntiguoViewController.detalleProyectoV0.listaPersonal.size() eq 0 or
proyectoAntiguoViewController.detalleProyectoV0.listaPersonal == null}">

                #{msg['gestion.proyectosProgramas.proyectosAntiguos.personal.empty']}
            </h:panelGroup>
            <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoAntiguoViewController.detalleProyectoV0.listaPersonal.size() gt 0}">

                <p:dataTable
id="#{proyectoAntiguoViewController.modulo}tablePersonal"
                    var="persona"

value="#{proyectoAntiguoViewController.detalleProyectoV0.listaPersonal}"
                    paginator="true"
                    selectionMode="single"
                    reflow="true"
                    rowKey="#{persona.personaProyectoId}"
                    currentPageReportTemplate="Total de

registros: {totalRecords}"
                    paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
                    paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"
                    rows="10"
                    styleClass="table table-striped table-
bordered table-hover"
                    rowIndexVar="rowIndex"
                >
                    <p:column headerText="#"
                        width="5%"
                        style="text-align: center;"
                    >
                        #{rowIndex+1}
                    </p:column>
                    <p:column>
                        <!-- column header -->
                        <f:facet
name="header">#{msg['gestion.proyectosProgramas.proyectosAntiguos.personal.persona']}</f:facet
et>
                            <!-- row record -->
                            #{persona.nombreCompleto}
                        </p:column>
                        <p:column width="30%" style="text-align: center;">
                            <!-- column header -->
                            <f:facet
name="header">#{msg['gestion.proyectosProgramas.proyectosAntiguos.personal.tipo']}</f:facet>
                            <!-- row record -->
                            #{persona.rolPersonaId != null
?
persona.rolPersonaId.descripcionValorDefinicion
: persona.registroRolId != null
?
persona.registroRolId.descripcionValorDefinicion

```

```

                                :
proyectoAntiguoViewController.cargarRolesPersona(persona.personaProyectoId)}
                                </p:column>
                                <p:column width="20%" style="text-align: center;">
                                    <!-- column header -->
                                    <f:facet
name="header">#{msg['gestion.proyectosProgramas.proyectosAntiguos.personal.identificacion']}
                                </f:facet>

                                    <!-- row record -->
                                    #{persona.numeroIdentificacion}
                                </p:column>
                                <p:column width="10%" style="text-align: center;">
                                    <!-- column header -->
                                    <f:facet
name="header">#{msg['sb.component.boton.detalles']}

```

◆ Carga de datos

Se usa un `<p:remoteCommand>` para invocar el método de carga desde el ViewController:

```

<p:remoteCommand id="cargarPersonasRC"
                name="cargarPersonas"
                update="personasProyecto"
                actionListener="#{proyectoAntiguoViewController.cargarPersonasProyecto()}"
/>

```


Durante esta carga se bloquea visualmente la interfaz:

```
<p:blockUI block="projecChildrens" trigger="cargarPersonasRC" widgetVar="demoBlock">
```

◆ Visualización

Dentro de:

```
<h:panelGroup id="personasProyecto"
rendered="#{proyectoAntiguoViewController.proyectoAntiguo}">
```

- Se muestra un mensaje de lista vacía si no hay personas:

```
<h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoAntiguoViewController.detalleProyectoV0.listaPersonal.size() eq 0 or
proyectoAntiguoViewController.detalleProyectoV0.listaPersonal == null}">
```

- Si hay datos, se renderiza una tabla <p:dataTable> paginada, responsiva y con estilo Bootstrap.

◆ Estructura de la tabla

Cada fila representa a una persona asignada al proyecto, con las siguientes columnas:

#	Campo	Descripción	Fuente
1	# (Índice)	Número de fila incremental	<code>{rowIndex+1}</code>
2	Nombre completo	Nombre completo de la persona asignada	<code>{persona.nombreCompleto}</code>

#	Campo	Descripción	Fuente
3	Tipo / Rol	Rol del personal (usa dos fuentes posibles o invoca un método de carga)	persona.rolPersonalId.descripcionValorDefinicion o persona.registroRolId.descripcionValorDefinicion o proyectoAntiguoViewController.cargarRolesPersona()
4	Identificación	Número de documento o cédula	<code>#{persona.numeroIdentificacion}</code>
5	Detalles	Botón que abre un diálogo para ver detalles de la persona	<code>p:commandLink con actionListener="#{proyectoAntiguoConvController.cargarInfoPersonal(persona)}"</code>

◆ **Funcionalidad adicional**

-  **Selección de fila única** (selectionMode="single")
-  **Paginación activa** (paginator="true")
-  **Responsivo** (reflow="true")
-  **Invocación de diálogo de detalles personales**
-  **JS personalizado para inicializar DataTables internacionalizados:**

```
initTablaNobar("#{request.contextPath}/resources/js/plugins/datatables-  
plugins/i18n/#{i18nBean.currentLocale}.lang", ".dataTables-personal");
```

Subpestaña: Marco Lógico

◆ Propósito

Visualizar la estructura del marco lógico del proyecto antiguo, mostrando:

- Objetivo general del proyecto
- Meta y fuente de verificación
- Objetivos específicos
- Actividades por objetivo
- Resultados por actividad
- Indicadores por resultado

```
<!-- Pestaña Marco Logico -->  
<div id="tab-marco" class="tab-pane" >  
  <div class="panel-body">  
    <p:remoteCommand id="cargarMarcoRC" name="cargarMarcoLogico"  
update="marcoLogicoProyecto"  
actionListener="#{proyectoAntiguoViewController.cargarMarcoLogico()}" />  
    <p:blockUI block="projecChildrens" trigger="cargarMarcoRC"  
widgetVar="demoBlock">  
      #{msg['sb.mensajes.cargando.label']}<br/>  
      <div class="progress progress-striped active m-b-sm">  
        <div style="width: 100%;" class="progress-bar"></div>  
      </div>  
    </p:blockUI>  
    <h2 style="text-align: center;">  
  
    <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.marcoLogico']}</b>  
    </h2>  
    <br/>  
    <h:panelGroup id="marcoLogicoProyecto">  
      <h:panelGroup layout="block" styleClass="well"  
rendered="#{proyectoAntiguoViewController.marcoObjGeneralDTO.objetivo == null}">  
  
      #{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoGeneral.empty']}  
      </h:panelGroup>  
      <ui:fragment  
rendered="#{proyectoAntiguoViewController.marcoObjGeneralDTO.objetivo != null}">  
        <h:panelGroup layout="block" styleClass="well" >  
          <div class="row">  
            <div class="col-lg-12">  
              <dl class="dl-horizontal">  
  
              <dt>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoGral']}</dt> <dd  
class="text-justify">#{proyectoAntiguoViewController.marcoObjGeneralDTO.objetivo}</dd>
```

```

<br/>

<dt>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.meta']}</dt> <dd
class="text-justify">#{proyectoAntiguoViewController.marcoObjGeneralDT0.meta}</dd>
<br/>

<dt>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.fuenteVerificacion']}</d
t> <dd class="text-
justify">#{proyectoAntiguoViewController.marcoObjGeneralDT0.fuenteVerificacion}</dd>
</dl>

</div>
</div>
</h:panelGroup>
</ui:fragment>

<div class="row">
<div class="col-lg-12">

<h2>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivosEspecificos.labe
l']}</h2>

<br/>
</div>
</div>
<h:panelGroup layout="block" styleClass="well"
rendered=("#{proyectoAntiguoViewController.detalleProyectoV0.listaObjetivosEspecificos.size()
eq 0}")>

    #{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivosEspecificos.empty']}
    </h:panelGroup>
    <h:panelGroup layout="block"
rendered=("#{proyectoAntiguoViewController.detalleProyectoV0.listaObjetivosEspecificos.size()
gt 0}")>

        <div class="panel-group" id="accordionObjEspecificos">
            <ui:repeat
value=("#{proyectoAntiguoViewController.detalleProyectoV0.listaObjetivosEspecificos}"
var="objetivoEspecifico">
                <div class="panel panel-default">
                    <div class="panel-heading">
                        <h5 class="panel-title">
                            <a data-
toggle="collapse" data-parent="#accordionObjEspecificos"
href="#objetivoEspecifico#{objetivoEspecifico.numeroObjetivo}">i class="fa fa-arrows-
v"></i>
                            #{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.label']}
                            #{objetivoEspecifico.numeroObjetivo}</a>
                        </h5>
                    </div>
                    <div
id="objetivoEspecifico#{objetivoEspecifico.numeroObjetivo}"
class="panel-collapse
collapse">
                        <div class="panel-body">
                            <h:panelGroup
layout="block" styleClass="well">
                                <div
class="row">
                                    <div
class="col-lg-12">
                                        <dl class="dl-horizontal">

```

```

<dt>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.objeti
vo.label']}</dt> <dd class="text-justify">#{objetivoEspecifico.objetivo}</dd>

</dl>

</div>
</div>
</h:panelGroup>

<h4>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.resultados.label']}</h4>
<br/>
<div class="dd"
id="nestableActividades">
<ol class="dd-
list">

<h:panelGroup layout="block" styleClass="well" rendered="#{empty
objetivoEspecifico.marcoActividadList}">

  #{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.actividade
s.empty']}

</h:panelGroup>

<ui:repeat value="#{objetivoEspecifico.marcoActividadList}"

  rendered="#{not empty objetivoEspecifico.marcoActividadList}"

  var="actividad">

<li class="dd-item" data-id="#{actividad.numeroActividad}">

  <div class="dd-handle">

    <span class="label label-info" data-toggle="collapse" data-
parent="#accordionActividades"></span> Actividad - #{actividad.numeroActividad} :
#{actividad.actividad}

  </div>

  <ol class="dd-list">

    <h:panelGroup layout="block" styleClass="well">

      <div class="row">

        <div class="col-lg-12" style="height: 40px;">

          <dl class="dl-horizontal">

            <dt>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.period
oInicial.label']}</dt> <dd class="text-justify">#{actividad.periodoInicial}</dd>

            <dt>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.period
oFinal.label']}</dt> <dd class="text-justify">#{actividad.periodoFinal}</dd>

```

```

<dt>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.duraci
on.label']}</dt>

<dd class="text-
justify">#{actividad.cronograma == 'M'

    ? msg['listaDuracion.M']

    : actividad.cronograma == 'S'

    ? msg['listaDuracion.S']

    : msg['listaDuracion.D']}</dd>

</dl>

</div>

</div>

</h:panelGroup>

<h:panelGroup layout="block" styleClass="well" rendered="#{empty
actividad.marcoResultadoList}">

    #{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.resultados
.empty']}

</h:panelGroup>

<ui:repeat value="#{actividad.marcoResultadoList}"

    rendered="#{not empty actividad.marcoResultadoList}"

    var="resultado">

    <li class="dd-item" data-id="#{resultado.numero}">

        <div class="dd-handle" >

            <span class="label label-info"></span>
Resultado - #{resultado.numero} : #{resultado.descripcion}

        </div>

        <ol class="dd-list">

            <h:panelGroup layout="block" styleClass="well"
rendered="#{empty resultado.marcoIndicadorList}">

                #{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.indicadore
s.empty']}

            </h:panelGroup>

```

```

value="#{resultado.marcoIndicadorList}"
<ui:repeat
resultado.marcoIndicadorList}"
rendered="#{not empty
var="indicador">
<li class="dd-item" data-
id="#{indicador.numeroIndicador}">
<div class="dd-handle">
<span class="label
label-info"></span> Indicador - #{indicador.numeroIndicador} : #{indicador.indicador}
</div>
<ol class="dd-list">
<li class="dd-item">
<h:panelGroup
layout="block" styleClass="well">
<div
class="row">
<div class="col-lg-12">
<dl class="dl-horizontal">
<dt>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.medioV
erif.label']}:</dt> <dd class="text-justify">#{indicador.medioVerificacion}</dd>
<br/>
<dt>#{msg['gestion.proyectosProgramas.proyectosAntiguos.marco.objetivoEspecifico.mesCum
pli.label']}:</dt> <dd class="text-justify">#{indicador.mesCumplimiento}</dd>
</dl>
</div>
</div>
</h:panelGroup>
</li>
</ol>

```

```
</li>
</ui:repeat>
</ol>
</li>
</ui:repeat>
</ol>
</li>
</ui:repeat>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
```

- ◆ **Carga de datos**

Al cambiar de pestaña se ejecuta un `remoteCommand` que invoca el método de carga correspondiente:

```
<p:remoteCommand id="cargarMarcoRC" name="cargarMarcoLogico"
    update="marcoLogicoProyecto"
    actionListener="#{proyectoAntiguoViewController.cargarMarcoLogico()}" />
```

Se acompaña con una animación de carga usando blockUI:

```
<p:blockUI block="projecChildrens" trigger="cargarMarcoRC" widgetVar="demoBlock">
```

◆ Visualización del Marco Lógico

Dentro de:

```
<h:panelGroup id="marcoLogicoProyecto">
```


✓ 1. Objetivo general del proyecto

Se muestra si existe el valor proyectoAntiguoViewController.marcoObjGeneralDTO.objetivo. Si no existe, se muestra un mensaje de alerta (well Bootstrap).

Si existe, se despliega un resumen del objetivo general, meta y fuente de verificación en un bloque dl-horizontal.

✓ 2. Objetivos Específicos

- Se muestran en un componente tipo accordion colapsable.
- Cada **objetivo específico** tiene una sección con:
 - Descripción del objetivo (objetivoEspecifico.objetivo)
 - Lista de **actividades** relacionadas.

✓ 3. Actividades

Cada actividad incluye:

- Número y descripción: #{actividad.numeroActividad} : #{actividad.actividad}
- Periodo inicial y final
- Duración en texto, basada en el tipo de cronograma:

```
actividad.cronograma == 'M' ? msg['listaDuracion.M'] : ...
```

✓ 4. Resultados por actividad

Anidados dentro de cada actividad:

- Lista de resultados, con número y descripción.
- Si no hay resultados: mensaje de lista vacía (well).

✓ 5. Indicadores por resultado

Cada resultado puede tener varios indicadores, cada uno incluye:

- Número y descripción
- Medio de verificación
- Mes estimado de cumplimiento

◆ Estructura Jerárquica

El contenido se construye con `ui:repeat` y `<ol class="dd-list">` para representar una **estructura tipo árbol**:

plaintext

CopiarEditar

Objetivo General

└─ Objetivos Específicos

└─ Actividades

└─ Resultados

└─ Indicadores

Cada nodo tiene su estructura visual (`dd-handle`, `well`, etc.) para mejorar la presentación y navegación.

✓ Funcionalidades clave

Elemento	Tipo	Descripción
blockUI	Bloqueo visual	Para mostrar carga mientras se procesan los datos
remoteCommand	Llamado AJAX	Ejecuta cargarMarcoLogico() desde el Controller
ui:repeat	Repetición de estructuras	Para recorrer objetivos, actividades, resultados e indicadores
dl-horizontal	Etiquetas y valores	Para mostrar pares "etiqueta: valor" de forma legible
collapse	Interacción visual	Para contraer/expandir objetivos específicos y actividades

2. Pestaña de Evaluación

- Carga mediante **cargarEvaluacion();**
- Contiene detalles del proceso de revisión del proyecto, evaluadores, observaciones técnicas, conceptos o resultados del análisis técnico.
- Esta pestaña está diseñada para presentar al usuario final la información detallada sobre las evaluaciones realizadas a un proyecto antiguo. Se aprovechan los componentes de PrimeFaces para interacción dinámica (AJAX), manejo de bloques durante procesos y uso de bibliotecas JS externas (DataTables) para una mejor experiencia de usuario.

```
<!-- Pestaña Evaluacion -->
<div class="tab-pane" id="tab-evaluacion">
  <p:remoteCommand id="cargarEvaluacionRC" name="cargarEvaluacion"
update="infoEvaluacion"
actionListener="#{proyectoAntiguoViewController.cargarEvaluacionesProyecto()}" />
  <p:blockUI block="projecChildrens" trigger="cargarEvaluacionRC" widgetVar="demoBlock">
    #{msg['sb.mensajes.cargando.label']}<br/>
    <div class="progress progress-striped active m-b-sm">
      <div style="width: 100%;" class="progress-bar"></div>
    </div>
  </p:blockUI>
  <h2 style="text-align: center">

  <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.revisionEvaluacion']}</b>
</h2>
<br/>
<h:panelGroup id="infoEvaluacion" >
```

```

        <h:panelGroup layout="block" styleClass="well"

rendered="#{proyectoAntiguoViewController.listaEvaluaciones.size() eq 0
or
proyectoAntiguoViewController.listaEvaluaciones == null}">
    #{msg['gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.emptyTable']}
    </h:panelGroup>
    <h:panelGroup styleClass="animated fadeInLeft" layout="block"

rendered="#{proyectoAntiguoViewController.listaEvaluaciones.size() gt 0}">
        <h:dataTable value="#{proyectoAntiguoViewController.listaEvaluaciones}"
            var="evaluacion"

id="#{proyectoAntiguoViewController.modulo}tablaEvaluaciones"
            styleClass="table table-striped table-bordered
table-hover dataTables-evaluaciones"
            >
                <h:column headerClass="column1">
                    <!-- column header -->
                    <f:facet
name="header">#{msg['gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.tip
oEvaluacion']}

```

```

<f:facet
name="header">#{msg['sb.component.boton.opciones']}</f:facet>
<!-- row record -->
<p:commandLink style="color: #fff;"

title="#{msg['sb.component.boton.detalles']}"

actionListener="#{proyectoAntiguoConvController.prepararEditar(evaluacion)}"
styleClass="btn btn-xs
btn-primary sbTooltip"

oncomplete="$('#dlgDetallesEvaluacion').modal();"

update="formDetallesEvaluacion"

process="@this"
>
<i class="fa fa-list"></i>
</p:commandLink>
<p:spacer/>
<p:commandLink
title="#{msg['sb.component.boton.download']}"
styleClass=" btn btn-xs
btn-primary sbTooltip"

actionListener="#{proyectoAntiguoConvController.downloadReporteEvaluacion('calificacion
', evaluacion.evaluacionProyectoId)}"

immediate="true"
ajax="false"
>
<i class="fa fa-file-pdf-o"></i>
</p:commandLink>
</h:column>
</h:dataTable>
<script>
// A $( document ).ready() block.
$(document).ready(function () {
    $('#dataTables-evaluaciones').DataTable({
        "language": {
            "url":
"#{request.contextPath}/resources/js/plugins/datatables-
plugins/i18n/#{i18nBean.currentLocale}.lang"
        }
    });
});
</script>
</h:panelGroup>
</h:panelGroup>
</div>

```

1. Propósito del componente

La pestaña de evaluación permite consultar las evaluaciones asociadas a un proyecto antiguo. A través de esta interfaz, el usuario puede:

- Visualizar una tabla con todas las evaluaciones realizadas al proyecto.

- Identificar el tipo de evaluación y evaluador.
- Ver los nombres de los evaluadores asignados.
- Descargar el informe de evaluación.
- Acceder a los detalles de cada evaluación en una ventana modal.

◆ 2. Estructura General

```
<div class="tab-pane" id="tab-evaluacion">
  ...
</div>
```

El componente está contenido en un div con clase tab-pane y id="tab-evaluacion", lo cual permite integrarlo en una estructura de pestañas tipo Bootstrap o similar.

◆ 3. Carga dinámica de información

```
<p:remoteCommand id="cargarEvaluacionRC"
  name="cargarEvaluacion"
  update="infoEvaluacion"
```

```
  actionListener="#{proyectoAntiguoViewController.cargarEvaluacionesProyecto()}" />
```

- **Propósito:** permite cargar de forma dinámica la información de evaluaciones del proyecto seleccionado.
- **Uso:** este comando se invoca desde JavaScript, normalmente cuando el usuario accede a esta pestaña.
- **Acción asociada:** cargarEvaluacionesProyecto() – método del controlador que consulta la base de datos o backend para obtener la lista de evaluaciones.

◆ 4. Bloqueo de interfaz mientras carga

```
<p:blockUI block="projecChildrens" trigger="cargarEvaluacionRC" widgetVar="demoBlock">
  #{msg['sb.mensajes.cargando.label']}
  ...
</p:blockUI>
```

- **Componente:** <p:blockUI> de PrimeFaces
- **Función:** bloquea la sección identificada con block="projecChildrens" mientras se ejecuta el remoteCommand, mostrando un mensaje de carga con una barra de progreso animada.
- **Internacionalización:** el mensaje es dinámico y extraído desde un ResourceBundle.

◆ 5. Encabezado de sección

```
<h2 style="text-align: center">
  <b>#{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.tab.revisionEvaluacion' ]}</b>
</h2>
```

Muestra el título de la pestaña, usando ResourceBundle para soportar múltiples idiomas.

◆ 6. Contenedor de datos de evaluación

```
<h:panelGroup id="infoEvaluacion">
  ...
</h:panelGroup>
```

Agrupar toda la información evaluativa. Este contenedor es el que se actualiza dinámicamente desde el remoteCommand.

◆ 7. Condicional sin evaluaciones

```
<h:panelGroup layout="block" styleClass="well"
  rendered="#{proyectoAntiguoViewController.listaEvaluaciones.size() eq 0
    or proyectoAntiguoViewController.listaEvaluaciones == null}">
  #{msg[ 'gestion.proyectosProgramas.proyectosAntiguos.evaluacion.evaluadores.emptyTable' ]}
</h:panelGroup>
```

- Muestra un mensaje informativo cuando no hay evaluaciones registradas para el proyecto.
- Se utiliza una condición rendered para evaluar si la lista está vacía o es null.

◆ 8. Tabla de Evaluaciones

```
<h:dataTable value="#{proyectoAntiguoViewController.listaEvaluaciones}" var="evaluacion"
...>
...
</h:dataTable>
```

Columnas:

1. Tipo Evaluación

- Obtenido mediante método consultarValorTipoEvaluacion() del proyectoAntiguoConvController.

2. Tipo Evaluador

- Muestra "Interno", "Externo" o vacío según el valor de evaluacion.tipoEvaluador.

3. Nombres Evaluador

- Si es tipo "Panel", muestra nombrePanel.
- Si es interno, muestra funcionario.
- Si es externo, muestra nombre completo usando nombreCompletoPersona().

4. Usuario SIGP

- Muestra el usuario del sistema asignado a la evaluación.

5. Opciones

- Botón "Detalles": abre modal dlgDetallesEvaluacion.
- Botón "Descargar": descarga el PDF de calificación, vía método downloadReporteEvaluacion().

◆ 9. Inicialización de tabla con DataTables.js

```
$(document).ready(function () {
    $('#dataTables-evaluaciones').DataTable({
        "language": {
            "url": "#{request.contextPath}/resources/js/plugins/datatables-
plugins/i18n/#{i18nBean.currentLocale}.lang"
        }
    })
})
```



```
});  
});
```

- Se utiliza jQuery para aplicar el plugin DataTables a la tabla.
- El archivo de lenguaje se selecciona dinámicamente según el idioma actual del sistema.

Integración con lógica de negocio

- **Carga de Evaluaciones:**
proyectoAntiguoViewController.cargarEvaluacionesProyecto()
→ Consulta a la base de datos las evaluaciones asignadas al proyecto actual.
- **Obtención de valores de campos específicos:**
proyectoAntiguoConvController.consultarValorTipoEvaluacion(...)
proyectoAntiguoConvController.nombreCompletoPersona(...)
- **Acciones de botones:**
 - prepararEditar(...): configura los datos necesarios para ver detalles en la ventana modal.
 - downloadReporteEvaluacion(...): genera y entrega el archivo PDF de la evaluación.

3. Pestaña de Documentos

- Carga mediante **cargarDocumentos()**;
- Muestra los archivos adjuntos, plantillas cargadas, fichas del proyecto, resoluciones, anexos, etc.
- La pestaña **Documentos** permite al usuario final acceder a los archivos cargados para el proyecto de una forma visual y sencilla. El diseño basado en tarjetas mejora la usabilidad, mientras que el uso de PrimeFaces garantiza la integración fluida con la lógica del backend para la descarga directa de archivos.

```
<!-- Pestaña Documentos -->
```

```
<div class="tab-pane" id="tab-documentos">
  <p:remoteCommand id="cargarDocumentosRC" name="cargarDocumentos"
update="infoDocumentos"
actionListener="#{proyectoAntiguoViewController.cargarDocumentosProyecto()}" />
  <p:blockUI block="projecChildrens" trigger="cargarDocumentosRC" widgetVar="demoBlock">
    #{msg['sb.mensajes.cargando.label']}<br/>
    <div class="progress progress-striped active m-b-sm">
      <div style="width: 100%;" class="progress-bar"></div>
    </div>
  </p:blockUI>
  <h2 style="text-align: center">
    <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.documentos']}</b>
  </h2>
  <br/>
  <h:panelGroup id="infoDocumentos" >
    <h:panelGroup layout="block" styleClass="well"

rendered="#{proyectoAntiguoViewController.listaDocumentos.size() eq 0
or
proyectoAntiguoViewController.listaDocumentos == null}">
  #{msg['gestion.proyectosProgramas.proyectosAntiguos.documentos.emptyTable']}
  </h:panelGroup>
  <h:panelGroup styleClass="animated fadeInLeft" layout="block"

rendered="#{proyectoAntiguoViewController.listaDocumentos.size() gt 0}">
  <ui:repeat var="documento"
value="#{proyectoAntiguoViewController.listaDocumentos}">
    <div id="documento#{documento.documentoId}" class="col-lg-12
mb-3">
      <div class="card">
        <div class="card-body">
          <div class="row align-items-center">
            <div class="col-lg-10">
              <h5 class="card-title">
                <i class="fa
fa-file"></i>
                #{documento.nombre}
              </h5>
            </div>
            <div class="col-lg-2 text-
right">
              <p:commandLink
id="descargarDoc"
title="#{msg['sb.mensajes.descargar.label']} - #{documento.nombre}"
process="@this"
update="@this"
ajax="false"
rendered="#{documento.nombre != null}"
immediate="true"
actionListener="#{proyectoAntiguoConvController.descargarArchivoDocumentos(documento)}">
```


- Ejecuta el método cargarDocumentosProyecto() en el ViewController.

◆ 4. Bloqueo durante la carga

```
<p:blockUI block="projecChildrens" trigger="cargarDocumentosRC" widgetVar="demoBlock">
  #{msg['sb.mensajes.cargando.label']}<br/>
  <div class="progress progress-striped active m-b-sm">
    <div style="width: 100%;" class="progress-bar"></div>
  </div>
</p:blockUI>
```

- Muestra una animación de carga mientras se ejecuta la acción de carga de documentos.
- Bloquea el componente projecChildrens para evitar interacción del usuario durante el proceso.

◆ 5. Encabezado de sección

```
<h2 style="text-align: center">
  <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.documentos']}</b>
</h2>
```

- Muestra el título principal de la pestaña.
- El texto se obtiene del archivo de propiedades (ResourceBundle) para permitir internacionalización.

◆ 6. Contenedor de la información

```
<h:panelGroup id="infoDocumentos">
  ...
</h:panelGroup>
```

Contiene todo el bloque que muestra los documentos o un mensaje si no existen.

◆ 7. Mensaje sin documentos

```
<h:panelGroup layout="block" styleClass="well"
  rendered="#{proyectoAntiguoViewController.listaDocumentos.size() eq 0}
```

```
or proyectoAntiguoViewController.listaDocumentos == null})">
    #{msg['gestion.proyectosProgramas.proyectosAntiguos.documentos.emptyTable']}
</h:panelGroup>
```

- Muestra un mensaje en pantalla cuando no existen documentos registrados para el proyecto.
- El mensaje se encuentra centralizado en los archivos de propiedades.

◆ 8. Listado de documentos

```
<ui:repeat var="documento" value="#{proyectoAntiguoViewController.listaDocumentos}">
    <div id="documento#{documento.documentoId}" class="col-lg-12 mb-3">
        <div class="card">
            <div class="card-body">
                <div class="row align-items-center">
                    <div class="col-lg-10">
                        <h5 class="card-title">
                            <i class="fa fa-file"></i>
                            #{documento.nombre}
                        </h5>
                    </div>
                    <div class="col-lg-2 text-right">
                        ...
                    </div>
                </div>
            </div>
        </div>
    </div>
</ui:repeat>
```

- Usa ui:repeat para recorrer la lista listaDocumentos.
- Cada documento se renderiza como una tarjeta (card) estilo Bootstrap.
- Muestra el nombre del documento con ícono de archivo.

◆ 9. Descarga de documento

```
<p:commandLink id="descargarDoc"
    title="#{msg['sb.mensajes.descargar.label']} - #{documento.nombre}"
    process="@this"
    update="@this"
    ajax="false"
    rendered="#{documento.nombre != null}"
    immediate="true"

    actionListener="#{proyectoAntiguoConvController.descargarArchivoDocumentos(documento)}"
    styleClass="btn btn-primary">
```

```
<i class="fa fa-download"></i> Descargar  
<p:fileDownload value="#{proyectoAntiguoViewController.archivoDocumento}"/>  
</p:commandLink>
```

- **Acción:** llama a `descargarArchivoDocumentos(documento)` para generar el archivo para descarga.
- **Componente clave:** `<p:fileDownload>` de PrimeFaces. Se asocia al bean que expone el archivo `archivoDocumento`.
- **Propiedades importantes:**
 - `ajax="false"`: se requiere una petición completa (no AJAX) para iniciar la descarga.
 - `immediate="true"`: evita validaciones JSF antes de ejecutar la acción.
 - `process="@this"` y `update="@this"` aseguran procesamiento mínimo.

✿ Integración con la lógica de negocio

- `proyectoAntiguoViewController.cargarDocumentosProyecto()`
→ Se encarga de cargar desde la base de datos los documentos relacionados al proyecto.
- `proyectoAntiguoConvController.descargarArchivoDocumentos(documento)`
→ Crea y expone el objeto `StreamedContent` necesario para el componente `<p:fileDownload>`.
- `proyectoAntiguoViewController.archivoDocumento`
→ Atributo tipo `StreamedContent` que contiene el archivo actual a descargar.

4. Pestaña de Observaciones

- Carga mediante `cargarObservaciones();`
- Permite registrar, editar o visualizar observaciones de los diferentes revisores o entidades.

- La pestaña **Observaciones** proporciona una visualización detallada del historial de actividades realizadas sobre un proyecto. Está diseñada para ser visualmente clara y centrada en la experiencia del usuario, integrando funcionalidades de carga remota, formateo de fechas y scroll dinámico. Esto facilita a los usuarios rastrear los cambios y comentarios importantes realizados por otros participantes.

```
<!-- Pestaña Observaciones -->
<div class="tab-pane" id="tab-observaciones">
  <p:remoteCommand id="cargarObservacionesRC" name="cargarObservaciones"
update="infoObservaciones"
actionListener="#{proyectoAntiguoViewController.cargarObservacionesProyecto()}" />
  <p:blockUI block="projecChildrens" trigger="cargarObservacionesRC"
widgetVar="demoBlock">
    #{msg['sb.mensajes.cargando.label']}<br/>
    <div class="progress progress-striped active m-b-sm">
      <div style="width: 100%;" class="progress-bar"></div>
    </div>
  </p:blockUI>
  <h2 style="text-align: center">
    <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.observaciones']}</b>
  </h2>
  <br/>
  <h:panelGroup id="infoObservaciones" >
    <h:panelGroup layout="block" styleClass="well"

rendered="#{proyectoAntiguoViewController.listaObservacionProyecto.size() eq 0
or
proyectoAntiguoViewController.listaObservacionProyecto == null}">

  #{msg['gestion.proyectosProgramas.proyectosAntiguos.observaciones.emptyTable']}
  </h:panelGroup>
  <h:panelGroup styleClass="feed-activity-list" layout="block"

rendered="#{proyectoAntiguoViewController.listaObservacionProyecto.size() gt 0}">
    <ui:repeat
value="#{proyectoAntiguoViewController.listaObservacionProyecto}"
var="observacion">
      <div class="feed-element" style="margin: 14px">
        <a href="#" class="pull-left">
          
        </a>
        <div class="media-body ">
          <small class="pull-right">
            <div class="ibox-tools open">
            </div>
          </small>
          <strong>#{observacion.nombreUsuario}</strong>

#{observacion.log}. <br/>

          <small class="text-muted">
            Creado
            #{proyectoAntiguoConvController.isBeforeToday(observacion.fechaCreacion)} :
            <h:outputText
value="#{observacion.fechaCreacion}">
```

```

                                <f:convertDateTime
pattern="dd/MM/yyyy hh:mm a" timeZone="EST"/>
                                </h:outputText>
                                <ui:fragment
rendered="#{observacion.fechaModificacion ne null}">
                                - Modificado
                                #{proyectoAntiguoConvController.isBeforeToday(observacion.fechaModificacion)} :
                                <h:outputText
value="#{observacion.fechaModificacion}">
                                <f:convertDateTime
pattern="dd/MM/yyyy hh:mm a" timeZone="EST"/>
                                </h:outputText>
                                </ui:fragment>
                                </small>
                                <ui:fragment
rendered="#{observacion.observacion ne null}">
                                <div class="well">
                                    #{observacion.observacion}
                                </div>
                                </ui:fragment>
                                </div>
                                </div>
                                </ui:repeat>
                                <script>
                                    var scrollear = function scrollearF() {
                                        $(".feed-activity-list").each(function (index) {
                                            if ($(this).height() > 500) {
                                                $(this).slimScroll({
                                                    height: '500px',
                                                    color: '#1AB394'
                                                });
                                            }
                                        });
                                    };
                                    scrollear();
                                </script>
                                </h:panelGroup>
                                </h:panelGroup>
                                </div>

```

◆ 1. Propósito del componente

La pestaña **Observaciones** permite mostrar un historial de mensajes u observaciones realizadas por los usuarios sobre el proyecto. Cada observación incluye el nombre del usuario, la descripción de la acción, la fecha de creación y, si aplica, la de modificación.

◆ 2. Estructura general

```

<div class="tab-pane" id="tab-observaciones">
    ...

```


</div>

Define el contenedor de la pestaña de observaciones con el identificador tab-observaciones, para ser manejado por un sistema de navegación por pestañas.

◆ 3. Carga dinámica de observaciones

```
<p:remoteCommand id="cargarObservacionesRC"
                 name="cargarObservaciones"
                 update="infoObservaciones"

actionListener="#{proyectoAntiguoViewController.cargarObservacionesProyecto()}" />
```

- Permite invocar vía JavaScript el método cargarObservacionesProyecto() para cargar las observaciones del proyecto.
- Actualiza el componente infoObservaciones con los resultados obtenidos.

◆ 4. Indicador de carga (bloqueo de UI)

```
<p:blockUI block="projecChildrens" trigger="cargarObservacionesRC" widgetVar="demoBlock">
    ...
</p:blockUI>
```

- Muestra una animación de carga mientras se procesan las observaciones.
- Bloquea el componente identificado por projecChildrens para evitar interacciones durante la operación.

◆ 5. Título de la sección

```
<h2 style="text-align: center">
    <b>#{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.observaciones']}</b>
</h2>
```

Presenta el título de la pestaña, obtenido del archivo de propiedades para soportar internacionalización.

◆ 6. Contenedor de contenido

```
<h:panelGroup id="infoObservaciones">
    ...
</h:panelGroup>
```

Agrega dinámicamente el contenido que será actualizado con el resultado de la carga remota.

◆ 7. Mensaje de "sin observaciones"

```
<h:panelGroup layout="block" styleClass="well"
    rendered="#{proyectoAntiguoViewController.listaObservacionProyecto.size() eq 0
        or proyectoAntiguoViewController.listaObservacionProyecto ==
        null}">
    #{msg['gestion.proyectosProgramas.proyectosAntiguos.observaciones.emptyTable']}
</h:panelGroup>
```

- Muestra un mensaje informativo si no existen observaciones registradas.
- Se utiliza la clase well de Bootstrap para destacar el mensaje.

◆ 8. Listado de observaciones

```
<h:panelGroup styleClass="feed-activity-list" layout="block"
    rendered="#{proyectoAntiguoViewController.listaObservacionProyecto.size() gt 0}">
    <ui:repeat value="#{proyectoAntiguoViewController.listaObservacionProyecto}"
        var="observacion">
        ...
    </ui:repeat>
</h:panelGroup>
```

- Se utiliza ui:repeat para iterar sobre la lista listaObservacionProyecto.
- Cada observación se presenta con formato tipo "actividad reciente" usando la clase feed-element.

◆ 9. Detalles de cada observación

```
<div class="feed-element">
    ...
    <strong>#{observacion.nombreUsuario}</strong> #{observacion.log}
    ...
    Creado #{proyectoAntiguoConvController.isBeforeToday(observacion.fechaCreacion)} :
```

```
<h:outputText value="#{observacion.fechaCreacion}">
    <f:convertDateTime pattern="dd/MM/yyyy hh:mm a" timeZone="EST"/>
</h:outputText>
...
</div>
```

- Muestra:
 - Usuario que generó la observación (nombreUsuario).
 - Descripción del evento (log).
 - Fecha de creación con formato amigable.
 - Si existe una modificación, también se muestra con su respectiva fecha.
- Se utiliza un método auxiliar isBeforeToday(...) para comparar la fecha contra la fecha actual (ej. mostrar "hoy", "ayer", etc.).

◆ 10. Visualización del contenido de la observación

```
<ui:fragment rendered="#{observacion.observacion ne null}">
    <div class="well">
        #{observacion.observacion}
    </div>
</ui:fragment>
```

- Si la observación contiene un texto adicional, este se muestra enmarcado con estilo well.
- El uso de ui:fragment evita que se renderice un div innecesario si el contenido está vacío.

◆ 11. Estilos y experiencia de usuario

```
<script>
    var scrollear = function scrollearF() {
        $(".feed-activity-list").each(function (index) {
            if ($(this).height() > 500) {
                $(this).slimScroll({
                    height: '500px',
                    color: '#1AB394'
                });
            }
        });
    };
    scrollear();
</script>
```

- Si el contenido de observaciones sobrepasa los 500px de alto, se activa automáticamente un scroll vertical con estilo personalizado usando **jQuery SlimScroll**.
- Mejora la usabilidad manteniendo el contenido dentro de un espacio fijo.

✿ Integración con la lógica del backend

- `proyectoAntiguoViewController.cargarObservacionesProyecto()`
→ Carga desde la base de datos las observaciones asociadas al proyecto.
- `proyectoAntiguoConvController.isBeforeToday(Date)`
→ Método utilitario que permite evaluar y formatear fechas en función del día actual, para enriquecer el contexto de cada observación.

Requisitos de Proyectos Antiguos

Ubicación del componente:

Pantalla asociada a requisitos del proyecto antiguo.

Controladores implicados:

- `requisitoProyectoAntiguoViewController`
- `requisitoProyectoAntiguoConvController`

◆ 1. Propósito del componente

Esta pantalla permite visualizar los **requisitos de la convocatoria** para el proyecto antiguo y sus respectivos documentos adjuntos. Se presenta una tabla de requisitos y un conjunto de acciones asociadas a cada documento, como descarga o visualización.

◆ 2. Estructura general

```
<div class="ibox-content">
  <h:form id="#{requisitoProyectoAntiguoViewController.modulo}Form" styleClass="form-
horizontal">
```

```

<p:remoteCommand name="rc"

update="#{requisitoProyectoAntiguoViewController.modulo}Form:requisitos"

actionListener="#{requisitoProyectoAntiguoViewController.cargarListas()}"
                    process="@this"/>
    <div class="row">
        <div class="col-lg-12" style="overflow-x: scroll;margin-bottom: 40px;">
            <p:dataTable id="requisitos"
                var="item"

value="#{requisitoProyectoAntiguoViewController.requisitosConvocatoria}"

rowKey="#{item.convocatoriaRequisitoDTO.convocatoriaRequisitoId}"
                    reflow="true"
                    editable="true"
                    style="font-size: 13px;"
                    currentPageReportTemplate="Total de

registros: {totalRecords}"

                    paginator="true"
                    paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"

                    paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"
                >
                    <p:column width="30%">
                        <f:facet name="header">
                            <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.requisito.label']}" />
                        </f:facet>
                        <h:outputText
value="#{item.convocatoriaRequisitoDTO.subsanable == 'N' ? '*' : ''}"
#{item.convocatoriaRequisitoDTO.requisitoId.descripcionValorDefinicion}/>
                        </p:column>
                        <p:column width="20%">
                            <f:facet name="header">
                                <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.nombre.archivo.label']}" />
                            </f:facet>
                            <h:outputText
value="#{item.documentoDTO.nombre}" title="#{item.documentoDTO.nombre}"/>
                        </p:column>
                        <p:column width="10%">
                            <f:facet name="header">
                                <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.tamano.label']}" />
                            </f:facet>
                            <h:outputText
value="#{requisitoProyectoAntiguoConvController.calcularTamañoKb(item.documentoDTO.docSize)}"
"

rendered="#{item.documentoDTO ne null}"

                    styleClass="pull-
right"/>
                </p:column>
                <p:column width="12%" style="text-align: center;">
                    <f:facet name="header">

```

```

                                <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.progreso.label']}" />
                                </f:facet>
                                <h:outputText
styleClass="#{item.documentoDTO.nombre != null ? 'badge badge-primary' : ''}"
value="#{item.documentoDTO.nombre != null ?
msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.progreso.ok'] : 'Sin archivo
adjunto'}"
                                style="white-space:
initial;"/>
                                </p:column>
                                <p:column width="10%"
                                style="text-align: center;"
                                rendered="false"
                                >
                                <f:facet name="header">
                                <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.cumple.label']}" />
                                </f:facet>
                                <f:facet name="output">
                                <h:outputText
styleClass="#{item.requisitoDTO.cumple eq 'S' ? 'badge badge-success'
: item.requisitoDTO.cumple eq 'N' ? 'badge badge-danger'
: item.requisitoDTO.cumple eq 'P' ? 'badge badge-default' : ''}"
value="#{msg['listaCumple.'.concat(item.requisitoDTO.cumple ne null ?
item.requisitoDTO.cumple : 'X')]}"/>
                                </f:facet>
                                </p:column>
                                <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.fechaCumplimiento
.label']}"
                                width="10%"
                                rendered="false"
                                >
                                <f:facet name="output">
                                <h:outputText
value="#{item.requisitoDTO.fechaCumplimiento}"
                                <f:convertDateTime
pattern="dd/MM/yyyy" timeZone="EST"/>
                                </h:outputText>
                                </f:facet>
                                </p:column>
                                <p:column exportable="false"
                                width="10%"
                                style="text-align: center;"
                                >
                                <f:facet name="header">
                                <h:outputText
value="#{msg['sb.datatable.registro.opciones']}" />
                                </f:facet>
                                <p:commandLink
                                rendered="false"
                                style="color: #fff"
                                actionListener="#{requisitoProyectoAntiguoConvController.onRequisitoSelect(item)}"
                                update="dlgEditarRequisito"

```

```

        disabled="#{SeguridadSessController.noEsUsuarioRegistro()}"
        title="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.tooltip.cumplimiento']}"
        sbTooltip"
        styleClass="btn btn-xs btn-primary
        process="@this"

        oncomplete="$('#editRequisito').modal();"
        <i class="fa fa-check"></i>
        </p:commandLink>
        <p:spacer width="10"></p:spacer>
        <p:commandLink
title="#{msg['sb.component.boton.adjuntar.add']}"
actionListener="#{requisitoProyectoAntiguoConvController.onRequisitoSelect(item)}"
update="formAdjRequisito"
oncomplete="$('#adjuntarDlg').modal()"
disabled="#{SeguridadSessController.noEsUsuarioRegistro()}"
rendered="false"
styleClass="btn btn-xs btn-primary">
        <i class="fa fa-upload"></i>
        </p:commandLink>
        <p:spacer width="10"></p:spacer>
        <p:commandLink
title="#{msg['sb.component.boton.download']}"
rendered="#{item.documentoDTO ne null}"
onclick="PrimeFaces.monitorDownload(start, stop)" ajax="false"
actionListener="#{requisitoProyectoAntiguoConvController.descargarArchivo(item)}"
        immediate="true"
        styleClass="btn btn-
xs btn-primary"
disabled="#{item.documentoDTO.documentoId == null}">
        <i class="fa fa-download"></i>
        <p:fileDownload
value="#{requisitoProyectoAntiguoViewController.archivo}"/>
        </p:commandLink>
        <p:spacer width="10"></p:spacer>
        <p:commandLink
title="#{msg['sb.component.boton.visualizar']}"
update="dlgVisorRequisitos"
        styleClass="btn btn-
xs btn-primary"
actionListener="#{requisitoProyectoAntiguoConvController.descargarArchivo(item)}"
rendered="#{item.documentoDTO ne null}"
oncomplete="$('#visorRequisitos').modal()"

```

```
disabled="#{item.documentoDTO.documentoId == null}">
    <i class="fa fa-eye"></i>
    </p:commandLink>
  </p:column>
</p:dataTable>
</div>
</div>
<p:spacer></p:spacer>
</h:form>
</div>
```

- El formulario se genera dinámicamente usando el identificador modulo, lo cual permite reutilizar la vista para distintos tipos de proyectos o contextos.
- Contiene una tabla de requisitos y comandos remotos para su carga.

◆ 3. Carga dinámica de datos

```
<p:remoteCommand name="rc"
  update="#{requisitoProyectoAntiguoViewController.modulo}Form:requisitos"
  actionListener="#{requisitoProyectoAntiguoViewController.cargarListas()}"
  process="@this"/>
```

- Comando remoto utilizado para cargar la lista de requisitos desde el backend cuando se invoque por JavaScript.
- Se actualiza únicamente el componente requisitos.

◆ 4. Tabla de requisitos

```
<p:dataTable id="requisitos"
  value="#{requisitoProyectoAntiguoViewController.requisitosConvocatoria}"
  var="item"
  paginator="true"
  editable="true"
  emptyMessage="#{msg['sb.datatable.registro.empty']}"
  ... />
```

- Muestra la lista requisitosConvocatoria.
- Usa paginación y formato adaptativo (reflow).
- Cada fila representa un requisito y su archivo relacionado (si lo tiene).

◆ 5. Columnas de la tabla

✓ Requisito (obligatorio o subsanable)

```
<h:outputText value="#{item.convocatoriaRequisitoDTO.subsanable == 'N' ? '*':''}  
#{item.convocatoriaRequisitoDTO.requisitoId.descripcionValorDefinicion}" />
```

- Si el requisito no es subsanable, se muestra con asterisco *.

📄 Nombre del archivo adjunto

```
<h:outputText value="#{item.documentoDTO.nombre}" title="#{item.documentoDTO.nombre}" />
```

- Muestra el nombre del archivo cargado, si existe.

📦 Tamaño del archivo

```
<h:outputText  
value="#{requisitoProyectoAntiguoConvController.calcularTamañoKb(item.documentoDTO.docSize)}"  
/>
```

- Convierte el tamaño en bytes a kilobytes utilizando una función del controlador.

✓ Estado del archivo (progreso)

```
<h:outputText value="#{item.documentoDTO.nombre != null ? msg['...ok'] : 'Sin archivo  
adjunto'}" />
```

- Muestra un texto con badge indicando si hay archivo cargado.

✗ Columnas ocultas (cumplimiento y fecha de cumplimiento)

```
rendered="false"
```

- Columnas relacionadas con evaluación de cumplimiento, actualmente deshabilitadas.

◆ 6. Acciones por cada requisito

Se agrupan al final en una columna con íconos de acción:

✓ Validación manual (deshabilitada actualmente)

```
<p:commandLink ... oncomplete="$('#editRequisito').modal();">  
<i class="fa fa-check"></i>
```

```
</p:commandLink>
```

Adjuntar nuevo archivo (deshabilitada)

```
<p:commandLink oncomplete="$('#adjuntarDlg').modal()" ...>
  <i class="fa fa-upload"></i>
</p:commandLink>
```

↓ Descargar archivo

```
<p:commandLink
actionListener="#{requisitoProyectoAntiguoConvController.descargarArchivo(item)}">
  <i class="fa fa-download"></i>
  <p:fileDownload value="#{requisitoProyectoAntiguoViewController.archivo}" />
</p:commandLink>
```

- Usa PrimeFaces.monitorDownload(start, stop) para indicar progreso.
- La acción descarga el archivo que fue preparado por el controller.

Visualizar archivo

```
<p:commandLink update="dlgVisorRequisitos" oncomplete="$('#visorRequisitos').modal()">
  <i class="fa fa-eye"></i>
</p:commandLink>
```

- Llama el mismo método descargarArchivo(item) pero orientado a mostrar el documento en un visor (ej. PDF viewer embebido).

7. Consideraciones de seguridad

- El botón de adjuntar y validar está condicionados por SeguridadSessController.noEsUsuarioRegistro().
- Esto asegura que solo los usuarios con permisos puedan realizar acciones sobre los requisitos.

Integración con la lógica del backend

- requisitoProyectoAntiguoViewController.cargarListas()
→ Carga inicial de los requisitos desde la base de datos.

- requisitoProyectoAntiguoConvController.descargarArchivo(item)
→ Recupera el archivo asociado al requisito seleccionado y lo asigna al bean archivo.
- requisitoProyectoAntiguoConvController.calcularTamañoKb(size)
→ Método utilitario que convierte el tamaño del archivo a una unidad legible.

La pantalla de **Requisitos** proporciona una vista clara y funcional de los requisitos solicitados en la convocatoria, mostrando archivos adjuntos y su estado. Aunque algunos botones están deshabilitados, el diseño está preparado para permitir la carga, descarga y validación de documentos de forma controlada. La integración con seguridad por sesión y visualización modal hacen de esta pantalla un componente clave para la trazabilidad documental del proyecto.

[Pantalla Histórico de Proyectos de Beneficios Tributarios generados en años anteriores a 2017 extrayendo la información del SIGP Antiguo](#)

Función:

La pantalla tiene como función principal facilitar la consulta, visualización y descarga de información histórica relacionada con los proyectos y programas generados antes del año 2017, cuya fuente de datos corresponde al sistema SIGP antiguo.

Esta funcionalidad permitirá a los usuarios acceder de manera estructurada y centralizada a registros previos a la migración del sistema actual, conservando el detalle técnico y documental de cada proyecto o beneficio tributario.

Funcionalidades clave:

- Consulta detallada de proyectos de beneficios tributarios antiguos mediante filtros y listado.
- Visualización de información general del proyecto: convocatoria, estado, tipo, entidad, fechas clave, lugar de ejecución, entre otros campos relevantes.
- Acceso a pestañas relacionadas, como:
 - Requisitos del proyecto
 - Detalles del proyecto
 - Descarga de documentos asociados, tales como:
 - Ficha técnica del proyecto
 - Reporte de calificación
 - Reporte de evaluación con información de los evaluadores

Con esta pantalla, se asegura la preservación, trazabilidad y disponibilidad de la información histórica de CTel, facilitando procesos de consulta, auditoría y análisis institucional.

Creación nuevo menú de Proyectos Antiguos

```
--funciones
INSERT INTO HERCULES.SB_FUNCIONES (funcion_id, aplicacion, funcion, nombre_funcion,
tipo_funcion, menu, opcion, url, url_ayuda, funcion_padre, icono, usuario_creador,
fecha_creacion, usuario_modificador, fecha_modificacion)
VALUES (null, 'SIGP_WEB', 'GEST3100', 'Proyectos Antiguos BT', 'F', null, null,
'/app/beneficiosTributarios/proyectosAntiguosBt.xhtml', null, 'GEST3000', null,
'SIGP_WEB', to_date('25-09-2023 08:06:40', 'dd-mm-yyyy hh24:mi:ss'), null, null);

INSERT INTO HERCULES.SB_FUNCIONES (funcion_id, aplicacion, funcion, nombre_funcion,
tipo_funcion, menu, opcion, url, url_ayuda, funcion_padre, icono, usuario_creador,
fecha_creacion, usuario_modificador, fecha_modificacion)
VALUES (null, 'SIGP_WEB', 'GEST3110', 'Requisitos de Proyectos Antiguos BT', 'F', null,
null, '/app/beneficiosTributarios/requisitosAntiguosBt.xhtml', null, 'GEST3100', null,
'SIGP_WEB', to_date('25-09-2023 08:06:40', 'dd-mm-yyyy hh24:mi:ss'), null, null);

--Derechos funciones
INSERT INTO HERCULES.SB_DERECHOS_FUNCIONES (derecho_funcion_id, aplicacion, funcion,
grupo, estado, usuario_creador, fecha_creacion, usuario_modificador, fecha_modificacion)
values (null, 'SIGP_WEB', 'GEST3100', 'SGPS ADMINISTRADOR', 'H', 'SIGP_WEB',
to_date('22-02-2023 07:25:36', 'dd-mm-yyyy hh24:mi:ss'), null, null);

INSERT INTO HERCULES.SB_DERECHOS_FUNCIONES (derecho_funcion_id, aplicacion, funcion,
grupo, estado, usuario_creador, fecha_creacion, usuario_modificador, fecha_modificacion)
values (null, 'SIGP_WEB', 'GEST3110', 'SGPS ADMINISTRADOR', 'H', 'SIGP_WEB',
to_date('22-02-2023 07:25:36', 'dd-mm-yyyy hh24:mi:ss'), null, null);
```

*Crear nuevo campo de entidad antigua en la tabla **SGP_PROYECTOS** en la base de datos sigp*

```
@JoinColumn(name = "ENTIDAD_ID", referencedColumnName = "ENTIDAD_ID")
@ManyToOne
private EntidadAntigua entidadAntiguaId;
```

Este campo establece una relación **ManyToOne** con la entidad EntidadAntigua, permitiendo acceder a los datos históricos de la entidad que presentó el proyecto en convocatorias pasadas.

También se incluyeron sus respectivos **getters y setters**:

```
public EntidadAntigua getEntidadAntiguaId() {
    return entidadAntiguaId;
}

public void setEntidadAntiguaId(EntidadAntigua entidadAntiguaId) {
    this.entidadAntiguaId = entidadAntiguaId;
}
```

Nota: Se usa una clase distinta (EntidadAntigua) para evitar conflictos con la entidad actual (Entidad) y separar la lógica de datos históricos.

Actualización del DTO: ProyectoDTO.java

Para reflejar esta información en el frontend (JSF), se adicionó también el campo correspondiente en el DTO:

```
@Bind(name = "entidadAntiguaId", returnType = EntidadAntiguaDTO.class, type =
EntidadAntigua.class)
private EntidadAntiguaDTO entidadAntiguaId;
```

Con sus métodos de acceso:

```
public EntidadAntiguaDTO getEntidadAntiguaId() {
    return entidadAntiguaId;
}

public void setEntidadAntiguaId(EntidadAntiguaDTO entidadAntiguaId) {
    this.entidadAntiguaId = entidadAntiguaId;
}
```

- Adición del menú **"Proyectos Antiguos"** en la plantilla del menú de Beneficios Tributarios – Sistema del aplicativo de gestión

El siguiente código define un botón en el menú de navegación de la aplicación que permite acceder a la pantalla de proyectos Antiguos del sistema.

```
<!-- Proyectos Antiguos -->
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST3100')}">
    <li class="#{selectedPage eq 'proyectosAntiguosBt' ? 'active':''}">
        <h:commandLink
            actionListener="#{proyectoBeneficioAntiguoSessController.restablecerLista()}"
            immediate="true">
            <i class="ui-link ui-widget" style="font-style:
normal;">#{msg['sb.menu.gestion.beneficiosTributarios.proyectosAntiguos']}</i>
        </h:commandLink>
    </li>
</ui:fragment>
```

El acceso al módulo de histórico de proyectos fue agregado dentro del **template principal** del sistema, en el bloque de opciones de menú, utilizando JSF (<ui:fragment>, h:commandLink) y control de seguridad a través del SeguridadSessController.

Código del Menú

```
<!-- Proyectos Antiguos -->
```

```
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST3100')}">
    <li class="#{selectedPage eq 'proyectosAntiguosBt' ? 'active':''}">
        <h:commandLink
            actionListener="#{proyectoBeneficioAntiguoSessController.restablecerLista()}"
            immediate="true">
            <i class="ui-link ui-widget" style="font-style:
                normal;">#{msg['sb.menu.gestion.beneficiosTributarios.proyectosAntiguos']}</i>
        </h:commandLink>
    </li>
</ui:fragment>
```

Explicación Técnica

- **Control de acceso:** El ítem de menú solo se renderiza si el usuario tiene el permiso 'GEST3100', verificado a través de SeguridadSessController.mapaMenu.
- **Selección activa:** Se aplica la clase CSS active si la página actual es proyectosAntiguos, permitiendo resaltar visualmente el ítem seleccionado en el menú.
- **Acción de navegación:** El menú ejecuta el método restablecerLista() del controlador proyectoBeneficioAntiguoSessController para inicializar o refrescar la lista de proyectos antiguos.
- **Internacionalización:** El nombre del ítem se obtiene desde un archivo de mensajes (#{msg[...]}), permitiendo compatibilidad multilinguaje.

Controlador Referenciado

El método llamado al hacer clic:

```
public void restablecerLista() {
    buscarVisible = true;
    verProyecto = false;
    init();
    this.redirectPage("app/proyectosBeneficios/proyectosAntiguosBt.xhtml");
}
```

Este método se asegura de cargar la información actualizada de los proyectos antiguos antes de mostrar la vista asociada.

Observaciones

- Es importante usar immediate="true" en el h:commandLink para evitar validaciones JSF innecesarias antes de la carga.
- Esta opción de menú da acceso a la vista proyectosAntiguos.xhtml, ubicada en el módulo gestion/beneficiosTributarios.

Controlador de Vista ProyectoBeneficioAntiguoSessController

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.sb.colciencias.gestion.web.controller.beneficiosTributarios.session;

import com.sb.colciencias.gestion.web.controller.proyectos.session.*;
import com.sb.colciencias.gestion.business.domain.scienti.ScientiPersonaFormacion;
import com.sb.colciencias.gestion.business.facade.administracion.AdministracionServiceFacade;
import com.sb.colciencias.gestion.business.facade.beneficiosTributarios.BeneficiosServiceFacade;
import com.sb.colciencias.gestion.business.facade.configuracion.ConfiguracionServiceFacade;
import com.sb.colciencias.gestion.business.facade.proyectos.ProyectosServiceFacade;
import com.sb.colciencias.gestion.business.transfer.ConvocatoriaCampoDTO;
import com.sb.colciencias.gestion.business.transfer.ConvocatoriaDTO;
import com.sb.colciencias.gestion.business.transfer.ConvocatoriaMenuDTO;
import com.sb.colciencias.gestion.business.transfer.DescripcionDTO;
import com.sb.colciencias.gestion.business.transfer.DocumentoDTO;
import com.sb.colciencias.gestion.business.transfer.FocoDTO;
import com.sb.colciencias.gestion.business.transfer.ImportacionDetalladaDTO;
import com.sb.colciencias.gestion.business.transfer.MarcoObjGeneralDTO;
import com.sb.colciencias.gestion.business.transfer.ObjetivoEspecificoDTO;
import com.sb.colciencias.gestion.business.transfer.ObservacionProyectoDTO;
import com.sb.colciencias.gestion.business.transfer.OtroResultadoDTO;
import com.sb.colciencias.gestion.business.transfer.PermisoDTO;
import com.sb.colciencias.gestion.business.transfer.PersonaProgramaDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoEntidadDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoEvaluacionDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoPersonaDTO;
import com.sb.colciencias.gestion.business.transfer.RegionImpactoDTO;
import com.sb.colciencias.gestion.business.transfer.RegistroImportacionDTO;
import com.sb.colciencias.gestion.business.transfer.RubroDTO;
import com.sb.colciencias.gestion.business.transfer.TipoPersonalDTO;
import com.sb.colciencias.gestion.business.transfer.TramiteEstimuloDTO;
import com.sb.colciencias.gestion.business.transfer.UsuarioDTO;
import com.sb.colciencias.gestion.business.transfer.ValorDefinicionDTO;
import com.sb.colciencias.gestion.business.valueObjects.DetalleProyectoVO;
import com.sb.colciencias.gestion.business.valueObjects.ValoresRubroVO;
import com.sb.colciencias.gestion.web.controller.application.ConfigApplController;
import com.sb.colciencias.gestion.web.controller.proyectos.view.ProyectoAntiguoViewController;
import com.sb.colciencias.gestion.web.util.security.SeguridadConvController;
import com.sb.standard.web.controller.session.GenericSessionController;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Collection;
import java.util.Date;
import java.util.HashMap;
import java.util.List;

```

```
import java.util.Map;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.annotation.PostConstruct;
import javax.ejb.EJB;
import javax.enterprise.context.SessionScoped;
import javax.faces.event.ValueChangeEvent;
import javax.inject.Inject;
import javax.inject.Named;
import org.primefaces.PrimeFaces;
import org.primefaces.model.LazyDataModel;
import org.primefaces.model.SortOrder;
import org.primefaces.model.FilterMeta;
import java.util.TreeMap;
import org.primefaces.model.StreamedContent;

/**
 *
 * @author lsuarez
 */
@Named(value = "proyectoBeneficioAntiguoSessController")
@SessionScoped
public class ProyectoBeneficioAntiguoSessController extends
GenericSessionController<ProyectoDTO> implements Serializable {

    @EJB
    private BeneficiosServiceFacade beneficiosServiceFacade;
    @EJB
    private ProyectosServiceFacade proyectosServiceFacade;
    @EJB
    private ConfiguracionServiceFacade configuracionServiceFacade;
    @EJB
    private AdministracionServiceFacade administracionServiceFacade;
    @Inject
    private EvaluarProyectoSessController evaluarProyectoSessController;
    @Inject
    private SeguridadConvController seguridadConvController;
    @Inject
    private ConfigApplController configApplController;
    //
    private LazyDataModel<ProyectoDTO> model;
    private ValoresRubroVO valoresRubroVO;
    private Map<String, ValorDefinicionDTO> listaEstados;
    private List<ValorDefinicionDTO> listaTiposTramites;
    private List<ValorDefinicionDTO> listaTramites;
    private List<ObservacionProyectoDTO> listaObservacionProyecto;
    private String urlReportes;
    private List<ConvocatoriaDTO> listaConvocatorias;
    private ProyectoDTO proyectoFilter;
    private ValorDefinicionDTO tipoTramite;
    private DetalleProyectoVO detalleProyectoVO;
    private List<Object[]> listRubros;
    private List<RubroDTO> rubrosDetalleProyectoList;
    private String descripcionRubro;
    private MarcoObjGeneralDTO marcoObjGeneralDTO;
    private List<PermisoDTO> listaPermisos;
    private Map<String, ValorDefinicionDTO> listaPermisosContrato;
    private ProyectoDTO programaSelected;
    ///
    private List<ProyectoDTO> listaProyectoTramites;
    private boolean tramitesSecundarios;
```



```
private boolean buscarVisible;
private String tipoExencion;

private String nombreDependenciaConvocatoria;
private List<DescripcionDTO> listaDescripciones;
private DescripcionDTO descripcion;
private RubroDTO rubroSelected;
private List<OtroResultadoDTO> listResultadoDesarrolloTec;
private List<OtroResultadoDTO> listResultadoGeneracion;
private List<RegionImpactoDTO> listCoberturas;
private List<TipoPersonalDTO> listaTipoPersonal;
private HashMap<String, HashMap<String, String>> mapaCampos;
private Map<String, ConvocatoriaMenuDTO> listaMenu;
private ProyectoPersonaDTO proyectoPersonaDTO;
private PersonaProgramaDTO personaProgramaDTO;
private List<UsuarioDTO> listaUsuarios;
private List<ValorDefinicionDTO> listRolPersonas;
private String descRolPersona;
private boolean verProyecto;
private String parameterUrlCvLAC;
private List<ScientiPersonaFormacion> listaScientiPersonaFormacions;
private List<ProyectoEvaluacionDTO> listaEvaluaciones;
private List<DocumentoDTO> listaDocumentos;
private List<ProyectoEntidadDTO> listaProyectoEntidades;
private ProyectoEntidadDTO proyectoEntidadSelected;

// Evaluacion
private Integer countEvaluaciones;
private ProyectoEvaluacionDTO proyectoEvaluacionSelected;
private DocumentoDTO documentoEvaluacion;
private StreamedContent archivoEvaluacion;
private ValorDefinicionDTO documentoEvaluacionTipo;

//Documentos
private ValorDefinicionDTO documentoTipoSgpProyectos;
private ValorDefinicionDTO documentoTipoSgpProyectosXml;
private ValorDefinicionDTO documentoTipoSgpRegistroProyectos;
private StreamedContent archivoDocumento;

// Documento Hoja de Vida Personal
private ValorDefinicionDTO tipoDocHojaVida;
private StreamedContent archivoHojaVida;

//Importaciones
private List<RegistroImportacionDTO> registroImportacionesList;
private RegistroImportacionDTO registroImportacionSelected;
private ValorDefinicionDTO documentoImportacionTipo;
private StreamedContent archivoImportacion;

// Documento Soportes título de doctorado
private ValorDefinicionDTO tipoDocSoporteTituloDoc;
private StreamedContent archivoDocSoporteTituloDoc;

// Documento Soportes de funciones del personal con título de doctorado
private ValorDefinicionDTO tipoDocSoporteFuncionesDoc;
private StreamedContent archivoDocSoporteFuncionesDoc;

private List<FocoDTO> listaFocos;
private List<TramiteEstimuloDTO> listaTramiteEstimulos;
//
List<ObjetivoEspecificoDTO> listaObjetivosEspecificos;
```

```
//
private ArrayList<Integer> listAños;
// Totales
private double totalMonInstitucionSol;
private double totalConEspecieSol;
private double totalConEfectivoSol;
private double totalSol;

private static final org.apache.log4j.Logger LOG =
org.apache.log4j.Logger.getLogger(ProyectoBeneficioAntiguoSessController.class);

private boolean filtroActivo;

public ProyectoBeneficioAntiguoSessController() {
    super(ProyectoDTO.class);
    this.proyectoFilter = new ProyectoDTO();
    listaMenu = new TreeMap<>();
}

@PostConstruct
public void init() {

    this.model = new LazyDataModel<ProyectoDTO>() {

        @Override
        public List<ProyectoDTO> load(int first, int pageSize, String sortField,
SortOrder sortOrder, Map<String, FilterMeta> filterBy) {
            Map<String, Object> filters = new TreeMap<>();
            filterBy.forEach((k, v) -> {
                if (k.equals("globalFilter")) {
                    if (v.getFilterValue() != null &&
!v.getFilterValue().toString().isEmpty()) {
                        filters.put(k, v.getFilterValue());
                    }
                }
            });
            List<ProyectoDTO> proyectoList = new ArrayList<>();
            try {
                //filtro por búsqueda avanzada
                if (proyectoFilter != null) {
                    filters.put("proyectoFiltro", proyectoFilter);
                }
                filters.put("proyectosAntiguos", true); //filtro por proyectos antiguos
                filters.put("claveNull", true); //filtro por clave nula (ya que los
proyectos antiguos no la registraban)
                int count =
beneficiosServiceFacade.proyectosBeneficiosListCount(filters);
                if (count > 0) {
                    proyectoList =
beneficiosServiceFacade.proyectosBeneficiosList(first, pageSize, sortField,
sortOrder.equals(SortOrder.ASCENDING) ? "A" : "D", filters);
                }
                model.setRowCount(count);
            } catch (Exception ex) {
                LOG.error("Error al cargar el listado de proyectos antiguos ", ex);
            }
            return proyectoList;
        }
    };
}
```

```
@Inject
public void cargarListas() {
    try {
        listaEstados = configApplController.getMapValorDefinicion().get("ESTADO");
        // System.out.println("listaEstados SIZE: "+ listaEstados.size());
        listaConvocatorias =
beneficiosServiceFacade.obtenerConvocatoriasEstimulosAntiguos();
        // System.out.println("listaConvocatorias Antiguas SIZE: " +
listaConvocatorias.size());
        listaPermisosContrato =
configApplController.getMapValorDefinicion().get("CONTPERM");
        urlReportes =
beneficiosServiceFacade.obtenerParametroSistema("URLREPORTE").getCadena();
        documentoImportacionTipo =
configApplController.getMapValorDefinicion().get("TIPDOCREC").get("Plan de importación para
solicitudes de exencion de IVA");
        tipoDocSoporteTituloDoc =
configApplController.getMapValorDefinicion().get("TIPDOCPRY").get("Soportes título de
doctorado");
        tipoDocSoporteFuncionesDoc =
configApplController.getMapValorDefinicion().get("TIPDOCPRY").get("Soportes de funciones del
personal con título de doctorado");
        parameterUrlCvLAC =
beneficiosServiceFacade.obtenerParametroSistema("URLCVLAC2").getCadena();
        //
        documentoEvaluacionTipo =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("PRY_EVALUACION");
        documentoTipoSgpProyectos =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_PROYECTOS");
        documentoTipoSgpProyectosXml =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_PROYECTOS_XML");
        documentoTipoSgpRegistroProyectos =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_REGISTRO_PROYECTOS");
        //
        cargarTiposTramites();
        // cargamos el listado de años de los proyectos
        Date fechaFinProy =
beneficiosServiceFacade.obtenerParametroSistema("PRBENFCHAI").getFecha();
        Calendar fechaFinal = Calendar.getInstance();
        fechaFinal.setTime(fechaFinProy);
        int anioInicio = 2000;
        //
        int añoFinal = fechaFinal.get(Calendar.YEAR);
        listAnios = new ArrayList<>();
        for (int i = anioInicio; i <= añoFinal; i++) {
            listAnios.add(i);
        }

    } catch (Exception ex) {

}

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public void setTipoProyecto(ValueChangeEvent event) {
    String dato = String.valueOf(event.getNewValue());
    switch (dato) {
        case "F":
            this.proyectoFilter.setAnteproyecto("S");
            this.proyectoFilter.setExencion("R");
```

```
        break;
    case "R":
        this.proyectoFilter.setExencion("R");
        this.proyectoFilter.setAnteproyecto("N");
        break;
    case "P":
        this.proyectoFilter.setExencion("P");
        this.proyectoFilter.setAnteproyecto(null);
        break;
    default:
        break;
    }
}

public LazyDataModel<ProyectoDTO> getModel() {
    return model;
}

public void setModel(LazyDataModel<ProyectoDTO> model) {
    this.model = model;
}

public ValoresRubroVO getValoresRubroVO() {
    return valoresRubroVO;
}

public void setValoresRubroVO(ValoresRubroVO valoresRubroVO) {
    this.valoresRubroVO = valoresRubroVO;
}

public void cargarDetallesProyecto(ProyectoDTO proyectoDTO) {
    try {
        this.selected = proyectoDTO;

this.selected.setFormularioProyectoId(beneficiosServiceFacade.obtenerFormularioProyectoId(se
lected.getProyectoId()));
        valoresRubroVO =
beneficiosServiceFacade.montosRubroProyecto(proyectoDTO.getProyectoId(), null);
        listaObservacionProyecto =
beneficiosServiceFacade.obtenerObservaciones(proyectoDTO.getProyectoId());

selected.setPalabraClaveList(beneficiosServiceFacade.obtenerPalabrasClave(selected.getProyec
toId()));
        cargarCamposConvocatoria();
        cargarPersonasProyecto();
        cargarListaMenusConvocatoria();
        cargarPresupuestoProyecto();
        Map<String, Object> filters = new HashMap<>();
        filters.put("proyectoId", this.selected.getProyectoId());
        countEvaluaciones =
beneficiosServiceFacade.proyectosEvaluacionListCount(filters);

        // Objetivos especificos (Menu)
        listaObjetivosEspecificos =
beneficiosServiceFacade.objetivosEspecificosList(this.selected.getProyectoId());

        cargarInformacionBasicaProyecto();

    } catch (Exception ex) {
```

```
Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

    public void cargarInformacionBasicaProyecto() {
        try {
//
selected.setRolEntidadList(beneficiosServiceFacade.obtenerRolesEntidad(selected.getProyectoId()));
                nombreDependenciaConvocatoria =
proyectosServiceFacade.obtenerDependenciaConvocatoria(selected.getConvocatoriaId().getConvocatoriaId());
                detalleProyectoVO =
beneficiosServiceFacade.obtenerDetallesProyecto(selected.getProyectoId());
                cargarTramitesEstimulos();
        } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

    public void cargarEntidadesProyecto() {
        try {
            listaProyectoEntidades = new ArrayList<>();
            listaProyectoEntidades =
beneficiosServiceFacade.obtenerProyectoEntidades(selected.getProyectoId());
        } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

    public void cargarRolEntidadesProyecto() {
        try {

selected.setRolEntidadList(beneficiosServiceFacade.obtenerRolesEntidad(selected.getProyectoId()));
        } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

    public void cargarPresupuestoProyecto() {
        try {
            listRubros = new ArrayList<>();
            listRubros =
beneficiosServiceFacade.obtenerRubrosProyAntiguos(selected.getProyectoId());
            calcularTotales();
        } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
```

```
// Método para calcular los totales
public void calcularTotales() {
    totalMonInstitucionSol = 0;
    totalConEspecieSol = 0;
    totalConEfectivoSol = 0;
    totalSol = 0;

    if (listRubros != null && !listRubros.isEmpty()) {
        for (Object[] rubro : listRubros) {
            totalMonInstitucionSol += ((Number) rubro[1]).doubleValue();
            totalConEspecieSol += ((Number) rubro[2]).doubleValue();
            totalConEfectivoSol += ((Number) rubro[3]).doubleValue();
            totalSol += ((Number) rubro[4]).doubleValue();
        }
    }
}

public void cargarImpactosProyecto() {
    try {
        detalleProyectoVO =
projectosServiceFacade.obtenerImpactosProyecto(selected.getProjectoId());
    } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public void cargarCoberturasProyecto() {
    try {
        listCoberturas =
projectosServiceFacade.obtenerCoberturas(selected.getProjectoId());
    } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public void cargarMarcoLogico() {
    try {
        detalleProyectoVO =
projectosServiceFacade.cargarMarcoLogicoProyecto(selected.getProjectoId());
        marcoObjGeneralDTO = detalleProyectoVO.getListaObjetivosGeneral() != null &&
!detalleProyectoVO.getListaObjetivosGeneral().isEmpty() ?
detalleProyectoVO.getListaObjetivosGeneral().get(0) : null;
    } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public void cargarTipoPersonal() {
    try {
        listaTipoPersonal =
projectosServiceFacade.obtenerTipoPersonal(selected.getProjectoId());
    } catch (Exception ex) {

Logger.getLogger(ProyectoAntiguoViewController.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

```

    }
}

public void cargarPersonasProyecto() {
    try {
        //Tipo Documento para Hoja de Vida
        tipoDocHojaVida =
configAppController.getMapValorDefinicion().get("TIPDOCPRY").get("Documento Hoja de Vida
Personal No CvLAC");

        detalleProyectoVO =
projectosServiceFacade.cargarPersonasProyecto(selected.getProjectoId());
    } catch (Exception ex) {
        LOG.error("Error al cargar las personas del proyecto", ex);
    }
}

public void cargarCronogramaProyecto() {
    try {
        detalleProyectoVO =
projectosServiceFacade.cargarCronogramaProyecto(selected.getProjectoId());
    } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public void cargarMontosProyecto(ProyectoDTO proyectoDTO) {
    try {
        this.selected = proyectoDTO;

        cargarListaProyectosTramites(proyectoDTO.getProjectoId());
        valoresRubroVO =
beneficiosServiceFacade.montosRubroProyecto(proyectoDTO.getProjectoId(), tipoExencion);
    } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public boolean renderCampo(String campo, String pantalla) {
    HashMap<String, String> camposValor;
    String render = "";
    if (mapaCampos != null) {
        if (mapaCampos.containsKey(pantalla)) {
            camposValor = mapaCampos.get(pantalla);
            if (camposValor.containsKey(campo)) {
                render = camposValor.get(campo);
            } else {
                render = "S";
            }
        }
    }
    return render.equals("S");
}

private void cargarCamposConvocatoria() {
    try {

```

```

        List<ConvocatoriaCampoDTO> listConvocatoriaCampos =
projectosServiceFacade.buscarCamposConvocatoria(selected.getConvocatoriaId().getConvocatoria
Id());
        HashMap<String, String> camposValor;
        mapaCampos = new HashMap<>();
        if (listConvocatoriaCampos != null && !listConvocatoriaCampos.isEmpty()) {
            for (ConvocatoriaCampoDTO campo : listConvocatoriaCampos) {
                if
(!mapaCampos.containsKey(campo.getPantallaId().getTipoValidacion().getTipoDefinicion())) {
                    camposValor = new HashMap<>();
                    if (campo.getCampoId() != null) {
                        camposValor.put(campo.getCampoId().getValorDefinicion(),
campo.getVisible());
                    }
                    mapaCampos.put(campo.getPantallaId().getTipoValidacion().getTipoDefinicion(), camposValor);
                }
                } else if (campo.getCampoId() != null) {
                    camposValor =
mapaCampos.get(campo.getPantallaId().getTipoValidacion().getTipoDefinicion());
                    camposValor.put(campo.getCampoId().getValorDefinicion(),
campo.getVisible());
                    mapaCampos.put(campo.getPantallaId().getTipoValidacion().getTipoDefinicion(), camposValor);
                }
            }
        } catch (Exception ex) {
            LOG.error("Error al consultar los campos de la convocatoria", ex);
        }
    }

    public boolean existeMenu(String funcion) {
        return listaMenu.containsKey(funcion);
    }

    private void cargarListaMenusConvocatoria() {
        try {
            listaMenu = new HashMap<>();
            List<ConvocatoriaMenuDTO> listaMenuDTO =
projectosServiceFacade.obtenerConvocatoriaMenu(this.selected.getConvocatoriaId());
            if (listaMenuDTO != null && !listaMenuDTO.isEmpty()) {
                for (ConvocatoriaMenuDTO convocatoriaMenuDTO : listaMenuDTO) {
                    if (convocatoriaMenuDTO.getMenuId() != null) {
                        String funcion = convocatoriaMenuDTO.getMenuId().getComentario();
                        if (funcion != null && !funcion.equals("")) {
                            listaMenu.put(funcion, convocatoriaMenuDTO);
                        }
                    }
                }
            }
        } catch (Exception ex) {
            LOG.error("Error al cargar la lista de menus");
        }
    }

    public String cargarRolesPersona(Long personaProyectoId) {
        try {
            if (proyectoPersonaDTO != null && personaProyectoId != null) {

```



```
        listRolPersonas =
proyectosServiceFacade.obtenerRolesPersona(proyectoPersonaDTO.getPersonaProyectoId());
        descRolPersona = new String();
        if (listRolPersonas != null && !listRolPersonas.isEmpty()) {
            listRolPersonas.forEach((valorDefinicionDTO) -> {
                descRolPersona += valorDefinicionDTO.getDescripcionValorDefinicion()
+ ", ";
            });
        }
        return descRolPersona;
    }
} catch (Exception ex) {
    LOG.error("Error al cargar los roles de la persona", ex);
}
return descRolPersona;
}

public Map<String, ValorDefinicionDTO> getListaEstados() {
    return listaEstados;
}

public void setListaEstados(Map<String, ValorDefinicionDTO> listaEstados) {
    this.listaEstados = listaEstados;
}

public List<ObservacionProyectoDTO> getListaObservacionProyecto() {
    return listaObservacionProyecto;
}

public void setListaObservacionProyecto(List<ObservacionProyectoDTO>
listaObservacionProyecto) {
    this.listaObservacionProyecto = listaObservacionProyecto;
}

public String getUrlReportes() {
    return urlReportes;
}

public void setUrlReportes(String urlReportes) {
    this.urlReportes = urlReportes;
}

public List<ValorDefinicionDTO> getListaTramites() {
    return listaTramites;
}

public void setListaTramites(List<ValorDefinicionDTO> listaTramites) {
    this.listaTramites = listaTramites;
}

public List<ConvocatoriaDTO> getListaConvocatorias() {
    return listaConvocatorias;
}

public void setListaConvocatorias(List<ConvocatoriaDTO> listaConvocatorias) {
    this.listaConvocatorias = listaConvocatorias;
}

public ProyectoDTO getProyectoFilter() {
    return proyectoFilter;
}
}
```

```
public void setProyectoFilter(ProyectoDTO proyectoFilter) {
    this.proyectoFilter = proyectoFilter;
}

public ValorDefinicionDTO getTipoTramite() {
    return tipoTramite;
}

public void setTipoTramite(ValorDefinicionDTO tipoTramite) {
    this.tipoTramite = tipoTramite;
}

public List<ValorDefinicionDTO> getListaTiposTramites() {
    return listaTiposTramites;
}

public void setListaTiposTramites(List<ValorDefinicionDTO> listaTiposTramites) {
    this.listaTiposTramites = listaTiposTramites;
}

public List<ProyectoDTO> getListaProyectoTramites() {
    return listaProyectoTramites;
}

public void setListaProyectoTramites(List<ProyectoDTO> listaProyectoTramites) {
    this.listaProyectoTramites = listaProyectoTramites;
}

public boolean isTramitesSecundarios() {
    return tramitesSecundarios;
}

public void setTramitesSecundarios(boolean tramitesSecundarios) {
    this.tramitesSecundarios = tramitesSecundarios;
}

public boolean isBuscarVisible() {
    return buscarVisible;
}

public void setBuscarVisible(boolean buscarVisible) {
    this.buscarVisible = buscarVisible;
}

public String getTipoExencion() {
    return tipoExencion;
}

public void setTipoExencion(String tipoExencion) {
    this.tipoExencion = tipoExencion;
}

public void cargarListaProyectosTramites(Long proyectoId) {
    try {
        listaProyectoTramites =
beneficiosServiceFacade.obtenerProyectosTramites(proyectoId);
        if (listaProyectoTramites != null && !listaProyectoTramites.isEmpty()) {
            tramitesSecundarios = true;
        }
    }
}
```

```
    } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public void actualizarLista(Long proyectoId) {
    proyectoFilter.setProyectoPadreId(proyectoId);
    buscarVisible = false;
    init();
}

public void restablecerLista() {
    buscarVisible = true;
    verProyecto = false;
    init();
    this.redirectPage("app/beneficiosTributarios/proyectosAntiguosBt.xhtml");
}

public DetalleProyectoVO getDetalleProyectoVO() {
    return detalleProyectoVO;
}

public void setDetalleProyectoVO(DetalleProyectoVO detalleProyectoVO) {
    this.detalleProyectoVO = detalleProyectoVO;
}

public List<Object[]> getListRubros() {
    return listRubros;
}

public void setListRubros(List<Object[]> listRubros) {
    this.listRubros = listRubros;
}

public List<RubroDTO> getRubrosDetalleProyectoList() {
    return rubrosDetalleProyectoList;
}

public void setRubrosDetalleProyectoList(List<RubroDTO> rubrosDetalleProyectoList) {
    this.rubrosDetalleProyectoList = rubrosDetalleProyectoList;
}

public String getDescripcionRubro() {
    return descripcionRubro;
}

public void setDescripcionRubro(String descripcionRubro) {
    this.descripcionRubro = descripcionRubro;
}

public MarcoObjGeneralDTO getMarcoObjGeneralDTO() {
    return marcoObjGeneralDTO;
}

public void setMarcoObjGeneralDTO(MarcoObjGeneralDTO marcoObjGeneralDTO) {
    this.marcoObjGeneralDTO = marcoObjGeneralDTO;
}

public List<PermisoDTO> getListaPermisos() {
```

```
        return listaPermisos;
    }

    public void setListaPermisos(List<PermisoDTO> listaPermisos) {
        this.listaPermisos = listaPermisos;
    }

    public void cargarPermisos() {
        listaPermisos = this.selected.getPermisoList() != null ? new
        ArrayList<>(this.selected.getPermisoList()) : null;
        if (listaPermisosContrato != null && !listaPermisosContrato.isEmpty()) {
            if (listaPermisos == null) {
                listaPermisos = new ArrayList<>();
            }
            for (Map.Entry<String, ValorDefinicionDTO> entry :
        listaPermisosContrato.entrySet()) {
                PermisoDTO permisoDTO = new PermisoDTO();
                permisoDTO.setPermisoId(entry.getValue());
                if (!listaPermisos.contains(permisoDTO)) {
                    listaPermisos.add(permisoDTO);
                }
            }
        }
    }

    public Map<String, ValorDefinicionDTO> getListaPermisosContrato() {
        return listaPermisosContrato;
    }

    public void setListaPermisosContrato(Map<String, ValorDefinicionDTO>
        listaPermisosContrato) {
        this.listaPermisosContrato = listaPermisosContrato;
    }

    public ProyectoDTO getProgramaSelected() {
        return programaSelected;
    }

    public void setProgramaSelected(ProyectoDTO programaSelected) {
        this.programaSelected = programaSelected;
    }

    public String getNombreDependenciaConvocatoria() {
        return nombreDependenciaConvocatoria;
    }

    public void setNombreDependenciaConvocatoria(String nombreDependenciaConvocatoria) {
        this.nombreDependenciaConvocatoria = nombreDependenciaConvocatoria;
    }

    public List<DescripcionDTO> getListaDescripciones() {
        return listaDescripciones;
    }

    public void setListaDescripciones(List<DescripcionDTO> listaDescripciones) {
        this.listaDescripciones = listaDescripciones;
    }

    public DescripcionDTO getDescripcion() {
        return descripcion;
    }
}
```

```
public void setDescripcion(DescripcionDTO descripcion) {
    this.descripcion = descripcion;
}

public RubroDTO getRubroSelected() {
    return rubroSelected;
}

public void setRubroSelected(RubroDTO rubroSelected) {
    this.rubroSelected = rubroSelected;
}

public List<OtroResultadoDTO> getListResultadoDesarrolloTec() {
    return listResultadoDesarrolloTec;
}

public void setListResultadoDesarrolloTec(List<OtroResultadoDTO>
listResultadoDesarrolloTec) {
    this.listResultadoDesarrolloTec = listResultadoDesarrolloTec;
}

public List<OtroResultadoDTO> getListResultadoGeneracion() {
    return listResultadoGeneracion;
}

public void setListResultadoGeneracion(List<OtroResultadoDTO> listResultadoGeneracion) {
    this.listResultadoGeneracion = listResultadoGeneracion;
}

public ProyectoPersonaDTO getProyectoPersonaDTO() {
    return proyectoPersonaDTO;
}

public void setProyectoPersonaDTO(ProyectoPersonaDTO proyectoPersonaDTO) {
    this.proyectoPersonaDTO = proyectoPersonaDTO;
}

public PersonaProgramaDTO getPersonaProgramaDTO() {
    return personaProgramaDTO;
}

public void setPersonaProgramaDTO(PersonaProgramaDTO personaProgramaDTO) {
    this.personaProgramaDTO = personaProgramaDTO;
}

public List<RegionImpactoDTO> getListCoberturas() {
    return listCoberturas;
}

public void setListCoberturas(List<RegionImpactoDTO> listCoberturas) {
    this.listCoberturas = listCoberturas;
}

public List<TipoPersonalDTO> getListaTipoPersonal() {
    return listaTipoPersonal;
}

public void setListaTipoPersonal(List<TipoPersonalDTO> listaTipoPersonal) {
    this.listaTipoPersonal = listaTipoPersonal;
}
```

```
public List<UsuarioDTO> getListaUsuarios() {
    return listaUsuarios;
}

public void setListaUsuarios(List<UsuarioDTO> listaUsuarios) {
    this.listaUsuarios = listaUsuarios;
}

public boolean isFiltroActivo() {
    return filtroActivo;
}

public void setFiltroActivo(boolean filtroActivo) {
    this.filtroActivo = filtroActivo;
}

public Collection<UsuarioDTO> getListaUsuariosSigp(String query) {
    try {
        listaUsuarios =
beneficiosServiceFacade.obtenerUsuariosSigp(query.replaceAll("^\\s*", ""));
    } catch (Exception ex) {
        LOG.error("Error al obtener el listado de usuarios ", ex);
    }
    return listaUsuarios;
}

public void generarFicha(ProyectoDTO proyecto) {
    String pryId =
seguridadConvController.encriptarCodigo(proyecto.getProyectoId().toString());
    PrimeFaces.current().executeScript("descargarFicha('" + pryId + "')");
}

public boolean isVerProyecto() {
    return verProyecto;
}

public void setVerProyecto(boolean verProyecto) {
    this.verProyecto = verProyecto;
}

public HashMap<String, HashMap<String, String>> getMapaCampos() {
    return mapaCampos;
}

public void setMapaCampos(HashMap<String, HashMap<String, String>> mapaCampos) {
    this.mapaCampos = mapaCampos;
}

public Map<String, ConvocatoriaMenuDTO> getListaMenu() {
    return listaMenu;
}

public void setListaMenu(Map<String, ConvocatoriaMenuDTO> listaMenu) {
    this.listaMenu = listaMenu;
}

public List<ValorDefinicionDTO> getListRolPersonas() {
    return listRolPersonas;
}
```

```
public void setListRolPersonas(List<ValorDefinicionDTO> listRolPersonas) {
    this.listRolPersonas = listRolPersonas;
}

public String getDescRolPersona() {
    return descRolPersona;
}

public void setDescRolPersona(String descRolPersona) {
    this.descRolPersona = descRolPersona;
}

public ValorDefinicionDTO getTipoDocHojaVida() {
    return tipoDocHojaVida;
}

public void setTipoDocHojaVida(ValorDefinicionDTO tipoDocHojaVida) {
    this.tipoDocHojaVida = tipoDocHojaVida;
}

public StreamedContent getArchivoHojaVida() {
    return archivoHojaVida;
}

public void setArchivoHojaVida(StreamedContent archivoHojaVida) {
    this.archivoHojaVida = archivoHojaVida;
}

public List<FocoDTO> getListaFocos() {
    return listaFocos;
}

public void setListaFocos(List<FocoDTO> listaFocos) {
    this.listaFocos = listaFocos;
}

public List<TramiteEstimuloDTO> getListaTramiteEstimulos() {
    return listaTramiteEstimulos;
}

public void setListaTramiteEstimulos(List<TramiteEstimuloDTO> listaTramiteEstimulos) {
    this.listaTramiteEstimulos = listaTramiteEstimulos;
}

private void cargarTramitesEstimulos() {
    try {
        listaTramiteEstimulos = new ArrayList<>();

        Map<String, Object> filters = new HashMap<>();
        //Filtro por Proyecto ID
        filters.put("proyectoId", selected.getProyectoId());
        int countTramites = proyectosServiceFacade.tramiteEstimulosListCount(filters);
        if (countTramites > 0) {
            listaTramiteEstimulos = proyectosServiceFacade.tramiteEstimulosList(-1, -1,
"tramiteId.descripcionValorDefinicion", "A", filters);
        }
    } catch (Exception ex) {
        LOG.error("Error al obtener el listado de tramites de beneficios tributarios del
proyecto ", ex);
    }
}
```

```
/**
 * Metodo para consultar la información de un proyecto
 *
 * @param proyectoId
 * @return
 */
public ProyectoDTO consultarInfoProyecto(Long proyectoId) {
    try {
        return beneficiosServiceFacade.obtenerProyecto(proyectoId);
    } catch (Exception ex) {
        LOG.error("Error al consultar la información del proyecto ", ex);
    }
    return new ProyectoDTO();
}

public String getParameterUrlCvLAC() {
    return parameterUrlCvLAC;
}

public void setParameterUrlCvLAC(String parameterUrlCvLAC) {
    this.parameterUrlCvLAC = parameterUrlCvLAC;
}

public List<ScientiPersonaFormacion> getListScientiPersonaFormacions() {
    return listaScientiPersonaFormacions;
}

public void setListScientiPersonaFormacions(List<ScientiPersonaFormacion>
listaScientiPersonaFormacions) {
    this.listaScientiPersonaFormacions = listaScientiPersonaFormacions;
}

public Integer getCountEvaluaciones() {
    return countEvaluaciones;
}

public void setCountEvaluaciones(Integer countEvaluaciones) {
    this.countEvaluaciones = countEvaluaciones;
}

public List<ProyectoEvaluacionDTO> getListEvaluaciones() {
    return listaEvaluaciones;
}

public void setListEvaluaciones(List<ProyectoEvaluacionDTO> listaEvaluaciones) {
    this.listaEvaluaciones = listaEvaluaciones;
}

public ProyectoEvaluacionDTO getProyectoEvaluacionSelected() {
    return proyectoEvaluacionSelected;
}

public void setProyectoEvaluacionSelected(ProyectoEvaluacionDTO
proyectoEvaluacionSelected) {
    this.proyectoEvaluacionSelected = proyectoEvaluacionSelected;
}

public DocumentoDTO getDocumentoEvaluacion() {
    return documentoEvaluacion;
}
```



```

public void setDocumentoEvaluacion(DocumentoDTO documentoEvaluacion) {
    this.documentoEvaluacion = documentoEvaluacion;
}

public ValorDefinicionDTO getDocumentoEvaluacionTipo() {
    return documentoEvaluacionTipo;
}

public void setDocumentoEvaluacionTipo(ValorDefinicionDTO documentoEvaluacionTipo) {
    this.documentoEvaluacionTipo = documentoEvaluacionTipo;
}

public StreamedContent getArchivoEvaluacion() {
    return archivoEvaluacion;
}

public void setArchivoEvaluacion(StreamedContent archivoEvaluacion) {
    this.archivoEvaluacion = archivoEvaluacion;
}

/**
 * Metodo para cargar las evaluaciones del proyecto
 */
public void cargarEvaluacionesProyecto() {
    try {
        listaEvaluaciones = new ArrayList();

        Map<String, Object> filters = new HashMap<>();
        filters.put("proyectoId", this.selected.getProyectoId());
        int countEva = beneficiosServiceFacade.proyectosEvaluacionListCount(filters);
        // System.out.println("countEvaluaciones: " + countEva);
        if (countEva > 0) {
            listaEvaluaciones = beneficiosServiceFacade.proyectosEvaluacionList(-1, -1,
"evaluacionProyectoId", "D", filters);
        }

    } catch (Exception ex) {
        LOG.error("Error al cargar las evaluaciones de proyectos antiguos ", ex);
    }
}

public void cargarDatosProyectoEval(ProyectoEvaluacionDTO proyectoEvaluacionDTO) {
    try {
        evaluarProyectoSessController.setSelected(proyectoEvaluacionDTO);
        cargarMontosProyecto(proyectoEvaluacionDTO.getProyectoId());
        evaluarProyectoSessController.obtenerDocumentoEvaluacion(proyectoEvaluacionDTO);
        evaluarProyectoSessController.obtenerDocumentoPanel(proyectoEvaluacionDTO);
    } catch (Exception ex) {

        Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public List<DocumentoDTO> getListaDocumentos() {
    return listaDocumentos;
}

public void setListaDocumentos(List<DocumentoDTO> listaDocumentos) {
    this.listaDocumentos = listaDocumentos;
}

```

```
}

public ValorDefinicionDTO getDocumentoTipoSgpProyectos() {
    return documentoTipoSgpProyectos;
}

public void setDocumentoTipoSgpProyectos(ValorDefinicionDTO documentoTipoSgpProyectos) {
    this.documentoTipoSgpProyectos = documentoTipoSgpProyectos;
}

public ValorDefinicionDTO getDocumentoTipoSgpProyectosXml() {
    return documentoTipoSgpProyectosXml;
}

public void setDocumentoTipoSgpProyectosXml(ValorDefinicionDTO
documentoTipoSgpProyectosXml) {
    this.documentoTipoSgpProyectosXml = documentoTipoSgpProyectosXml;
}

public ValorDefinicionDTO getDocumentoTipoSgpRegistroProyectos() {
    return documentoTipoSgpRegistroProyectos;
}

public void setDocumentoTipoSgpRegistroProyectos(ValorDefinicionDTO
documentoTipoSgpRegistroProyectos) {
    this.documentoTipoSgpRegistroProyectos = documentoTipoSgpRegistroProyectos;
}

public StreamedContent getArchivoDocumento() {
    return archivoDocumento;
}

public void setArchivoDocumento(StreamedContent archivoDocumento) {
    this.archivoDocumento = archivoDocumento;
}

public List<RegistroImportacionDTO> getRegistroImportacionesList() {
    return registroImportacionesList;
}

public void setRegistroImportacionesList(List<RegistroImportacionDTO>
registroImportacionesList) {
    this.registroImportacionesList = registroImportacionesList;
}

public RegistroImportacionDTO getRegistroImportacionSelected() {
    return registroImportacionSelected;
}

public void setRegistroImportacionSelected(RegistroImportacionDTO
registroImportacionSelected) {
    this.registroImportacionSelected = registroImportacionSelected;
}

public ValorDefinicionDTO getDocumentoImportacionTipo() {
    return documentoImportacionTipo;
}

public void setDocumentoImportacionTipo(ValorDefinicionDTO documentoImportacionTipo) {
    this.documentoImportacionTipo = documentoImportacionTipo;
}
```

```
public StreamedContent getArchivoImportacion() {
    return archivoImportacion;
}

public void setArchivoImportacion(StreamedContent archivoImportacion) {
    this.archivoImportacion = archivoImportacion;
}

public ValorDefinicionDTO getTipoDocSoporteTituloDoc() {
    return tipoDocSoporteTituloDoc;
}

public void setTipoDocSoporteTituloDoc(ValorDefinicionDTO tipoDocSoporteTituloDoc) {
    this.tipoDocSoporteTituloDoc = tipoDocSoporteTituloDoc;
}

public StreamedContent getArchivoDocSoporteTituloDoc() {
    return archivoDocSoporteTituloDoc;
}

public void setArchivoDocSoporteTituloDoc(StreamedContent archivoDocSoporteTituloDoc) {
    this.archivoDocSoporteTituloDoc = archivoDocSoporteTituloDoc;
}

public ValorDefinicionDTO getTipoDocSoporteFuncionesDoc() {
    return tipoDocSoporteFuncionesDoc;
}

public void setTipoDocSoporteFuncionesDoc(ValorDefinicionDTO tipoDocSoporteFuncionesDoc)
{
    this.tipoDocSoporteFuncionesDoc = tipoDocSoporteFuncionesDoc;
}

public StreamedContent getArchivoDocSoporteFuncionesDoc() {
    return archivoDocSoporteFuncionesDoc;
}

public void setArchivoDocSoporteFuncionesDoc(StreamedContent
archivoDocSoporteFuncionesDoc) {
    this.archivoDocSoporteFuncionesDoc = archivoDocSoporteFuncionesDoc;
}

public List<ObjetivoEspecificoDTO> getListaObjetivosEspecificos() {
    return listaObjetivosEspecificos;
}

public void setListaObjetivosEspecificos(List<ObjetivoEspecificoDTO>
listaObjetivosEspecificos) {
    this.listaObjetivosEspecificos = listaObjetivosEspecificos;
}

public ArrayList<Integer> getListAnios() {
    return listAnios;
}

public void setListAnios(ArrayList<Integer> listAnios) {
    this.listAnios = listAnios;
}

public List<ProyectoEntidadDTO> getListProyectoEntidades() {
```

```

        return listaProyectoEntidades;
    }

    public void setListaProyectoEntidades(List<ProyectoEntidadDTO> listaProyectoEntidades) {
        this.listaProyectoEntidades = listaProyectoEntidades;
    }

    public ProyectoEntidadDTO getProyectoEntidadSelected() {
        return proyectoEntidadSelected;
    }

    public void setProyectoEntidadSelected(ProyectoEntidadDTO proyectoEntidadSelected) {
        this.proyectoEntidadSelected = proyectoEntidadSelected;
    }

    public double getTotalMonInstitucionSol() {
        return totalMonInstitucionSol;
    }

    public void setTotalMonInstitucionSol(double totalMonInstitucionSol) {
        this.totalMonInstitucionSol = totalMonInstitucionSol;
    }

    public double getTotalConEspecieSol() {
        return totalConEspecieSol;
    }

    public void setTotalConEspecieSol(double totalConEspecieSol) {
        this.totalConEspecieSol = totalConEspecieSol;
    }

    public double getTotalConEfectivoSol() {
        return totalConEfectivoSol;
    }

    public void setTotalConEfectivoSol(double totalConEfectivoSol) {
        this.totalConEfectivoSol = totalConEfectivoSol;
    }

    public double getTotalSol() {
        return totalSol;
    }

    public void setTotalSol(double totalSol) {
        this.totalSol = totalSol;
    }

    /**
     * Metodo para cargar los documentos del proyecto
     */
    public void cargarDocumentosProyecto() {
        try {
            listaDocumentos = new ArrayList();
            List<DocumentoDTO> documentosSgpProyectos = new ArrayList();
            List<DocumentoDTO> documentosSgpProyectosXml = new ArrayList();
            List<DocumentoDTO> documentosSgpRegistroProyectos = new ArrayList();

            //--Documentos asociados al tipo de origen SGP_PROYECTOS
            Map<String, Object> filtersSgpProyectos = new HashMap<>();
            filtersSgpProyectos.put("origenId", this.selected.getProyectoId());

```

```

        filtersSgpProyectos.put("tipoOrigenId",
documentoTipoSgpProyectos.getValorDefinicionId());
        int countDocumentosSgpProyectos =
configuracionServiceFacade.documentosListCount(filtersSgpProyectos);
//        System.out.println("countDocumentosSgpProyectos: " +
countDocumentosSgpProyectos);
        if (countDocumentosSgpProyectos > 0) {
            documentosSgpProyectos = configuracionServiceFacade.documentosList(-1, -1,
"documentoId", "A", filtersSgpProyectos);
        }

        //--Documentos asociados al tipo de origen SGP_PROYECTOS_XML
//        System.out.println("formularioProyectoId: " +
selected.getFormularioProyectoId());
        if (selected.getFormularioProyectoId() != null) {
            Map<String, Object> filtersSgpProyectosXML = new HashMap<>();
            filtersSgpProyectosXML.put("origenId",
this.selected.getFormularioProyectoId());
            filtersSgpProyectosXML.put("tipoOrigenId",
documentoTipoSgpProyectosXml.getValorDefinicionId());
            int countDocumentosProyectosXML =
configuracionServiceFacade.documentosListCount(filtersSgpProyectosXML);
//        System.out.println("countDocumentosProyectosXML: " +
countDocumentosProyectosXML);
            if (countDocumentosProyectosXML > 0) {
                documentosSgpProyectosXml = configuracionServiceFacade.documentosList(-
1, -1, "documentoId", "A", filtersSgpProyectosXML);
            }
        }

        //--Documentos asociados al tipo de origen SGP_REGISTRO_PROYECTOS
Map<String, Object> filtersSgpRegistroProyectos = new HashMap<>();
        filtersSgpRegistroProyectos.put("origenId", this.selected.getProyectoId());
        filtersSgpRegistroProyectos.put("tipoOrigenId",
documentoTipoSgpRegistroProyectos.getValorDefinicionId());
        int countDocumentosRegistro =
configuracionServiceFacade.documentosListCount(filtersSgpRegistroProyectos);
//        System.out.println("countDocumentosRegistro: " + countDocumentosRegistro);
        if (countDocumentosRegistro > 0) {
            documentosSgpRegistroProyectos = configuracionServiceFacade.documentosList(-
1, -1, "documentoId", "A", filtersSgpRegistroProyectos);
        }

        listaDocumentos.addAll(documentosSgpProyectos);
        listaDocumentos.addAll(documentosSgpProyectosXml);
        listaDocumentos.addAll(documentosSgpRegistroProyectos);

    } catch (Exception ex) {
        LOG.error("Error al cargar los documentos de proyectos antiguos ", ex);
    }
}

/**
 * Metodo para cargar las observaciones de los proyectos
 */
public void cargarObservacionesProyecto() {
    try {
        listaObservacionProyecto =
beneficiosServiceFacade.obtenerObservaciones(this.selected.getProyectoId());
    } catch (Exception ex) {
        LOG.error("Error al cargar las observaciones de proyectos antiguos ", ex);
    }
}

```

```

    }
}

/**
 * Metodo para cargar las importaciones
 */
public void cargarRegistroImportaciones() {
    try {
        List<RegistroImportacionDTO> registroImportacionList =
beneficiosServiceFacade.obtenerRegistroImportaciones(selected.getProyectoId());
        if (registroImportacionList != null && !registroImportacionList.isEmpty()) {
            //Consultamos los detalles de importaciones
            for (RegistroImportacionDTO registroImportacionDTO :
registroImportacionList) {
                List<ImportacionDetalleDTO> listImportacionDetalles =
beneficiosServiceFacade.obtenerImportacionDetalles(registroImportacionDTO.getRegistroImporta
cionId());

registroImportacionDTO.setImportacionDetalleList(listImportacionDetalles);
            }
        }
        this.setRegistroImportacionesList(registroImportacionList);
    } catch (Exception ex) {

Logger.getLogger(ProyectoBeneficiosAllSessController.class.getName()).log(Level.SEVERE,
null, ex);
    }
}

public void cargarTiposTramites() {
    try {
        Map<String, Object> filters = new HashMap<>();
        filters.put("tipoDefinicion", "SOLEXENC");
        listaTiposTramites = administracionServiceFacade.valorDefinicionList(-1, -1,
"descripcionValorDefinicion", "A", filters);
    } catch (Exception ex) {
        LOG.error("Error al cargar los tipos de tramites ", ex);
    }
}
}
}

```

El controlador ProyectoBeneficioAntiguoSessController es un *managed bean* con alcance de sesión (@SessionScoped) que gestiona la lógica de presentación y navegación para la vista de proyectos antiguos en el sistema. Utiliza tecnologías JSF, PrimeFaces, EJB y se integra con distintos servicios y controladores del sistema.

```

@Named(value = "proyectoBeneficioAntiguoSessController ")
@SessionScoped
public class ProyectoBeneficioAntiguoSessController extends
GenericSessionController<ProyectoDTO> implements Serializable

```

2. Inyección de Dependencias

El controlador inyecta varios EJBs y beans de control de sesión:

- `beneficiosServiceFacade`: Acceso a servicios de negocio relacionados con proyectos.
- `configuracionServiceFacade`: Acceso a configuraciones globales.
- `evaluarProyectoSessController`, `seguridadConvController`, `configApplController`: Beans auxiliares del contexto de sesión.

3. Atributos Principales

El controlador define un conjunto amplio de atributos:

- **Modelo de datos lazy (`LazyDataModel<ProyectoDTO>`)** para la tabla de proyectos antiguos con paginación y filtros.
- **Filtros y DTOs auxiliares** para mantener el estado de búsqueda, configuraciones y selección.
- **Listas y mapas** de configuraciones, evaluaciones, rubros, entidades, etc.

Esto permite mantener un estado completo de navegación y gestión del proyecto antiguo seleccionado.

4. Método `init()`

```
@PostConstruct
public void init() {

    this.model = new LazyDataModel<ProyectoDTO>() {

        @Override
        public List<ProyectoDTO> load(int first, int pageSize, String sortField,
SortOrder sortOrder, Map<String, FilterMeta> filterBy) {
            Map<String, Object> filters = new TreeMap<>();
            filterBy.forEach((k, v) -> {
                if (k.equals("globalFilter")) {
                    if (v.getFilterValue() != null &&
!v.getFilterValue().toString().isEmpty()) {
                        filters.put(k, v.getFilterValue());
                    }
                }
            });
            List<ProyectoDTO> proyectoList = new ArrayList<>();
            try {
                //filtro por busqueda avanzada
                if (proyectoFilter != null) {
                    filters.put("proyectoFiltro", proyectoFilter);
                }
                filters.put("proyectosAntiguos", true); //filtro por proyectos antiguos
```

```

        filters.put("claveNull", true); //filtro por clave nula (ya que los
        proyectos antiguos no la registraban)
        int count =
        beneficiosServiceFacade.proyectosBeneficiosListCount(filters);
        if (count > 0) {
            proyectoList =
        beneficiosServiceFacade.proyectosBeneficiosList(first, pageSize, sortField,
        sortOrder.equals(SortOrder.ASCENDING) ? "A" : "D", filters);
        }
        model.setRowCount(count);
    } catch (Exception ex) {
        LOG.error("Error al cargar el listado de proyectos antiguos ", ex);
    }
    return proyectoList;
}
};
}

```

Este método se ejecuta automáticamente después de construir el bean. Dentro de él:

- Se inicializa el modelo LazyDataModel, necesario para paginar los resultados del DataTable en la vista.
- Se definen filtros automáticos para solo cargar proyectos antiguos (clave nula y flag proyectosAntiguos = true).
- Se llama el servicio para contar y listar registros con paginación.

5. Método cargarListas()

```

@Inject
public void cargarListas() {
    try {
        listaEstados = configApplController.getMapValorDefinicion().get("ESTADO");
        // System.out.println("listaEstados SIZE: "+ listaEstados.size());
        listaConvocatorias =
        beneficiosServiceFacade.obtenerConvocatoriasEstimulosAntiguos();
        // System.out.println("listaConvocatorias Antiguas SIZE: " +
        listaConvocatorias.size());
        listaPermisosContrato =
        configApplController.getMapValorDefinicion().get("CONTPERM");
        urlReportes =
        beneficiosServiceFacade.obtenerParametroSistema("URLREPORTE").getCadena();
        documentoImportacionTipo =
        configApplController.getMapValorDefinicion().get("TIPDOCREC").get("Plan de importación para
        solicitudes de exencion de IVA");
        tipoDocSoporteTituloDoc =
        configApplController.getMapValorDefinicion().get("TIPDOCPRY").get("Soportes título de
        doctorado");
        tipoDocSoporteFuncionesDoc =
        configApplController.getMapValorDefinicion().get("TIPDOCPRY").get("Soportes de funciones del
        personal con título de doctorado");
    }
}

```



```

        parameterUrlCvLAC =
beneficiosServiceFacade.obtenerParametroSistema("URLCVLAC2").getCadena();
        //
        documentoEvaluacionTipo =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("PRY_EVALUACION");
        documentoTipoSgpProyectos =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_PROYECTOS");
        documentoTipoSgpProyectosXml =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_PROYECTOS_XML");
        documentoTipoSgpRegistroProyectos =
configApplController.getMapValorDefinicion().get("TIPORIGEN").get("SGP_REGISTRO_PROYECTOS");
        //
        cargarTiposTramites();
        // cargamos el listado de años de los proyectos
        Date fechaFinProy =
beneficiosServiceFacade.obtenerParametroSistema("PRBENFCHAI").getFecha();
        Calendar fechaFinal = Calendar.getInstance();
        fechaFinal.setTime(fechaFinProy);
        int anioInicio = 2000;
        //
        int añoFinal = fechaFinal.get(Calendar.YEAR);
        listAnios = new ArrayList<>();
        for (int i = anioInicio; i <= añoFinal; i++) {
            listAnios.add(i);
        }

    } catch (Exception ex) {

        Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
        null, ex);
    }
}

```

Este método inicializa listas necesarias para la vista:

- Estados (listaEstados), convocatorias antiguas (listaConvocatorias), permisos de contrato, URLs de reportes y CVLAC.
- Tipos de documentos utilizados en el módulo.
- Años desde el 2000 hasta el último año con proyectos antiguos.

6. Método setTipoProyecto()

```

public void setTipoProyecto(ValueChangeEvent event) {
    String dato = String.valueOf(event.getNewValue());
    switch (dato) {
        case "F":
            this.proyectoFilter.setAnteproyecto("S");
            this.proyectoFilter.setExencion("R");
            break;
        case "R":
            this.proyectoFilter.setExencion("R");
            this.proyectoFilter.setAnteproyecto("N");

```

```
        break;
    case "P":
        this.proyectoFilter.setExencion("P");
        this.proyectoFilter.setAnteproyecto(null);
        break;
    default:
        break;
    }
}
```

Permite actualizar el filtro de proyecto (proyectoFilter) basado en el tipo de trámite seleccionado por el usuario.

7. Método cargarDetallesProyecto()

```
public void cargarDetallesProyecto(ProyectoDTO proyectoDTO) {
    try {
        this.selected = proyectoDTO;

        this.selected.setFormularioProyectoId(beneficiosServiceFacade.obtenerFormularioProyectoId(selected.getProjectoId()));
        valoresRubroVO =
        beneficiosServiceFacade.montosRubroProyecto(proyectoDTO.getProjectoId(), null);
        listaObservacionProyecto =
        beneficiosServiceFacade.obtenerObservaciones(proyectoDTO.getProjectoId());

        selected.setPalabraClaveList(beneficiosServiceFacade.obtenerPalabrasClave(selected.getProjectoId()));

        cargarCamposConvocatoria();
        cargarPersonasProyecto();
        cargarListaMenusConvocatoria();
        cargarPresupuestoProyecto();
        Map<String, Object> filters = new HashMap<>();
        filters.put("proyectoId", this.selected.getProjectoId());
        countEvaluaciones =
        beneficiosServiceFacade.proyectosEvaluacionListCount(filters);

        // Objetivos especificos (Menu)
        listaObjetivosEspecificos =
        beneficiosServiceFacade.objetivosEspecificosList(this.selected.getProjectoId());

        cargarInformacionBasicaProyecto();

    } catch (Exception ex) {

        Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
        null, ex);
    }
}
```

Carga todos los datos relevantes del proyecto seleccionado:

- Formulario asociado, valores de rubros, observaciones, palabras clave, ficha técnica, evaluaciones.
- También llama métodos auxiliares para cargar personas, menú de convocatoria y entidades.

8. Método cargarInformacionBasicaProyecto()

```
public void cargarInformacionBasicaProyecto() {
    try {
        //
        selected.setRolEntidadList(beneficiosServiceFacade.obtenerRolesEntidad(selected.getProjectoId()));
        nombreDependenciaConvocatoria =
        proyectosServiceFacade.obtenerDependenciaConvocatoria(selected.getConvocatoriaId().getConvocatoriaId());
        detalleProyectoVO =
        beneficiosServiceFacade.obtenerDetallesProyecto(selected.getProjectoId());
        cargarTramitesEstimulos();
    } catch (Exception ex) {

        Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
        null, ex);
    }
}
```

```
private void cargarTramitesEstimulos() {
    try {
        listaTramiteEstimulos = new ArrayList<>();

        Map<String, Object> filters = new HashMap<>();
        //Filtro por Proyecto ID
        filters.put("proyectoId", selected.getProjectoId());
        int countTramites = proyectosServiceFacade.tramiteEstimulosListCount(filters);
        if (countTramites > 0) {
            listaTramiteEstimulos = proyectosServiceFacade.tramiteEstimulosList(-1, -1,
            "tramiteId.descripcionValorDefinicion", "A", filters);
        }
    } catch (Exception ex) {
        LOG.error("Error al obtener el listado de tramites de beneficios tributarios del
        proyecto ", ex);
    }
}
```

Este método profundiza en la información del proyecto seleccionado:

- Roles, trámites, etc.
- Carga el nombre de la dependencia de la convocatoria y el componente relacionado.

Notas Importantes

- Toda la lógica está encapsulada adecuadamente, siguiendo el patrón MVC.
- El uso de LazyDataModel es clave para evitar problemas de rendimiento con grandes volúmenes de datos.
- Se implementan buenas prácticas como @PostConstruct, @EJB, @Inject, manejo de excepciones y separación de lógica por métodos.

◆ public void cargarEntidadesProyecto()

¿Qué hace?

Carga la lista de **entidades asociadas** al proyecto seleccionado (selected) desde el servicio proyectosServiceFacade.

Detalle:

```
listaProyectoEntidades =  
beneficiosServiceFacade.obtenerProyectoEntidades(selected.getProyectoId());
```

Esto llena listaProyectoEntidades con los datos retornados por el servicio, usando el ID del proyecto seleccionado.

◆ public void cargarRolEntidadesProyecto()

¿Qué hace?

Carga los **roles de cada entidad** en el proyecto y los guarda directamente en el objeto selected.

```
selected.setRolEntidadList(beneficiosServiceFacade.obtenerRolesEntidad(selected.getProyectoId()));
```

◆ public void cargarPresupuestoProyecto()

¿Qué hace?

Carga los **rubros presupuestales** del proyecto antiguo y calcula los totales por tipo de aporte (institución, especie, efectivo).

```
listRubros = beneficiosServiceFacade.obtenerRubrosProyAntiguos(selected.getProyectoId());  
calcularTotales();
```

◆ **public void calcularTotales()**

¿Qué hace?

Suma los valores presupuestales de los rubros cargados en listRubros, dividiendo por tipo:

- totalMonInstitucionSol: aporte institucional
- totalConEspecieSol: aporte en especie
- totalConEfectivoSol: aporte en efectivo
- totalSol: total solicitado

```
for (Object[] rubro : listRubros) {  
    totalMonInstitucionSol += ((Number) rubro[1]).doubleValue();  
    ...  
}
```

◆ **public void cargarResultadosProyecto()**

¿Qué hace?

Carga los **resultados del proyecto** (científicos y de desarrollo tecnológico) en los siguientes atributos:

- detalleProyectoVO (resultado principal)
- listResultadoGeneracion (otros resultados tipo "G")
- listResultadoDesarrolloTec (otros resultados tipo "D")

```
detalleProyectoVO =  
beneficiosServiceFacade.cargarResultadosProyecto(selected.getProyectoId());
```

◆ **public void cargarImpactosProyecto()**

¿Qué hace?

Carga los **impactos del proyecto** y los guarda en detalleProyectoVO.

```
detalleProyectoVO =  
beneficiosServiceFacade.obtenerImpactosProyecto(selected.getProyectoId());
```

◆ public void cargarCoberturasProyecto()

¿Qué hace?

Carga las **coberturas geográficas o poblacionales** del proyecto.

```
listCoberturas = beneficiosServiceFacade.obtenerCoberturas(selected.getProyectoId());
```

◆ public void cargarMarcoLogico()

¿Qué hace?

Carga la información del **marco lógico** del proyecto, incluyendo objetivos generales.

```
detalleProyectoVO =  
beneficiosServiceFacade.cargarMarcoLogicoProyecto(selected.getProyectoId());  
marcoObjGeneralDTO = detalleProyectoVO.getListaObjetivosGeneral().get(0);
```

◆ public void cargarTipoPersonal()

¿Qué hace?

Carga la lista de tipos de personal que participaron en el proyecto (investigadores, administrativos, etc.).

```
listaTipoPersonal = beneficiosServiceFacade.obtenerTipoPersonal(selected.getProyectoId());
```

◆ public void cargarPersonasProyecto()

¿Qué hace?

Carga las **personas que participaron** en el proyecto.

5. Primero obtiene el tipo de documento definido para hoja de vida.

6. Verifica si el proyecto es antiguo usando verificarProyectoPrograma(...).
7. Si es antiguo, carga desde detalleProyectoVO.
8. Si **no** es antiguo, carga desde obtenerPryPersonasPrograma(...) con paginación.

```
proyectoAntiguo = proyectosServiceFacade.verificarProyectoPrograma(parametros);
if (proyectoAntiguo) {
    detalleProyectoVO =
beneficiosServiceFacade.cargarPersonasProyecto(selected.getProyectoId());
} else {
    listaPersonaPrograma = proyectosServiceFacade.obtenerPryPersonasPrograma(...);
}
```

◆ **public void cargarCronogramaProyecto()**

¿Qué hace?

Carga el **cronograma de actividades del proyecto** y lo asigna a detalleProyectoVO.

```
detalleProyectoVO =
beneficiosServiceFacade.cargarCronogramaProyecto(selected.getProyectoId());
```

◆ **public void cargarMontosProyecto(ProyectoDTO proyectoDTO)**

¿Qué hace?

Carga los **montos por rubro** del proyecto (para exención tributaria) y también carga trámites asociados.

```
cargarListaProyectosTramites(proyectoDTO.getProyectoId());
valoresRubroVO = beneficiosServiceFacade.montosRubroProyecto(proyectoDTO.getProyectoId(),
tipoExencion);
```

◆ **public boolean renderCampo(String campo, String pantalla)**

```
public boolean renderCampo(String campo, String pantalla) {
    HashMap<String, String> camposValor;
    String render = "";
    if (mapaCampos != null) {
        if (mapaCampos.containsKey(pantalla)) {
            camposValor = mapaCampos.get(pantalla);
            if (camposValor.containsKey(campo)) {
                render = camposValor.get(campo);
            } else {
```

```

        render = "S";
    }
}
return render.equals("S");
}

```

¿Qué hace?

Verifica si un campo específico debe renderizarse en una pantalla según un mapa de configuración (mapaCampos).

- Si no encuentra la configuración, por defecto retorna true ("S").

```
return render.equals("S");
```

◆ private void cargarCamposConvocatoria()

```

private void cargarCamposConvocatoria() {
    try {
        List<ConvocatoriaCampoDTO> listConvocatoriaCampos =
        proyectosServiceFacade.buscarCamposConvocatoria(selected.getConvocatoriaId().getConvocatoria
        Id());
        HashMap<String, String> camposValor;
        mapaCampos = new HashMap<>();
        if (listConvocatoriaCampos != null && !listConvocatoriaCampos.isEmpty()) {
            for (ConvocatoriaCampoDTO campo : listConvocatoriaCampos) {
                if
                (!mapaCampos.containsKey(campo.getPantallaId().getTipoValidacion().getTipoDefinicion())) {
                    camposValor = new HashMap<>();
                    if (campo.getCampoId() != null) {
                        camposValor.put(campo.getCampoId().getValorDefinicion(),
                        campo.getVisible());
                    }
                    mapaCampos.put(campo.getPantallaId().getTipoValidacion().getTipoDefinicion(), camposValor);
                }
                } else if (campo.getCampoId() != null) {
                    camposValor =
                    mapaCampos.get(campo.getPantallaId().getTipoValidacion().getTipoDefinicion());
                    camposValor.put(campo.getCampoId().getValorDefinicion(),
                    campo.getVisible());
                    mapaCampos.put(campo.getPantallaId().getTipoValidacion().getTipoDefinicion(), camposValor);
                }
            }
        } catch (Exception ex) {
            LOG.error("Error al consultar los campos de la convocatoria", ex);
        }
    }
}

```

¿Qué hace?

Carga la configuración de **campos visibles** para una convocatoria específica y los organiza en un mapa (mapaCampos) para controlar qué se muestra por pantalla.

- Usa ConvocatoriaCampoDTO para consultar visibilidad por pantalla y campo.
- Llena mapaCampos<pantalla, Map<campo, visible>>.

◆ **public boolean existeMenu(String funcion)**

```
public boolean existeMenu(String funcion) {  
    return listaMenu.containsKey(funcion);  
}
```

¿Qué hace?

Verifica si una **función o menú específico existe** en el mapa listaMenu (cargado previamente).

◆ **private void cargarListaMenusConvocatoria()**

```
private void cargarListaMenusConvocatoria() {  
    try {  
        listaMenu = new HashMap<>();  
        List<ConvocatoriaMenuDTO> listaMenuDTO =  
        proyectosServiceFacade.obtenerConvocatoriaMenu(this.selected.getConvocatoriaId());  
        if (listaMenuDTO != null && !listaMenuDTO.isEmpty()) {  
            for (ConvocatoriaMenuDTO convocatoriaMenuDTO : listaMenuDTO) {  
                if (convocatoriaMenuDTO.getMenuId() != null) {  
                    String funcion = convocatoriaMenuDTO.getMenuId().getComentario();  
                    if (funcion != null && !funcion.equals("")) {  
                        listaMenu.put(funcion, convocatoriaMenuDTO);  
                    }  
                }  
            }  
        }  
    } catch (Exception ex) {  
        LOG.error("Error al cargar la lista de menus");  
    }  
}
```

¿Qué hace?

Carga los **menús configurados para la convocatoria actual**, y los guarda en listaMenu.

- Se usa para decidir qué pestañas o secciones mostrar en la UI.

◆ **public String cargarRolesPersona(Long personaProyectoId)**

```
public String cargarRolesPersona(Long personaProyectoId) {
```

```
try {
    if (proyectoPersonaDTO != null && personaProyectoId != null) {
        listRolPersonas =
proyectosServiceFacade.obtenerRolesPersona(proyectoPersonaDTO.getPersonaProyectoId());
        descRolPersona = new String();
        if (listRolPersonas != null && !listRolPersonas.isEmpty()) {
            listRolPersonas.forEach((valorDefinicionDTO) -> {
                descRolPersona += valorDefinicionDTO.getDescripcionValorDefinicion()
+ ", ";
            });
        }
        return descRolPersona;
    }
} catch (Exception ex) {
    LOG.error("Error al cargar los roles de la persona", ex);
}
return descRolPersona;
}
```

¿Qué hace?

Carga los **roles de una persona** asociada al proyecto (por ejemplo: investigador, coordinador).

- Retorna un String con la concatenación de descripciones de roles.

◆ public void cargarListaProyectosTramites(Long proyectoId)

```
public void cargarListaProyectosTramites(Long proyectoId) {
    try {
        listaProyectoTramites =
beneficiosServiceFacade.obtenerProyectosTramites(proyectoId);
        if (listaProyectoTramites != null && !listaProyectoTramites.isEmpty()) {
            tramitesSecundarios = true;
        }

    } catch (Exception ex) {

        Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
        null, ex);
    }
}
```

¿Qué hace?

Consulta los **trámites asociados al proyecto**, y si existen, activa la bandera tramitesSecundarios.

◆ public void actualizarLista(Long proyectoId)

```
public void actualizarLista(Long proyectoId) {
    proyectoFilter.setProyectoPadreId(proyectoId);
}
```

```
        buscarVisible = false;  
        init();  
    }
```

¿Qué hace?

Actualiza el filtro proyectoFilter para listar los proyectos hijos de un proyecto padre. Llama a init() para recargar la vista.

◆ public void restablecerLista()

```
public void restablecerLista() {  
    buscarVisible = true;  
    verProyecto = false;  
    init();  
    this.redirectPage("app/beneficiosTributarios/proyectosAntiguosBt.xhtml");  
}
```

¿Qué hace?

Restaura la vista a su estado inicial, con la lista de proyectos. También hace una redirección a la página principal de proyectos antiguos.

◆ public void cargarPermisos()

```
public void cargarPermisos() {  
    listaPermisos = this.selected.getPermisoList() != null ? new  
    ArrayList<>(this.selected.getPermisoList()) : null;  
    if (listaPermisosContrato != null && !listaPermisosContrato.isEmpty()) {  
        if (listaPermisos == null) {  
            listaPermisos = new ArrayList<>();  
        }  
        for (Map.Entry<String, ValorDefinicionDTO> entry :  
        listaPermisosContrato.entrySet()) {  
            PermisoDTO permisoDTO = new PermisoDTO();  
            permisoDTO.setPermisoId(entry.getValue());  
            if (!listaPermisos.contains(permisoDTO)) {  
                listaPermisos.add(permisoDTO);  
            }  
        }  
    }  
}
```

¿Qué hace?

Carga los permisos actuales del proyecto (selected.getPermisoList()), y además **agrega permisos adicionales** si vienen desde listaPermisosContrato.

◆ **public Collection<UsuarioDTO> getListaUsuariosSigp(String query)**

```
public Collection<UsuarioDTO> getListaUsuariosSigp(String query) {  
    try {  
        listaUsuarios =  
beneficiosServiceFacade.obtenerUsuariosSigp(query.replaceAll("^\\s*", ""));  
    } catch (Exception ex) {  
        LOG.error("Error al obtener el listado de usuarios ", ex);  
    }  
    return listaUsuarios;  
}
```

¿Qué hace?

Busca usuarios del SIGP según un texto de búsqueda (query) ingresado, y los retorna.

◆ **public void generarFicha(ProyectoDTO proyecto)**

```
public void generarFicha(ProyectoDTO proyecto) {  
    String pryId =  
seguridadConvController.encryptedCodigo(proyecto.getProyectoId().toString());  
    PrimeFaces.current().executeScript("descargarFicha('" + pryId + "')");  
}
```

¿Qué hace?

Genera una ficha del proyecto usando su ID encriptado. Ejecuta un JavaScript (descargarFicha(...)) en el navegador.

◆ **private void cargarTramitesEstimulos()**

```
private void cargarTramitesEstimulos() {  
    try {  
        listaTramiteEstimulos = new ArrayList<>();  
  
        Map<String, Object> filters = new HashMap<>();  
        //Filtro por Proyecto ID  
        filters.put("proyectoId", selected.getProyectoId());  
        int countTramites = proyectosServiceFacade.tramiteEstimulosListCount(filters);  
        if (countTramites > 0) {  
            listaTramiteEstimulos = proyectosServiceFacade.tramiteEstimulosList(-1, -1,  
"tramiteId.descripcionValorDefinicion", "A", filters);  
        }  
    } catch (Exception ex) {  
        LOG.error("Error al obtener el listado de tramites de beneficios tributarios del  
proyecto ", ex);  
    }  
}
```

¿Qué hace?

Carga una lista de **trámites de estímulos tributarios** asociados al proyecto, si existen.

◆ **public ProyectoDTO consultarInfoProyecto(Long proyectoId)**

```
/**
 * Metodo para consultar la información de un proyecto
 *
 * @param proyectoId
 * @return
 */
public ProyectoDTO consultarInfoProyecto(Long proyectoId) {
    try {
        return beneficiosServiceFacade.obtenerProyecto(proyectoId);
    } catch (Exception ex) {
        LOG.error("Error al consultar la información del proyecto ", ex);
    }
    return new ProyectoDTO();
}
```

¿Qué hace?

Consulta toda la información del proyecto por su ID y la retorna como ProyectoDTO.

◆ **public void cargarEvaluacionesProyecto()**

```
/**
 * Metodo para cargar las evaluaciones del proyecto
 */
public void cargarEvaluacionesProyecto() {
    try {
        listaEvaluaciones = new ArrayList();

        Map<String, Object> filters = new HashMap<>();
        filters.put("proyectoId", this.selected.getProyectoId());
        int countEva = beneficiosServiceFacade.proyectosEvaluacionListCount(filters);
        // System.out.println("countEvaluaciones: " + countEva);
        if (countEva > 0) {
            listaEvaluaciones = beneficiosServiceFacade.proyectosEvaluacionList(-1, -1,
"evaluacionProyectoId", "D", filters);
        }

    } catch (Exception ex) {
        LOG.error("Error al cargar las evaluaciones de proyectos antiguos ", ex);
    }
}
```

¿Qué hace?

Consulta las **evaluaciones del proyecto** (si existen), y las guarda en listaEvaluaciones.

◆ **public void cargarDatosProyectoEval(ProyectoEvaluacionDTO proyectoEvaluacionDTO)**

```
public void cargarDatosProyectoEval(ProyectoEvaluacionDTO proyectoEvaluacionDTO) {
    try {
        evaluarProyectoSessController.setSelected(proyectoEvaluacionDTO);
        cargarMontosProyecto(proyectoEvaluacionDTO.getProyectoId());
        evaluarProyectoSessController.obtenerDocumentoEvaluacion(proyectoEvaluacionDTO);
        evaluarProyectoSessController.obtenerDocumentoPanel(proyectoEvaluacionDTO);
    } catch (Exception ex) {

        Logger.getLogger(ProyectoBeneficioAntiguoSessController.class.getName()).log(Level.SEVERE,
        null, ex);
    }
}
```

¿Qué hace?

Carga todos los datos necesarios para mostrar la evaluación seleccionada:

4. Montos del proyecto.
5. Documento de evaluación.
6. Documento del panel evaluador.

◆ **public void cargarDocumentosProyecto()**

```
/**
 * Metodo para cargar los documentos del proyecto
 */
public void cargarDocumentosProyecto() {
    try {
        listaDocumentos = new ArrayList();
        List<DocumentoDTO> documentosSgpProyectos = new ArrayList();
        List<DocumentoDTO> documentosSgpProyectosXml = new ArrayList();
        List<DocumentoDTO> documentosSgpRegistroProyectos = new ArrayList();

        //--Documentos asociados al tipo de origen SGP_PROYECTOS
        Map<String, Object> filtersSgpProyectos = new HashMap<>();
        filtersSgpProyectos.put("origenId", this.selected.getProyectoId());
        filtersSgpProyectos.put("tipoOrigenId",
        documentoTipoSgpProyectos.getValorDefinicionId());
        int countDocumentosSgpProyectos =
        configuracionServiceFacade.documentosListCount(filtersSgpProyectos);
        // System.out.println("countDocumentosSgpProyectos: " +
        countDocumentosSgpProyectos);
        if (countDocumentosSgpProyectos > 0) {
            documentosSgpProyectos = configuracionServiceFacade.documentosList(-1, -1,
            "documentoId", "A", filtersSgpProyectos);
        }
    }
}
```

```
//--Documentos asociados al tipo de origen SGP_PROYECTOS_XML
//      System.out.println("formularioProyectoId: " +
selected.getFormularioProyectoId());
      if (selected.getFormularioProyectoId() != null) {
        Map<String, Object> filtersSgpProyectosXML = new HashMap<>();
        filtersSgpProyectosXML.put("origenId",
this.selected.getFormularioProyectoId());
        filtersSgpProyectosXML.put("tipoOrigenId",
documentoTipoSgpProyectosXml.getValorDefinicionId());
        int countDocumentosProyectosXML =
configuracionServiceFacade.documentosListCount(filtersSgpProyectosXML);
//      System.out.println("countDocumentosProyectosXML: " +
countDocumentosProyectosXML);
        if (countDocumentosProyectosXML > 0) {
          documentosSgpProyectosXml = configuracionServiceFacade.documentosList(-
1, -1, "documentoId", "A", filtersSgpProyectosXML);
        }
      }

//--Documentos asociados al tipo de origen SGP_REGISTRO_PROYECTOS
Map<String, Object> filtersSgpRegistroProyectos = new HashMap<>();
filtersSgpRegistroProyectos.put("origenId", this.selected.getProyectoId());
filtersSgpRegistroProyectos.put("tipoOrigenId",
documentoTipoSgpRegistroProyectos.getValorDefinicionId());
int countDocumentosRegistro =
configuracionServiceFacade.documentosListCount(filtersSgpRegistroProyectos);
//      System.out.println("countDocumentosRegistro: " + countDocumentosRegistro);
      if (countDocumentosRegistro > 0) {
        documentosSgpRegistroProyectos = configuracionServiceFacade.documentosList(-
1, -1, "documentoId", "A", filtersSgpRegistroProyectos);
      }

      listaDocumentos.addAll(documentosSgpProyectos);
      listaDocumentos.addAll(documentosSgpProyectosXml);
      listaDocumentos.addAll(documentosSgpRegistroProyectos);

    } catch (Exception ex) {
      LOG.error("Error al cargar los documentos de proyectos antiguos ", ex);
    }
  }
}
```

¿Qué hace?

Carga todos los **documentos asociados** al proyecto desde tres fuentes distintas:

- SGP_PROYECTOS
- SGP_PROYECTOS_XML (si tiene formularioProyectoId)
- SGP_REGISTRO_PROYECTOS

Agrupar todo en listaDocumentos.

◆ **public void cargarObservacionesProyecto()**

```
/**
 * Metodo para cargar las observaciones de los proyectos
 */
public void cargarObservacionesProyecto() {
    try {
        listaObservacionProyecto =
beneficiosServiceFacade.obtenerObservaciones(this.selected.getProyectoId());
    } catch (Exception ex) {
        LOG.error("Error al cargar las observaciones de proyectos antiguos ", ex);
    }
}
```

¿Qué hace?

Consulta y carga las **observaciones** que se han hecho al proyecto.

Controlador de Conversación ProyectoBeneficioAntiguoConvController.java

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.sb.colciencias.gestion.web.controller.beneficiosTributarios.conversation;

import com.sb.colciencias.gestion.business.domain.scienti.ScientiPersona;
import com.sb.colciencias.gestion.business.facade.beneficiosTributarios.BeneficiosServiceFacade;
import com.sb.colciencias.gestion.business.facade.convocatoria.ConvocatoriaServiceFacade;
import com.sb.colciencias.gestion.business.facade.scienti.ScientiServiceFacade;
import com.sb.colciencias.gestion.business.transfer.ConvocatoriaDTO;
import com.sb.colciencias.gestion.business.transfer.DocumentoDTO;
import com.sb.colciencias.gestion.business.transfer.EntidadDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoEvaluacionDTO;
import com.sb.colciencias.gestion.business.transfer.ProyectoPersonaDTO;
import com.sb.colciencias.gestion.business.transfer.RegistroImportacionDTO;
import com.sb.colciencias.gestion.business.transfer.RubroDTO;
import com.sb.colciencias.gestion.business.transfer.ValorDefinicionDTO;
import com.sb.colciencias.gestion.business.util.SystemException;
import com.sb.colciencias.gestion.web.controller.application.ConfigAppController;
import com.sb.colciencias.gestion.web.controller.beneficiosTributarios.session.ProyectoBeneficioAntiguoSessController;
import com.sb.colciencias.gestion.web.controller.proyectos.session.ProyectoAllSessController;
import com.sb.colciencias.gestion.web.util.component.ReporteUtil;
import com.sb.colciencias.gestion.web.util.poi.ReporteExcelMarcoLogico;
import com.sb.colciencias.gestion.web.util.security.SeguridadSessController;
import com.sb.standard.web.controller.conversation.GenericConversationController;
import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.Serializable;
```



```
import java.sql.Blob;
import java.sql.SQLException;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.logging.Level;
import javax.ejb.EJB;
import javax.enterprise.context.ConversationScoped;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;
import javax.faces.event.ActionEvent;
import javax.faces.model.SelectItem;
import javax.inject.Inject;
import javax.inject.Named;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServletRequest;
import org.apache.log4j.Logger;
import org.primefaces.PrimeFaces;
import org.primefaces.component.datatable.DataTable;
import org.primefaces.model.DefaultStreamedContent;

/**
 *
 * @author lsuarez
 */
@Named(value = "proyectoBeneficioAntiguoConvController")
@ConversationScoped
public class ProyectoBeneficioAntiguoConvController extends
GenericConversationController<EntidadDTO> implements Serializable {

    @EJB
    private BeneficiosServiceFacade beneficiosServiceFacade;
    @EJB
    private ScientiServiceFacade scientiServiceFacade;
    @EJB
    private ConvocatoriaServiceFacade convocatoriaServiceFacade;
    @Inject
    private ProyectoBeneficioAntiguoSessController proyectoBeneficioAntiguoSessController;
    @Inject
    private ProyectoAllSessController proyectoAllSessController;
    @Inject
    private SeguridadSessController seguridadSessController;
    @Inject
    private ReporteExcelMarcoLogico reporteExcelMarcoLogico;
    @Inject
    private ConfigApplController configApplController;

    private Boolean cambioEstado;
    private String estadoAnterior;
    private String observacionJustificacion;
    private String estadoRetiro;

    private final String programa = "GEST3100";

    /**
     * Define el log del sistema
     */
    private static final Logger LOG =
Logger.getLogger(ProyectoBeneficioAntiguoConvController.class);
```

```
public ProyectoBeneficioAntiguoConvController() {
    this.selected = new EntidadDTO();
    cambioEstado = false;
}

@Override
public void guardar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void buscar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void eliminar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void editar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void mostrar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void goNuevo(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void goEditar(ActionEvent ae) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void goBuscar(int i) {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

@Override
public void goEliminar() {
    throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
}

public void cambiarEstado(String estado) throws SystemException {
```

```

ProyectoDTO proyectoDTO = proyectoAllSessController.getSelected();
switch (estado) {
    case "REVISADO":
        String codigoEntidad = proyectoDTO.getEntidadNid().getCodigoEntidad();
        if (codigoEntidad != null) {
            if (codigoEntidad.length() < 4) {
                codigoEntidad = "000" + codigoEntidad;
                codigoEntidad = codigoEntidad.substring(codigoEntidad.length(),
codigoEntidad.length() - 4);
            }
            proyectoAllSessController.getSelected().setCodigoProyecto(codigoEntidad
+ proyectoDTO.getConvocatoriaId().getCodigoConvocatoria() + proyectoDTO.getProyectoId());
            cambioEstado = true;
        } else {
            cambioEstado = false;
            throw new
SystemException("#{msg['gestion.beneficiosTributarios.evaluadores.error.codigoEntidad']}");
        }
        break;
    case "DILIGENCIANDO":
        if (!convocatoriaActiva()) {
            throw new
SystemException("#{msg['gestion.beneficiosTributarios.evaluadores.error.diligenciado']}");
        } else {
            proyectoDTO.setFechaEnvio(null);
            cambioEstado = true;
        }
        break;
    default:
        cambioEstado = true;
        break;
}
if (cambioEstado) {
    estadoAnterior =
proyectoAllSessController.getSelected().getEstadoProyectoId().getDescripcionValorDefinicion(
);
    proyectoDTO.setEstadoProyectoId(proyectoAllSessController.getListEstados().get(estado));
    this.editar(null);
}

}

public String isBeforeToday(Date d) {
    Calendar cal = Calendar.getInstance();
    cal.set(Calendar.HOUR_OF_DAY, 0);
    cal.set(Calendar.MINUTE, 0);
    cal.set(Calendar.SECOND, 0);
    cal.set(Calendar.MILLISECOND, 0);
    Date today = cal.getTime();
    if (d.before(today)) {
        cal.setTime(today);
        cal.add(Calendar.DATE, -1);
        Date ayer = cal.getTime();
        if (d.before(ayer)) {
            long startTime = d.getTime();
            long endTime = today.getTime();
            long diffTime = endTime - startTime;
            long diffDays = diffTime / (1000 * 60 * 60 * 24);
            return diffDays + " " +
this.resolveExpression("#{msg['gestion.beneficiosTributarios.proyectos.diasAtras']}");
        } else {

```

```

        return
this.resolveExpression("#{msg['gestion.beneficiosTributarios.proyectos.ayer']}");
    }
    } else {
        return
this.resolveExpression("#{msg['gestion.beneficiosTributarios.proyectos.hoy']}");
    }
}

    public Boolean convocatoriaActiva() {
        ConvocatoriaDTO convocatoriaDTO =
projectoAllSessController.getSelected().getConvocatoriaId();
        Date dt = new Date();
        Calendar c = Calendar.getInstance();
        c.setTime(dt);
        c.add(Calendar.DATE, 1);
        dt = c.getTime();
        return !(convocatoriaDTO.getFechaCierre().before(dt)
            || convocatoriaDTO.getActivo().equals("N")
            || convocatoriaDTO.getPublicar().equals("N"));
    }

    public void lazyLoad() {
//        System.out.println("entra a lazyLoad");
        proyectoBeneficioAntiguoSessController.setFiltroActivo(true);
        DataTable dataTable = (DataTable)
FacesContext.getCurrentInstance().getViewRoot().findComponent("proyectodtoForm:Table");
        dataTable.setFilterBy(null);
        dataTable.loadLazyData();
    }

    public void lazyLoadReg() {
        proyectoAllSessController.setFiltroActivoReg(true);
        DataTable dataTable = (DataTable)
FacesContext.getCurrentInstance().getViewRoot().findComponent("proyectodtoForm:Table");
        dataTable.setFilterBy(null);
        dataTable.loadLazyData();
    }

    public void limpiarFiltros() {
        proyectoBeneficioAntiguoSessController.setProyectoFilter(new ProyectoDTO());
        proyectoBeneficioAntiguoSessController.setTipoTramite(new ValorDefinicionDTO());
        proyectoBeneficioAntiguoSessController.setProgramaSelected(null);
        proyectoBeneficioAntiguoSessController.setFiltroActivo(false);
        System.out.println("limpiando");
    }

    public void limpiarFiltrosReg() {
        proyectoAllSessController.setProyectoFilterReg(new ProyectoDTO());
        proyectoAllSessController.setTipoTramite(new ValorDefinicionDTO());
        proyectoAllSessController.setFiltroActivoReg(false);
        proyectoAllSessController.setProgramaSelected(null);
        System.out.println("limpiando");
    }

    public void filtrarHijos(ProyectoDTO proyectoDTO) {
        if (proyectoDTO != null && proyectoDTO.getProyectoId() != null) {
            ProyectoDTO filtro = new ProyectoDTO(proyectoDTO.getProyectoId());
            filtro.setExencion("H");
            proyectoBeneficioAntiguoSessController.setProgramaSelected(proyectoDTO);
            proyectoBeneficioAntiguoSessController.setProyectoFilter(filtro);
        }
    }

```

```

        proyectoBeneficioAntiguoSessController.setFiltroActivo(false);
        this.lazyLoadSinFiltroActivo();
    }
}

public void lazyLoadSinFiltroActivo() {
    DataTable dataTable = (DataTable)
FacesContext.getCurrentInstance().getViewRoot().findComponent("proyectodtoForm:Table");
    dataTable.setFilterBy(null);
    dataTable.loadLazyData();
}

public void filtrarHijosReg(ProyectoDTO proyectoDTO) {
    if (proyectoDTO != null && proyectoDTO.getProyectoId() != null) {
        ProyectoDTO filtro = new ProyectoDTO(proyectoDTO.getProyectoId());
        filtro.setExencion("H");
        proyectoAllSessController.setProgramaSelected(proyectoDTO);
        proyectoAllSessController.setProyectoFilterReg(filtro);
        proyectoAllSessController.setFiltroActivoReg(false);
        this.lazyLoadSinFiltroActivo();
    }
}

public void cargarProgramas() {
    proyectoAllSessController.setProgramaSelected(null);
    ProyectoDTO filtro = new ProyectoDTO();
    filtro.setExencion("R");
    filtro.setAnteproyecto("N");
    proyectoAllSessController.setProyectoFilterReg(filtro);
    this.lazyLoadReg();
}

public String getObservacionJustificacion() {
    return observacionJustificacion;
}

public void setObservacionJustificacion(String observacionJustificacion) {
    this.observacionJustificacion = observacionJustificacion;
}

public void cargarDescripcionesProyecto() {
    try {

proyectoBeneficioAntiguoSessController.setListaDescripciones(beneficiosServiceFacade.obtener
Descripciones(proyectoBeneficioAntiguoSessController.getSelected().getProyectoId()));
    } catch (Exception ex) {

java.util.logging.Logger.getLogger(ProyectoBeneficioAntiguoConvController.class.getName()).l
og(Level.SEVERE, null, ex);
    }
}

public void cargarDetallesRubrosProyecto(Long rubroId) {
    try {

        Long proyectoId =
proyectoBeneficioAntiguoSessController.getSelected().getProyectoId();
        String tipoExencion = proyectoBeneficioAntiguoSessController.getTipoExencion();

        List<RubroDTO> listaRubros = null;
        Map<String, Object> filters = new HashMap<>();
    }
}

```

```

        filters.put("proyectoId", proyectoId);
        filters.put("rubroId", rubroId);

        if (proyectoBeneficioAntiguoSessController.getTipoExencion() != null) {
            filters.put("tipoExencion", tipoExencion);
        }
        int count = beneficiosServiceFacade.countRubros(filters);
        if (count > 0) {
            listaRubros = beneficiosServiceFacade.obtenerRubros(-1, -1,
"presupuestoRubroId", "A", filters);
        }

proyectoBeneficioAntiguoSessController.setRubrosDetalleProyectoList(listaRubros);
    } catch (Exception ex) {

java.util.logging.Logger.getLogger(ProyectoBeneficioAntiguoConvController.class.getName()).log(Level.SEVERE, null, ex);
    }
}

    public Long calcularTotalRubroConvocatoria(Long cuadroPrespuestoId) {
        Long total = null;
        try {
            total =
beneficiosServiceFacade.obtenerTotalRubroConvocatoriaAntiguas(proyectoBeneficioAntiguoSessCo
ntroller.getSelected().getProyectoId(), cuadroPrespuestoId,
proyectoBeneficioAntiguoSessController.getTipoExencion());
        } catch (Exception e) {
            LOG.error("Error al calcular total rubro cuadro " + cuadroPrespuestoId, e);
        }
        return total == null ? 0L : total;
    }

    public void crearReporteMarcoLogico(ProyectoDTO proyectoSelected) {
        try {
            byte reporte[] = reporteExcelMarcoLogico.generarReporte(proyectoSelected);
            HttpServletResponse response = (HttpServletResponse)
FacesContext.getCurrentInstance().getExternalContext().getResponse();
            response.setContentType(reporteExcelMarcoLogico.isFormatoXlsx() ?
"application/x-xls" : "application/xls");
            response.setHeader("Content-Disposition", "attachment;
filename=\"reporteMarcoLogico_\" + proyectoSelected.getProyectoId() + \".\" +
(reporteExcelMarcoLogico.isFormatoXlsx() ? "xlsx" : "xls") + "\"");
            response.setContentLength(reporte.length);
            response.getOutputStream().write(reporte, 0, reporte.length);
            response.getOutputStream().flush();
            response.getOutputStream().close();
            FacesContext.getCurrentInstance().responseComplete();
        } catch (Exception ex) {
            LOG.error("Error al generar el reporte de marco lógico ", ex);
        }
    }

    public void cargarInfoPersonal(ProyectoPersonaDTO proyectoPersonaDTO) {
        try {
            if (proyectoPersonaDTO.getRegistroCvllac() != null &&
proyectoPersonaDTO.getRegistroCvllac().equals("S")) {
                //Consultamos la informacion de la persona en CvLAC
                proyectoPersonaDTO = consultarInformacionCvllac(proyectoPersonaDTO);
            }
        }
    }

```

```
//Agregamos el documento de Hoja de vida (No CvLAC)
if (proyectoPersonaDTO.getRegistroCvLac() == null ||
proyectoPersonaDTO.getRegistroCvLac().equals("N")) {

proyectoPersonaDTO.setDocumentoHojaDeVida(beneficiosServiceFacade.obtenerDocumento(proyectoB
eneficioAntiguoSessController.getTipoDocHojaVida(),
proyectoPersonaDTO.getPersonaProyectoId()));
}

proyectoBeneficioAntiguoSessController.setProyectoPersonaDTO(proyectoPersonaDTO);
} catch (Exception ex) {
LOG.error("Error al cargar la información de la persona", ex);
}
}

/**
 * Metodo para consultar la información de la persona en CvLAC
 *
 * @throws SystemException
 */
private ProyectoPersonaDTO consultarInformacionCvLac(ProyectoPersonaDTO
proyectoPersonaDTO) throws SystemException {
try {
if (proyectoPersonaDTO.getRegistroCvLac().equals("S")) {
String tipoDoc = "";
/* 010 = Cedula de Ciudadania
020 = Tarjeta de Identidad
050 = Cedula de Extranjeria
*/
if
(proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("050")) {
tipoDoc = "E";
} else if
(proyectoPersonaDTO.getTipoIdentificacionId().getDescripcionValorDefinicion().contains("PASA
PORTE")) {
tipoDoc = "P";
} else if
(proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("010")) {
tipoDoc = "C";
} else if
(proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("020")) {
tipoDoc = "T";
}
ScientiPersona scientiPersona =
scientiServiceFacade.obtenerScientiPersona(tipoDoc,
String.valueOf(proyectoPersonaDTO.getNumeroIdentificacion() == null ? "" :
proyectoPersonaDTO.getNumeroIdentificacion()));
if (scientiPersona != null) {
//Enfoque Diferencial
if (scientiPersona.getStaEnfDiferencial().equals("T")) {
if (scientiPersona.getTxtDscCategoria() != null &&
!scientiPersona.getTxtDscCategoria().isEmpty()) {
proyectoPersonaDTO.setCategoria scientiPersona.getTxtDscCategoria();
}
if (scientiPersona.getStaVictimaConflicto() != null &&
!scientiPersona.getStaVictimaConflicto().isEmpty()) {
proyectoPersonaDTO.setPoblacionVictimaConflicto (scientiPersona.getStaVictimaConflicto().equa
ls("T") ? "S" : "N");
}
}
}
}
```

```

        if (scientiPersona.getTxtPoblacionDisc() != null &&
!scientiPersona.getTxtPoblacionDisc().isEmpty()) {
proyectoPersonaDTO.setPoblacionCondicionDisc scientiPersona.getDescPoblacionDisc());
        }
        if (scientiPersona.getTxtGrupoEtnico() != null &&
!scientiPersona.getTxtGrupoEtnico().isEmpty()) {
proyectoPersonaDTO.setGrupoEtnico scientiPersona.getDescGrupoEtnico());
        }
        if (scientiPersona.getDescGrupoEtario() != null &&
!scientiPersona.getDescGrupoEtario().isEmpty()) {
proyectoPersonaDTO.setGrupoEtario scientiPersona.getDescGrupoEtario());
        }
    }
    }
    return proyectoPersonaDTO;
}
} catch (Exception ex) {
    LOG.error("Error al consultar la información de la persona en CvLAC ", ex);
}
return proyectoPersonaDTO;
}

/**
 * Metodo para obtener a hoja de vida del personal de CvLAC
 *
 * @param personaDTO
 * @return
 */
public String obtenerUrlHojaVidaCvLAC(ProyectoPersonaDTO personaDTO) {
    String url = "";
    if (personaDTO != null) {
        if (personaDTO.getTipoIdentificacionId() != null &&
personaDTO.getNumeroIdentificacion() != null
            && personaDTO.getRegistroCvLac() != null &&
personaDTO.getRegistroCvLac().equals("S")) {
            try {
                ProyectoPersonaDTO proyectoPersonaDTO =
proyectoBeneficioAntiguoSessController.getProyectoPersonaDTO();
                String tipoDoc = "";
                /* 010 = Cedula de Ciudadania
                020 = Tarjeta de Identidad
                050 = Cedula de Extranjeria
                */
                if
(proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("050")) {
                    tipoDoc = "E";
                } else if
(proyectoPersonaDTO.getTipoIdentificacionId().getDescripcionValorDefinicion().contains("PASA
PORTE")) {
                    tipoDoc = "P";
                } else if
(proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("010")) {
                    tipoDoc = "C";
                } else if
(proyectoPersonaDTO.getTipoIdentificacionId().getValorDefinicion().equals("020")) {
                    tipoDoc = "T";
                }
            }
        }
    }
}

```



```

        String numeroIdentificacion =
proyectoPersonaDTO.getNumeroIdentificacion().trim();
        ScientiPersona scientiPersona =
scientiServiceFacade.obtenerScientiPersona(tipoDoc, numeroIdentificacion != null ?
numeroIdentificacion : "");
        if (scientiPersona != null) {
            url = proyectoBeneficioAntiguoSessController.getParameterUrICvLAC()
+ scientiPersona.getCodRh();
        }
        } catch (Exception ex) {
            LOG.error("Error al obtener la url de la hoja de vida de CvLAC ", ex);
        }
    }
    }
    return url;
}

public void descargarArchivoHojaVida(DocumentoDTO documentoTO) {
    ByteArrayInputStream bis = new ByteArrayInputStream(documentoTO.getArchivo());
    proyectoBeneficioAntiguoSessController.setArchivoHojaVida(new
DefaultStreamedContent(bis, documentoTO.getMime(), documentoTO.getNombre()));
}

public String getEstadoRetiro() {
    return estadoRetiro;
}

public void setEstadoRetiro(String estadoRetiro) {
    this.estadoRetiro = estadoRetiro;
}

public String consultarProgramaPadre(Long proyectoId) throws Exception {
    ProyectoDTO proyectoPadre =
beneficiosServiceFacade.consultarProyectoPadre(proyectoId);
    return proyectoPadre.getDependenciaId() != null ?
proyectoPadre.getDependenciaId().getNombreDependencia() : "";
}

public String consultarValorTipoEvaluacion(String valor) {
    List<SelectItem> lista = configApplController.getListTipoEvaluacion();
    if (lista != null && !lista.isEmpty()) {
        for (SelectItem selectItem : lista) {
            if (selectItem.getValue().equals(valor)) {
                return selectItem.getLabel();
            }
        }
    }
    return "";
}

public String nombreCompletoPersona(ProyectoEvaluacionDTO proyectoEvaluacionDTO) {
    String nombre = "";
    if (proyectoEvaluacionDTO != null) {
        if (proyectoEvaluacionDTO.getPersonaId() != null) {
            nombre = proyectoEvaluacionDTO.getPersonaId().getNombreCompletoPersona();
        } else if (proyectoEvaluacionDTO.getEvaluadorScientiId() != null) {
            nombre = proyectoEvaluacionDTO.getEvaluadorScientiId().getNombreCompleto();
        }
    }
    return nombre;
}

```

```

    }

    public void prepararEditar(ProyectoEvaluacionDTO proyectoEvaluacionSeleccionado) {
        try {
            //Se consulta la evaluacion por si se actualizaron los datos
            ProyectoEvaluacionDTO proyectoEvaluacionDTO =
            beneficiosServiceFacade.queryProyectoEvaluacion(proyectoEvaluacionSeleccionado.getEvaluacion
            ProyectoId());
            // ProyectoEvaluacionDTO proyectoEvaluacionDTO =
            proyectoServiceFacade.obtenerEvaluacion(proyectoEvaluacionSeleccionado.getEvaluacionProyecto
            Id(), proyectoEvaluacionSeleccionado.getProyectoId().getProyectoId());
            obtenerDocumentoEvaluacion(proyectoEvaluacionDTO);

            proyectoBeneficioAntiguoSessController.setProyectoEvaluacionSelected(proyectoEvaluacionDTO);
        } catch (Exception ex) {
            LOG.error("Error al preparaEditar la Evaluacion ", ex);
        }
    }

    /**
     * Metodo para obtener el documento de evaluacion
     *
     * @param evaluacionSeleccionada
     */
    public void obtenerDocumentoEvaluacion(ProyectoEvaluacionDTO evaluacionSeleccionada) {
        try {
            DocumentoDTO documentoEvaluacion =
            beneficiosServiceFacade.obtenerDocumento(proyectoBeneficioAntiguoSessController.getDocumento
            EvaluacionTipo(), evaluacionSeleccionada.getEvaluacionProyectoId());

            proyectoBeneficioAntiguoSessController.setDocumentoEvaluacion(documentoEvaluacion);
        } catch (Exception ex) {
            LOG.error("Error al obtener el documento de la Evaluacion ", ex);
        }
    }

    /**
     * Metodo que descarga el documento de evaluacion
     */
    public void descargarArchivoEvaluacion() {
        ByteArrayInputStream bis = new
        ByteArrayInputStream(proyectoBeneficioAntiguoSessController.getDocumentoEvaluacion().getArch
        ivo());
        proyectoBeneficioAntiguoSessController.setArchivoEvaluacion(
            new DefaultStreamedContent(
                bis,
                proyectoBeneficioAntiguoSessController.getDocumentoEvaluacion().getMime(),
                proyectoBeneficioAntiguoSessController.getDocumentoEvaluacion().getNombre()));
    }

    /**
     * Metodo para descargar el reporte de evaluacion calificacion
     *
     * @param tipoReporte
     * @param evaluacionProyectoId
     */
    public void downloadReporteEvaluacion(String tipoReporte, Integer evaluacionProyectoId)
    {
        FacesContext context = FacesContext.getCurrentInstance();
    }

```

```

        HttpServletResponse response = (HttpServletResponse)
context.getExternalContext().getResponse();
        String proyectoId =
proyectoBeneficioAntiguoSessController.getSelected().getProyectoId().toString();
        try {
            ReporteUtil reporteUtil = new ReporteUtil();
            HashMap hashMap = new HashMap();
            hashMap.put("PROYECTO_ID", proyectoId);
            if (evaluacionProyectoId != null) {
                hashMap.put("evaluacion_proyecto_id", String.valueOf(evaluacionProyectoId));
            }

            byte reporte[];

            //carpeta por defecto
            reporteUtil.setNomCarpeta("ReporteEvaluacionHistorico");
            String nombreReporte;
            String headerReporte;
            //Reporte de Evaluacion
            if (tipoReporte.equals("evaluacion")) {
                nombreReporte = "ReporteEvaluacionHistorico";
                headerReporte = nombreReporte + "_" + proyectoId;
            } else {
                //Reporte de Evaluacion Calificacion
                nombreReporte = "ReporteEvaluacionCalificacionHistorico";
                headerReporte = nombreReporte + "_" + proyectoId + "_" +
(evaluacionProyectoId != null ? evaluacionProyectoId.toString() : "");
            }

            //Se setea la imagen del logo parametrizada, guardada en la base de datos
            String imagenLogoBase64 =
beneficiosServiceFacade.obtenerParametroSistema("LOGIN_LOGO").getDetalle();
            String[] parts = imagenLogoBase64.split(",");
            String part1 = parts[0]; // data:image/jpeg;base64
            String imagen = parts[1]; // IMAGEN
            hashMap.put("IMG_BASE64", imagen);
            //
            reporte = reporteUtil.generarReporte(hashMap, nombreReporte, "PDF", context);
            response.setContentType("application/pdf");
            response.setHeader("Content-Disposition", "attachment; filename=\"" +
headerReporte + ".pdf\"");
            response.setContentLength(reporte.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(reporte, 0, reporte.length);
            ouputStream.flush();
            ouputStream.close();
        } catch (Exception ex) {
            FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
            LOG.error("Error al generar el reporte de evaluacion calificación", ex);

            FacesContext.getCurrentInstance().addMessage(null, message);
        }
    }

    /**
     * Metodo para descargar documentos adjuntos
     *
     * @param documentoDTO
     */
    public void descargarArchivoDocumentos(DocumentoDTO documentoDTO) {

```

```

        try {
            //Para estar seguros antes, vuelvo y consulto el documento
            DocumentoDTO documentoConsultado = documentoDTO;
            ByteArrayInputStream bis = new
            ByteArrayInputStream(documentoConsultado.getArchivo());
            proyectoBeneficioAntiguoSessController.setArchivoDocumento(new
            DefaultStreamedContent(bis, documentoConsultado.getMime(),
            documentoConsultado.getNombre()));
        } catch (Exception ex) {
            LOG.error("Error al descargar el documento " + documentoDTO.getNombre() + " con
            tipo origen ID " + documentoDTO.getTipoOrigenId().getValorDefinicionId() + " - " +
            documentoDTO.getTipoOrigenId().getDescripcionValorDefinicion() + " en los documentos del
            proyecto ID " + proyectoBeneficioAntiguoSessController.getSelected().getProyectoId() + " ",
            ex);
        }
    }

    /**
     * Metodo para cargar los detalles del registro de importación
     *
     * @param registroImportacionDTO
     */
    public void cargarDetallesRegistroImportacion(RegistroImportacionDTO
    registroImportacionDTO) {
        try {

            proyectoBeneficioAntiguoSessController.setRegistroImportacionSelected(registroImportacionDTO
            );

            PrimeFaces.current().executeScript("#{ 'dlgImportacionDetalle' }.modal()");
        } catch (Exception ex) {
            LOG.error("Error al cargar los detalles de importación ", ex);
        }
    }

    public void generarReporteCalificacion() {
        FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
            resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
            resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));
        FacesContext context = FacesContext.getCurrentInstance();
        try {
            HttpServletResponse response = (HttpServletResponse)
            context.getExternalContext().getResponse();

            ProyectoDTO proyectoDTO = proyectoBeneficioAntiguoSessController.getSelected();
            String tipoTramiteReporte =
            proyectoDTO.getTramiteExencionId().getValorDefinicion();

            //          System.out.println("proyectoId: " + proyectoDTO.getProyectoId());
            //          System.out.println("tipoTramiteReporte: " + tipoTramiteReporte);
            //          System.out.println("usuarioCreador: " +
            seguridadSessController.getSelected().getUsuario());
            Blob resultadoBlob =
            beneficiosServiceFacade.callSBGenerarReporteEvaluacionCalificacion(proyectoDTO.getProyectoId
            (), tipoTramiteReporte, seguridadSessController.getSelected().getUsuario());

            if (resultadoBlob != null) {

                String codProyectoStr = proyectoDTO.getCodigoProyecto() != null
                    ? proyectoDTO.getCodigoProyecto()
                    : proyectoDTO.getProyectoId().toString();
            }
        }
    }

```

```

        try (InputStream inputStream = resultadoBlob.getBinaryStream();
            ServletOutputStream outputStream = response.getOutputStream()) {

            // Configura el tipo de contenido y encabezados para la respuesta
            response.setContentType("application/rtf");
            response.setHeader("Content-Disposition", "attachment;
filename=\"documentoCalificacion_" + codProyectoStr + ".rtf\"");
            response.setContentLength((int) resultadoBlob.length());

            // Lee el BLOB y escribe el contenido en la respuesta HTTP
            byte[] buffer = new byte[4096];
            int bytesRead;
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                outputStream.write(buffer, 0, bytesRead);
            }

            outputStream.flush();
            outputStream.close();
            FacesContext.getCurrentInstance().responseComplete();

        } catch (SQLException | IOException e) {
            e.printStackTrace();
            // Maneja los errores apropiadamente
        }
    } else {
        throw new SystemException("El reporte de calificación está vacío");
    }

    } catch (Exception ex) {
        LOG.error("Error al generar el documento de calificacion evaluacion ", ex);
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        FacesContext.getCurrentInstance().addMessage(null, message);
    }
}

/**
 * Metodo para consultar y obtener la dependencia asociada
 *
 * @param dependenciaId
 * @return
 */
public String consultarDependencia(Long dependenciaId) {
    if (dependenciaId != null) {
        try {
            return
convocatoriaServiceFacade.consultarProgramaConvocatoria(dependenciaId);
        } catch (Exception ex) {
            LOG.error("Error al consultar la dependencia ", ex);
        }
    }
    return null;
}
}

```

ProyectoBeneficioAntiguoConvController es un controlador de conversación encargado de manejar la lógica de negocio relacionada con la visualización, edición, filtrado y reporte de proyectos antiguos vinculados a convocatorias en el sistema SIGP. Este controlador trabaja de

forma coordinada con otros controladores y servicios para garantizar la gestión eficiente de los proyectos según su estado, tipo de exención y reglas de negocio del sistema.

Inyecciones de Dependencia

```
@EJB
private BeneficiosServiceFacade beneficiosServiceFacade;
@EJB
private ScientiServiceFacade scientiServiceFacade;
@EJB
private ConvocatoriaServiceFacade convocatoriaServiceFacade;
@Inject
private ProyectoBeneficioAntiguoSessController proyectoBeneficioAntiguoSessController;
@Inject
private ProyectoAllSessController proyectoAllSessController;
@Inject
private SeguridadSessController seguridadSessController;
@Inject
private ReporteExcelMarcoLogico reporteExcelMarcoLogico;
@Inject
private ConfigApplController configApplController;
```

Descripción:

Proveen acceso a la lógica de negocio (EJBs) y otros controladores de sesión o visualización para operaciones sobre proyectos y convocatorias.

Atributos Principales

- **cambioEstado (Boolean):** Bandera para determinar si se realizó un cambio de estado del proyecto.
- **estadoAnterior (String):** Almacena el estado anterior del proyecto antes de realizar el cambio.
- **observacionJustificacion (String):** Campo utilizado para registrar observaciones justificativas del cambio de estado.
- **estadoRetiro (String):** Valor de estado de retiro (pendiente por uso en métodos posteriores).
- **programa (String):** Código de programa utilizado en filtros y validaciones (GEST3100).

Métodos Clave

cambiarEstado(String estado)

Modifica el estado del proyecto seleccionado dependiendo de la lógica definida para cada estado:

- En estado **REVISADO**: genera el código del proyecto basado en el código de la entidad y la convocatoria.
- En estado **DILIGENCIANDO**: valida que la convocatoria esté activa y limpia la fecha de envío.
- Otros estados: permite el cambio directo.

isBeforeToday(Date d)

Devuelve un texto indicando si la fecha dada corresponde a:

- Hoy
- Ayer
- N días atrás

convocatoriaActiva()

Verifica si una convocatoria está activa, evaluando:

- Fecha de cierre
- Estado de publicación
- Estado de activación

Carga y Filtros

- lazyLoad(), lazyLoadReg()
Recargan los datos de proyectos aplicando el filtro correspondiente (activo o registro).
- limpiarFiltros(), limpiarFiltrosReg()
Limpian los filtros de búsqueda de los proyectos y tipo de trámite.

- `filtrarHijos(...)`, `filtrarHijosReg(...)`
Filtran los proyectos hijos asociados al proyecto padre seleccionado, utilizando exención tipo H.
- `lazyLoadSinFiltroActivo()`
Carga datos sin aplicar filtros activos.
- `cargarProgramas()`
Inicializa los proyectos con exención tipo R (registro), sin anteproyecto, para filtrado.

Operaciones Adicionales

`cargarDescripcionesProyecto()`

Consulta y asigna las descripciones del proyecto seleccionado al ViewController.

`cargarDetallesRubrosProyecto(Long rubroId)`

Obtiene los detalles del rubro presupuestal de un proyecto específico, filtrando por `proyectoId`, `rubroId` y `tipoExencion`.

`calcularTotalRubroConvocatoria(Long cuadroPrespuestId)`

Calcula el total asignado a un rubro específico en convocatorias antiguas.

`crearReporteMarcoLogico(ProyectoDTO proyectoSelected)`

Genera y descarga un reporte en Excel (XLS/XLSX) con la estructura del marco lógico del proyecto seleccionado.

Observaciones de Diseño

- Se hace uso intensivo del patrón de inyección de dependencias (`@Inject`, `@EJB`) para mantener el controlador limpio y desacoplado.
- El controlador está diseñado para manejar tanto proyectos antiguos como registros activos, diferenciados por banderas (`filtroActivo`, `filtroActivoReg`).
- Se integra con componentes JSF (`DataTable`) para recarga de datos con lazy loading.

Este controlador es responsable de gestionar la información relacionada con la evaluación de proyectos antiguos, incluyendo datos personales, documentos y reportes.

1. Carga de información personal del investigador

- El método **cargarInfoPersonal** determina si se debe consultar la información desde CvLAC (registroCvLac == 'S') o desde la base de datos local.
- Si no se usa CvLAC, se carga el documento de hoja de vida almacenado.
- **consultarInformacionCvLac** obtiene datos poblacionales y de enfoque diferencial desde SCIENTI.

2. Hoja de vida

- **obtenerUrlHojaVidaCvLAC**: construye la URL del CvLAC de la persona, si es evaluador registrado.
- **descargarArchivoHojaVida**: descarga el archivo cargado como hoja de vida (no-CvLAC).

3. Evaluaciones

- **prepararEditar**: vuelve a cargar los datos actualizados de una evaluación seleccionada.
- **obtenerDocumentoEvaluacion**: recupera el archivo asociado a una evaluación.
- **descargarArchivoEvaluacion**: permite descargar el archivo PDF de la evaluación.

4. Reporte de Evaluación

- **downloadReporteEvaluacion**: genera y descarga un reporte PDF de evaluación o de calificación usando ReporteUtil.
 - Incluye logo parametrizado (LOGIN_LOGO).
 - Usa parámetros como PROYECTO_ID y evaluacion_proyecto_id.

5. Descarga de documentos

- **descargarArchivoDocumentos**: permite descargar cualquier documento adjunto al proyecto, como soportes.

6. Dependencia y evaluador

- **consultarProgramaPadre:** obtiene el nombre del programa padre del proyecto.
- **consultarValorTipoEvaluacion:** devuelve la descripción asociada a un tipo de evaluación.
- **nombreCompletoPersona:** arma el nombre completo del evaluador según la fuente (local o SCIENTI).
- **consultarDependencia:** obtiene el nombre de la dependencia académica asociada a una convocatoria.

Observaciones Técnicas

- Usa EJBs como `proyectoServiceFacade`, `scientiServiceFacade`, y `convocatoriaServiceFacade` para acceder a la lógica de negocio.
- Controla errores con logs (`LOG.error(...)`) y muestra mensajes con `FacesMessage` en el contexto JSF.
- Emplea clases DTO (`ProyectoPersonaDTO`, `ScientiPersona`, `DocumentoDTO`, `ProyectoEvaluacionDTO`) para el transporte de datos.

*Se definen métodos de consulta en Facade EJB (**BeneficiosServiceFacade.java**) para funcionalidad de pantalla*

```
//  
//PROYECTOS BENEFICIOS TRIBUTARIOS  
/**  
 * Consulta el listado de proyectos de Beneficios Tributarios registrados en  
 * la base de datos  
 *  
 * @param first indica desde donde empieza la consulta  
 * @param pageSize indica el final de la consulta (first + pageSize)  
 * @param sortField indica el campo por el que se realiza el ordenamiento  
 * @param sortOrder indica el orden (ascendente o descendente)  
 * @param filters contiene los filtros al momento de solicitar el listado de  
 * proyectos de Beneficios Tributarios  
 * @return lista de proyectos de Beneficios Tributarios encontrados en el  
 * rango enviado y basado en los filtros enviados  
 * @throws Exception  
 */
```

```
List<ProyectoDTO> proyectosBeneficiosList(int first, int pageSize, String sortField,
String sortOrder, Map<String, Object> filters) throws Exception;

/**
 * Cuenta la cantidad de proyectos de Beneficios Tributarios actualmente
 * registrados en la base de datos
 *
 * @param filters
 * @return
 * @throws Exception
 */
int proyectosBeneficiosListCount(Map<String, Object> filters) throws Exception;
```

*Se implementan métodos de consulta en facade EJB
(BeneficiosFacadeServiceBean.java) para funcionalidad de pantalla*

```
//----- Proyectos de Beneficios Tributarios -----
-----//
@Override
public List<ProyectoDTO> proyectosBeneficiosList(int first, int pageSize, String
sortField, String sortOrder, Map<String, Object> filters) throws Exception {
    ObjectConverter<ProyectoDTO, Proyecto> converter =
objectConverterFactory.checkOut();
    List<ProyectoDTO> listProyectoDTOS =
converter.toTransferList(beneficiosService.proyectosBeneficiosList(first, pageSize,
sortField, sortOrder, filters), ProyectoDTO.class);
    objectConverterFactory.checkIn(converter);
    return listProyectoDTOS;
}

@Override
public int proyectosBeneficiosListCount(Map<String, Object> filters) throws Exception {
    ProyectoDTO proyectoFiltroDTO = (ProyectoDTO) filters.get("proyectoFiltro");
    ObjectConverter<ProyectoDTO, Proyecto> converter =
objectConverterFactory.checkOut();
    Proyecto proyectoFiltro = converter.toEntity(proyectoFiltroDTO, Proyecto.class);
    filters.put("proyectoFiltro", proyectoFiltro);
    return beneficiosService.proyectosBeneficiosListCount(filters);
}
```

*Se ajusta interfaz BeneficioService.java que define métodos de consulta en Service EJB
para funcionalidad de pantalla*

```
//----- Proyectos de Beneficios Tributarios -----
-----//
List<Proyecto> proyectosBeneficiosList(int first, int pageSize, String sortField, String
sortOrder, Map<String, Object> filters) throws SystemException;

int proyectosBeneficiosListCount(Map<String, Object> filters) throws SystemException;
```

*Se ajusta **BeneficioServiceBean.java** donde implementan métodos consulta en Service EJB para funcionalidad de pantalla*

```
//----- Proyectos de Beneficios Tributarios -----
----//
@Override
public List<Proyecto> proyectosBeneficiosList(int first, int pageSize, String sortField,
String sortOrder, Map<String, Object> filters) throws SystemException {
    return
this.getStoreManager(BeneficiosStoreManager.class).proyectosBeneficiosList(first, pageSize,
sortField, sortOrder, filters);
}

@Override
public int proyectosBeneficiosListCount(Map<String, Object> filters) throws
SystemException {
    return
this.getStoreManager(BeneficiosStoreManager.class).proyectosBeneficiosListCount(filters);
}
```

Internacionalización en español e inglés

```
#ES
##### Proyectos Beneficios Tributarios ANTIGUOS
#####

gestion.beneficiosTributarios.proyectosAntiguos.header =
Proyectos Beneficios Tributarios Antiguos
gestion.beneficiosTributarios.proyectosAntiguos.breadcrum =
Proyectos Antiguos
gestion.beneficiosTributarios.proyectosAntiguos.title =
Proyectos Antiguos
gestion.beneficiosTributarios.proyectosAntiguos.formularioProyectoId =
Formulario Proyecto ID
gestion.beneficiosTributarios.proyectosAntiguos.codigoProyecto =
Código Registro
gestion.beneficiosTributarios.proyectosAntiguos.codigoProyecto.placeholder =
Ingrese el Código del Registro
gestion.beneficiosTributarios.proyectosAntiguos.codigoProyectoId =
Código Proyecto
gestion.beneficiosTributarios.proyectosAntiguos.codigoProyectoId.placeholder =
Ingrese el Código del Proyecto
gestion.beneficiosTributarios.proyectosAntiguos.titulo =
Título Proyecto
gestion.beneficiosTributarios.proyectosAntiguos.titulo.placeholder =
Ingrese el Título del Proyecto
gestion.beneficiosTributarios.proyectosAntiguos.tipoProyecto =
Tipo Proyecto
gestion.beneficiosTributarios.proyectosAntiguos.anioRegistro =
Año Registro
gestion.beneficiosTributarios.proyectosAntiguos.fechaRegistro =
Fecha de Registro
gestion.beneficiosTributarios.proyectosAntiguos.fechaEnvio =
Fecha de Envío
gestion.beneficiosTributarios.proyectosAntiguos.fechaRevision =
Fecha Revisión
```

gestion.beneficiosTributarios.proyectosAntiguos.tipoTramite	=
Trámite	
gestion.beneficiosTributarios.proyectosAntiguos.convocatoria	=
Convocatoria	
gestion.beneficiosTributarios.proyectosAntiguos.programa	=
Prog Nacional de CTeI	
gestion.beneficiosTributarios.proyectosAntiguos.programa1	=
Programa Nacional 1	
gestion.beneficiosTributarios.proyectosAntiguos.programa2	=
Programa Nacional 2	
gestion.beneficiosTributarios.proyectosAntiguos.estado	=
Estado	
gestion.beneficiosTributarios.proyectosAntiguos.estado.proy	=
Estado Proyecto	
gestion.beneficiosTributarios.proyectosAntiguos.entidad	=
Entidad Ejecutora	
gestion.beneficiosTributarios.proyectosAntiguos.lugar	=
Lugar de Ejecución	
gestion.beneficiosTributarios.proyectosAntiguos.fechaModificacion	=
Ultima Modificación	
gestion.beneficiosTributarios.proyectosAntiguos.fechaCreacion	=
Fecha Creación	
gestion.beneficiosTributarios.proyectosAntiguos.financiacion	=
Tipo Financiación	
gestion.beneficiosTributarios.proyectosAntiguos.tipoGestion	=
Tipo Gestión	
gestion.beneficiosTributarios.proyectosAntiguos.duracion	=
Duración	
gestion.beneficiosTributarios.proyectosAntiguos.ejecucion	=
Ejecución	
gestion.beneficiosTributarios.proyectosAntiguos.meses	=
Meses	
gestion.beneficiosTributarios.proyectosAntiguos.fechaInicioProyecto	=
Fecha Inicio Proyecto	
gestion.beneficiosTributarios.proyectosAntiguos.fechaFinalizacionProyecto	=
Fecha Finalización Proyecto	
gestion.beneficiosTributarios.proyectosAntiguos.palabras	=
Palabras Claves	
gestion.beneficiosTributarios.proyectosAntiguos.montos	=
Montos del proyecto	
gestion.beneficiosTributarios.proyectosAntiguos.montosProg	=
Montos del programa	
gestion.beneficiosTributarios.proyectosAntiguos.montosFicTec	=
Montos de la ficha Técnica	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos	=
Requisitos Adjuntos	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.btn	=
Requisitos	
gestion.beneficiosTributarios.proyectosAntiguos.especie	=
Especie	
gestion.beneficiosTributarios.proyectosAntiguos.dinero	=
Dinero	
gestion.beneficiosTributarios.proyectosAntiguos.solicitado	=
Solicitado	
gestion.beneficiosTributarios.proyectosAntiguos.total	=
Total	
gestion.beneficiosTributarios.proyectosAntiguos.radicadoBtn	=
Cambiar a Radicado	
gestion.beneficiosTributarios.proyectosAntiguos.rechazarBtn	=
Rechazar	

gestion.beneficiosTributarios.proyectosAntiguos.evaluacionBtn	=
Enviar a Evaluación	
gestion.beneficiosTributarios.proyectosAntiguos.registro.requisitos.error	=
Debe revisar los requisitos antes de pasar el proyecto a la etapa de evaluación	
gestion.beneficiosTributarios.proyectosAntiguos.registro.requisitos.error2	= No
se puede pasar el proyecto a la etapa de evaluación porque no cumple uno o más requisitos obligatorios	
gestion.beneficiosTributarios.proyectosAntiguos.pasarDiligenciado	=
Pasar a Diligenciado	
gestion.beneficiosTributarios.proyectosAntiguos.pasarDecision	=
Enviar a Decisión	
gestion.beneficiosTributarios.proyectosAntiguos.devolverEvaluacion	=
Devolver a Evaluación	
gestion.beneficiosTributarios.proyectosAntiguos.devolverSecretaria	=
Devolver a la Secretaría Técnica CNBT	
gestion.beneficiosTributarios.proyectosAntiguos.devolverRegistro	=
Devolver a Registro	
gestion.beneficiosTributarios.proyectosAntiguos.subsanable	=
Subsanable	
gestion.beneficiosTributarios.proyectosAntiguos.agregarObservacion	=
Agregar Observación	
gestion.beneficiosTributarios.proyectosAntiguos.editarObservacion	=
Editar Observación	
gestion.beneficiosTributarios.proyectosAntiguos.hoy	=
Hoy	
gestion.beneficiosTributarios.proyectosAntiguos.ayer	=
Ayer	
gestion.beneficiosTributarios.proyectosAntiguos.diasAtras	=
Días atrás a las	
gestion.beneficiosTributarios.proyectosAntiguos.fechaObservacion	=
Fecha Observación	
gestion.beneficiosTributarios.proyectosAntiguos.fechaObservacion.required	=
Debe Seleccionar Fecha Observación	
#Requisitos	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.header	=
Requisitos	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.obligatorio.msg	=
Recuerde que los requisitos que tengan asterisco (*) son requeridos.	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.requisito.label	=
Requisito	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.nombre.archivo.label	=
Nombre Archivo	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.tamano.label	=
Tamaño	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.progreso.label	=
Progreso	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.progreso.ok	=
DOCUMENTO CARGADO	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.cumple.label	=
Verificación Requisito	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.cumple.required	=
Debe ingresar el tipo de cumplimiento del requisito	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.observaciones.label	=
Observaciones	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.certificado.error	=
Debe adjuntar todos los requisitos requeridos	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.vacio	=
No hay Adjunto	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.vacio.mensaje	=
Debe adjuntar un archivo para poder agregarle una observación	

gestion.beneficiosTributarios.proyectosAntiguos.requisitos.concepto.label	=
Concepto panel de Evaluación	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.puntaje.label	=
Puntaje	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.elegible.label	=
Elegible	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.boton.generar	=
Generar Certificado	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.fechaCumplimiento.label	=
Fecha Cumplimiento	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.modal.label	=
Cumplimiento del requisito	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.tooltip.cumplimiento	=
Cumplimiento del requisito	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.irDetallesPry.btn	=
Ir a Detalles	
gestion.beneficiosTributarios.proyectosAntiguos.observaciones.requisitos.title	=
Observaciones Generales de los Requisitos	
gestion.beneficiosTributarios.proyectosAntiguos.observaciones.requisitos.label	=
Observaciones	
#pestañas	
gestion.beneficiosTributarios.proyectosAntiguos.tab.infoBasica	=
Info Básica	
gestion.beneficiosTributarios.proyectosAntiguos.tab.observaciones	=
Observaciones	
gestion.beneficiosTributarios.proyectosAntiguos.tab.revisionEvaluacion	=
Evaluación	
gestion.beneficiosTributarios.proyectosAntiguos.tab.exencion	=
Exención	
gestion.beneficiosTributarios.proyectosAntiguos.tab.entidades	=
Entidades	
gestion.beneficiosTributarios.proyectosAntiguos.tab.rolEntidades	=
Roles Entidades	
gestion.beneficiosTributarios.proyectosAntiguos.tab.presupuesto	=
Presupuesto	
gestion.beneficiosTributarios.proyectosAntiguos.tab.descripciones	=
Descripciones	
gestion.beneficiosTributarios.proyectosAntiguos.tab.impactos	=
Impactos	
gestion.beneficiosTributarios.proyectosAntiguos.tab.coberturas	=
Coberturas	
gestion.beneficiosTributarios.proyectosAntiguos.coberturas.label	=
Coberturas	
gestion.beneficiosTributarios.proyectosAntiguos.tab.productos	=
Productos	
gestion.beneficiosTributarios.proyectosAntiguos.tab.cronograma	=
Actividades Cronograma	
gestion.beneficiosTributarios.proyectosAntiguos.tab.tipoPersonal	=
Tipo de Personal	
gestion.beneficiosTributarios.proyectosAntiguos.tab.personal	=
Personal	
gestion.beneficiosTributarios.proyectosAntiguos.tab.marcoLogico	=
Marco Lógico	
gestion.beneficiosTributarios.proyectosAntiguos.tab.registroImportaciones	=
Importaciones	
gestion.beneficiosTributarios.proyectosAntiguos.tab.documentos	=
Documentos	
#	
gestion.beneficiosTributarios.proyectosAntiguos.entidades.empty	= No
se encontraron Entidades	

gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.empty	= No
se encontraron Rol Entidades	
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.empty	= No
se encontraron Presupuesto	
gestion.beneficiosTributarios.proyectosAntiguos.impactos.empty	= No
se encontraron Impactos	
gestion.beneficiosTributarios.proyectosAntiguos.coberturas.empty	= No
se encontraron Coberturas	
gestion.beneficiosTributarios.proyectosAntiguos.productos.empty	= No
se encontraron Productos	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.empty	= No
se encontraron Actividades Cronograma	
gestion.beneficiosTributarios.proyectosAntiguos.personal.empty	= No
se encontraron Personal	
gestion.beneficiosTributarios.proyectosAntiguos.marcoLogico.empty	= No
se encontraron Marco Lógico	
gestion.beneficiosTributarios.proyectosAntiguos.descripcion.empty	= No
se encontraron Descripciones	
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.empty	= No
se encontraron Importaciones	
#	
gestion.beneficiosTributarios.proyectosAntiguos.column.convocatoria	=
Convocatoria	
gestion.beneficiosTributarios.proyectosAntiguos.column.programa	=
Programa	
gestion.beneficiosTributarios.proyectosAntiguos.column.tramite	=
Trámite	
gestion.beneficiosTributarios.proyectosAntiguos.column.fichaProyecto	=
Ficha Proyecto	
gestion.beneficiosTributarios.proyectosAntiguos.column.fichaPrograma	=
Ficha Programa	
gestion.beneficiosTributarios.proyectosAntiguos.column.busquedaAvanzada	=
Búsqueda Avanzada	
gestion.beneficiosTributarios.proyectosAntiguos.reporteRubrosPry	=
Reporte Rubros y Proyectos	
gestion.beneficiosTributarios.proyectosAntiguos.reporteMarcoLogico	=
Reporte Marco Lógico	
gestion.beneficiosTributarios.proyectosAntiguos.reporteEvaluacion	=
Reporte Evaluación	
gestion.beneficiosTributarios.proyectosAntiguos.nivelMadurezTecnologica	=
Nivel de madurez tecnológica	
gestion.beneficiosTributarios.proyectosAntiguos.tipoPersona	=
Tipo persona	
gestion.beneficiosTributarios.proyectosAntiguos.CiudadFormacion	=
Ciudad de formación	
gestion.beneficiosTributarios.proyectosAntiguos.valorInscripcion	=
Valor de la inscripción	
gestion.beneficiosTributarios.proyectosAntiguos.viaPresentacion	=
Vía de Presentación	
gestion.beneficiosTributarios.proyectosAntiguos.origenInvencion	=
Información de Titularidad	
gestion.beneficiosTributarios.proyectosAntiguos.vinculoLaboral	=
Vínculo Laboral	
#Exencion	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.fechaSolicitud	=
Fecha Solicitud	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.codigoProyectoExencion	=
Código Proyecto Exención	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.fechaInicio	=
Fecha Inicio	

gestion.beneficiosTributarios.proyectosAntiguos.exencion.fechaFin	=
Fecha Fin	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.caracter	=
Carácter Proyecto	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.descripcion	=
Carácter	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.justificacion	=
Justificación Cácter	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.valor	=
Valor Solicitado	
#Entidades	
gestion.beneficiosTributarios.proyectosAntiguos.entidades.entidad	=
Entidad	
gestion.beneficiosTributarios.proyectosAntiguos.entidades.nit	=
NIT	
#Rol Entidades	
gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.entidad	=
Entidad	
gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.rol	=
Rol	
#Descripciones	
gestion.beneficiosTributarios.proyectosAntiguos.tabla.orden	=
Orden	
gestion.beneficiosTributarios.proyectosAntiguos.tabla.descripcion	=
Descripción	
gestion.beneficiosTributarios.proyectosAntiguos.cargando	=
CARGANDO DESCRIPCIONES...	
gestion.beneficiosTributarios.proyectosAntiguos.btn.ver	=
Ver descripción	
gestion.beneficiosTributarios.proyectosAntiguos.descripcion.header	=
Descripción	
#Impactos	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.label	=
= Impactos	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.tipo	=
= Tipo Impacto	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.descripcion	=
= Descripción	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.año	=
= Año	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.empty	=
= No se encontraron Impactos para este proyecto	
#Cronograma	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.label	=
= Actividades Cronograma	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.empty	=
= No se encontraron Actividades	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.nActividad	=
= N° Actividad	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.actividad	=
= Actividad	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.inicio	=
= Inicio	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.fin	=
= Fin	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.ejecucion	=
= Ejecución En	
#Tipo de Personal	
gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.label	=
= Tipo de Personal	

```

gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.empty
= No se encontraron tipos de personal
gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.tipoPersonal
= Rol en el Proyecto
gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.numeroPersonas
= Número Personas
#Personal
gestion.beneficiosTributarios.proyectosAntiguos.personal.label
= Personal
gestion.beneficiosTributarios.proyectosAntiguos.personal.empty
= No se encontraron Personas Registradas
gestion.beneficiosTributarios.proyectosAntiguos.personal.persona
= Persona
gestion.beneficiosTributarios.proyectosAntiguos.personal.tipo
= Vinculación Proyecto
gestion.beneficiosTributarios.proyectosAntiguos.personal.identificacion
= N° Identificación
gestion.beneficiosTributarios.proyectosAntiguos.personal.entidad
= Entidad
gestion.beneficiosTributarios.proyectosAntiguos.personal.vinculo
= Vinculo
gestion.beneficiosTributarios.proyectosAntiguos.personal.fechaNacimiento
= Fecha Nacimiento
gestion.beneficiosTributarios.proyectosAntiguos.personal.pais
= País
gestion.beneficiosTributarios.proyectosAntiguos.personal.email
= Email
gestion.beneficiosTributarios.proyectosAntiguos.personal.tipoIdentificación
= Tipo Identificación
gestion.beneficiosTributarios.proyectosAntiguos.personal.numeroIdentificación
= Número Identificación
gestion.beneficiosTributarios.proyectosAntiguos.personal.genero
= Género
gestion.beneficiosTributarios.proyectosAntiguos.personal.dedicacion
= Dedicación (h/semana)
gestion.beneficiosTributarios.proyectosAntiguos.personal.nMeses
= Número Meses
gestion.beneficiosTributarios.proyectosAntiguos.personal.funcion
= Función en el proyecto
gestion.beneficiosTributarios.proyectosAntiguos.personal.registradoCvLAC
= ¿Registrado en CvLAC?
gestion.beneficiosTributarios.proyectosAntiguos.personal.documentoHojaVida
= Documento Adjunto Hoja de Vida
gestion.beneficiosTributarios.proyectosAntiguos.personal.poblacionVictimaConflicto
= Población Víctima del Conflicto Armado
gestion.beneficiosTributarios.proyectosAntiguos.personal.poblacionCondicionDisc
= Población en Situación de Discapacidad
gestion.beneficiosTributarios.proyectosAntiguos.personal.grupoEtnico
= Grupo Étnico
gestion.beneficiosTributarios.proyectosAntiguos.personal.grupoEtario
= Grupo Etario
gestion.beneficiosTributarios.proyectosAntiguos.personal.categoria
= Categoría
#Marco Logico
gestion.beneficiosTributarios.proyectosAntiguos.marco.label
= Marco Lógico
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoGral
= Objetivo General
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoGeneral.empty
= No se encontró Objetivo General

```

```

gestion.beneficiosTributarios.proyectosAntiguos.marco.meta
= Meta
gestion.beneficiosTributarios.proyectosAntiguos.marco.fuenteVerificacion
= Fuente Verificación
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivosEspecificos.label
= Objetivos Específicos
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivosEspecificos.empty
= No se encontraron Objetivos Específicos
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.label
= Objetivo Específico N°
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.objetivo.label
= Objetivo Específico
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.meta.label
= Meta
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.fuente.label
= Fuente de verificación
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.indicador.label
= Indicador
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.actividades.empty
= Este Objetivo no tiene Actividades
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.resultados.empty
= Esta Actividad no tiene Resultados
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.indicadores.empty
= Este Resultado no tiene Indicadores
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.periodoInicial.labe
l = Periodo Inicial
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.periodoFinal.label
= Periodo Final
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.duracion.label
= Duración
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.medioVerif.label
= Medio Verificación
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.mesCumpli.label
= Mes de Cumplimiento
gestion.beneficiosTributarios.proyectosAntiguos.marco.resultados.label
= Actividades, Resultados e Indicadores
#Presupuesto
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tab
= Presupuesto
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.loading
= CARGANDO RUBROS
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.rubro.header
= Rubro
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.monInstitucionSol.header
= Monto financiado solicitado
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEspecieSol.header
= Contrapartida especie solicitado
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEfectivoSol.header
= Contrapartida efectivo solicitado
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.totalSol.header
= Total solicitado
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.monInstitucionAprob.header
= Monto financiado aprobado
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEspecieAprob.header
= Contrapartida especie aprobado
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEfectivoAprob.header
= Contrapartida efectivo aprobado
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.totalAprob.header
= Total aprobado

```

```

gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.entidad
= Entidad
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.anio
= Año
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tipoPresupuesto
= Tipo Presupuesto
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.modalidad
= Modalidad
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.financiado
= Financiado
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.contrapartida
= Contrapartida
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.especie
= Especie
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.efectivo
= Efectivo
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tipoFinanciacion
= Tipo Financiación
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.propuesto
= Monto Propuesto
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.aprobado
= Monto Aprobado
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.rubro
= Rubro
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.nDias
= # Días
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.nPersonas
= # Personas
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.cantidad
= Cantidad
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.costoPersona
= Costo por Persona
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.valorUnitario
= Valor Unitario
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tema
= Tema
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.descripcion
= Descripción
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.lugarViaje
= Lugar de viaje
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.lugarJustificacion
= Lugar / Justificación
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.justificacion
= Justificación
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.nombres
= Nombres y Apellidos
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.funcion
= Función
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.dedicacionMeses
= Dedicación Meses
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.dedicacionSemanas
= Dedicación Semanas
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.proveedor
= Proveedor
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.institucion
= Institución Financiadora
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tipoFinanciacion
= Tipo Financiación
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.total
= Total

```

```

gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.modificar
= Modificar Rubros
gestion.beneficiosTributarios.proyectosAntiguos.ficha.loading
= Cargando Ficha, espere por favor...
#Registro Importaciones
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.header
= Registro de Importaciones
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.modalidad.label
= Modalidad de Importación
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.numeroIncomex.label
= No. Incomex
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaInicioPlan.label
= Fecha Inicio Plan
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaFinalPlan.label
= Fecha Fin Plan
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.numeroLicencia.label
= N° Licencia
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.planImportacion.label
= Plan Importación
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaCertificado.label
= Fecha Certificado
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaVencimiento.label
= Fecha Vencimiento
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.valorLicencia.label
= Valor Importación
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.valorAprobado.label
= Valor Aprobado
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.valorExcluido.label
= Valor Excluido
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.aprobado.label
= Aprobado
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.modificable.label
= Con Modificación
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.numeroActaDecision.labe
el
= No. Acta Decisión
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaActaDecision.labe
l
= Fecha Acta Decisión
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.anoResolucionExencion.
label
= Año Resolución Exención
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.numeroResolucionExenci
on.label
= No. Resolución Exención
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaNotificacionExenc
ion.label
= Fecha Notificacion Exencion
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.planImportacion.error
= El plan de Importacion no posee documento adjunto
#Registro Detalles Importaciones
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.button
= Detalle Importación
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.header
= Detalles Importación
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.cantidad.label
= Cantidad
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.posicionArance
laria.label
= Posición Arancelaria
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.valorSubpartid
a.label
= Valor Total Subpartida
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.descripcionEqu
ipo.label
= Equipo o Elemento a Importar
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.impactoAmbient
al.label
= Impacto Ambiental del Equipo

```

```

gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.aprobado.label
= Aprobado
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.valorAprobado.
label
= Valor Aprobado
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.observaciones.
label
= Justificación Excluidos
#Evaluacion
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.emptyTable
= No se encontraron Evaluadores Asignados
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.infoGeneral
= Información General
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.tipoEvaluacion
= Tipo Evaluación
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.estadoEvaluador
= Estado Evaluador
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.panel
= Panel de Evaluación
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.panel.nombrePanel
= Nombre Panel
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.panel.actaPanel
= Acta panel
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.infoEvaluador
= Información Evaluador
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.tipoEvaluador
= Tipo Evaluador
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.dependencia
= Dependencia
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.funcionario
= Funcionario
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.nombres
= Nombres
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.usuarioSigp
= Usuario SIGP
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.evaluacion
= Evaluación
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.fechaAsignacionEvalua
dor
= Fecha Asignación Evaluador
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.fechaEvaluacion
= Fecha Evaluación
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.duracionEvaluacion
= Duración Evaluación (días)
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.conceptoFinal
= Concepto Final
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.recomenFinal
= Recomendación Final
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.concepto
= Concepto
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.recomendacionEvaluado
r
= Recomendación Evaluador
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.puntaje
= Puntaje
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.documentoEvaluacion
= Documento Evaluación
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.pago
= Pago
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.valorEvaluacion
= Valor Evaluación
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.fechaPago
= Fecha Pago
#Documentos

```

```
gestion.beneficiosTributarios.proyectosAntiguos.documentos.emptyTable
= No se encontraron Documentos asociados
#Observaciones
gestion.beneficiosTributarios.proyectosAntiguos.observaciones.emptyTable
= No se encontraron Observaciones
```

```
#EN
##### Proyectos Beneficios Tributarios ANTIGUOS
#####

gestion.beneficiosTributarios.proyectosAntiguos.header =
Old Tax Benefits Projects
gestion.beneficiosTributarios.proyectosAntiguos.breadcrum =
Old Projects
gestion.beneficiosTributarios.proyectosAntiguos.title =
Old Projects
gestion.beneficiosTributarios.proyectosAntiguos.formularioProyectoId =
Project ID Form
gestion.beneficiosTributarios.proyectosAntiguos.codigoProyecto =
Registration Code
gestion.beneficiosTributarios.proyectosAntiguos.codigoProyecto.placeholder =
Enter the Registration Code
gestion.beneficiosTributarios.proyectosAntiguos.codigoProyectoId =
Project Code
gestion.beneficiosTributarios.proyectosAntiguos.codigoProyectoId.placeholder =
Enter the Project Code
gestion.beneficiosTributarios.proyectosAntiguos.titulo =
Title Project
gestion.beneficiosTributarios.proyectosAntiguos.titulo.placeholder =
Enter the Project Title
gestion.beneficiosTributarios.proyectosAntiguos.tipoProyecto =
Type Project
gestion.beneficiosTributarios.proyectosAntiguos.anioRegistro =
Year of Registration
gestion.beneficiosTributarios.proyectosAntiguos.fechaRegistro =
Registration Date
gestion.beneficiosTributarios.proyectosAntiguos.fechaEnvio =
Shipping Date
gestion.beneficiosTributarios.proyectosAntiguos.fechaRevision =
Date Review
gestion.beneficiosTributarios.proyectosAntiguos.tipoTramite =
Procedure
gestion.beneficiosTributarios.proyectosAntiguos.convocatoria =
Call
gestion.beneficiosTributarios.proyectosAntiguos.programa =
National Prog of CTeI
gestion.beneficiosTributarios.proyectosAntiguos.programa1 =
National Program 1
gestion.beneficiosTributarios.proyectosAntiguos.programa2 =
National Program 2
gestion.beneficiosTributarios.proyectosAntiguos.estado =
State
gestion.beneficiosTributarios.proyectosAntiguos.estado.proy =
Project Status
gestion.beneficiosTributarios.proyectosAntiguos.entidad =
Executing Entity
gestion.beneficiosTributarios.proyectosAntiguos.lugar =
Place of Execution
```


gestion.beneficiosTributarios.proyectosAntiguos.fechaModificacion	=
Last Modification	
gestion.beneficiosTributarios.proyectosAntiguos.fechaCreacion	=
Creation Date	
gestion.beneficiosTributarios.proyectosAntiguos.financiacion	=
Type of financing	
gestion.beneficiosTributarios.proyectosAntiguos.tipoGestion	=
Type Management	
gestion.beneficiosTributarios.proyectosAntiguos.duración	=
Duration	
gestion.beneficiosTributarios.proyectosAntiguos.ejecucion	=
Execution	
gestion.beneficiosTributarios.proyectosAntiguos.meses	=
Months	
gestion.beneficiosTributarios.proyectosAntiguos.fechaInicioProyecto	=
Project Start Date	
gestion.beneficiosTributarios.proyectosAntiguos.fechaFinalizacionProyecto	=
Project Completion Date	
gestion.beneficiosTributarios.proyectosAntiguos.palabras	=
Keywords	
gestion.beneficiosTributarios.proyectosAntiguos.montos	=
Project amounts	
gestion.beneficiosTributarios.proyectosAntiguos.montosProg	=
Amounts of the program	
gestion.beneficiosTributarios.proyectosAntiguos.montosFicTec	=
Amounts of the technical file	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos	=
Attached Requirements	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.btn	=
Requirements	
gestion.beneficiosTributarios.proyectosAntiguos.especie	=
Species	
gestion.beneficiosTributarios.proyectosAntiguos.dinero	=
Money	
gestion.beneficiosTributarios.proyectosAntiguos.solicitado	=
Requested	
gestion.beneficiosTributarios.proyectosAntiguos.total	=
Total	
gestion.beneficiosTributarios.proyectosAntiguos.radicadoBtn	=
Switch to filed	
gestion.beneficiosTributarios.proyectosAntiguos.rechazarBtn	=
Reject	
gestion.beneficiosTributarios.proyectosAntiguos.evaluacionBtn	=
Send to Evaluation	
gestion.beneficiosTributarios.proyectosAntiguos.registro.requisitos.error	=
You must review the requirements before moving the project to the evaluation stage	
gestion.beneficiosTributarios.proyectosAntiguos.registro.requisitos.error2	=
The project can not be passed to the evaluation stage because it does not meet one or more mandatory requirements	
gestion.beneficiosTributarios.proyectosAntiguos.pasarDiligenciado	= Go
to Diligence	
gestion.beneficiosTributarios.proyectosAntiguos.pasarDecision	=
Send to Decision	
gestion.beneficiosTributarios.proyectosAntiguos.devolverEvaluacion	=
Return to Evaluation	
gestion.beneficiosTributarios.proyectosAntiguos.devolverSecretaria	=
Return to the Technical Secretariat CNBT	
gestion.beneficiosTributarios.proyectosAntiguos.devolverRegistro	=
Return to Register	
gestion.beneficiosTributarios.proyectosAntiguos.subsanable	=
Subsanable	

gestion.beneficiosTributarios.proyectosAntiguos.agregarObservacion	=
Add Observation	
gestion.beneficiosTributarios.proyectosAntiguos.editarObservacion	=
Edit Observation	
gestion.beneficiosTributarios.proyectosAntiguos.hoy	=
Today	
gestion.beneficiosTributarios.proyectosAntiguos.ayer	=
Yesterday	
gestion.beneficiosTributarios.proyectosAntiguos.diasAtras	=
Days ago at	
gestion.beneficiosTributarios.proyectosAntiguos.fechaObservacion	=
Date Observation	
gestion.beneficiosTributarios.proyectosAntiguos.fechaObservacion.required	=
You Must Select Date Observation	
#Requisitos	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.header	=
Requirements	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.obligatorio.msg	=
Remember that requirements with an asterisk (*) are required.	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.requisito.label	=
Requirement	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.nombre.archivo.label	=
File Name	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.tamano.label	=
Size	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.progreso.label	=
Progress	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.progreso.ok	=
UPLOADED DOCUMENT	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.cumple.label	=
Verification Requirement	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.cumple.required	=
You must enter the type of compliance with the requirement	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.observaciones.label	=
Observations	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.certificado.error	=
You must attach all the required requirements	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.vacio	=
No Attachment	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.vacio.mensaje	=
You must attach a file to be able to add an observation	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.concepto.label	=
Evaluation panel concept	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.puntaje.label	=
Score	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.elegible.label	=
Eligible	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.boton.generar	=
Generate Certificate	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.fechaCumplimiento.label	=
Date Compliance	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.modal.label	=
Compliance with the requirement	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.tooltip.cumplimiento	=
Compliance with the requirement	
gestion.beneficiosTributarios.proyectosAntiguos.requisitos.irDetallesPry.btn	=
Go to Details	
gestion.beneficiosTributarios.proyectosAntiguos.observaciones.requisitos.title	=
General Comments of the Requirements	
gestion.beneficiosTributarios.proyectosAntiguos.observaciones.requisitos.label	=
Observations	

#pestañas	
gestion.beneficiosTributarios.proyectosAntiguos.tab.infoBasica	=
Basic Info	
gestion.beneficiosTributarios.proyectosAntiguos.tab.observaciones	=
Observations	
gestion.beneficiosTributarios.proyectosAntiguos.tab.revisionEvaluacion	=
Evaluation	
gestion.beneficiosTributarios.proyectosAntiguos.tab.exencion	=
Exemption	
gestion.beneficiosTributarios.proyectosAntiguos.tab.entidades	=
Entities	
gestion.beneficiosTributarios.proyectosAntiguos.tab.rolEntidades	=
Entities Roles	
gestion.beneficiosTributarios.proyectosAntiguos.tab.presupuesto	=
Budget	
gestion.beneficiosTributarios.proyectosAntiguos.tab.descripciones	=
Descriptions	
gestion.beneficiosTributarios.proyectosAntiguos.tab.impactos	=
Impacts	
gestion.beneficiosTributarios.proyectosAntiguos.tab.coberturas	=
Coverages	
gestion.beneficiosTributarios.proyectosAntiguos.coberturas.label	=
Coverages	
gestion.beneficiosTributarios.proyectosAntiguos.tab.productos	=
Products	
gestion.beneficiosTributarios.proyectosAntiguos.tab.cronograma	=
Activities Schedule	
gestion.beneficiosTributarios.proyectosAntiguos.tab.tipoPersonal	=
Type of Staff	
gestion.beneficiosTributarios.proyectosAntiguos.tab.personal	=
Personal	
gestion.beneficiosTributarios.proyectosAntiguos.tab.marcoLogico	=
Logical Framework	
gestion.beneficiosTributarios.proyectosAntiguos.tab.registroImportaciones	=
Imports	
gestion.beneficiosTributarios.proyectosAntiguos.tab.documentos	=
Documents	
#	
gestion.beneficiosTributarios.proyectosAntiguos.entidades.empty	= No
Entities found	
gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.empty	= No
Entities Roles found	
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.empty	= No
Budget found	
gestion.beneficiosTributarios.proyectosAntiguos.impactos.empty	= No
Impacts found	
gestion.beneficiosTributarios.proyectosAntiguos.coberturas.empty	= No
Coverages found	
gestion.beneficiosTributarios.proyectosAntiguos.productos.empty	= No
Products found	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.empty	= No
Activities Schedule found	
gestion.beneficiosTributarios.proyectosAntiguos.personal.empty	= No
Personnel found	
gestion.beneficiosTributarios.proyectosAntiguos.marcoLogico.empty	= No
Logical Framework found	
gestion.beneficiosTributarios.proyectosAntiguos.descripcion.empty	= No
descriptions found	
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.empty	= No
Registered Imports were found	
#	

gestion.beneficiosTributarios.proyectosAntiguos.column.convocatoria	=
Call	
gestion.beneficiosTributarios.proyectosAntiguos.column.programa	=
Program	
gestion.beneficiosTributarios.proyectosAntiguos.column.tramite	=
Procedure	
gestion.beneficiosTributarios.proyectosAntiguos.column.fichaProyecto	=
Project tab	
gestion.beneficiosTributarios.proyectosAntiguos.column.fichaPrograma	=
Program tab	
gestion.beneficiosTributarios.proyectosAntiguos.column.busquedaAvanzada	=
Advanced Search	
gestion.beneficiosTributarios.proyectosAntiguos.reporteRubrosPry	=
Items and Projects Report	
gestion.beneficiosTributarios.proyectosAntiguos.reporteMarcoLogico	=
Logical Framework Report	
gestion.beneficiosTributarios.proyectosAntiguos.reporteEvaluacion	=
Evaluation Report	
gestion.beneficiosTributarios.proyectosAntiguos.nivelMadurezTecnologica	=
Level of technological maturity	
gestion.beneficiosTributarios.proyectosAntiguos.tipoPersona	=
Type person	
gestion.beneficiosTributarios.proyectosAntiguos.CiudadFormacion	=
Formation city	
gestion.beneficiosTributarios.proyectosAntiguos.valorInscripcion	=
Registration value	
gestion.beneficiosTributarios.proyectosAntiguos.viaPresentacion	=
Presentation Route	
gestion.beneficiosTributarios.proyectosAntiguos.origenInversion	=
Ownership Information	
gestion.beneficiosTributarios.proyectosAntiguos.vinculoLaboral	=
Employment relationship	
#Exencion	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.fechaSolicitud	=
Date Request	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.codigoProyectoExencion	=
Project Code Exemption	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.fechaInicio	=
Date Start	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.fechaFin	=
End Date	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.caracter	=
Character Project	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.descripcion	=
Character	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.justificacion	=
Justification Character	
gestion.beneficiosTributarios.proyectosAntiguos.exencion.valor	=
Requested value	
#Entidades	
gestion.beneficiosTributarios.proyectosAntiguos.entidades.entidad	=
Entity	
gestion.beneficiosTributarios.proyectosAntiguos.entidades.nit	=
NIT	
#Rol Entidades	
gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.entidad	=
Entity	
gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.rol	=
Role	
#Descripciones	

gestion.beneficiosTributarios.proyectosAntiguos.tabla.orden	=
Order	
gestion.beneficiosTributarios.proyectosAntiguos.tabla.descripcion	=
Description	
gestion.beneficiosTributarios.proyectosAntiguos.cargando	=
LOADING DESCRIPTIONS...	
gestion.beneficiosTributarios.proyectosAntiguos.btn.ver	=
See description	
gestion.beneficiosTributarios.proyectosAntiguos.descripcion.header	=
Description	
#Impactos	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.label	
= Impacts	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.tipo	
= Type Impact	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.descripcion	
= Description	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.año	
= Year	
gestion.beneficiosTributarios.proyectosAntiguos.impacto.empty	
= No impacts found for this project	
#Cronograma	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.label	
= Activities Schedule	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.empty	
= No activities found	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.nActividad	
= N° Activity	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.actividad	
= Activity	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.inicio	
= Start	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.fin	
= End	
gestion.beneficiosTributarios.proyectosAntiguos.cronograma.ejecucion	
= Execution In	
#Tipo de Personal	
gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.label	
= Type of Staff	
gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.empty	
= No staff types found	
gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.tipoPersonal	
= Role in the Project	
gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.numeroPersonas	
= Number of People	
#Personal	
gestion.beneficiosTributarios.proyectosAntiguos.personal.label	
= Personal	
gestion.beneficiosTributarios.proyectosAntiguos.personal.empty	
= No Registered Persons were found	
gestion.beneficiosTributarios.proyectosAntiguos.personal.persona	
= Person	
gestion.beneficiosTributarios.proyectosAntiguos.personal.tipo	
= Project Linkage	
gestion.beneficiosTributarios.proyectosAntiguos.personal.identificacion	
= N° Identification	
gestion.beneficiosTributarios.proyectosAntiguos.personal.entidad	
= Entity	
gestion.beneficiosTributarios.proyectosAntiguos.personal.vinculo	
= Link	

```

gestion.beneficiosTributarios.proyectosAntiguos.personal.fechaNacimiento
= Date Birth
gestion.beneficiosTributarios.proyectosAntiguos.personal.pais
= Country
gestion.beneficiosTributarios.proyectosAntiguos.personal.email
= Email
gestion.beneficiosTributarios.proyectosAntiguos.personal.tipoIdentificación
= Type Identification
gestion.beneficiosTributarios.proyectosAntiguos.personal.numeroIdentificación
= Identification number
gestion.beneficiosTributarios.proyectosAntiguos.personal.genero
= Gender
gestion.beneficiosTributarios.proyectosAntiguos.personal.dedicacion
= Dedication (h / week)
gestion.beneficiosTributarios.proyectosAntiguos.personal.nMeses
= Number Months
gestion.beneficiosTributarios.proyectosAntiguos.personal.funcion
= Function in the project
gestion.beneficiosTributarios.proyectosAntiguos.personal.registradoCvLAC
= Registered in CvLAC?
gestion.beneficiosTributarios.proyectosAntiguos.personal.documentoHojaVida
= Attached Document Curriculum vitae
gestion.beneficiosTributarios.proyectosAntiguos.personal.poblacionVictimaConflicto
= Population Victim of the Armed Conflict
gestion.beneficiosTributarios.proyectosAntiguos.personal.poblacionCondicionDisc
= Population in a Situation of Disability
gestion.beneficiosTributarios.proyectosAntiguos.personal.grupoEtnico
= Ethnic Group
gestion.beneficiosTributarios.proyectosAntiguos.personal.grupoEtario
= Age Group
gestion.beneficiosTributarios.proyectosAntiguos.personal.categoria
= Category
#Marco Logico
gestion.beneficiosTributarios.proyectosAntiguos.marco.label
= Logical Framework
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoGral
= General Objective
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoGeneral.empty
= Not found General Objective
gestion.beneficiosTributarios.proyectosAntiguos.marco.meta
= Goal
gestion.beneficiosTributarios.proyectosAntiguos.marco.fuenteVerificacion
= Source Verification
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivosEspecificos.label
= Specific Objectives
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivosEspecificos.empty
= No Specific Objectives Found
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.label
= Specific Objective N°
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.objetivo.label
= Specific Objective
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.meta.label
= Goal
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.fuente.label
= Source Verification
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.indicador.label
= Indicator
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.actividades.empty
= This Objective has no Activities
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.resultados.empty
= This Activity has no Results

```

```

gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.indicadores.empty
= This Result does not have Indicators
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.periodoInicial.labe
l = Initial Period
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.periodoFinal.label
= Final Period
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.duracion.label
= Duration
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.medioVerif.label
= Medium Verification
gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.mesCumpli.label
= Month of Compliance
gestion.beneficiosTributarios.proyectosAntiguos.marco.resultados.label
= Activities, Results and Indicators
#Presupuesto
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tab
= Budget
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.loading
= LOADING ITEMS
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.rubro.header
= Item
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.monInstitucionSol.header
= Requested financed amount
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEspecieSol.header
= Species counterpart requested
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEfectivoSol.header
= Cash counterpart requested
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.totalSol.header
= Total requested
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.monInstitucionAprob.header
= Approved financed amount
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEspecieAprob.header
= Species counterpart approved
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEfectivoAprob.header
= Approved cash counterpart
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.totalAprob.header
= Approved total

gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.entidad
= Entity
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.anio
= Year
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tipoPresupuesto
= Budget Type
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.modalidad
= Mode
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.financiado
= Funded
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.contrapartida
= Counterpart
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.especie
= Species
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.efectivo
= Cash
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tipoFinanciacion
= Financing Type
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.propuesto
= Proposed Amount
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.aprobado
= Approved Amount

```

```

gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.rubro
= Item
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.nDias
= # Days
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.nPersonas
= # People
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.cantidad
= Quantity
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.costoPersona
= Cost per Person
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.valorUnitario
= Unit Value
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tema
= Theme
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.descripcion
= Description
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.lugarViaje
= Travel place
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.lugarJustificacion
= Place / Justification
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.justificacion
= Justification
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.nombres
= First and Last Names
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.funcion
= Function
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.dedicacionMeses
= Dedication Months
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.dedicacionSemanas
= Dedication Weeks
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.proveedor
= Provider
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.institucion
= Financing Institution
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.tipoFinanciacion
= Financing Type
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.total
= Total
gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.modificar
= Modify Items
gestion.beneficiosTributarios.proyectosAntiguos.ficha.loading
= Loading File, please wait...
#Registro Importaciones
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.header
= Import Registry
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.modalidad.label
= Import Modality
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.numeroIncomex.label
= No. Incomex
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaInicioPlan.label
= Plan Start Date
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaFinalPlan.label
= Plan End Date
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.numeroLicense.label
= License Number
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.planImportacion.label
= Import Plan
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaCertificado.label
= Certificate Date
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaVencimiento.label
= Expiration Date

```



```

gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.valorLicencia.label
= Value Import
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.valorAprobado.label
= Approved Value
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.valorExcluido.label
= Excluded Value
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.aprobado.label
= Approved
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.modificable.label
= With Modification
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.numeroActaDecision.labe
el          = Decision Minutes Number
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaActaDecision.labe
l          = Decision Minutes Date
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.anoResolucionExencion.
label      = Year Resolution Exemption
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.numeroResolucionExenci
on.label   = Exemption Resolution Number
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.fechaNotificacionExenc
ion.label  = Exemption Notification Date
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.planImportacion.error
= The import plan has no attached document
#Registro Detalles Importaciones
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.button
= Detail Import
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.header
= Import Details
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.cantidad.label
= Amount
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.posicionArance
laria.label = Tariff Position
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.valorSubpartid
a.label     = Total Value Subheading
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.descripcionEqu
ipo.label   = Equipment or Item to Import
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.impactoAmbient
al.label    = Environmental Impact of the Team
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.aprobado.label
= Approved
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.valorAprobado.
label      = Approved Value
gestion.beneficiosTributarios.proyectosAntiguos.registroImportaciones.detalle.observaciones.
label      = Justification Excluded
#Evaluacion
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.emptyTable
= No Assigned Evaluators Found
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.infoGeneral
= General Information
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.tipoEvaluacion
= Evaluation Type
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.estadoEvaluador
= Evaluator Status
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.panel
= Evaluation Panel
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.panel.nombrePanel
= Panel Name
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.panel.actaPanel
= Panel Minutes
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.infoEvaluador
= Evaluator Information

```



```
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.tipoEvaluador
= Evaluator Type
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.dependencia
= Dependency
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.funcionario
= Official
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.nombres
= Names
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.usuarioSigp
= SIGP User
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.evaluacion
= Evaluation
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.fechaAsignacionEvalua
dor = Evaluator Assignment Date
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.fechaEvaluacion
= Evaluation Date
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.duracionEvaluacion
= Evaluation Duration (days)
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.conceptoFinal
= Final Concept
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.recomenFinal
= Final Recommendation
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.concepto
= Concept
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.recomendacionEvaluado
r = Evaluator Recommendation
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.puntaje
= Score
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.documentoEvaluacion
= Evaluation Document
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.pago
= Payment
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.valorEvaluacion
= Evaluation Value
gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.fechaPago
= Payment Date
#Documentos
gestion.beneficiosTributarios.proyectosAntiguos.documentos.emptyTable
= No associated documents found
#Observaciones
gestion.beneficiosTributarios.proyectosAntiguos.observaciones.emptyTable
= No Comments Found
```

Tabla de visualización de la información en la pantalla de Proyectos Antiguos de Beneficios Tributarios

El siguiente representa una tabla de visualización de datos utilizando **PrimeFaces** dentro de una aplicación basada en **JSF (JavaServer Faces)**. La tabla está diseñada para mostrar información sobre **auditorías** dentro del sistema y cuenta con funcionalidades avanzadas como **lazy loading**, **filtros**, **ordenamiento**, **selección de filas**, **expansión de filas**, y **acciones personalizadas**.

```
<!-- TABLA PRINCIPAL -->
<div class="row">
  <div class="col-lg-12" style="overflow-x: scroll;margin-bottom: 40px;">
```

```
<p:dataTable id="Table"
value="#{proyectoBeneficioAntiguoSessController.modelo}"
var="proyecto"
lazy="true"
sortBy="#{proyecto.proyectoId}"
sortOrder="DESCENDING"
currentPageReportTemplate="Total de registros:
{totalRecords}"
styleClass="table table-striped table-bordered table-
hover dataTable-entidades"
reflow="true"
rowKey="#{proyecto.proyectoId}"
paginator="true"
paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
paginatorPosition="bottom"
emptyMessage="#{msg['sb.datatable.registro.empty']}"
widgetVar="items#{proyectoBeneficioAntiguoSessController.modulo}"
rows="10">
<f:facet name="header">
<p:outputPanel style="text-align: right;"
rendered="#{proyectoBeneficioAntiguoSessController.buscarVisible}">
<h:outputText
value="#{msg['sb.component.tableHeader.buscar']}" />
<p:inputText id="globalFilter"
placeholder="#{msg['sb.component.tableHeader.buscar.placeholder']}"
onkeyup="PF('items#{proyectoBeneficioAntiguoSessController.modulo}').filter()"
style="width:200px" disabled="#{proyectoBeneficioAntiguoSessController.programaSelected !=
null or proyectoBeneficioAntiguoSessController.filtroActivo eq 'true'}"/>
</p:outputPanel>
</f:facet>
<p:column style="text-align: center; padding: 5px;"
width="3%">
<p:rowToggler/>
</p:column>
<p:column filterBy="#{solicitud.fechaRegistro}"
filterMatchMode="contains"
filterStyle="display:none;"
width="5%"
style="text-align: center"
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.anioRegistro']}">
<h:outputText value="#{proyecto.fechaRegistro != null ?
proyecto.fechaRegistro : ''}">
<f:convertDateTime type="date" pattern="yyyy"/>
</h:outputText>
</p:column>
<p:column filterBy="#{proyecto.proyectoId}" filterMatchMode="contains"
rendered="false"
filterStyle="display:none;"
width="8%"
style="text-align: center"
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.codigoProyecto']}">
<h:outputText value="#{proyecto.proyectoId}" />
</p:column>
<p:column filterBy="#{proyecto.codigoProyecto}"
filterMatchMode="contains">
```

```

filterStyle="display:none;"
width="12%"
style="text-align: center"

headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.codigoProyectoId']}">
    <h:outputText value="#{proyecto.codigoProyecto}" />
</p:column>
<p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.titulo']}"
    filterBy="#{proyecto.titulo}"
filterMatchMode="contains"
    filterStyle="display:none;"
    >
        <h:outputText id="tituloProyecto" value="#{proyecto.titulo}" />
        <p:tooltip id="tituloProyectoTooltip" for="tituloProyecto"
position="right">
            <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.column.convocatoria']} : </b>
            #{proyecto.convocatoriaId.descripcionConvocatoria}<br/>
            <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.column.programa']} : </b>
            #{proyecto.proyectoPadreId eq null ? proyecto.dependenciaId.nombreDependencia :
            proyectoAntiguoConvSessController.consultarProgramaPadre(proyecto.proyectoPadreId)}<br/>
            </p:tooltip>
        </p:column>
</p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.entidad']}"
    filterBy="#{proyecto.entidadAntiguaId.nombreEntidad}"
    filterMatchMode="contains"
    filterStyle="display:none;"
    >
        <h:outputText value="#{proyecto.entidadAntiguaId ne null ?
proyecto.entidadAntiguaId.nombreEntidad : '' }"/>
    </p:column>
<p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.estado']}"
filterBy="#{proyecto.estadoProyectoId.descripcionValorDefinicion}"
    filterMatchMode="contains"
    filterStyle="display:none;"
    style="text-align: center;"
    width="10%"
    >
        <small style="white-space: normal;"
            class="label label-
            #{proyecto.estadoProyectoId.descripcionValorDefinicion eq 'DILIGENCIANDO' ? 'primary'
:
proyecto.estadoProyectoId.descripcionValorDefinicion eq 'RADICADO' ? 'warning'
:
(proyecto.estadoProyectoId.descripcionValorDefinicion eq 'RECHAZADO'
or
proyecto.estadoProyectoId.descripcionValorDefinicion.contains('RETIRADO')) ? 'danger' :
''}">
        #{proyecto.estadoProyectoId.descripcionValorDefinicion}
    </small>
    </p:column>
</p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.column.tramite']}"
filterBy="#{proyecto.tramiteExencionId.descripcionValorDefinicion}"

```

```

filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center;"
width="8%"
>
<small style="white-space: normal;">

#{proyecto.tramiteExencionId.descripcionValorDefinicion}
</small>
</p:column>
<p:column headerText="#{msg['sb.component.boton.opciones']}"
width="8%"
>
<p:commandLink
update=":#{proyectoBeneficioAntiguoSessController.modulo}FormDetalle"
style="color: #fff"
process="@this"
id="goDetalles"

onstart="PF('blockTable').show();"

actionListener="#{proyectoBeneficioAntiguoSessController.cargarDetallesProyecto(proyecto)}"

title="#{msg['sb.component.boton.detalles']}"

primary sbTooltip"
styleClass=" btn btn-xs btn-
oncomplete="cambiarTab();
scrolllear();PF('blockTable').hide();">
<i class="fa fa-list"></i>
</p:commandLink>
<p:spacer></p:spacer>
<p:commandLink
title="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.requisitos.btn']}"
styleClass=" btn btn-xs btn-
primary sbTooltip"

action="/app/beneficiosTributarios/requisitosAntiguosBt"
ajax="false"

actionListener="#{requisitoAntiguoSessController.cargarRequisitosProyecto(proyecto)}"
process="@this"
immediate="true"

rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST5410')}"
>
<f:setPropertyActionListener
target="#{requisitoAntiguoSessController.proyectoSelected}" value="#{proyecto}"/>
<i class="fa fa-cubes"></i>
</p:commandLink>
<p:spacer></p:spacer>
<p:commandLink title="#{msg['sb.component.boton.download']}" -
#{msg['gestion.beneficiosTributarios.proyectosAntiguos.column.fichaProyecto']}"
styleClass=" btn btn-xs btn-primary
sbTooltip"

onstart="descargarFicha('#{proyectoBeneficioAntiguoSessController.urlReportes}GenerarReporte
ProyectosAntiguos?proyecto_id=#{proyecto.proyectoId}&amp;formato=#{SeguridadSessController.f
ormatoReporte}')"
>
<i class="fa fa-download"></i>

```

```

        <f:setPropertyActionListener
value="#{proyecto.proyectoId}" target="#{SeguridadSessController.proyectoId}"/>
        </p:commandLink>
        <p:spacer></p:spacer>
    </p:column>
    <p:rowExpansion>
        <div class="mt-element-ribbon bg-grey-steel">
            <div class="row">
                <div class="col-lg-8 col-lg-offset-2">
                    <p:panelGrid columns="2"

columnClasses="label,value"
                                >

                                <h:outputText
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.codigoProyecto']}" />
                                <h:outputText
value="#{proyecto.proyectoId}" />
                                <h:outputText
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.fechaRegistro']}" />
                                <h:outputText
value="#{proyecto.fechaRegistro != null ? proyecto.fechaRegistro : ''}" />
                                <f:convertDateTime
type="date" pattern="dd/MM/yyyy"/>
                                </h:outputText>
                                <h:outputText
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.column.convocatoria']}" />
                                <h:outputText
value="#{proyecto.convocatoriaId.descripcionConvocatoria}" />
                                <h:outputText
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.column.programa']}" />
                                <h:outputText
value="#{proyecto.proyectoPadreId eq null
                                ? proyecto.dependenciaId.nombreDependencia
                                :
                                proyectoBeneficioAntiguoConvController.consultarProgramaPadre(proyecto.proyectoPadreId)}" />
                                <h:outputText
value="#{msg['sb.menu.gestion.usuarioCreador']}" />
                                <h:outputText
value="#{proyecto.usuarioCreador}" />
                                <h:outputText
value="#{msg['sb.menu.gestion.fechaCreacion']}" />
                                <h:outputText
value="#{proyecto.fechaCreacion}" />
                                <f:convertDateTime
locale="es" pattern="dd/MM/yyyy"/>
                                </h:outputText>
                                </p:panelGrid>
                            </div>
                        </div>
                    <p:spacer/>
                </div>
            </p:rowExpansion>
        </p:dataTable>
    <p:blockUI block="Table" widgetVar="blockTable">
        #{msg['sb.mensajes.cargando.label']}<br />
    </p:blockUI>

```

```

        <div class="progress progress-striped active m-b-sm">
            <div style="width: 100%;" class="progress-bar"></div>
        </div>
    </p:blockUI>
    <p:blockUI block="Table" widgetVar="blockFicha">

    #{msg['gestion.beneficiosTributarios.proyectosAntiguos.ficha.loading']}<br />
        <div class="progress progress-striped active m-b-sm">
            <div style="width: 100%;" class="progress-bar"></div>
        </div>
    </p:blockUI>
    <p:blockUI block="contenidoPagina" widgetVar="blockContent">
        #{msg['sb.mensajes.descargando.espera.label']}<br />
        <div class="progress progress-striped active m-b-sm">
            <div style="width: 100%;" class="progress-bar"></div>
        </div>
    </p:blockUI>
    </div>
</div>

```

◆ Sección: Contenedor Principal

```

<div class="row">
    <div class="col-lg-12" style="overflow-x: scroll;margin-bottom: 40px;">

```

- Se utiliza Bootstrap para definir el layout.
- overflow-x: scroll permite visualizar toda la tabla en pantallas pequeñas, forzando scroll horizontal si hay muchas columnas.

◆ p:dataTable – Definición de Tabla

```

<p:dataTable id="Table"
    value="#{ proyectoBeneficioAntiguoSessController.model}"
    var="proyecto"
    lazy="true"
    ...
    rows="10">

```

- id="Table": identificador del componente.
- value="#{ proyectoBeneficioAntiguoSessController.model}": binding a la lista de datos (LazyDataModel), lo cual activa la carga bajo demanda (paginada).
- var="proyecto": nombre de la variable de iteración.
- lazy="true": activa la carga perezosa, optimizando el rendimiento en listas grandes.
- rows="10": muestra 10 registros por página.

◆ Orden y Paginación

```
sortBy="#{proyecto.proyectoId}"
sortOrder="DESCENDING"
paginator="true"
paginatorTemplate="..."
```

- Ordena inicialmente por ID del proyecto, de forma descendente.
- Se habilita paginación con una plantilla personalizada ({FirstPageLink} ... {RowsPerPageDropdown}).

◆ Filtro Global

```
<f:facet name="header">
  <p:outputPanel rendered="#{ proyectoBeneficioAntiguoSessController.buscarVisible}">
    ...
    <p:inputText id="globalFilter"
      onkeyup="PF('items#{
proyectoBeneficioAntiguoSessController.modulo}').filter()"
      disabled="#{
proyectoBeneficioAntiguoSessController.programaSelected != null or
proyectoBeneficioAntiguoSessController.filtroActivo eq 'true'}"/>
  </p:outputPanel>
</f:facet>
```

- Campo de texto para búsqueda global (no por columna).
- Se desactiva cuando hay un programa seleccionado o un filtro activo, evitando sobrecargar los filtros.

◆ Columnas

◆ Columna: Expandir Fila

```
<p:column style="text-align: center;" width="3%">
  <p:rowToggler/>
</p:column>
```

- Agrega ícono de expandir para mostrar información detallada del proyecto (ver p:rowExpansion abajo).

♦ **Columna: Año de Registro**

```
<p:column filterBy="#{solicitud.fechaRegistro}" ...>
  <h:outputText value="#{proyecto.fechaRegistro}">
    <f:convertDateTime pattern="yyyy"/>
  </h:outputText>
</p:column>
```

- Muestra solo el año de la fecha de registro.
- **Nota:** el filtro se realiza sobre solicitud.fechaRegistro, pero el valor mostrado es proyecto.fechaRegistro → esto podría ser un bug o error de referencia.

♦ **Columna: Código del Proyecto**

```
<p:column filterBy="#{proyecto.codigoProyecto}" ...>
  <h:outputText value="#{proyecto.codigoProyecto}" />
</p:column>
```

- Muestra el código asignado al proyecto (posiblemente diferente del ID).

♦ **Columna: Título con Tooltip**

```
<p:column filterBy="#{proyecto.titulo}" ...>
  <h:outputText id="tituloProyecto" value="#{proyecto.titulo}" />
  <p:tooltip for="tituloProyecto">
    <b>Convocatoria: </b> #{proyecto.convocatoriaId.descripcionConvocatoria}<br/>
    <b>Programa: </b>
    #{proyectoBeneficioAntiguoConvController.consultarProgramaPadre(...)}
  </p:tooltip>
</p:column>
```

- Muestra el título del proyecto con un tooltip contextual, enriquecido con datos relacionados (convocatoria y programa).

♦ **Columna: Estado del Proyecto con Estilo**

```
<small class="label label-#{...}">
  #{proyecto.estadoProyectoId.descripcionValorDefinicion}
</small>
```


- Se asigna un color al estado usando clases de Bootstrap:
 - primary: DILIGENCIANDO
 - warning: RADICADO
 - danger: RECHAZADO o RETIRADO
- Ayuda visualmente a identificar el estado.

◆ Columna: Tramite

```
<small style="white-space: normal;">
#{proyecto.tramiteExencionId.descripcionValorDefinicion}
</small>
```

- Se traduce el campo como si es tramite de exención, deducción, etc.

◆ Columna de Opciones

```
<p:column headerText="Opciones">
  <p:commandLink
    actionListener="#{proyectoAntiguoViewController.cargarDetallesProyecto(proyecto)}" ...>
    <i class="fa fa-list"></i>
  </p:commandLink>
  ...
  <p:commandLink action="/app/ beneficiosTributarios/requisitosAntiguosBt "
    actionListener="#{
requisitoAntiguoSessController.cargarRequisitosProyecto(proyecto)}"
    rendered="#{SeguridadSessController.mapaMenu.containsKey(GEST3110)}">
    <i class="fa fa-cubes"></i>
  </p:commandLink>
  ...
  <p:commandLink onstart="descargarFicha(...)">
    <i class="fa fa-download"></i>
  </p:commandLink>
</p:column>
```

- **Detalle (fa-list):** Carga detalle y cambia de pestaña.
- **Requisitos (fa-cubes):** Redirige a requisitos (solo si el usuario tiene permiso).
- **Descarga (fa-download):** Inicia descarga del informe PDF o RTF.

◆ Expansión de Fila (p:rowExpansion)

```
<p:rowExpansion>
  <p:panelGrid columns="2">
    <h:outputText value="Código Proyecto"/>
    <h:outputText value="#{proyecto.proyectoId}"/>
    ...
  </p:panelGrid>
</p:rowExpansion>
```

- Se muestra un resumen expandido del proyecto: ID, fecha registro, convocatoria, dependencia, usuario creador y fecha creación.

◆ p:blockUI – Indicadores de Carga

```
<p:blockUI block="Table" widgetVar="blockTable">...</p:blockUI>
```

- Bloquea la tabla durante procesos largos (carga de detalles, descarga).
- Mejora experiencia de usuario evitando múltiples clics y errores.

• Botón de Filtro por Búsqueda Avanzada

```
<!-- FILTRO AVANZADO -->
<h:panelGroup layout="block" id="formaFiltro" class="row">
  <div class="col-lg-12">
    <ui:fragment rendered="#{proyectoBeneficioAntiguoSessController.buscarVisible
and proyectoBeneficioAntiguoSessController.programaSelected == null}">
      <button type="button" class="btn btn-white btn-bitbucket" data-
toggle="collapse" data-target="#demo"><i class="fa fa-filter"></i>
      <#{msg['gestion.beneficiosTributarios.proyectosAntiguos.column.busquedaAvanzada']}</button>
    </ui:fragment>
    <div id="demo" class="collapse
#{proyectoBeneficioAntiguoSessController.filtroActivo ? 'in' : ''} well form-horizontal"
style="padding: 20px">
      <div class="row">
        <div class="col-lg-6" jsf:rendered="false">
          <div class="form-group">
            <p:outputLabel
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.codigoProyecto']}"
styleClass="col-lg-4
control-label"
for="codigoRegistroBusq"/>
          <div class="col-lg-8">
```

```

                                <p:inputText
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.proyectoId}"
styleClass="form-control"
id="codigoRegistroBusq"
placeholder="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.codigoProyecto.placeholder']}"
                                />
                                </div>
                                </div>
                                </div>
                                <div class="col-lg-6">
                                <div class="form-group">
                                <p:outputLabel
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.codigoProyectoId']}"
                                styleClass="col-lg-4
control-label"
for="codigoProyectoBusq"/>
                                <div class="col-lg-8">
                                <p:inputText
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.codigoProyecto}"
styleClass="form-control"
id="codigoProyectoBusq"
placeholder="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.codigoProyectoId.placeholder']}"
                                />
                                </div>
                                </div>
                                </div>
                                <div class="col-lg-6">
                                <div class="form-group">
                                <p:outputLabel
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.anioRegistro']}"
                                styleClass="col-lg-4
control-label"
for="anioRegistroBusq"/>
                                <div class="col-lg-8">
                                <p:selectOneMenu
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.anioRegistro}"
id="anioRegistroBusq"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains"

```

```

itemLabel="#{msg['sb.selectOneMenu.registro']}" />
value="#{proyectoBeneficioAntiguoSessController.listAnios}" />
</p:selectOneMenu>
</div>
</div>
</div>
<div class="row">
  <div class="col-lg-6">
    <div class="form-group">
      <p:outputLabel
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.estado']}"
control-label"
styleClass="col-lg-4
for="estadoBusq-
#{msg['sb.convert.selectOne.estadoProyecto.ID']}" />
      <div class="col-lg-8">
        <p:selectOneMenu
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.estadoProyectoId}"
id="estadoBusq-#{msg['sb.convert.selectOne.estadoProyecto.ID']}"
converter="valorDefinicionConverter"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains"
      >
        <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
        <f:selectItems
value="#{proyectoBeneficioAntiguoSessController.listaEstados}"
var="valorDef"
itemLabel="#{valorDef.descripcionValorDefinicion}"
itemValue="#{valorDef}"
      />
      </p:selectOneMenu>
    </div>
  </div>
  <div class="col-lg-6">
    <div class="form-group">
      <p:outputLabel
value="#{msg['gestion.beneficiosTributarios.proyectos.tipoTramite']}"
control-label"
styleClass="col-lg-4
for="tipoTramite-
AD"/>

```

```

<div class="col-lg-8">
    <p:selectOneMenu id="tipoTramite-AD"
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.tramiteExencionId}"
converter="tipoTramiteBTConverter"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains"
    >
        <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
        <f:selectItems
value="#{proyectoBeneficioAntiguoSessController.listaTiposTramites}"
var="valorDef"
itemLabel="#{valorDef.descripcionValorDefinicion}"
itemValue="#{valorDef}"
    />
    </p:selectOneMenu>
</div>
</div>
</div>
<div class="row">
    <div class="col-lg-12">
        <div class="form-group">
            <p:outputLabel
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.programa']}"
styleClass="col-lg-2
control-label"
            for="programaBusq"/>
            <div class="col-lg-10">
                <p:selectOneMenu
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.dependenciaId}"
id="programaBusq"
converter="dependenciaConverter"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"

```

```

filterMatchMode="contains"
<f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
<f:selectItems
value="#{proyectoBeneficiosEvaluadoresSessController.listaDependencias}"
var="dependencia"
itemDescription="#{dependencia.nombreDependencia}"
itemLabel="#{dependencia.nombreDependencia}"
itemValue="#{dependencia}"
/>
</p:selectOneMenu>
</div>
</div>
</div>
<div class="row">
<div class="col-lg-12">
<div class="form-group">
<p:outputLabel
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.convocatoria']}"
styleClass="col-lg-2
control-label"
for="convocatoriaBusq-AntiguosBT"/>
<div class="col-lg-10">
<p:selectOneMenu
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.convocatoriaId}"
id="convocatoriaBusq-AntiguosBT"
converter="convocatoriaProyectosConverter"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains"
<f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
<f:selectItems
value="#{proyectoBeneficioAntiguoSessController.listaConvocatorias}"
var="convocatoria"
itemLabel="#{convocatoria.descripcionConvocatoria}"
itemValue="#{convocatoria}"
/>

```

```

                                </p:selectOneMenu>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>
                                <div class="row">
                                    <div class="col-lg-12">
                                        <div class="form-group">
                                            <p:outputLabel
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.titulo']}"
styleClass="col-lg-2
control-label"
for="tituloProyectoBusq"/>
                                <div class="col-lg-10">
                                    <p:inputText
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.titulo}"
styleClass="form-control"
id="tituloProyectoBusq"
placeholder="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.titulo.placeholder']}"
                                />
                                </div>
                                </div>
                                </div>
                                </div>
                                <div class="hr-line-dashed"></div>
                                <div class="row">
                                    <div class="col-lg-6">
                                        <div class="form-group">
                                            <div class="col-lg-8 col-lg-offset-6">
                                                <p:commandLink id="buscar"

process="formaFiltro"
update="Table"

actionListener="#{proyectoBeneficioAntiguoConvController.lazyLoad}"
styleClass="btn btn btn-primary pull-left"

                                >
                                <i class="fa fa-search"></i>

                                #{msg['sb.component.boton.filtrar']}
                                </p:commandLink>
                                </div>
                                </div>
                                </div>
                                <div class="col-lg-6">
                                    <div class="form-group">
                                        <div class="col-lg-8 col-lg-offset-4">
                                            <p:commandLink id="limpiar"

update="formaFiltro,Table"
process="@this"
oncomplete="initChosen();"

```

```

actionListener="#{proyectoBeneficioAntiguoConvController.limpiarFiltros}"
styleClass="btn btn-primary pull-right"
        <i class="fa fa-times"></i>
        </p:commandLink>
    </div>
    </div>
</div>
</div>
</h:panelGroup>

```

Esta sección se encuentra dentro del h:panelGroup con id="formaFiltro", y corresponde a una **interfaz colapsable** que permite aplicar filtros avanzados sobre la tabla p:dataTable de proyectos.

◆ Estructura General del Filtro

```
<h:panelGroup layout="block" id="formaFiltro" class="row">
```

- Contenedor visual con layout en bloque que agrupa todos los filtros.
- Usado como objetivo para process y update desde los botones de acción ("Filtrar", "Limpiar").

◆ Botón para Expandir/Contraer

```

<ui:fragment rendered="#{proyectoBeneficioAntiguoSessController.buscarVisible and
proyectoBeneficioAntiguoSessController.programaSelected == null}">
    <button type="button" class="btn btn-white btn-bitbucket" data-toggle="collapse" data-
target="#demo">
        <i class="fa fa-filter"></i> #{msg['gestion.
beneficiosTributarios.proyectosAntiguos.column.busquedaAvanzada']}
    </button>
</ui:fragment>

```

- Este botón permite al usuario expandir o colapsar los filtros.
- Solo se muestra si buscarVisible es true y no hay programa seleccionado.

◆ Panel de Filtros (id="demo")

```
<div id="demo" class="collapse #{ proyectoBeneficioAntiguoSessController.filtroActivo ? 'in' : '' } well form-horizontal" style="padding: 20px">
```

- collapse in: hace que el panel esté expandido si filtroActivo es true.
- Se aplica el estilo form-horizontal para distribuir etiquetas y campos.

Campos de Filtro

A continuación se explican los filtros implementados:

1. ID del Proyecto (Técnico)

```
<p:inputText value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.proyectoId}" .../>
```

- Permite buscar directamente por el ID técnico del proyecto (clave primaria en BD).

2. Código de Proyecto (Funcional)

```
<p:inputText value="#{ proyectoBeneficioAntiguoSessController.proyectoFilter.codigoProyecto}" .../>
```

- Busca por el código asignado al proyecto, distinto del ID interno.

3. Año de Registro

```
<p:selectOneMenu value="#{ proyectoBeneficioAntiguoSessController.proyectoFilter.anioRegistro}" ...>  
  <f:selectItems value="#{ proyectoBeneficioAntiguoSessController.listAnios}"/>  
</p:selectOneMenu>
```

- Selector desplegable con los años disponibles cargados desde el backend.

4. 📌 Estado del Proyecto

```
<p:selectOneMenu value="#{
proyectoBeneficioAntiguoSessController.proyectoFilter.estadoProyectoId}"
converter="valorDefinicionConverter" ...>
    <f:selectItems value="#{ proyectoBeneficioAntiguoSessController.listaEstados}" ... />
</p:selectOneMenu>
```

- Lista desplegable para seleccionar el estado del proyecto.
- Usa un converter para mapear entre la entidad y el valor en la vista.

5. 🏷️ Tipo de Proyecto

```
<p:selectOneMenu id="tipoTramite-AD"
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.tramiteExencionId}"
converter="tipoTramiteBTConverter"
style="width:100%;"
autoWidth="false"
panelStyle="width:180px"
effect="fade"
filter="true"
filterMatchMode="contains"
>
    <f:selectItem itemLabel="#{msg['sb.selectOneMenu.registro']}" />
    <f:selectItems value="#{proyectoBeneficioAntiguoSessController.listaTiposTramites}"
        var="valorDef"
        itemLabel="#{valorDef.descripcionValorDefinicion}"
        itemValue="#{valorDef}"
    />
</p:selectOneMenu>
```

- Determina el tipo de tramite

6. 🏠 Programa / Dependencia

```
<p:selectOneMenu value="#{proyectoAntiguoViewController.proyectoFilter.dependenciaId}"
converter="dependenciaConverter">
    <f:selectItems value="#{proyectoBeneficiosEvaluadoresSessController.listaDependencias}"
    ... />
</p:selectOneMenu>
```

- Filtro por la unidad académica o administrativa asociada.
- Usa un converter específico para dependencias.

7. Convocatoria

```
<p:selectOneMenu  
value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.convocatoriaId}"  
converter="convocatoriaProyectosConverter">  
  <f:selectItems value="#{proyectoBeneficioAntiguoSessController.listaConvocatorias}" ...  
</f:selectItems>  
</p:selectOneMenu>
```

- Filtro por convocatoria asociada al proyecto.
- Permite identificar la “ronda” o periodo de participación.

8. Título del Proyecto

```
<p:inputText value="#{proyectoBeneficioAntiguoSessController.proyectoFilter.titulo}" ... />
```

- Permite realizar búsqueda por coincidencia del texto del título.

Acciones del Filtro

Botón Filtrar

```
<p:commandLink id="buscar"  
  process="formaFiltro"  
  update="Table"  
  actionListener="#{ proyectoBeneficioAntiguoSessController.lazyLoad}"  
  styleClass="btn btn-primary">  
  <i class="fa fa-search"></i> #{msg['sb.component.boton.filtrar']}  
</p:commandLink>
```

- Procesa el formulario formaFiltro.
- Llama al método lazyLoad que probablemente recarga el LazyDataModel según los filtros actuales.
- Actualiza la tabla principal Table.

Botón Limpiar Filtros

```
<p:commandLink id="limpiar"
    process="@this"
    update="formaFiltro,Table"
    oncomplete="initChoosen();"

    actionListener="#{proyectoBeneficioAntiguoConvController.limpiarFiltros}"
    styleClass="btn btn-primary">
    <i class="fa fa-times"></i> #{msg['sb.component.boton.cancelar']}
</p:commandLink>
```

- Limpia todos los campos del filtro (se asume que limpiarFiltros reinicia el objeto proyectoFilter).
- Recarga el formulario y la tabla.
- Ejecuta JavaScript initChoosen() para volver a aplicar estilos o plugins (como Select2 o Chosen).

La pantalla se ve de la siguiente forma:

Antiguos proyectos de beneficios fiscales

Hogar / Beneficios fiscales / Proyectos antiguos

Proyectos antiguos

Búsqueda avanzada

ENCONTRAR TODO: <input type="text" value="Filtrar por palabra clave"/>							
	Año de registro	Código del proyecto	Título del proyecto	Entidad Ejecutora	Estado	Procedimiento	Opciones
☐	2015	53640406	Proyecto pruebas Luigi y Natalia 12345	desarrolladores de software	REVISADO	Ejecución de IVA	<input type="button" value="🔍"/> <input type="button" value="🔄"/> <input type="button" value="🗑️"/>
☐	2015	481553640404	Proyecto pruebas Luigi y Natalia 1234	A SERVIR YA CONSTRUIR "ASECO"	REVISADO	Ejecución de IVA	<input type="button" value="🔍"/> <input type="button" value="🔄"/> <input type="button" value="🗑️"/>
☐	2015	120324040396	Ejercicio Físico en sujetos sanos no entrenados nativos a altitud Moderada en comparación con Crónicamente aclimatados	PONTIFICIA UNIVERSIDAD JAVERIANA - PUJ - SEDE BOGOTÁ	REVISADO	Ejecución de IVA	<input type="button" value="🔍"/> <input type="button" value="🔄"/> <input type="button" value="🗑️"/>
☐	2013	324324040110	Prueba de estímulos Luigi Peña	ACADEMIA COLOMBIANA DE LA LENGUA	REVISADO	Ingresos no constitutivos de renta o ganancia ocasional	<input type="button" value="🔍"/> <input type="button" value="🔄"/> <input type="button" value="🗑️"/>
☐	2012	481554840086	PRUEBA	UNIVERSIDAD ANTONIO NARIÑO - SEDE BOGOTÁ - LIAN	REVISADO	Ejecución de IVA	<input type="button" value="🔍"/> <input type="button" value="🔄"/> <input type="button" value="🗑️"/>
☐	2011	502254831820	Desarrollo de materiales con propiedades de aislamiento térmico y acústico para aplicación en el sector de la construcción.	ACEROS DIACO SA	REVISADO	Deducción por donación	<input type="button" value="🔍"/> <input type="button" value="🔄"/> <input type="button" value="🗑️"/>
☐	2011	12302402319	Producción de ácido láctico por biotransformación a partir de residuos agroindustriales	UNIVERSIDAD DE LA SABANA - UNISABANA	COORDINADO	Ingresos no constitutivos de renta o ganancia ocasional	<input type="button" value="🔍"/> <input type="button" value="🔄"/> <input type="button" value="🗑️"/>
☐	2011	12302402333-2	Evaluación de la capacidad de captura de CO2 en medios líquidos y sólidos	UNIVERSIDAD DE LA SABANA - UNISABANA	COORDINADO	Deducción por inversión	<input type="button" value="🔍"/> <input type="button" value="🔄"/> <input type="button" value="🗑️"/>
☐	2011	511173405517	Evaluación de los fenómenos de glicosilación y sus implicaciones reproductivas en un modelo	UNIVERSIDAD VITTO	COORDINADO		<input type="button" value="🔍"/> <input type="button" value="🔄"/> <input type="button" value="🗑️"/>

Detalles de Proyectos Antiguos

◆ Funcionalidad General

El panel de detalles se muestra cuando el usuario hace clic en el botón “Detalles” desde la tabla principal de proyectos antiguos. Se despliega un formulario dividido en secciones que presentan información clave del proyecto y enlaces a acciones adicionales, como descarga de reportes y requisitos asociados.

◆ Estructura del Panel

🗑 Layout Principal

- Utiliza un `div.row` con dos columnas:
 - **Columna Izquierda (col-lg-10):** Contiene la información detallada del proyecto.
 - **Columna Derecha (col-lg-2):** Lista de palabras clave asociadas al proyecto.

✚ Encabezado del Panel

- Título del proyecto (`{{proyectoBeneficioAntiguoSessController.selected.titulo}}`).
- Botón “Volver” que ejecuta la función `cambiarTabInicial()` para regresar a la tabla principal.

📄 Información General

Usa elementos `<dl class="dl-horizontal">` para mostrar pares de datos (dt / dd) como:

- Código de proyecto
- Convocatoria
- Dependencia
- Estado del proyecto (con estilos condicionales para resaltar estado)
- Tipo de proyecto (Ficha técnica, Proyecto, Programa)
- Entidad ejecutora
- Duración, lugar de ejecución, ejecución del cronograma, tipo de financiación (si aplica)

📅 Fechas Importantes

- Registro
- Revisión
- Modificación (formateadas con f:convertDateTime)

 Subsanabilidad

- Representada con un h:selectBooleanCheckbox deshabilitado.

 Acciones Adicionales

 Requisitos del Proyecto

- Si el usuario tiene permisos (GEST5410 en SeguridadSessController.mapaMenu), se presenta un botón que redirige a /app/beneficiosTributarios/requisitosAntiguosBt con carga de requisitos mediante:

```
actionListener="#{requisitoAntiguoSessController.cargarRequisitosProyecto(...)}"
```

 Ficha del Proyecto/Programa

- Se permite descarga en formato PDF o Word vía descargarFicha(...), con parámetros dinámicos (proyecto_id y formato).

 Reporte Marco Lógico

- Se muestra si el menú FOR1060 está disponible.
- Acción delegada a proyectoBeneficioAntiguoConvController.crearReporteMarcoLogico(...).

 Reporte de Evaluación

- Disponible si countEvaluaciones > 0.
- Acción delegada a proyectoBeneficioAntiguoConvController.downloadReporteEvaluacion(...).

Palabras Clave

En la columna derecha, se listan las palabras clave del proyecto:

```
<ui:repeat value="#{proyectoBeneficioAntiguoSessController.selected.palabraClaveList}"
var="palabra">
  <li><a><i class="fa fa-tag"></i> #{palabra.palabra}</a></li>
</ui:repeat>
```

Consideraciones Técnicas

- La visibilidad de secciones depende de múltiples condiciones (rendered, valores nulos, tipo de proyecto).
- Uso de utilidades JS (cambiarTabInicial(), descargarFicha(...)).
- La vista depende de un backing bean llamado proyectoBeneficioAntiguoSessController y también interactúa con otros controladores como requisitoAntiguoSessController, proyectoBeneficioAntiguoConvController y SeguridadSessController.

2. Pestaña de Información Básica

Esta pestaña contiene información estructurada del proyecto y está organizada en subtabs o secciones internas, cada una con funcionalidades específicas:

- **Entidades**
- **Roles Entidades**
- **Presupuesto**
- **Descripciones**
- **Impactos**
- **Coberturas**
- **Actividades Cronograma**
- **Tipo de Personal**
- **Personal**

Cada una de estas secciones probablemente se carga de forma dinámica o condicional con su propia lógica en el controlador.

Las pestañas principales se gestionan con Bootstrap (<ul class="nav nav-tabs">) y los eventos de cambio de pestaña llaman funciones JavaScript:

```
<a href="#tab-basica" data-toggle="tab" onclick="cargarInformacionBasica();">
```

Las demás pestañas también siguen esta lógica.

Subpestaña Exención

◆ 1. Propósito del componente

Esta subpestaña permite **visualizar la información registrada de la exención tributaria** solicitada para el proyecto antiguo, incluyendo fechas, valores, códigos, y descripciones asociadas al beneficio.

```
<!-- Pestaña Exencion -->
<div id="tab-exencion" class="tab-pane active" >
  <div class="panel-body">
    <p:remoteCommand id="cargarExencionesRC" name="cargarExenciones"
update="exencionesProyecto" />
    <p:blockUI block="projecChildrens" trigger="cargarExencionesRC"
widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{projectoBeneficioAntiguoSessController.selected.convocatoriaId.tipoTramiteId !=
null
      ?
projectoBeneficioAntiguoSessController.selected.convocatoriaId.tipoTramiteId.descripcionValorDefinicion
      :
msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.exencion']}</b>
    </h2>
    <br/>
    <h:panelGroup id="exencionesProyecto">
      <div class="panel-group" id="accordionExencion">
        <h:panelGroup id="exencion" styleClass="animated fadeInLeft">
          <h:panelGroup layout="block" styleClass="well" >
            <div class="row">
              <div class="col-lg-12">
                <dl class="dl-horizontal">

                  <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.exencion.fechaSolicitud']}:<
/dt> <dd><h:outputText
value="#{projectoBeneficioAntiguoSessController.selected.fechaSolicitudExencion}"><f:convert
DateTime pattern="dd-MMM-yyyy" timeZone="EST"/></h:outputText></dd>
```



```

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.exencion.valor']}</dt>
    <dd><h:outputText
value="#{proyectoBeneficioAntiguoSessController.selected.valorSolicitadoExencion}"><f:conver
tNumber type="currency" currencySymbol="$" locale="es"/></h:outputText></dd>

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.exencion.codigoProyectoExenc
ion']}</dt>
    <dd>#{proyectoBeneficioAntiguoSessController.selected.codigoProyectoExencion}</dd>
        <p:separator/>

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.exencion.caracter']}</dt>
    <dd>#{proyectoBeneficioAntiguoSessController.selected.caracterProyectoId.descripcionValorDef
inicion}</dd>

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.exencion.fechaInicio']}</dt>
    <dd><h:outputText
value="#{proyectoBeneficioAntiguoSessController.selected.fechaInicioProyectoExencion}"><f:co
nvertDateTime pattern="dd-MMM-yyyy" timeZone="EST"/></h:outputText></dd>

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.exencion.fechaFin']}</dt>
    <dd><h:outputText
value="#{proyectoBeneficioAntiguoSessController.selected.fechaFinalProyectoExencion}"><f:con
vertDateTime pattern="dd-MMM-yyyy" timeZone="EST"/></h:outputText></dd>
        <p:separator/>

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.exencion.descripcion']}</dt>
    <dd style="text-align:
justify;">#{proyectoBeneficioAntiguoSessController.selected.descripcion}</dd>

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.exencion.justificacion']}</dt>
    <dd style="text-align:
justify;">#{proyectoBeneficioAntiguoSessController.selected.justificacion}</dd>
        </dl>
    </div>
</h:panelGroup>
</h:panelGroup>
</div>
</h:panelGroup>
</div>
</div>

```

◆ 2. Estructura general

```

<div id="tab-exencion" class="tab-pane active">
    <div class="panel-body">
        ...
    </div>
</div>

```

- La subpestaña está definida en un div con clase tab-pane, lo que indica que es parte de un sistema de pestañas.

- Tiene un panel-body que contiene el contenido visible.

◆ 3. Carga dinámica de datos

```
<p:remoteCommand id="cargarExencionesRC" name="cargarExenciones" update="exencionesProyecto" />
```

- Permite cargar los datos de exención dinámicamente vía JavaScript.
- Actualiza el componente exencionesProyecto.

🌀 Bloqueo visual con blockUI

```
<p:blockUI block="projecChildrens" trigger="cargarExencionesRC">
    #{msg['sb.mensajes.cargando.label']}
</p:blockUI>
```

- Muestra un indicador de carga mientras se ejecuta cargarExencionesRC.
- Bloquea visualmente el componente con id="projecChildrens".

◆ 4. Encabezado dinámico

```
<h2>
    <b>#{projecBeneficioAntiguoSessController.selected.convocatoriaId.tipoTramiteId !=
    null
        ?
        projecBeneficioAntiguoSessController.selected.convocatoriaId.tipoTramiteId.descripcionValorDefinicion
        : msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.exencion']}</b>
</h2>
```

- Muestra el nombre del tipo de trámite si está definido, o un texto por defecto del archivo de propiedades (.properties).

◆ 5. Datos mostrados (accordion principal)

```
<h:panelGroup id="exencionesProyecto">
    <div class="panel-group" id="accordionExencion">
        <h:panelGroup id="exencion" class="animated fadeInLeft">
```

- Agrupación principal que muestra los datos en un accordion.

- Tiene animación CSS fadeInLeft.

◆ 6. Contenido del formulario

Utiliza una estructura tipo dl (definition list) con etiquetas <dt> y <dd> para presentar los campos:

Fechas y valores

```
<dt>Fecha de solicitud:</dt> <dd><h:outputText value="#{...fechaSolicitudExencion}" /></dd>
<dt>Valor solicitado:</dt> <dd><h:outputText value="#{...valorSolicitadoExencion}" /></dd>
<dt>Fecha inicio / fin:</dt> <dd><h:outputText value="#{...fechaInicioProyectoExencion}" /></dd>
```

- Las fechas son convertidas con f:convertDateTime.
- Los valores monetarios usan f:convertNumber con símbolo \$ y configuración regional en español (locale="es").

Datos del proyecto

```
<dt>Código del proyecto de exención:</dt> <dd>#{...codigoProyectoExencion}</dd>
<dt>Carácter del proyecto:</dt> <dd>#{...caracterProyectoId.descripcionValorDefinicion}</dd>
```

- Muestra información cualitativa registrada en el proyecto, como su código y tipo.

Descripciones

```
<dt>Descripción:</dt> <dd style="text-align: justify;">#{...descripcion}</dd>
<dt>Justificación:</dt> <dd style="text-align: justify;">#{...justificacion}</dd>
```

- Información narrativa del proyecto exento.

◆ 7. Estética y diseño

- Usa class="well" para encuadrar visualmente los datos.
- Estilo responsivo con col-lg-12.
- Textos justificados para mejorar legibilidad en campos largos (style="text-align: justify;").

- Incluye separadores visuales con <p:separator/> para dividir secciones temáticamente.

✿ Integración con la lógica del backend

- **Bean:** proyectoBeneficioAntiguoSessController.selected
→ Contiene toda la información de la exención del proyecto cargada previamente al abrir la pestaña.
- **Método remoto:** cargarExenciones()
→ Se invoca desde JavaScript o al cambiar de pestaña para actualizar los datos dinámicamente.
- **Control visual de carga:**
→ p:blockUI se activa durante la ejecución para bloquear la pantalla y evitar interacciones innecesarias.

Subpestaña: Entidades

◆ Propósito

Esta subpestaña muestra las **entidades vinculadas al proyecto** antiguo, incluyendo su nombre, NIT y dígito de verificación, y permite consultar sus detalles.

```
<!-- Pestaña Entidades -->
<div id="tab-entidades" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarEntidadesRC" name="cargarEntidades"
update="entidadesProyecto"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarEntidadesProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarEntidadesRC"
widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.entidades']}</b>
    </h2>
    <br/>
    <h:panelGroup id="entidadesProyecto">
      <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoBeneficioAntiguoSessController.listaProyectoEntidades.size() eq 0 or
proyectoBeneficioAntiguoSessController.listaProyectoEntidades == null}">

    #{msg['gestion.beneficiosTributarios.proyectosAntiguos.entidades.empty']}
```

```

        </h:panelGroup>
        <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoBeneficioAntiguoSessController.listaProyectoEntidades.size() gt 0}">
            <p:dataTable
id="#{proyectoBeneficioAntiguoSessController.modulo}tableEntidades"
                var="entidad"

value="#{proyectoBeneficioAntiguoSessController.listaProyectoEntidades}"
                    paginator="true"
                    selectionMode="single"
                    reflow="true"
                    rowKey="#{entidad.entidadProyectoId}"
                    currentPageReportTemplate="Total de

registros: {totalRecords}"
                        paginatorTemplate="{CurrentPageReport}"
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
                            paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"

widgetVar="items#{proyectoBeneficioAntiguoSessController.modulo}"
                                rows="10"
                                >
                                    <p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.entidades.entidad']}"
                                        filterStyle="display:none;"
                                        >
                                            <h:outputText
value="#{entidad.nombreEntidad}"/>
                                                </p:column>
                                                <p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.entidades.nit']}"
                                                    filterStyle="display:none;"
                                                    width="30%"
                                                    style="text-align: center;"
                                                    >
                                                        <h:outputText value="#{entidad.nit} -
#{entidad.digitoVerificacion}"/>
                                                            </p:column>
                                                            <p:column
headerText="#{msg['sb.component.boton.detalles']}"
                                                                width="15%"
                                                                style="text-align: center;"
                                                                >
                                                                    <!-- row record -->
                                                                    <p:commandLink
title="#{msg['sb.component.boton.detalles']}"
                                                                        oncomplete="$('#dlgDetallesEntidad').modal();"
                                                                        update="formDetallesEntidad"
                                                                        styleClass="btn
btn-sm btn-primary"
                                                                            >
                                                                                <f:setPropertyActionListener
target="#{proyectoBeneficioAntiguoSessController.proyectoEntidadSelected}"
value="#{entidad}"/>
                                                                                    <i class="fa fa-list"></i>
                                                                                        </p:commandLink>
                                                                                        </p:column>

```

```
</p:dataTable>
    </h:panelGroup>
</h:panelGroup>
</div>
</div>
```

◆ Estructura

- **Activación de contenido:** Al hacer clic en la pestaña Entidades, se ejecuta la función cargarEntidades() a través de un <p:remoteCommand> con update="entidadesProyecto" y acción en proyectoAntiguoViewController.cargarEntidadesProyecto().

```
<p:remoteCommand id="cargarEntidadesRC" name="cargarEntidades"
    update="entidadesProyecto"
    actionListener="#{proyectoBeneficioAntiguoSessController.cargarEntidadesProyecto()}" />
```

- **Carga visual (bloqueo):** Se emplea un p:blockUI con efecto de carga mientras se ejecuta la llamada remota.
- **Vista sin resultados:**
Si listaProyectoEntidades está vacía o null, se muestra un mensaje indicando que no hay entidades registradas.
- **Vista con resultados:**
Si hay entidades, se presenta una tabla (p:dataTable) con paginación y filtrado oculto.

◆ Componentes de la tabla (p:dataTable)

- value="#{proyectoBeneficioAntiguoSessController.listaProyectoEntidades}"
- selectionMode="single" – permite seleccionar una fila.
- widgetVar="items#{proyectoBeneficioAntiguoSessController.modulo}" – permite interactuar con JavaScript.
- rows="10" – se muestran 10 registros por página.

Columnas:

4. **Entidad**
Muestra el nombre de la entidad (#{entidad.nombreEntidad})
5. **NIT**
Muestra el NIT con dígito de verificación (#{entidad.nit} - #{entidad.digitoVerificacion})

6. Botón de Detalles

- p:commandLink que abre el modal #dlgDetallesEntidad
- Actualiza formDetallesEntidad
- Usa f:setPropertyActionListener para pasar el objeto entidad seleccionado a proyectoBeneficioAntiguoSessController.proyectoEntidadSelected

◆ Interfaz visual

- La pestaña está identificada por el ícono fa fa-building y el label dinámico:

```
{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.entidades']}
```

◆ Modal relacionado

- Se abre #dlgDetallesEntidad (no incluido en el código, pero referenciado desde el botón de detalles).

Subpestaña: Rol Entidades

◆ Propósito

Esta sección muestra el **rol desempeñado por cada entidad** participante dentro del proyecto antiguo, especificando tanto la entidad como su rol asociado.

```
<!-- Pestaña Rol Entidades -->
<div id="tab-rolEntidades" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarRolEntidadesRC" name="cargarRolEntidades"
update="rolEntidadesProyecto"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarRolEntidadesProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarRolEntidadesRC"
widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">
      <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.rolEntidades']}</b>
    </h2>
    <br/>
    <h:panelGroup id="rolEntidadesProyecto">
      <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoBeneficioAntiguoSessController.selected.rolEntidadList.size() eq 0 or
proyectoBeneficioAntiguoSessController.selected.rolEntidadList == null}">
```

```

    #{msg['gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.empty']}
    </h:panelGroup>
    <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoBeneficioAntiguoSessController.selected.rolEntidadList.size() gt 0}">
        <p:dataTable
id="#{proyectoBeneficioAntiguoSessController.modulo}tableRolEntidades"
var="rolEntidad"

value="#{proyectoBeneficioAntiguoSessController.selected.rolEntidadList}"
paginator="true"
selectionMode="single"
reflow="true"
rowKey="#{rolEntidad.rolEntidadId}"
currentPageReportTemplate="Total de

registros: {totalRecords}"
paginatorTemplate="{CurrentPageReport}"
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"

widgetVar="items#{proyectoBeneficioAntiguoSessController.modulo}"
rows="10"
>
        <p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.entidad']}"
filterStyle="display:none;"
>
            <h:outputText
value="#{rolEntidad.entidadProyectoId.nombreEntidad}"/>
        </p:column>
        <p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.rol']}"
filterStyle="display:none;"
width="30%"
style="text-align: center;"
>
            <h:outputText
value="#{rolEntidad.rolId.descripcionValorDefinicion}"/>
        </p:column>
    </p:dataTable>
</h:panelGroup>
</h:panelGroup>
</div>
</div>

```

◆ Estructura y funcionamiento

- **Activación de contenido:** Al seleccionar la pestaña, se invoca el método cargarRolEntidades() definido como un <p:remoteCommand> que actualiza la sección rolEntidadesProyecto y ejecuta el método del controlador:

```

<p:remoteCommand id="cargarRolEntidadesRC" name="cargarRolEntidades"
update="rolEntidadesProyecto"

```



```
actionListener="#{ proyectoBeneficioAntiguoSessController.cargarRolEntidadesProyecto()}" />
```

- **Carga visual:** Se utiliza p:blockUI con animación de carga para el contenedor projecChildrens.
- **Mensajes vacíos:**
Si la lista rolEntidadList asociada al proyecto está vacía o nula, se muestra el mensaje:

```
{msg['gestion.beneficiosTributarios.proyectosAntiguos.rolEntidades.empty']}
```

- **Tabla de resultados:**
Cuando la lista rolEntidadList contiene registros, se presenta un p:dataTable con paginación y visualización responsive.

◆ Componentes principales (p:dataTable)

- value="#{proyectoBeneficioAntiguoSessController.selected.rolEntidadList}"
- selectionMode="single" – permite seleccionar una única fila.
- widgetVar="items#{proyectoBeneficioAntiguoSessController.modulo}" – identificador dinámico.
- rows="10" – número de registros por página.

Columnas:

3. **Entidad**
Muestra el nombre de la entidad asociada:
#{rolEntidad.entidadProyectoId.nombreEntidad}
4. **Rol**
Muestra el rol asignado a esa entidad:
#{rolEntidad.rolId.descripcionValorDefinicion}

◆ Interfaz visual

- Ícono: fa fa-building
- Etiqueta dinámica:

```
{msg['gestion.proyectosProgramas.proyectosAntiguos.tab.rolEntidades']}
```

Subpestaña: Presupuesto

◆ Propósito

Mostrar los rubros presupuestales solicitados y aprobados por el proyecto antiguo, discriminando los valores por fuente de financiación y en formato de tabla, junto con un acceso a los detalles de cada rubro.

```
<!-- Pestaña Presupuesto-->
<div id="tab-presupuesto" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarPresupuestoRC" name="cargarPresupuesto"
update="presupuestoProyecto"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarPresupuestoProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarPresupuestoRC"
widgetVar="demoBlock">
      #{msg['sb.mensajes.cargando.label']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.presupuesto']}</b>
    </h2>
    <br/>
    <h:panelGroup id="presupuestoProyecto">
      <div class="panel-group" id="accordionPresupuesto">
        <h:panelGroup id="presupuesto" styleClass="animated
fadeInLeft">
          <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoBeneficioAntiguoSessController.listRubros.size() lt 1 or
proyectoBeneficioAntiguoSessController.listRubros == null}">

            #{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.empty']}
            </h:panelGroup>
            <h:panelGroup layout="block"
rendered="#{proyectoBeneficioAntiguoSessController.listRubros.size() gt 0}">
              <p:dataTable id="Table"

value="#{proyectoBeneficioAntiguoSessController.listRubros}"

var="rubro"
sortOrder="descending"
styleClass="table"

table-striped table-bordered table-hover dataTables-entidades"

reflow="true"
rowKey="#{rubro[6]}"
paginator="true"

currentPageReportTemplate="Total de registros: {totalRecords}"

paginatorTemplate="{CurrentPageReport} {FirstPageLink} {PreviousPageLink} {PageLinks}
{NextPageLink} {LastPageLink} {RowsPerPageDropdown}"

paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}">
```

```

widgetVar="itemsRubros"

align: right;"
value="#{msg['sb.component.tableHeader.buscar']}" />
id="globalFilter" placeholder="#{msg['sb.component.tableHeader.buscar.placeholder']}"
onkeyup="PF('itemsRubros').filter()" style="width:200px"/>
filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
center"
ID">
value="#{rubro[9]}" />
filterMatchMode="contains"
filterStyle="display:none;"
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.rubro.header'
]]}">
value="#{rubro[0]}" />
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.total']}"
style="text-align: right; text-transform: uppercase;" />
filterMatchMode="contains"
filterStyle="display:none;"
right;"
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.monInstitucio
nSol.header']}">
value="#{rubro[1]}">
currencySymbol="$" type="currency" locale="es"/>
value="#{proyectoBeneficioAntiguoSessController.totalMonInstitucionSol}">
<f:convertNumber currencySymbol="$" type="currency" locale="es"/>

```

```

</f:facet>
</p:column>
<p:column filterBy="#{rubro[2]}"

filterMatchMode="contains"

filterStyle="display:none;"

right;"

style="text-align:

headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEspecieSol
.header']}">

<h:outputText

value="#{rubro[2]}">

<f:convertNumber

currencySymbol="$" type="currency" locale="es"/>

</h:outputText>
<f:facet name="footer">
<h:outputText

value="#{proyectoBeneficioAntiguoSessController.totalConEspecieSol}">

<f:convertNumber currencySymbol="$" type="currency" locale="es"/>
</h:outputText>
</f:facet>
</p:column>
<p:column filterBy="#{rubro[3]}"

filterMatchMode="contains"

filterStyle="display:none;"

right;"

style="text-align:

headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEfectivoSo
l.header']}">

<h:outputText

value="#{rubro[3]}">

<f:convertNumber

currencySymbol="$" type="currency" locale="es"/>

</h:outputText>
<f:facet name="footer">
<h:outputText

value="#{proyectoBeneficioAntiguoSessController.totalConEfectivoSol}">

<f:convertNumber currencySymbol="$" type="currency" locale="es"/>
</h:outputText>
</f:facet>
</p:column>
<p:column filterBy="#{rubro[4]}"

filterMatchMode="contains"

filterStyle="display:none;"

right;"

style="text-align:

headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.totalSol.head
er']}">

<h:outputText

value="#{rubro[4]}">

<f:convertNumber

currencySymbol="$" type="currency" locale="es"/>

</h:outputText>
<f:facet name="footer">

```

```

value="#{proyectoBeneficioAntiguoSessController.totalSol}">
    <f:convertNumber currencySymbol="$" type="currency" locale="es"/>
    </h:outputText>
    </f:facet>
</p:column>
<p:column filterBy="#{rubro[5]}"
filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
right;"
style="text-align:
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.monInstitucionAprob.header']}">
    <h:outputText
value="#{rubro[5]}">
    <f:convertNumber
currencySymbol="$" type="currency" locale="es"/>
    </h:outputText>
</p:column>
<p:column filterBy="#{rubro[6]}"
filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
right;"
style="text-align:
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEspecieAprob.header']}">
    <h:outputText
value="#{rubro[6]}">
    <f:convertNumber
currencySymbol="$" type="currency" locale="es"/>
    </h:outputText>
</p:column>
<p:column filterBy="#{rubro[7]}"
filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
right;"
style="text-align:
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.conEfectivoAprob.header']}">
    <h:outputText
value="#{rubro[7]}">
    <f:convertNumber
currencySymbol="$" type="currency" locale="es"/>
    </h:outputText>
</p:column>
<p:column filterBy="#{rubro[8]}"
filterMatchMode="contains" rendered="false"
filterStyle="display:none;"
right;"
style="text-align:
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.presupuesto.totalAprob.
ader']}">

```

```

value="#{rubro[8]}">
currencySymbol="$" type="currency" locale="es"/>
<h:outputText
    <f:convertNumber
    </h:outputText>
</p:column>
<p:column width="6%">
    <p:commandLink styleClass=" btn
btn-xs btn-primary sbTooltip"
style="color: #fff;"
title="#{msg['sb.component.boton.detalles']}"
actionListener="#{proyectoBeneficioAntiguoConvController.cargarDetallesRubrosProyecto(rubro[
9])}"
oncomplete="openDialogRubros();"
update="formDialogRubros"
    <i class="fa fa-
money"></i>
    <f:setPropertyActionListener
target="#{proyectoBeneficioAntiguoSessController.descripcionRubro}" value="#{rubro[0]}" />
    </p:commandLink>
    </p:column>
</p:dataTable>
</h:panelGroup>
</h:panelGroup>
</div>
</h:panelGroup>
</div>
</div>

```

◆ Funcionamiento general

• Carga del contenido:

Al activarse la pestaña, se ejecuta un `<p:remoteCommand>` que:

- Llama al método `cargarPresupuestoProyecto()` del controlador `proyectoBeneficioAntiguoSessController`.
- Actualiza el panel `presupuestoProyecto`.

```

<p:remoteCommand id="cargarPresupuestoRC" name="cargarPresupuesto"
update="presupuestoProyecto"

```

```

actionListener="#{proyectoAntiguoViewController.cargarPresupuestoProyecto()}" />

```

- **Bloqueo de interfaz mientras carga:**

Se utiliza p:blockUI para bloquear visualmente projecChildrens con una barra de progreso mientras se carga la información.

♦ **Vista de resultados**

- **Condición vacía:**

Si listRubros está vacía o nula, se muestra un mensaje:

```
{msg['gestion.beneficiosProgramas.proyectosAntiguos.presupuesto.empty']}
```

- **Tabla principal:**

Si listRubros contiene elementos, se muestra un <p:dataTable> con paginación, ordenamiento y filtro global. Cada rubro es una fila representada por un arreglo de objetos (rubro[n]).

♦ **Columnas mostradas (visibles)**

Índice	Campo	Descripción
rubro[0]	Rubro	Nombre o descripción del rubro.
rubro[1]	Monto institución solicitante	Valor en efectivo solicitado por la institución.
rubro[2]	Contribución en especie solicitada	Aportes no monetarios solicitados.
rubro[3]	Contribución en efectivo solicitada	Aportes monetarios solicitados por terceros.
rubro[4]	Total solicitado	Suma total solicitada del rubro.

Todas estas columnas están configuradas con formato de moneda (f:convertNumber con locale="es" y currencySymbol="\$").

Además, cada una incluye su respectivo **footer** que muestra totales provenientes del controlador, por ejemplo:

```
<f:facet name="footer">
```

```
<h:outputText value="#{proyectoBeneficioAntiguoSessController.totalSol}">
  <f:convertNumber currencySymbol="$" type="currency" locale="es"/>
</h:outputText>
</f:facet>
```

◆ Columnas ocultas (rendered="false")

Estas columnas están presentes en el modelo pero no se muestran directamente:

- rubro[5] – Monto institución aprobado
- rubro[6] – Contribución especie aprobada
- rubro[7] – Contribución efectivo aprobada
- rubro[8] – Total aprobado
- rubro[9] – Rubro ID

◆ Detalles por rubro

En la última columna, cada fila incluye un botón tipo commandLink que:

- Llama al método cargarDetallesRubrosProyecto(rubro[9]) del controlador proyectoBeneficioAntiguoConvController.
- Abre un diálogo (openDialogRubros()) para mostrar el detalle del rubro seleccionado.
- Actualiza el formulario formDialogRubros.
- Establece el nombre del rubro en descripcionRubro.

```
<p:commandLink ...
actionListener="#{proyectoBeneficioAntiguoConvController.cargarDetallesRubrosProyecto(rubro[
9])}"
oncomplete="openDialogRubros();"
update="formDialogRubros">
  <i class="fa fa-money"></i>
  <f:setPropertyActionListener
target="#{proyectoBeneficioAntiguoSessController.descripcionRubro}" value="#{rubro[0]}" />
</p:commandLink>
```


◆ Filtro global

El encabezado de la tabla tiene un buscador global que filtra todas las columnas simultáneamente:

```
<p:inputText id="globalFilter" onkeyup="PF('itemsRubros').filter()" />
```

Subpestaña: Descripciones

◆ Propósito

Presentar las descripciones textuales configuradas para el proyecto antiguo, en el orden y clasificación establecidos, permitiendo visualizarlas a través de un diálogo.

```
<!-- Pestaña Descripciones -->
<div id="tab-descripciones" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarDescpcionesRC" name="cargarDescpciones"
update="descripciones"
actionListener="#{proyectoBeneficioAntiguoConvController.cargarDescpcionesProyecto}" />
    <p:blockUI block="projecChildrens" trigger="cargarDescpcionesRC"
widgetVar="demoBlock">
      #{msg['gestion.beneficiosTributarios.proyectosAntiguos.cargando']}<br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.descripciones']}</b>
    </h2>
    <br/>
    <h:panelGroup id="descripciones" styleClass="animated fadeInLeft">
      <h:panelGroup layout="block" id="empty" styleClass="well"
rendered="#{proyectoBeneficioAntiguoSessController.listaDescripcion.size() lt 1 or
proyectoBeneficioAntiguoSessController.listaDescripcion == null}">

        #{msg['gestion.beneficiosTributarios.proyectosAntiguos.descripcion.empty']}
        </h:panelGroup>
        <h:panelGroup layout="block"
rendered="#{proyectoBeneficioAntiguoSessController.listaDescripcion.size() gt 0}">
          <h:dataTable
value="#{proyectoBeneficioAntiguoSessController.listaDescripcion}"
var="descripcion"

id="#{proyectoBeneficioAntiguoSessController.modulo}Table"
styleClass="table table-striped table-
bordered table-hover"

          >
            <h:column>
              <!-- column header -->
              <f:facet
name="header">#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tabla.orden']}</f:facet
              >

              <!-- row record -->
```

```

        #{descripcion.configuracionDescripcionId.ordenPresentacion}
        </h:column>
        <h:column>
            <!-- column header -->
            <f:facet
name="header">#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tabla.descripcion']}</f
:facet>

            <!-- row record -->

        #{descripcion.configuracionDescripcionId.tipoDescripcionId.descripcionValorDefinicion}
        </h:column>

        <h:column>
            <!-- column header -->
            <f:facet
name="header">#{msg['sb.component.boton.opciones']}</f:facet>
            <!-- row record -->
            <p:commandLink
value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.btn.ver']}"

            oncomplete="openDialogDescripcion();"

            update="formDialogDescripcion"

            styleClass="btn
            btn-sm btn-primary"

            >
            <f:setPropertyActionListener
value="#{descripcion}" target="#{proyectoBeneficioAntiguoSessController.descripcion}"/>
            </p:commandLink>
        </h:column>

    </h:dataTable>
    <script>
        $(document).ready(function () {

            initTablaNobar("#{request.contextPath}/resources/js/plugins/datatables-
plugins/i18n/#{i18nBean.currentLocale}.lang", ".dataTables-descripciones");
        });

    </script>
    </h:panelGroup>
    </h:panelGroup>
</div>
</div>

```

◆ Funcionamiento general

- **Carga de datos:**

Se ejecuta el siguiente <p:remoteCommand> al activarse la pestaña, que:

- Llama al método cargarDescripcionesProyecto() del proyectoBeneficioAntiguoConvController.

- Actualiza el panel descripciones.

```
<p:remoteCommand id="cargarDescipcionesRC" name="cargarDescipciones"
    update="descripciones"
    actionListener="#{proyectoBeneficioAntiguoConvController.cargarDescipcionesProyecto}" />
```

- **Bloqueo de interfaz:**

El componente p:blockUI bloquea la sección projecChildrens mientras se ejecuta la carga de las descripciones.

◆ Visualización de contenido

- **Panel sin datos:**

Si listaDescripciones está vacía o es null, se muestra un mensaje indicando que no hay descripciones registradas:

```
{msg['gestion.beneficiosTributarios.proyectosAntiguos.descripcion.empty']}
```

- **Tabla de descripciones:**

Si existen registros en listaDescripciones, se despliega una tabla HTML (<h:dataTable>) con las siguientes columnas:

◆ Columnas de la tabla

Columna	Contenido
Orden	descripcion.configuracionDescripcionId.ordenPresentacion Número que indica la posición que debe ocupar la descripción.
Descripción	descripcion.configuracionDescripcionId.tipoDescripcionId.descripcionValorDefinicion Texto o tipo de contenido configurado para la descripción del proyecto.
Opciones	Botón "Ver", que abre un diálogo con el contenido completo de la descripción.

◆ Acceso al detalle

En la columna **Opciones**, cada fila tiene un botón commandLink que:

- Asigna la entidad descripcion seleccionada a la propiedad descripcion del proyectoBeneficioAntiguoSessController.
- Ejecuta oncomplete="openDialogDescripcion();" para abrir el diálogo modal con el detalle.
- Actualiza el formulario formDialogDescripcion.

```
<p:commandLink value="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.btn.ver']}"
oncomplete="openDialogDescripcion();"
update="formDialogDescripcion"
styleClass="btn btn-sm btn-primary">
  <f:setPropertyActionListener value="#{descripcion}"
target="#{proyectoBeneficioAntiguoSessController.descripcion}"/>
</p:commandLink>
```

◆ Componente adicional: inicialización JS

Se carga dinámicamente el idioma para DataTables usando jQuery (aunque la tabla no es un <p:dataTable> de PrimeFaces, se espera que se le haya aplicado una clase dataTables-descripciones):

```
$(document).ready(function () {
  initTablaNobar("#{request.contextPath}/resources/js/plugins/datatables-
plugins/i18n/#{i18nBean.currentLocale}.lang", ".dataTables-descripciones");
});
```

Este script una función personalizada initTablaNobar(...), que aplica funcionalidades de búsqueda, ordenamiento y traducción.

Subpestaña: Impactos

◆ Propósito

Mostrar la lista de **impactos esperados o generados** del proyecto antiguo, agrupados por tipo y año. Permite conocer los efectos previstos según la planificación del proyecto.

```
<!-- Pestaña Impactos -->
<div id="tab-impactos" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarImpactosRC" name="cargarImpactos"
update="impactosProyecto"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarImpactosProyecto()}" />
```

```

        <p:blockUI block="projecChildrens" trigger="cargarImpactosRC"
        widgetVar="demoBlock">
            #{msg['sb.mensajes.cargando.label']}<br/>
            <div class="progress progress-striped active m-b-sm">
                <div style="width: 100%;" class="progress-bar"></div>
            </div>
        </p:blockUI>
        <h2 style="text-align: center;">

        <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.impactos']}</b>
        </h2>
        <br/>
        <h:panelGroup id="impactosProyecto">
            <h:panelGroup layout="block" styleClass="well"
            rendered="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaImpactos.size() eq
            0 or proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaImpactos == null}">

                #{msg['gestion.beneficiosTributarios.proyectosAntiguos.impactos.empty']}
                </h:panelGroup>
                <h:panelGroup styleClass="animated fadeInLeft" layout="block"
            rendered="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaImpactos.size() gt
            0}">

                    <h:dataTable
            value="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaImpactos}"
                    var="impacto"

            id="#{proyectoBeneficioAntiguoSessController.modulo}tableImpactos"
                    styleClass="table table-striped table-
            bordered table-hover dataTables"

                    >
                        <h:column>
                            <!-- column header -->
                            <f:facet
            name="header">#{msg['gestion.beneficiosTributarios.proyectosAntiguos.impacto.tipo']}</f:face
            t>

                                <!-- row record -->
                                #{impacto.configuracionImpactoId.descripcion}
                            </h:column>
                            <h:column>
                                <!-- column header -->
                                <f:facet
            name="header">#{msg['gestion.beneficiosTributarios.proyectosAntiguos.impacto.descripcion']}</f:facet>

                                    <!-- row record -->
                                    #{impacto.descripcion != null ?
            mpacto.descripcion : impacto.impactoEsperado}
                                </h:column>
                                <h:column>
                                    <!-- column header -->
                                    <f:facet
            name="header">#{msg['gestion.beneficiosTributarios.proyectosAntiguos.impacto.año']}</f:facet>
                                    >

                                        <!-- row record -->
                                        #{impacto.anoEsperadoImpacto}
                                    </h:column>

                                </h:dataTable>
                            <script>
                                $(document).ready(function () {

```

```
initTablaNobar("#{request.contextPath}/resources/js/plugins/datatables-  
plugins/i18n/#{i18nBean.currentLocale}.lang", ".dataTables-impactos");  
});  
  
    </script>  
    </h:panelGroup>  
</h:panelGroup>  
</div>  
</div>
```

◆ **Funcionamiento general**

- **Carga de datos:**

Utiliza un `<p:remoteCommand>` que se ejecuta para inicializar la pestaña y cargar los impactos desde el backend:

```
<p:remoteCommand id="cargarImpactosRC" name="cargarImpactos"
```

```
update="impactosProyecto"
```

```
actionListener="#{proyectoBeneficioAntiguoSessController.cargarImpactosProyecto()}"
```

```
/>
```

- **Bloqueo visual durante la carga:**

Se emplea un `p:blockUI` que bloquea visualmente el componente `projecChildrens` mientras se obtienen los datos:

```
<p:blockUI block="projecChildrens" trigger="cargarImpactosRC" widgetVar="demoBlock">
```

◆ **Visualización del contenido**

- **Panel sin impactos:**

Si `listalImpactos` es null o está vacía (`size() eq 0`), se muestra un mensaje tipo well que indica que no hay impactos disponibles:

```
{msg['gestion.beneficiosTributarios.proyectosAntiguos.impactos.empty']}
```

- **Tabla de impactos:**

Si existen registros, se muestra una tabla `<h:dataTable>` con clase `dataTables`, lo cual sugiere que luego es transformada en tabla dinámica vía jQuery.

◆ **Columnas de la tabla**

Columna	Contenido
Tipo de Impacto	impacto.configuracionImpactoId.descripcion Define la categoría o naturaleza del impacto (económico, ambiental, social, etc.).
Descripción	Muestra impacto.descripcion, y si este campo está nulo, muestra impacto.impactoEsperado. Esto permite cubrir proyectos antiguos que solo registraban el campo impactoEsperado.
Año Esperado	impacto.anoEsperadoImpacto Corresponde al año en el cual se proyecta que se manifieste el impacto.

◆ **Componente adicional: inicialización DataTables**

Al igual que otras pestañas, el siguiente script inicializa el DataTable usando configuración de idioma localizada:

```
$(document).ready(function () {  
    initTablaNobar("#{request.contextPath}/resources/js/plugins/datatables-  
plugins/i18n/#{i18nBean.currentLocale}.lang", ".dataTables-impactos");  
});
```

Se infiere que:

- La clase dataTables-impactos debe aplicarse manualmente al componente.
- La tabla se convierte dinámicamente en una tabla interactiva con búsqueda, orden y paginación.

Subpestaña: Coberturas

◆ **Propósito**

Mostrar las **coberturas geográficas** del proyecto antiguo, indicando las **regiones** beneficiadas y su respectivo **porcentaje de impacto** sobre el total del proyecto.

```
<!-- Pestaña Coberturas -->
<div id="tab-coberturas" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarCoberturasRC" name="cargarCoberturas"
update="coberturasProyecto"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarCoberturasProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarCoberturasRC"
widgetVar="demoBlock">
      <#{msg['sb.mensajes.cargando.label']}><br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.coberturas']}</b>
    </h2>
    <br/>
    <h:panelGroup id="coberturasProyecto">
      <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoBeneficioAntiguoSessController.listCoberturas.size() eq 0 or
proyectoBeneficioAntiguoSessController.listCoberturas == null}">

      <#{msg['gestion.beneficiosTributarios.proyectosAntiguos.coberturas.empty']}>
      </h:panelGroup>
      <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoBeneficioAntiguoSessController.listCoberturas.size() gt 0}">
        <ul class="sortable-list connectList agile-list" id="todo">
          <ui:repeat var="cobertura">

            value="#{proyectoBeneficioAntiguoSessController.listCoberturas}">
              <li class="success-element"
id="cobertura#{cobertura.regionImpactoId}">
                <b>#{cobertura.porcentaje}%</b>
                <div class="agile-detail">
                  <i class="fa fa-clock-o"></i>
                  <h:outputText
value="#{cobertura.fechaCreacion}">
                    <f:convertDateTime
locale="es" pattern="dd.MM.yyyy"/>
                  </h:outputText>
                </div>
              </li>
            </ui:repeat>
          </ul>
          <script>
            <#{function () {
              if ($('#todo').height() > 400) {
                $('#todo').slimScroll({
                  height: '400px'
                });
              }
            }}>
          </script>
        </h:panelGroup>
      </h:panelGroup>
    </div>
  </div>
</div>
```


◆ Carga de datos

- Se utiliza un `<p:remoteCommand>` para invocar la acción en el servidor:

```
<p:remoteCommand id="cargarCoberturasRC"
    name="cargarCoberturas"
    update="coberturasProyecto"

    actionListener="#{proyectoBeneficioAntiguoSessController.cargarCoberturasProyecto()}" />
```

- Mientras se ejecuta la acción, se activa un `<p:blockUI>` sobre el contenedor `projecChildrens`, que muestra una barra de carga:

```
<p:blockUI block="projecChildrens" trigger="cargarCoberturasRC" widgetVar="demoBlock">
```

◆ Visualización del contenido

- El contenido principal se ubica en el componente `<h:panelGroup id="coberturasProyecto">`.
- Si `listCoberturas` está vacío o nulo, se muestra un mensaje de tipo `well` que indica que no existen coberturas registradas:

```
rendered="#{proyectoBeneficioAntiguoSessController.listCoberturas.size() eq 0 or
proyectoBeneficioAntiguoSessController.listCoberturas == null}"
```

- Si hay coberturas, se renderiza una **lista `<ul>` de elementos `<li>`**:

```
<ui:repeat var="cobertura" value="#{proyectoBeneficioAntiguoSessController.listCoberturas}">
```

◆ Estructura del elemento de cobertura (`<li>`)

Cada entrada representa una región de impacto con la siguiente información:

Elemento visual	Contenido
 Nombre de la región	<code>#{cobertura.regionId.nombreRegion}</code>

Elemento visual	Contenido
 Porcentaje de cobertura	<code>#{cobertura.porcentaje}%</code>
 Fecha de creación	<code>#{cobertura.fechaCreacion}</code> formateada con dd.MM.yyyy

```
<li class="success-element" id="cobertura#{cobertura.regionImpactoId}">
  #{cobertura.regionId.nombreRegion} <b>#{cobertura.porcentaje}%</b>
  <div class="agile-detail">
    <i class="fa fa-clock-o"></i>
    <h:outputText value="#{cobertura.fechaCreacion}">
      <f:convertDateTime locale="es" pattern="dd.MM.yyyy"/>
    </h:outputText>
  </div>
</li>
```

◆ Comportamiento dinámico (scroll)

- Si la altura del listado excede los **400px**, se activa un scroll vertical mediante la librería **jQuery SlimScroll**:

```
$(function () {
  if ($('#todo').height() > 400) {
    $('#todo').slimScroll({
      height: '400px'
    });
  }
});
```

Esto garantiza una buena experiencia visual sin alargar innecesariamente el panel.

Subpestaña: Cronograma de Actividades

◆ Propósito

Presentar de forma tabular el **cronograma de actividades** asociadas al proyecto antiguo, detallando el número de la actividad, su descripción, fechas de inicio y fin, y el tipo de ejecución.

```
<!-- Pestaña Cronograma de Actividades -->
<div id="tab-actividades" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarCronogramaRC" name="cargarCronograma"
      update="cronogramaProyecto"
      actionListener="#{proyectoBeneficioAntiguoSessController.cargarCronogramaProyecto()}" />
    <p:blockUI block="projecChildrens" trigger="cargarCronogramaRC"
      widgetVar="demoBlock">
```

```

        #{msg['sb.mensajes.cargando.label']}<br/>
        <div class="progress progress-striped active m-b-sm">
            <div style="width: 100%;" class="progress-bar"></div>
        </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.cronograma']}</b>
    </h2>
    <br/>
    <h:panelGroup id="cronogramaProyecto">
        <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaCronograma.size()
eq 0 or proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaCronograma == null}">

        #{msg['gestion.beneficiosTributarios.proyectosAntiguos.cronograma.empty']}
        </h:panelGroup>
        <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaCronograma.size()
gt 0}">
            <p:dataTable
id="#{proyectoBeneficioAntiguoSessController.modulo}tableCronogramas"
                var="actividadCronograma"

value="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaCronograma}"
                paginator="true"
                selectionMode="single"
                reflow="true"

rowKey="#{actividadCronograma.cronogramaProyectoId}"
                currentPageReportTemplate="Total de
registros: {totalRecords}"
                paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
                paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"
widgetVar="items#{proyectoBeneficioAntiguoSessController.modulo}"
                rows="10"
            >
                <p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.cronograma.nActividad']}"
                    filterStyle="display:none;"
                    width="10%"
                    style="text-align: center;"
                >
                    <h:outputText
value="#{actividadCronograma.numeroActividad}"/>
                </p:column>
                <p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.cronograma.actividad']}"
                    filterStyle="display:none;"
                >
                    <h:outputText
value="#{actividadCronograma.actividad}"/>
                </p:column>
                <p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.cronograma.inicio']}"
                    filterStyle="display:none;"

```

```

width="10%"
style="text-align: center;"
>
<h:outputText
rendered="#{actividadCronograma.fechaInicial != null}"
value="#{actividadCronograma.fechaInicial}"
>
<f:convertDateTime pattern="dd/MM/yyyy"
/>
</h:outputText>
<h:outputText
rendered="#{actividadCronograma.periodoInicial != null}"
value="#{actividadCronograma.periodoInicial}"
>
</h:outputText>
</p:column>
<p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.cronograma.fin']}"
filterStyle="display:none;"
width="10%"
style="text-align: center;"
>
<h:outputText
rendered="#{actividadCronograma.fechaFinal != null}"
value="#{actividadCronograma.fechaFinal}">
<f:convertDateTime pattern="dd/MM/yyyy"
/>
</h:outputText>
<h:outputText
rendered="#{actividadCronograma.periodoFinal != null}"
value="#{actividadCronograma.periodoFinal}"
>
</h:outputText>
</p:column>
<p:column
headerText="#{msg['gestion.beneficiosTributarios.proyectosAntiguos.cronograma.ejecucion']}"
filterStyle="display:none;"
width="10%"
style="text-align: center;"
>
<h:outputText
value="#{actividadCronograma.cronograma == 'D' ? msg['listaDuracion.D'] :
actividadCronograma.cronograma == 'S' ? msg['listaDuracion.S'] : msg['listaDuracion.M']}">
</p:column>
</p:dataTable>
</h:panelGroup>
</h:panelGroup>
</div>
</div>

```

◆ **Carga de datos**

Se realiza mediante un `<p:remoteCommand>` que invoca un método del ViewController:

```
<p:remoteCommand id="cargarCronogramaRC"
    name="cargarCronograma"
    update="cronogramaProyecto"

    actionListener="#{proyectoAntiguoViewController.cargarCronogramaProyecto()}" />
```

Este método se acompaña de un `<p:blockUI>` que bloquea la sección `projecChildrens` mientras se cargan los datos:

```
<p:blockUI block="projecChildrens" trigger="cargarCronogramaRC" widgetVar="demoBlock">
```

◆ Visualización

El contenedor principal es:

```
<h:panelGroup id="cronogramaProyecto">
```

- Si `listaCronograma` está vacío o nulo, se muestra un mensaje en un well:

```
rendered="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaCronograma.size()
eq 0 or proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaCronograma == null}"
```

- Si existen actividades, se renderiza un `<p:dataTable>` con paginación y filas dinámicas.

◆ Estructura de la tabla

Cada fila representa una actividad cronológica del proyecto. Los campos visualizados son:

Campo	Descripción	Fuente
 N. Actividad	Número secuencial de la actividad	<code>actividadCronograma.numeroActividad</code>
 Actividad	Nombre o descripción de la actividad	<code>actividadCronograma.actividad</code>
 Inicio	Fecha de inicio o período inicial	<code>fechaInicial o periodoInicial</code>

Campo	Descripción	Fuente
 Fin	Fecha de finalización o período final	fechaFinal o periodoFinal
 Ejecución	Tipo de duración: diaria, semanal o mensual	actividadCronograma.cronograma

Formato de fechas

Las fechas (fechaInicial y fechaFinal) se muestran en formato dd/MM/yyyy:

```
<f:convertDateTime pattern="dd/MM/yyyy" />
```

Tipo de ejecución

El tipo de cronograma se interpreta según el código:

- D → Diario (listaDuracion.D)
- S → Semanal (listaDuracion.S)
- M → Mensual (listaDuracion.M)

◆ **Funcionalidades adicionales de la tabla**

-  Paginación (paginator="true")
-  Reflujo para pantallas pequeñas (reflow="true")
-  Desactivación de filtros por columna (filterStyle="display:none;")
-  Selección de fila única (selectionMode="single")
-  Navegación entre páginas (paginatorTemplate)

Subpestaña: Tipo Personal

◆ **Propósito**

Visualizar de forma tabular la distribución del **personal asignado** al proyecto antiguo, detallando el tipo de personal y el número de personas involucradas en el proyecto.

```
<!-- Pestaña Tipo Personal -->
<div id="tab-tipoPersonal" class="tab-pane" >
  <div class="panel-body">
    <p:remoteCommand id="cargarTipoPersonalRC" name="cargarTipoPersonal"
update="tipoPersonalProyecto"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarTipoPersonal()}" />
    <p:blockUI block="projecChildrens" trigger="cargarTipoPersonalRC"
widgetVar="demoBlock">
      <#{msg['sb.mensajes.cargando.label']}><br/>
      <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
      </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.tipoPersonal']}</b>
    </h2>
    <br/>
    <h:panelGroup id="tipoPersonalProyecto" >
      <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoBeneficioAntiguoSessController.listaTipoPersonal.size() eq 0 or
proyectoBeneficioAntiguoSessController.listaTipoPersonal == null}">

      <#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.empty']}>
      </h:panelGroup>
      <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoBeneficioAntiguoSessController.listaTipoPersonal.size() gt 0}">

        <p:dataTable
id="#{proyectoBeneficioAntiguoSessController.modulo}tableTipoPersonal"
var="tipoPersonal"

value="#{proyectoBeneficioAntiguoSessController.listaTipoPersonal}"
paginator="true"
selectionMode="single"
reflow="true"

rowKey="#{tipoPersonal.personalProyectoId}"

registros: {totalRecords}"

currentPageReportTemplate="Total de
paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"

paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"

rows="10"
styleClass="table table-striped table-
bordered table-hover"

        >
        <p:column>
          <!-- column header -->
          <f:facet
name="header">#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.tipoPerson
al']}</f:facet>

          <!-- row record -->
```

```

        #{tipoPersonal.tipoPersonalId.descripcionValorDefinicion}
        </p:column>
        <p:column width="30%" style="text-align: center;">
            <!-- column header -->
            <f:facet
name="header">#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tipoPersonal.numeroPers
onas']}</f:facet>
            <!-- row record -->
            #{tipoPersonal.numeroPersonas}
        </p:column>
    </p:dataTable>
</h:panelGroup>
</h:panelGroup>
</div>
</div>

```

◆ Carga de datos

La carga se ejecuta mediante un `<p:remoteCommand>` que invoca un método del ViewController:

```

<p:remoteCommand id="cargarTipoPersonalRC"
    name="cargarTipoPersonal"
    update="tipoPersonalProyecto"

    actionListener="#{proyectoBeneficioAntiguoSessController.cargarTipoPersonal()}" />

```

Durante esta operación se bloquea visualmente el contenedor `projecChildrens` mediante un `<p:blockUI>`:

```

<p:blockUI block="projecChildrens" trigger="cargarTipoPersonalRC" widgetVar="demoBlock">

```

◆ Visualización

El contenido se agrupa dentro de:

```

<h:panelGroup id="tipoPersonalProyecto"
    rendered="#{proyectoBeneficioAntiguoSessController.proyectoAntiguo}">

```

- Si la lista está vacía o nula (`listaTipoPersonal`), se muestra un mensaje de vacío con estilo de resaltado:

```

<h:panelGroup layout="block" styleClass="well"
    rendered="#{proyectoBeneficioAntiguoSessController.listaTipoPersonal.size() eq
0 or proyectoBeneficioAntiguoSessController.listaTipoPersonal == null}">

```


- Si contiene datos, se renderiza un <p:dataTable> con paginación y formato responsivo.

♦ Estructura de la tabla

La tabla muestra dos columnas por registro de personal:

Campo	Descripción	Fuente
 Tipo de Personal	Rol o categoría del personal asignado	tipoPersonal.tipoPersonalId.descripcionValorDefinicion
 Número de Personas	Cantidad de personas asociadas al tipo de personal	tipoPersonal.numeroPersonas

♦ Características adicionales de la tabla

-  **Paginación** activada (paginator="true")
-  **Selección de fila única** (selectionMode="single")
-  **Diseño responsivo** (reflow="true")
-  **Mensaje personalizado** para tabla vacía (emptyMessage)
-  **Estilos de tabla Bootstrap** (table-striped, table-bordered, table-hover)

Subpestaña: Personal

♦ Propósito

Mostrar el listado de personas asignadas a un proyecto antiguo, incluyendo su rol, nombre completo, número de identificación y un acceso a los detalles individuales de cada participante.

```
<!-- Pestaña Personal -->
<div id="tab-personal" class="tab-pane" >
  <div class="panel-body">
```

```

        <p:remoteCommand id="cargarPersonasRC" name="cargarPersonas"
update="personasProyecto"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarPersonasProyecto()}" />
        <p:blockUI block="projecChildrens" trigger="cargarPersonasRC"
widgetVar="demoBlock">
            #{msg['sb.mensajes.cargando.label']}<br/>
            <div class="progress progress-striped active m-b-sm">
                <div style="width: 100%;" class="progress-bar"></div>
            </div>
        </p:blockUI>
        <h2 style="text-align: center">

        <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.personal']}</b>
        </h2>
        <br/>
        <h:panelGroup id="personasProyecto">
            <h:panelGroup layout="block" styleClass="well"
rendered="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaPersonal.size() eq
0 or proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaPersonal == null}">

            #{msg['gestion.beneficiosTributarios.proyectosAntiguos.personal.empty']}
            </h:panelGroup>
            <h:panelGroup styleClass="animated fadeInLeft" layout="block"
rendered="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaPersonal.size() gt
0}">

                <p:dataTable
id="#{proyectoBeneficioAntiguoSessController.modulo}tablePersonal"
var="persona"

value="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaPersonal}"
                    paginator="true"
                    selectionMode="single"
                    reflow="true"
                    rowKey="#{persona.personaProyectoId}"
                    currentPageReportTemplate="Total de

registros: {totalRecords}"
                    paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
                    paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"
                    rows="10"
                    styleClass="table table-striped table-
bordered table-hover"
                    rowIndexVar="rowIndex"
                >
                    <p:column headerText="#"
width="5%"
style="text-align: center;"
                >
                        #{rowIndex+1}
                    </p:column>
                    <p:column>
                        <!-- column header -->
                        <f:facet
name="header">#{msg['gestion.beneficiosTributarios.proyectosAntiguos.personal.persona']}</f:
facet>

                        <!-- row record -->
                        #{persona.nombreCompleto}
                    </p:column>

```

```

        <p:column width="30%" style="text-align: center;">
            <!-- column header -->
            <f:facet
name="header">#{msg['gestion.beneficiosTributarios.proyectosAntiguos.personal.tipo']}

```

◆ Carga de datos

Se usa un `<p:remoteCommand>` para invocar el método de carga desde el ViewController:

```
<p:remoteCommand id="cargarPersonasRC"
    name="cargarPersonas"
    update="personasProyecto"

    actionListener="#{proyectoBeneficioAntiguoSessController.cargarPersonasProyecto()}" />
```

Durante esta carga se bloquea visualmente la interfaz:

```
<p:blockUI block="projecChildrens" trigger="cargarPersonasRC" widgetVar="demoBlock">
```

◆ Visualización

Dentro de:

```
<h:panelGroup id="personasProyecto"
    rendered="#{proyectoBeneficioAntiguoSessController.proyectoAntiguo}">
```

- Se muestra un mensaje de lista vacía si no hay personas:

```
<h:panelGroup layout="block" styleClass="well"

    rendered="#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaPersonal.size() eq
    0 or proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaPersonal == null}">
```

- Si hay datos, se renderiza una tabla `<p:dataTable>` paginada, responsiva y con estilo Bootstrap.

◆ Estructura de la tabla

Cada fila representa a una persona asignada al proyecto, con las siguientes columnas:

#	Campo	Descripción	Fuente
1	# (Índice)	Número de fila incremental	<code>#{rowIndex+1}</code>

#	Campo	Descripción	Fuente
2	Nombre completo	Nombre completo de la persona asignada	<code>#{persona.nombreCompleto}</code>
3	Tipo / Rol	Rol del personal (usa dos fuentes posibles o invoca un método de carga)	<code>persona.rolPersonalId.descripcionValorDefinicion</code> <code>persona.registroRolId.descripcionValorDefinicion</code> <code>proyectoBeneficioAntiguoSessController.cargarRolesPersona()</code>
4	Identificación	Número de documento o cédula	<code>#{persona.numeroIdentificacion}</code>
5	Detalles	Botón que abre un diálogo para ver detalles de la persona	<code>p:commandLink</code> con <code>actionListener="#{proyectoBeneficioAntiguoConvController.cargarInfoPersonal(persona)}"</code>

◆ **Funcionalidad adicional**

-  **Selección de fila única** (`selectionMode="single"`)
-  **Paginación activa** (`paginator="true"`)
-  **Responsivo** (`reflow="true"`)
-  **Invocación de diálogo de detalles personales**

-  **JS personalizado para inicializar DataTables internacionalizados:**

```
initTablaNobar("#{request.contextPath}/resources/js/plugins/datatables-  
plugins/i18n/#{i18nBean.currentLocale}.lang", ".dataTables-personal");
```

Subpestaña: Marco Lógico

◆ Propósito

Visualizar la estructura del marco lógico del proyecto antiguo, mostrando:

- Objetivo general del proyecto
- Meta y fuente de verificación
- Objetivos específicos
- Actividades por objetivo
- Resultados por actividad
- Indicadores por resultado

```
<!-- Pestaña Marco Logico -->  
<div id="tab-marco" class="tab-pane" >  
  <div class="panel-body">  
    <p:remoteCommand id="cargarMarcoRC" name="cargarMarcoLogico"  
update="marcoLogicoProyecto"  
actionListener="#{proyectoBeneficioAntiguoSessController.cargarMarcoLogico()}" />  
    <p:blockUI block="projecChildrens" trigger="cargarMarcoRC"  
widgetVar="demoBlock">  
      #{msg['sb.mensajes.cargando.label']}<br/>  
      <div class="progress progress-striped active m-b-sm">  
        <div style="width: 100%;" class="progress-bar"></div>  
      </div>  
    </p:blockUI>  
    <h2 style="text-align: center">  
  
    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.marcoLogico']}</b>  
    </h2>  
    <br/>  
    <h:panelGroup id="marcoLogicoProyecto">  
      <h:panelGroup layout="block" styleClass="well"  
rendered="#{proyectoBeneficioAntiguoSessController.marcoObjGeneralDTO.objetivo == null}">  
  
      #{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoGeneral.empty']}  
    </h:panelGroup>  
    <ui:fragment  
rendered="#{proyectoBeneficioAntiguoSessController.marcoObjGeneralDTO.objetivo != null}">  
      <h:panelGroup layout="block" styleClass="well" >  
        <div class="row">  
          <div class="col-lg-12">  
            <dl class="dl-horizontal">
```

```

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoGral']}</dt>
    <dd class="text-
justify">#{proyectoBeneficioAntiguoSessController.marcoObjGeneralDTO.objetivo}</dd>
        <br/>

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.meta']}</dt> <dd
class="text-justify">#{proyectoBeneficioAntiguoSessController.marcoObjGeneralDTO.meta}</dd>
        <br/>

        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.fuenteVerificacion']}:
</dt> <dd class="text-
justify">#{proyectoBeneficioAntiguoSessController.marcoObjGeneralDTO.fuenteVerificacion}</dd
>
</dl>
    </div>
    </div>
    </h:panelGroup>
</ui:fragment>

    <div class="row">
        <div class="col-lg-12">

            <h2>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivosEspecificos.l
abel']}</h2>

                <br/>
            </div>
        </div>
        <h:panelGroup layout="block" styleClass="well"
rendered=("#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaObjetivosEspecific
os.size() eq 0}")>

            #{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivosEspecificos.empty
']}
                </h:panelGroup>
                <h:panelGroup layout="block"
rendered=("#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaObjetivosEspecific
os.size() gt 0}")>

                    <div class="panel-group" id="accordionObjEspecificos">
                        <ui:repeat
value=("#{proyectoBeneficioAntiguoSessController.detalleProyectoVO.listaObjetivosEspecificos}
"
                            var="objetivoEspecifico">
                            <div class="panel panel-default">
                                <div class="panel-heading">
                                    <h5 class="panel-title">
                                        <a data-
toggle="collapse" data-parent="#accordionObjEspecificos"
href="#objetivoEspecifico#{objetivoEspecifico.numeroObjetivo}"><i class="fa fa-arrows-
v"></i>
                                        </a>
                                    </h5>
                                </div>
                                <div
id="objetivoEspecifico#{objetivoEspecifico.numeroObjetivo}"
                                class="panel-collapse
collapse">
                                    <div class="panel-body">
                                        <h:panelGroup
layout="block" styleClass="well">

```

```

class="row">
class="col-lg-12">
    <dl class="dl-horizontal">
        <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.obj
etivo.label']}</dt> <dd class="text-justify">#{objetivoEspecifico.objetivo}</dd>
    </dl>
    </div>
    </div>
    </h:panelGroup>
    <h4>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.resultados.label']}</h
4>
    <br/>
    <div class="dd"
    <ol class="dd-
list">
    <h:panelGroup layout="block" styleClass="well" rendered="#{empty
objetivoEspecifico.marcoActividadList}">
        #{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.activid
ades.empty']}
    </h:panelGroup>
    <ui:repeat value="#{objetivoEspecifico.marcoActividadList}"
        rendered="#{not empty objetivoEspecifico.marcoActividadList}"
        var="actividad">
    <li class="dd-item" data-id="#{actividad.numeroActividad}">
        <div class="dd-handle">
            <span class="label label-info" data-toggle="collapse" data-
parent="#{accordionActividades}"></span> Actividad - #{actividad.numeroActividad} :
            #{actividad.actividad}
        </div>
        <ol class="dd-list">
            <h:panelGroup layout="block" styleClass="well">
                <div class="row">
                    <div class="col-lg-12" style="height: 40px;">
                        <dl class="dl-horizontal">

```



```

<dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.per
iodoInicial.label']}</dt> <dd class="text-justify">#{actividad.periodoInicial}</dd>

<dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.per
iodoFinal.label']}</dt> <dd class="text-justify">#{actividad.periodoFinal}</dd>

<dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.dur
acion.label']}</dt>

<dd class="text-
justify">#{actividad.cronograma == 'M'

? msg['listaDuracion.M']

: actividad.cronograma == 'S'

? msg['listaDuracion.S']

: msg['listaDuracion.D']}</dd>

</dl>

</div>

</div>

</h:panelGroup>

<h:panelGroup layout="block" styleClass="well" rendered="#{empty
actividad.marcoResultadoList}">

#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.resulta
dos.empty']}

</h:panelGroup>

<ui:repeat value="#{actividad.marcoResultadoList}"

rendered="#{not empty actividad.marcoResultadoList}"

var="resultado">

<li class="dd-item" data-id="#{resultado.numero}">

<div class="dd-handle" >

<span class="label label-info"></span>

Resultado - #{resultado.numero} : #{resultado.descripcion}

</div>

<ol class="dd-list">

```

```

<h:panelGroup layout="block" styleClass="well"
rendered="#{empty resultado.marcoIndicadorList}">

    #{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.indicad
ores.empty']}

</h:panelGroup>

<ui:repeat
value="#{resultado.marcoIndicadorList}"

    rendered="#{not empty
    resultado.marcoIndicadorList}"

    var="indicador">

        <li class="dd-item" data-
id="#{indicador.numeroIndicador}">

            <div class="dd-handle">

                <span class="label
label-info"></span> Indicador - #{indicador.numeroIndicador} : #{indicador.indicador}

            </div>

            <ol class="dd-list">

                <li class="dd-item">

                    <h:panelGroup
layout="block" styleClass="well">

                        <div
class="row">

                            <div class="col-lg-12">

                                <dl class="dl-horizontal">

                                    <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.med
ioVerif.label']}:</dt> <dd class="text-justify">#{indicador.medioVerificacion}</dd>

                                    <br/>

                                    <dt>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.marco.objetivoEspecifico.mes
Cumpli.label']}:</dt> <dd class="text-justify">#{indicador.mesCumplimiento}</dd>

                                </dl>

```

```
</div>
</div>
</h:panelGroup>
</li>
</ol>
</li>
</ui:repeat>
</ol>
</li>
</ui:repeat>
</ol>
</li>
</ui:repeat>
</ol>
</div>
</div>
</div>
</div>
</div>
</div>
```

- ◆ **Carga de datos**

Al cambiar de pestaña se ejecuta un `remoteCommand` que invoca el método de carga correspondiente:

```
<p:remoteCommand id="cargarMarcoRC" name="cargarMarcoLogico"
                update="marcoLogicoProyecto"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarMarcoLogico()}" />
```

Se acompaña con una animación de carga usando blockUI:

```
<p:blockUI block="projecChildrens" trigger="cargarMarcoRC" widgetVar="demoBlock">
```

◆ **Visualización del Marco Lógico**

Dentro de:

```
<h:panelGroup id="marcoLogicoProyecto">
```

✓ **1. Objetivo general del proyecto**

Se muestra si existe el valor `proyectoBeneficioAntiguoSessController.marcoObjGeneralDTO.objetivo`. Si no existe, se muestra un mensaje de alerta (well Bootstrap).

Si existe, se despliega un resumen del objetivo general, meta y fuente de verificación en un bloque dl-horizontal.

✓ **2. Objetivos Específicos**

- Se muestran en un componente tipo accordion colapsable.
- Cada **objetivo específico** tiene una sección con:
 - Descripción del objetivo (`objetivoEspecifico.objetivo`)
 - Lista de **actividades** relacionadas.

✓ **3. Actividades**

Cada actividad incluye:

- Número y descripción: `#{actividad.numeroActividad} : #{actividad.actividad}`
- Periodo inicial y final
- Duración en texto, basada en el tipo de cronograma:

```
actividad.cronograma == 'M' ? msg['listaDuracion.M'] : ...
```

✓ 4. Resultados por actividad

Anidados dentro de cada actividad:

- Lista de resultados, con número y descripción.
- Si no hay resultados: mensaje de lista vacía (well).

✓ 5. Indicadores por resultado

Cada resultado puede tener varios indicadores, cada uno incluye:

- Número y descripción
- Medio de verificación
- Mes estimado de cumplimiento

◆ Estructura Jerárquica

El contenido se construye con ui:repeat y <ol class="dd-list"> para representar una **estructura tipo árbol**:

plaintext

CopiarEditar

Objetivo General

└─ Objetivos Específicos

└─ Actividades

└─ Resultados

└─ Indicadores

Cada nodo tiene su estructura visual (dd-handle, well, etc.) para mejorar la presentación y navegación.

 Funcionalidades clave

Elemento	Tipo	Descripción
blockUI	Bloqueo visual	Para mostrar carga mientras se procesan los datos
remoteCommand	Llamado AJAX	Ejecuta cargarMarcoLogico() desde el Controller
ui:repeat	Repetición de estructuras	Para recorrer objetivos, actividades, resultados e indicadores
dl-horizontal	Etiquetas y valores	Para mostrar pares "etiqueta: valor" de forma legible
collapse	Interacción visual	Para contraer/expandir objetivos específicos y actividades

5. Pestaña de Evaluación

- Carga mediante **cargarEvaluacion()**;
- Contiene detalles del proceso de revisión del proyecto, evaluadores, observaciones técnicas, conceptos o resultados del análisis técnico.
- Esta pestaña está diseñada para presentar al usuario final la información detallada sobre las evaluaciones realizadas a un proyecto antiguo. Se aprovechan los componentes de PrimeFaces para interacción dinámica (AJAX), manejo de bloques durante procesos y uso de bibliotecas JS externas (DataTables) para una mejor experiencia de usuario.

```
<!-- Pestaña Evaluacion -->
<div class="tab-pane" id="tab-evaluacion">
  <p:remoteCommand id="cargarEvaluacionRC" name="cargarEvaluacion"
  update="infoEvaluacion"
  actionListener="#{proyectoBeneficioAntiguoSessController.cargarEvaluacionesProyecto()}" />
  <p:blockUI block="projecChildrens" trigger="cargarEvaluacionRC" widgetVar="demoBlock">
    #{msg['sb.mensajes.cargando.label']}<br/>
  </p>
</div>
```

```

        <div class="progress progress-striped active m-b-sm">
            <div style="width: 100%;" class="progress-bar"></div>
        </div>
    </p:blockUI>
    <h2 style="text-align: center">

    <b>#{msg[ 'gestion.beneficiosTributarios.proyectosAntiguos.tab.revisionEvaluacion' ]}</b>
    </h2>
    <br/>
    <h:panelGroup id="infoEvaluacion" >
        <h:panelGroup layout="block" styleClass="well"

rendered="#{proyectoBeneficioAntiguoSessController.listaEvaluaciones.size() eq 0
or
proyectoBeneficioAntiguoSessController.listaEvaluaciones == null}">

    #{msg[ 'gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.emptyTable' ]}
    </h:panelGroup>
    <h:panelGroup styleClass="animated fadeInLeft" layout="block"

rendered="#{proyectoBeneficioAntiguoSessController.listaEvaluaciones.size() gt 0}">
    <h:dataTable
value="#{proyectoBeneficioAntiguoSessController.listaEvaluaciones}"
var="evaluacion"

id="#{proyectoBeneficioAntiguoSessController.modulo}tablaEvaluaciones"
styleClass="table table-striped table-bordered
table-hover dataTables-evaluaciones"
    >
        <h:column headerClass="column1">
            <!-- column header -->
            <f:facet
name="header">#{msg[ 'gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.
tipoEvaluacion' ]}</f:facet>
            <!-- row record -->

            #{proyectoBeneficioAntiguoConvController.consultarValorTipoEvaluacion(evaluacion.tipoEva
aluacion)}
        </h:column>
        <h:column headerClass="column1">
            <!-- column header -->
            <f:facet
name="header">#{msg[ 'gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.
tipoEvaluador' ]}</f:facet>
            <!-- row record -->
            #{evaluacion.tipoEvaluador eq 'I' ? 'Interno' :
evaluacion.tipoEvaluador eq null ? '' : 'Externo'}
        </h:column>
        <h:column headerClass="column1">
            <!-- column header -->
            <f:facet
name="header">#{msg[ 'gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.
nombres' ]}</f:facet>
            <!-- row record -->
            #{evaluacion.nombrePanel != null
? evaluacion.nombrePanel
: evaluacion.tipoEvaluador eq 'I'
? evaluacion.funcionario
:
proyectoBeneficioAntiguoConvController.nombreCompletoPersona(evaluacion)}
        </h:column>

```

```

<h:column headerClass="column1">
    <!-- column header -->
    <f:facet
name="header">#{msg['gestion.reconocimientoActores.gestionar.solicitud.evaluacion.asignacion
Eval.usuarioSigp']}</f:facet>
    <!-- row record -->
    #{evaluacion.usuarioSigp}
</h:column>
<h:column headerClass="column1">
    <!-- column header -->
    <f:facet
name="header">#{msg['sb.component.boton.opciones']}</f:facet>
    <!-- row record -->
    <p:commandLink style="color: #fff;"

        title="#{msg['sb.component.boton.detalles']}"

        actionListener="#{proyectoBeneficioAntiguoConvController.prepararEditar(evaluacion)}"
                                styleClass="btn btn-xs
btn-primary sbTooltip"

        oncomplete="$('#dlgDetallesEvaluacion').modal();"

        update="formDetallesEvaluacion"

                                process="@this"
                                >
                                <i class="fa fa-list"></i>
                                </p:commandLink>
                                <p:spacer/>
                                <p:commandLink
title="#{msg['sb.component.boton.download']}"
                                styleClass=" btn btn-xs
btn-primary sbTooltip"

        actionListener="#{proyectoBeneficioAntiguoConvController.downloadReporteEvaluacion('cal
ificacion', evaluacion.evaluacionProyectoId)}"
                                immediate="true"
                                ajax="false"
                                >
                                <i class="fa fa-file-pdf-o"></i>
                                </p:commandLink>
</h:column>
</h:dataTable>
<script>
    // A $( document ).ready() block.
    $(document).ready(function () {
        $('#dataTables-evaluaciones').DataTable({
            "language": {
                "url":
                "#{request.contextPath}/resources/js/plugins/datatables-
plugins/i18n/#{i18nBean.currentLocale}.lang"
            }
        });
    });
</script>
</h:panelGroup>
</h:panelGroup>
</div>

```


1. Propósito del componente

La pestaña de evaluación permite consultar las evaluaciones asociadas a un proyecto antiguo. A través de esta interfaz, el usuario puede:

- Visualizar una tabla con todas las evaluaciones realizadas al proyecto.
- Identificar el tipo de evaluación y evaluador.
- Ver los nombres de los evaluadores asignados.
- Descargar el informe de evaluación.
- Acceder a los detalles de cada evaluación en una ventana modal.

◆ 2. Estructura General

```
<div class="tab-pane" id="tab-evaluacion">  
  ...  
</div>
```

El componente está contenido en un div con clase tab-pane y id="tab-evaluacion", lo cual permite integrarlo en una estructura de pestañas tipo Bootstrap o similar.

◆ 3. Carga dinámica de información

```
<p:remoteCommand id="cargarEvaluacionRC"  
  name="cargarEvaluacion"  
  update="infoEvaluacion"
```

```
actionListener="#{proyectoBeneficioAntiguoViewController.cargarEvaluacionesProyecto()}" />
```

- **Propósito:** permite cargar de forma dinámica la información de evaluaciones del proyecto seleccionado.
- **Uso:** este comando se invoca desde JavaScript, normalmente cuando el usuario accede a esta pestaña.
- **Acción asociada:** cargarEvaluacionesProyecto() – método del controlador que consulta la base de datos o backend para obtener la lista de evaluaciones.

◆ 4. Bloqueo de interfaz mientras carga

```
<p:blockUI block="projecChildrens" trigger="cargarEvaluacionRC" widgetVar="demoBlock">
    #{msg['sb.mensajes.cargando.label']}
    ...
</p:blockUI>
```

- **Componente:** <p:blockUI> de PrimeFaces
- **Función:** bloquea la sección identificada con block="projecChildrens" mientras se ejecuta el remoteCommand, mostrando un mensaje de carga con una barra de progreso animada.
- **Internacionalización:** el mensaje es dinámico y extraído desde un ResourceBundle.

◆ 5. Encabezado de sección

```
<h2 style="text-align: center">
    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.revisionEvaluacion']}</b>
</h2>
```

Muestra el título de la pestaña, usando ResourceBundle para soportar múltiples idiomas.

◆ 6. Contenedor de datos de evaluación

```
<h:panelGroup id="infoEvaluacion">
    ...
</h:panelGroup>
```

Agrupar toda la información evaluativa. Este contenedor es el que se actualiza dinámicamente desde el remoteCommand.

◆ 7. Condicional sin evaluaciones

```
<h:panelGroup layout="block" styleClass="well"
    rendered="#{proyectoBeneficioAntiguoViewController.listaEvaluaciones.size() eq 0
        or proyectoBeneficioAntiguoViewController.listaEvaluaciones ==
        null}">
```

```
{msg['gestion.beneficiosTributarios.proyectosAntiguos.evaluacion.evaluadores.emptyTable']}}  
</h:panelGroup>
```

- Muestra un mensaje informativo cuando no hay evaluaciones registradas para el proyecto.
- Se utiliza una condición rendered para evaluar si la lista está vacía o es null.

◆ 8. Tabla de Evaluaciones

```
<h:dataTable value="#{proyectoBeneficioAntiguoViewController.listaEvaluaciones}"  
var="evaluacion" ...>  
...  
</h:dataTable>
```

Columnas:

6. Tipo Evaluación

- Obtenido mediante método consultarValorTipoEvaluacion() del proyectoBeneficioAntiguoConvController.

7. Tipo Evaluador

- Muestra "Interno", "Externo" o vacío según el valor de evaluacion.tipoEvaluador.

8. Nombres Evaluador

- Si es tipo "Panel", muestra nombrePanel.
- Si es interno, muestra funcionario.
- Si es externo, muestra nombre completo usando nombreCompletoPersona().

9. Usuario SIGP

- Muestra el usuario del sistema asignado a la evaluación.

10. Opciones

- Botón "Detalles": abre modal dlgDetallesEvaluacion.

- Botón "Descargar": descarga el PDF de calificación, vía método `downloadReporteEvaluacion()`.

◆ 9. Inicialización de tabla con DataTables.js

```
$(document).ready(function () {  
    $('.dataTables-evaluaciones').DataTable({  
        "language": {  
            "url": "#{request.contextPath}/resources/js/plugins/datatables-  
plugins/i18n/#{i18nBean.currentLocale}.lang"  
        }  
    });  
});
```

- Se utiliza jQuery para aplicar el plugin DataTables a la tabla.
- El archivo de lenguaje se selecciona dinámicamente según el idioma actual del sistema.

✚ Integración con lógica de negocio

- **Carga de Evaluaciones:**
`proyectoBeneficioAntiguoViewController.cargarEvaluacionesProyecto()`
→ Consulta a la base de datos las evaluaciones asignadas al proyecto actual.
- **Obtención de valores de campos específicos:**
`proyectoBeneficioAntiguoConvController.consultarValorTipoEvaluacion(...)`
`proyectoBeneficioAntiguoConvController.nombreCompletoPersona(...)`
- **Acciones de botones:**
 - `prepararEditar(...)`: configura los datos necesarios para ver detalles en la ventana modal.
 - `downloadReporteEvaluacion(...)`: genera y entrega el archivo PDF de la evaluación.

6. Pestaña de Documentos

- Carga mediante **cargarDocumentos()**;
- Muestra los archivos adjuntos, plantillas cargadas, fichas del proyecto, resoluciones, anexos, etc.
- La pestaña **Documentos** permite al usuario final acceder a los archivos cargados para el proyecto de una forma visual y sencilla. El diseño basado en tarjetas mejora la usabilidad, mientras que el uso de PrimeFaces garantiza la integración fluida con la lógica del backend para la descarga directa de archivos.

```
<!-- Pestaña Documentos -->
<div class="tab-pane" id="tab-documentos">
  <p:remoteCommand id="cargarDocumentosRC" name="cargarDocumentos"
update="infoDocumentos"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarDocumentosProyecto()}" />
  <p:blockUI block="projecChildrens" trigger="cargarDocumentosRC" widgetVar="demoBlock">
    #{msg['sb.mensajes.cargando.label']}<br/>
    <div class="progress progress-striped active m-b-sm">
      <div style="width: 100%;" class="progress-bar"></div>
    </div>
  </p:blockUI>
  <h2 style="text-align: center">
    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.documentos']}</b>
  </h2>
  <br/>
  <h:panelGroup id="infoDocumentos" >
    <h:panelGroup layout="block" styleClass="well"

rendered="#{proyectoBeneficioAntiguoSessController.listaDocumentos.size() eq 0
or
proyectoBeneficioAntiguoSessController.listaDocumentos == null}">

  #{msg['gestion.beneficiosTributarios.proyectosAntiguos.documentos.emptyTable']}
  </h:panelGroup>
  <h:panelGroup styleClass="animated fadeInLeft" layout="block"

rendered="#{proyectoBeneficioAntiguoSessController.listaDocumentos.size() gt 0}">
    <ui:repeat var="documento"
value="#{proyectoBeneficioAntiguoSessController.listaDocumentos}">
      <div id="documento#{documento.documentoId}" class="col-lg-12
mb-3">
        <div class="card">
          <div class="card-body">
            <div class="row align-items-center">
              <div class="col-lg-10">
                <h5 class="card-title">
                  <i class="fa
fa-file"></i>
                  #{documento.nombre}
                </h5>
              </div>
              <div class="col-lg-2 text-
right">
                <p:commandLink
id="descargarDoc"
```

```
title="#{msg['sb.mensajes.descargar.label']}" - #{documento.nombre}"

process="@this"

update="@this"

ajax="false"

rendered="#{documento.nombre != null}"

immediate="true"

actionListener="#{proyectoBeneficioAntiguoConvController.descargarArchivoDocumentos(documento)}"

        styleClass="btn btn-primary">

                                                    <i class="fa
fa-download"></i> Descargar

                                                    <p:fileDownload
value="#{proyectoBeneficioAntiguoSessController.archivoDocumento}"/>

                                                    </p:commandLink>
        </div>
    </div>
</div>
</ui:repeat>

</h:panelGroup>
</h:panelGroup>

</div>
```

- ◆ **1. Propósito del componente**

La pestaña **Documentos** permite visualizar y descargar los documentos asociados a un proyecto antiguo. Cada documento se muestra de manera visual mediante tarjetas y se ofrece un botón de descarga individual.

◆ 2. Estructura General

```
<div class="tab-pane" id="tab-documentos">
    ...
</div>
```

Este bloque está contenido dentro de una pestaña (tab-pane) y se identifica con id="tab-documentos", lo cual permite su integración con un sistema de navegación por pestañas.

◆ 3. Carga dinámica de documentos

```
<p:remoteCommand id="cargarDocumentosRC"
    name="cargarDocumentos"
    update="infoDocumentos"

    actionListener="#{proyectoBeneficioAntiguoSessController.cargarDocumentosProyecto()}" />
```

- Permite cargar los documentos asociados al proyecto actual.
- Se invoca desde JavaScript cuando el usuario accede a esta pestaña.
- Ejecuta el método cargarDocumentosProyecto() en el ViewController.

◆ 4. Bloqueo durante la carga

```
<p:blockUI block="projecChildrens" trigger="cargarDocumentosRC" widgetVar="demoBlock">
    #{msg['sb.mensajes.cargando.label']}<br/>
    <div class="progress progress-striped active m-b-sm">
        <div style="width: 100%;" class="progress-bar"></div>
    </div>
</p:blockUI>
```

- Muestra una animación de carga mientras se ejecuta la acción de carga de documentos.
- Bloquea el componente projecChildrens para evitar interacción del usuario durante el proceso.

◆ 5. Encabezado de sección

```
<h2 style="text-align: center">
    <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.documentos']}</b>
</h2>
```

- Muestra el título principal de la pestaña.
- El texto se obtiene del archivo de propiedades (ResourceBundle) para permitir internacionalización.

◆ 6. Contenedor de la información

```
<h:panelGroup id="infoDocumentos">
    ...
</h:panelGroup>
```

Contiene todo el bloque que muestra los documentos o un mensaje si no existen.

◆ 7. Mensaje sin documentos

```
<h:panelGroup layout="block" styleClass="well"
    rendered="#{proyectoBeneficioAntiguoSessController.listaDocumentos.size() eq 0
    or proyectoBeneficioAntiguoSessController.listaDocumentos ==
    null}">
    #{msg['gestion.beneficiosTributarios.proyectosAntiguos.documentos.emptyTable']}
</h:panelGroup>
```

- Muestra un mensaje en pantalla cuando no existen documentos registrados para el proyecto.
- El mensaje se encuentra centralizado en los archivos de propiedades.

◆ 8. Listado de documentos

```
<ui:repeat var="documento"
    value="#{proyectoBeneficioAntiguoSessController.listaDocumentos}">
    <div id="documento#{documento.documentoId}" class="col-lg-12 mb-3">
        <div class="card">
            <div class="card-body">
                <div class="row align-items-center">
                    <div class="col-lg-10">
                        <h5 class="card-title">
                            <i class="fa fa-file"></i>
                            #{documento.nombre}
                        </h5>
                    </div>
                    <div class="col-lg-2 text-right">
                        ...
                    </div>
                </div>
            </div>
        </div>
    </div>
</ui:repeat>
```

- Usa ui:repeat para recorrer la lista listaDocumentos.
- Cada documento se renderiza como una tarjeta (card) estilo Bootstrap.
- Muestra el nombre del documento con ícono de archivo.

◆ 9. Descarga de documento

```
<p:commandLink id="descargarDoc"
                title="{msg['sb.mensajes.descargar.label']}" - #{documento.nombre}"
                process="@this"
                update="@this"
                ajax="false"
                rendered="{documento.nombre != null}"
                immediate="true"

                actionListener="{projectoBeneficioAntiguoConvController.descargarArchivoDocumentos(documento)}"
                styleClass="btn btn-primary">
    <i class="fa fa-download"></i> Descargar
    <p:fileDownload value="{projectoBeneficioAntiguoSessController.archivoDocumento}" />
</p:commandLink>
```

- **Acción:** llama a `descargarArchivoDocumentos(documento)` para generar el archivo para descarga.
- **Componente clave:** `<p:fileDownload>` de PrimeFaces. Se asocia al bean que expone el archivo `archivoDocumento`.
- **Propiedades importantes:**
 - `ajax="false"`: se requiere una petición completa (no AJAX) para iniciar la descarga.
 - `immediate="true"`: evita validaciones JSF antes de ejecutar la acción.
 - `process="@this"` y `update="@this"` aseguran procesamiento mínimo.

✿ Integración con la lógica de negocio

- `projectoBeneficioAntiguoSessController.cargarDocumentosProyecto()`
→ Se encarga de cargar desde la base de datos los documentos relacionados al proyecto.
- `projectoAntiguoConvController.descargarArchivoDocumentos(documento)`
→ Crea y expone el objeto `StreamedContent` necesario para el componente `<p:fileDownload>`.
- `projectoBeneficioAntiguoSessController.archivoDocumento`
→ Atributo tipo `StreamedContent` que contiene el archivo actual a descargar.

7. Pestaña de Observaciones

- Carga mediante **cargarObservaciones()**;
- Permite registrar, editar o visualizar observaciones de los diferentes revisores o entidades.
- La pestaña **Observaciones** proporciona una visualización detallada del historial de actividades realizadas sobre un proyecto. Está diseñada para ser visualmente clara y centrada en la experiencia del usuario, integrando funcionalidades de carga remota, formateo de fechas y scroll dinámico. Esto facilita a los usuarios rastrear los cambios y comentarios importantes realizados por otros participantes.

```
<!-- Pestaña Observaciones -->
<div class="tab-pane" id="tab-observaciones">
  <p:remoteCommand id="cargarObservacionesRC" name="cargarObservaciones"
update="infoObservaciones"
actionListener="#{proyectoBeneficioAntiguoSessController.cargarObservacionesProyecto()}" />
  <p:blockUI block="projecChildrens" trigger="cargarObservacionesRC"
widgetVar="demoBlock">
    #{msg['sb.mensajes.cargando.label']}<br/>
    <div class="progress progress-striped active m-b-sm">
      <div style="width: 100%;" class="progress-bar"></div>
    </div>
  </p:blockUI>
  <h2 style="text-align: center">

  <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.observaciones']}</b>
  </h2>
  <br/>
  <h:panelGroup id="infoObservaciones" >
    <h:panelGroup layout="block" styleClass="well"

rendered="#{proyectoBeneficioAntiguoSessController.listaObservacionProyecto.size() eq 0
or
proyectoBeneficioAntiguoSessController.listaObservacionProyecto == null}">

  #{msg['gestion.beneficiosTributarios.proyectosAntiguos.observaciones.emptyTable']}
  </h:panelGroup>
  <h:panelGroup styleClass="feed-activity-list" layout="block"

rendered="#{proyectoBeneficioAntiguoSessController.listaObservacionProyecto.size() gt 0}">
    <ui:repeat
value="#{proyectoBeneficioAntiguoSessController.listaObservacionProyecto}"
var="observacion">
      <div class="feed-element" style="margin: 14px">
        <a href="#" class="pull-left">
          
</a>
<div class="media-body ">
  <small class="pull-right">
    <div class="ibox-tools open">
      </div>
    </small>
    <strong>#{observacion.nombreUsuario}</strong>

    <small class="text-muted">
      Creado
      #{proyectoBeneficioAntiguoConvController.isBeforeToday(observacion.fechaCreacion)} :
      value="#{observacion.fechaCreacion}">
        <f:convertDateTime
          pattern="dd/MM/yyyy hh:mm a" timeZone="EST"/>
        </h:outputText>
      <ui:fragment
        rendered="#{observacion.fechaModificacion ne null}">
          - Modificado
          #{proyectoBeneficioAntiguoConvController.isBeforeToday(observacion.fechaModificacion)} :
          value="#{observacion.fechaModificacion}">
            <f:convertDateTime
              pattern="dd/MM/yyyy hh:mm a" timeZone="EST"/>
            </h:outputText>
          </ui:fragment>
        </small>
        <ui:fragment
          rendered="#{observacion.observacion ne null}">
            <div class="well">
              #{observacion.observacion}
            </div>
          </ui:fragment>
        </div>
      </div>
    </ui:repeat>
    <script>
      var scrollear = function scrollearF() {
        $(".feed-activity-list").each(function (index) {
          if ($(this).height() > 500) {
            $(this).slimScroll({
              height: '500px',
              color: '#1AB394'
            });
          }
        });
      };
      scrollear();
    </script>
  </h:panelGroup>
</h:panelGroup>
</div>

```

- ◆ **1. Propósito del componente**

La pestaña **Observaciones** permite mostrar un historial de mensajes u observaciones realizadas por los usuarios sobre el proyecto. Cada observación incluye el nombre del usuario, la descripción de la acción, la fecha de creación y, si aplica, la de modificación.

◆ 2. Estructura general

```
<div class="tab-pane" id="tab-observaciones">
    ...
</div>
```

Define el contenedor de la pestaña de observaciones con el identificador tab-observaciones, para ser manejado por un sistema de navegación por pestañas.

◆ 3. Carga dinámica de observaciones

```
<p:remoteCommand id="cargarObservacionesRC"
    name="cargarObservaciones"
    update="infoObservaciones"

    actionListener="#{proyectoBeneficioAntiguoSessController.cargarObservacionesProyecto()}" />
```

- Permite invocar vía JavaScript el método cargarObservacionesProyecto() para cargar las observaciones del proyecto.
- Actualiza el componente infoObservaciones con los resultados obtenidos.

◆ 4. Indicador de carga (bloqueo de UI)

```
<p:blockUI block="projecChildrens" trigger="cargarObservacionesRC" widgetVar="demoBlock">
    ...
</p:blockUI>
```

- Muestra una animación de carga mientras se procesan las observaciones.
- Bloquea el componente identificado por projecChildrens para evitar interacciones durante la operación.

◆ 5. Título de la sección

```
<h2 style="text-align: center">
  <b>#{msg['gestion.beneficiosTributarios.proyectosAntiguos.tab.observaciones']}</b>
</h2>
```

Presenta el título de la pestaña, obtenido del archivo de propiedades para soportar internacionalización.

◆ 6. Contenedor de contenido

```
<h:panelGroup id="infoObservaciones">
  ...
</h:panelGroup>
```

Agrega dinámicamente el contenido que será actualizado con el resultado de la carga remota.

◆ 7. Mensaje de "sin observaciones"

```
<h:panelGroup layout="block" styleClass="well"
  rendered="#{proyectoBeneficioAntiguoSessController.listaObservacionProyecto.size() eq 0
    or
    proyectoBeneficioAntiguoSessController.listaObservacionProyecto == null}">
  #{msg['gestion.beneficiosTributarios.proyectosAntiguos.observaciones.emptyTable']}
</h:panelGroup>
```

- Muestra un mensaje informativo si no existen observaciones registradas.
- Se utiliza la clase well de Bootstrap para destacar el mensaje.

◆ 8. Listado de observaciones

```
<h:panelGroup styleClass="feed-activity-list" layout="block"
  rendered="#{proyectoBeneficioAntiguoSessController.listaObservacionProyecto.size() gt
    0}">
  <ui:repeat value="#{proyectoBeneficioAntiguoSessController.listaObservacionProyecto}"
    var="observacion">
    ...
  </ui:repeat>
</h:panelGroup>
```

- Se utiliza ui:repeat para iterar sobre la lista listaObservacionProyecto.
- Cada observación se presenta con formato tipo "actividad reciente" usando la clase feed-element.

◆ 9. Detalles de cada observación

```
<div class="feed-element">
    ...
    <strong>#{observacion.nombreUsuario}</strong> #{observacion.log}
    ...
    Creado
    #{proyectoBeneficioAntiguoConvController.isBeforeToday(observacion.fechaCreacion)} :
    <h:outputText value="#{observacion.fechaCreacion}">
        <f:convertDateTime pattern="dd/MM/yyyy hh:mm a" timeZone="EST"/>
    </h:outputText>
    ...
</div>
```

- Muestra:
 - Usuario que generó la observación (nombreUsuario).
 - Descripción del evento (log).
 - Fecha de creación con formato amigable.
 - Si existe una modificación, también se muestra con su respectiva fecha.
- Se utiliza un método auxiliar isBeforeToday(...) para comparar la fecha contra la fecha actual (ej. mostrar "hoy", "ayer", etc.).

◆ 10. Visualización del contenido de la observación

```
<ui:fragment rendered="#{observacion.observacion ne null}">
    <div class="well">
        #{observacion.observacion}
    </div>
</ui:fragment>
```

- Si la observación contiene un texto adicional, este se muestra enmarcado con estilo well.
- El uso de ui:fragment evita que se renderice un div innecesario si el contenido está vacío.

◆ 11. Estilos y experiencia de usuario

```
<script>
    var scrollear = function scrollearF() {
```

```
$(".feed-activity-list").each(function (index) {  
    if ($(this).height() > 500) {  
        $(this).slimScroll({  
            height: '500px',  
            color: '#1AB394'  
        });  
    }  
});  
});  
scrolllear();  
</script>
```

- Si el contenido de observaciones sobrepasa los 500px de alto, se activa automáticamente un scroll vertical con estilo personalizado usando **jQuery SlimScroll**.
- Mejora la usabilidad manteniendo el contenido dentro de un espacio fijo.

Integración con la lógica del backend

- `proyectoBeneficioAntiguoSessController.cargarObservacionesProyecto()`
→ Carga desde la base de datos las observaciones asociadas al proyecto.
- `proyectoBeneficioAntiguoConvController.isBeforeToday(Date)`
→ Método utilitario que permite evaluar y formatear fechas en función del día actual, para enriquecer el contexto de cada observación.

Requisitos de Proyectos Antiguos Beneficios Tributarios

Ubicación del componente:

Pantalla asociada a requisitos del proyecto antiguo.

Controladores implicados:

- `requisitoAntiguoSessController`
- `requisitoAntiguoConvController`

1. Propósito del componente

Esta pantalla permite visualizar los **requisitos de la convocatoria** para el proyecto antiguo y sus respectivos documentos adjuntos. Se presenta una tabla de requisitos y un conjunto de acciones asociadas a cada documento, como descarga o visualización.

◆ 2. Estructura general

```
<div class="ibox-content">
  <h:form id="#{requisitoAntiguoSessController.modulo}Form" styleClass="form-horizontal">
    <p:remoteCommand name="rc"

update="#{requisitoAntiguoSessController.modulo}Form:requisitos"
actionListener="#{requisitoAntiguoSessController.cargarListas()}"
              process="@this"/>
    <div class="row">
      <div class="col-lg-12" style="overflow-x: scroll;margin-bottom: 40px;">
        <p:dataTable id="requisitos"
                      var="item"

value="#{requisitoAntiguoSessController.requisitosConvocatoria}"
rowKey="#{item.convocatoriaRequisitoDTO.convocatoriaRequisitoId}"
                      refrow="true"
                      editable="true"
                      style="font-size: 13px;"
                      currentPageReportTemplate="Total de
registros: {totalRecords}"
                      paginator="true"
                      paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
                      paginatorPosition="bottom"

emptyMessage="#{msg['sb.datatable.registro.empty']}"
                      >
        <p:column width="30%">
          <f:facet name="header">
            <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.requisito.label']}" />
          </f:facet>
          <h:outputText
value="#{item.convocatoriaRequisitoDTO.subsanable == 'N' ? '*' : ''}"
#{item.convocatoriaRequisitoDTO.requisitoId.descripcionValorDefinicion}" />
          </p:column>
          <p:column width="20%">
            <f:facet name="header">
              <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.nombre.archivo.label']}"
            </f:facet>
            <h:outputText
value="#{item.documentoDTO.nombre}" title="#{item.documentoDTO.nombre}" />
            </p:column>
            <p:column width="10%">
              <f:facet name="header">
```



```

                                <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.tamano.label']}" />
                                </f:facet>
                                <h:outputText
value="#{requisitoAntiguoConvController.calcularTamanoKb(item.documentoDTO.docSize)}"
rendered="#{item.documentoDTO ne null}"
                                styleClass="pull-
right"/>
                                </p:column>
                                <p:column width="12%" style="text-align: center;">
                                    <f:facet name="header">
                                        <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.progreso.label']}" />
                                        </f:facet>
                                        <h:outputText
styleClass="#{item.documentoDTO.nombre != null ? 'badge badge-primary' : ''}"
value="#{item.documentoDTO.nombre != null ?
msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.progreso.ok'] : 'Sin archivo
adjunto'}"
                                style="white-space:
initial;"/>
                                </p:column>
                                <p:column width="10%"
                                style="text-align: center;"
                                rendered="false"
                                >
                                    <f:facet name="header">
                                        <h:outputText
value="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.cumple.label']}" />
                                        </f:facet>
                                        <f:facet name="output">
                                            <h:outputText
styleClass="#{item.requisitoDTO.cumple eq 'S' ? 'badge badge-success'
: item.requisitoDTO.cumple eq 'N' ? 'badge badge-danger'
: item.requisitoDTO.cumple eq 'P' ? 'badge badge-default' : ''}"
value="#{msg['listaCumple.'.concat(item.requisitoDTO.cumple ne null ?
item.requisitoDTO.cumple : 'X')]}"/>
                                        </f:facet>
                                    </p:column>
                                    <p:column
headerText="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.fechaCumplimiento
.label']}"
                                width="10%"
                                rendered="false"
                                >
                                        <f:facet name="output">
                                            <h:outputText
value="#{item.requisitoDTO.fechaCumplimiento}"
                                <f:convertDateTime
pattern="dd/MM/yyyy" timeZone="EST"/>
                                </h:outputText>
                                        </f:facet>
                                    </p:column>
                                    <p:column exportable="false"
                                width="10%"
                                style="text-align: center;"

```

```

>
<f:facet name="header">
    <h:outputText
value="#{msg['sb.datatable.registro.opciones']}" />
    </f:facet>
    <p:commandLink
        rendered="false"
        style="color: #fff"

        actionListener="#{requisitoAntiguoConvController.onRequisitoSelect(item)}"
        update="dlgEditarRequisito"

        disabled="#{SeguridadSessController.noEsUsuarioRegistro()}"

        title="#{msg['gestion.proyectosProgramas.proyectosAntiguos.requisitos.tooltip.cumplimie
nto']}"

        styleClass="btn btn-xs btn-primary
sbTooltip"

        process="@this"

        oncomplete="$('#editRequisito').modal();">
            <i class="fa fa-check"></i>
        </p:commandLink>
        <p:spacer width="10"></p:spacer>
        <p:commandLink
title="#{msg['sb.component.boton.adjuntar.add']}">
actionListener="#{requisitoAntiguoConvController.onRequisitoSelect(item)}"
update="formAdjRequisito"
oncomplete="$('#adjuntarDlg').modal()"
disabled="#{SeguridadSessController.noEsUsuarioRegistro()}"
rendered="false"
styleClass="btn btn-xs btn-primary">
            <i class="fa fa-upload"></i>
        </p:commandLink>
        <p:spacer width="10"></p:spacer>
        <p:commandLink
title="#{msg['sb.component.boton.download']}">
rendered="#{item.documentoDTO ne null}"
onclick="PrimeFaces.monitorDownload(start, stop)" ajax="false"
actionListener="#{requisitoAntiguoConvController.descargarArchivo(item)}"
            immediate="true"
            styleClass="btn btn-
xs btn-primary"
disabled="#{item.documentoDTO.documentoId == null}">
            <i class="fa fa-download"></i>
        </p:commandLink>
        <p:fileDownload
value="#{requisitoAntiguoSessController.archivo}" />
        </p:commandLink>
        <p:spacer width="10"></p:spacer>
        <p:commandLink
title="#{msg['sb.component.boton.visualizar']}">

```

```
update="dlgVisorRequisitos"
xs btn-primary"
styleClass="btn btn-
actionListener="#{requisitoAntiguoConvController.descargarArchivo(item)}"
rendered="#{item.documentoDTO != null}"
oncomplete="$('#visorRequisitos').modal()"
disabled="#{item.documentoDTO.documentoId == null}">
<i class="fa fa-eye"></i>
</p:commandLink>
</p:column>
</p:dataTable>
</div>
</div>
<p:spacer></p:spacer>
</h:form>
</div>
```

- El formulario se genera dinámicamente usando el identificador modulo, lo cual permite reutilizar la vista para distintos tipos de proyectos o contextos.
- Contiene una tabla de requisitos y comandos remotos para su carga.

◆ 3. Carga dinámica de datos

```
<p:remoteCommand name="rc"
update="#{requisitoProyectoAntiguoViewController.modulo}Form:requisitos"
actionListener="#{requisitoProyectoAntiguoViewController.cargarListas()}"
process="@this"/>
```

- Comando remoto utilizado para cargar la lista de requisitos desde el backend cuando se invoque por JavaScript.
- Se actualiza únicamente el componente requisitos.

◆ 4. Tabla de requisitos

```
<p:dataTable id="requisitos"
value="#{requisitoProyectoAntiguoViewController.requisitosConvocatoria}"
var="item"
paginator="true"
editable="true"
emptyMessage="{msg['sb.datatable.registro.empty']}"
... />
```

- Muestra la lista requisitosConvocatoria.
- Usa paginación y formato adaptativo (reflow).
- Cada fila representa un requisito y su archivo relacionado (si lo tiene).

◆ 5. Columnas de la tabla

✓ Requisito (obligatorio o subsanable)

```
<h:outputText value="#{item.convocatoriaRequisitoDTO.subsanable == 'N' ? '*':''}"  
#{item.convocatoriaRequisitoDTO.requisitoId.descripcionValorDefinicion}" />
```

- Si el requisito no es subsanable, se muestra con asterisco *.

📄 Nombre del archivo adjunto

```
<h:outputText value="#{item.documentoDTO.nombre}" title="#{item.documentoDTO.nombre}" />
```

- Muestra el nombre del archivo cargado, si existe.

📦 Tamaño del archivo

```
<h:outputText  
value="#{requisitoAntiguoConvController.calcularTamañoKb(item.documentoDTO.docSize)}" />
```

- Convierte el tamaño en bytes a kilobytes utilizando una función del controlador.

✓ Estado del archivo (progreso)

```
<h:outputText value="#{item.documentoDTO.nombre != null ? msg['...ok'] : 'Sin archivo  
adjunto'}" />
```

- Muestra un texto con badge indicando si hay archivo cargado.

✗ Columnas ocultas (cumplimiento y fecha de cumplimiento)

```
rendered="false"
```

- Columnas relacionadas con evaluación de cumplimiento, actualmente deshabilitadas.

◆ 6. Acciones por cada requisito

Se agrupan al final en una columna con íconos de acción:

✓ Validación manual (deshabilitada actualmente)

```
<p:commandLink ... oncomplete="$('#editRequisito').modal();">
  <i class="fa fa-check"></i>
</p:commandLink>
```

Adjuntar nuevo archivo (deshabilitada)

```
<p:commandLink oncomplete="$('#adjuntarDlg').modal()" ...>
  <i class="fa fa-upload"></i>
</p:commandLink>
```

↓ Descargar archivo

```
<p:commandLink actionListener="#{requisitoAntiguoConvController.descargarArchivo(item)}">
  <i class="fa fa-download"></i>
  <p:fileDownload value="#{requisitoAntiguoSessController.archivo}" />
</p:commandLink>
```

- Usa PrimeFaces.monitorDownload(start, stop) para indicar progreso.
- La acción descarga el archivo que fue preparado por el controller.

Visualizar archivo

```
<p:commandLink update="dlgVisorRequisitos" oncomplete="$('#visorRequisitos').modal()">
  <i class="fa fa-eye"></i>
</p:commandLink>
```

- Llama el mismo método descargarArchivo(item) pero orientado a mostrar el documento en un visor (ej. PDF viewer embebido).

◆ 7. Consideraciones de seguridad

- El botón de adjuntar y validar está condicionados por SeguridadSessController.noEsUsuarioRegistro().
- Esto asegura que solo los usuarios con permisos puedan realizar acciones sobre los requisitos.

✚ Integración con la lógica del backend

- requisitoAntiguoSessController.cargarListas()
→ Carga inicial de los requisitos desde la base de datos.
- requisitoAntiguoConvController.descargarArchivo(item)
→ Recupera el archivo asociado al requisito seleccionado y lo asigna al bean archivo.
- requisitoAntiguoConvController.calcularTamañoKb(size)
→ Método utilitario que convierte el tamaño del archivo a una unidad legible.

La pantalla de **Requisitos** proporciona una vista clara y funcional de los requisitos solicitados en la convocatoria, mostrando archivos adjuntos y su estado. Aunque algunos botones están deshabilitados, el diseño está preparado para permitir la carga, descarga y validación de documentos de forma controlada. La integración con seguridad por sesión y visualización modal hacen de esta pantalla un componente clave para la trazabilidad documental del proyecto.

2. INFORME DE ACTIVIDADES DE SOPORTE EN EL USO DEL SIGP EN LA CREACION DE CONVOCATORIAS Y TRANSFERIR EL CONOCIMIENTO DEL SIGP A LOS FUNCIONARIOS DE MINCIENCIAS, (No. Obligación Contrato: 2.2.3, 2.2.4)

Fecha: 16/Abril/2025 – 16/Mayo/2025

Elaborado por: Leonardo Andres Suarez Baldovino

Entidad: Ministerio de Ciencias, Tecnología e Innovación - Minciencias

Introducción

El presente informe detalla las actividades de soporte realizadas en el uso del Sistema de Información para la Gestión de Proyectos (SIGP), abarcando la parametrización de convocatorias, la generación de reportes, la solución de requerimientos y la resolución de errores.

Durante este período, se atendieron diversas solicitudes reportadas por los usuarios en la plataforma. A continuación, se presenta el resumen de las actividades realizadas, agrupadas por tipo de solicitud.

Resumen de Actividades

Tipo de Solicitud	Cantidad de Casos
Parametrización de Convocatorias	2
Parametrización de Rúbricas de Evaluación	3
Generación de Reportes	35
Solución de Errores	1
Actualización de Datos	6
Requerimientos	1
Total	48

Detalle de Casos Atendidos

A continuación, se detallan las solicitudes gestionadas, incluyendo la descripción del caso, su estado y la acción realizada:

Parametrización de Convocatorias

Se atendieron las siguientes solicitudes relacionadas con la configuración y ajuste de convocatorias en la plataforma SIGP:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
155735	Parametrización: 965-2025 Convocatoria Movilidades Académicas con Europa 2025	Solucionado	Producción	[Ver detalle]
155965	Parametrización: 967-2025 Convocatoria ANH Jóvenes Investigadores e Innovadores	Solucionado	Producción	[Ver detalle]

Solicitud 155735

Asunto	Tiempo total de actividad
parametrizacion sigp	00:21:10
Descripción	

Cordial saludo. Desde la Dirección de Vocaciones y Formación se solicita el apoyo a la OTSI en la parametrización en el Sistema Integrado de Gestión de Proyectos (SIGP) la Convocatoria de movilidades académicas con Europa 2025 la cual tendrá apertura en el mes de abril 2025.

Ase adjunta a la presente solicitud, los términos de referencia y Anexos No. 1, 2 y 3 los cuales hacen parte integran de la convocatoria.

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
03/04/2025 12:26:00	12/05/2025 16:27:42	12/05/2025 16:16:12	12/05/2025 16:27:42

Acción Realizada 155735

1. Revisión de los términos de referencia

- Se analizan los términos de referencia de la Convocatoria *965-2025 Movilidades Académicas con Europa 2025* para comprender los requisitos y lineamientos.
- Se identifican las condiciones específicas que deben ser parametrizadas en el sistema.

2. Análisis e identificación de la información

- Se definen los datos que deben ser parametrizados en la convocatoria.
- Se establece la estructura del formulario de proyectos para capturar la información requerida.

3. Diligenciamiento de la matriz de parametrización

- Se completa la **matriz de parametrización**, documentando cada campo, regla de validación y configuración necesaria en la convocatoria.
- Se parametriza lo siguiente:
 - Programas

- DIRECCIÓN DE VOCACIONES Y FORMACIÓN
- Financiaciones
 - RECUPERACIÓN CONTINGENTE
- Menús
 - PRY - Función General Proyectos
 - PRY - Información General Proyecto
 - PRY - Cambiar Clave
 - PRY - Proyecto
 - PRY - Roles Entidad
 - PRY - Gran Área OCDE, Área OCDE y Disciplina OCDE
 - PTY – Palabras Claves
 - PRY – Descripciones
 - PRY - Cronograma
 - PRY – Personal
 - PRY – Tipo Personal
 - PRY - Generalidades Personal
 - PRY – Presupuesto
 - PRY – Rubros
 - PRY – Global Total
 - PRY – Globales Entidad
 - PRY – Documentos
 - PRY - Requisitos
 - PRY – Validaciones
 - PRY – Grupos
 - PRY - Información Adicional
 - PRY – Línea Temática – Sublinea Temática
- Rubros
 - RUBROS FINANCIABLES
 - RUBROS NO FINANCIABLES
- Descripciones
 - 1 RESUMEN EJECUTIVO
 - 2 OBJETIVO GENERAL
 - 3 OBJETIVOS ESPECÍFICOS

- 4 HIPOTESIS
 - 5 MARCO TEÓRICO Y/O CIENTÍFICO
 - 6 DISEÑO METODOLÓGICO
 - 7 PRODUCTOS ESPERADOS JUNTO CON EL INDICADOR A TRAVÉS DEL CUAL SE PODRÁN MEDIR LOS RESULTADOS
 - 8 PERTINENCIA DE LA PROPUESTA DE MOVILIDAD
 - 9 VALOR DE LA COOPERACIÓN
 - 10 BIBLIOGRAFÍA
- Campos (Se ocultan)
 - CADENA PRODUCTIVA
 - CIUDAD DE FORMACION
 - CLASIFICACIÓN
 - CODIGO CIU
 - DURACIÓN PROYECTO HORAS
 - ESTADO EMPRESA
 - NOMBRE CLÚSTER
 - PERMISOS
 - PROPÓSITO GENERAL
 - SECTOR
 - TEMÁTICAS ORIENTADORAS POR FOCO ESTRATÉGICO
 - TIPO DE FINANCIACIÓN
 - TIPO PERSONA
 - VALOR INSCRIPCION
 - Financiadoras
 - MINCIENCIAS
 - Requisitos
 - 8.13 Este requisito es opcional y no es de carácter inhabilitante. Aquellos investigadores colombianos, estudiantes de doctorado y/o profesionales con título de maestría, que pertenezcan a población con enfoque diferencial y/o interseccional en alguna de las categorías que se especifican en la Tabla No. 1 “Categorías de enfoque diferencial e interseccional, del numeral 7. ENFOQUE DIFERENCIAL E INTERSECCIONAL, deberán adjuntar en el SIGP la documentación que se especifica en la columna “Descripción” de la Tabla No. 1. Se debe aportar

el documento de cada persona postulante a la movilidad internacional según corresponda. (Este requisito SI es subsanable)

- 8.12 Los profesionales graduados con título de maestría, deberán anexar el diploma de grado de la maestría y/o el acta de grado. Se debe aportar el documento de cada persona postulante a la movilidad internacional (Este requisito SI es subsanable).
- 8.11 Los profesionales graduados con título de maestría deben ser ciudadanos colombianos. Adjuntar copia de la cédula de ciudadanía legible por ambas caras. Se debe aportar el documento de cada persona postulante a la movilidad internacional (Este requisito SI es subsanable).
- 8.10 Los estudiantes de doctorado deberán anexar un certificado de la Institución de Educación Superior en el cual se certifique como estudiante activo del programa de doctorado a la presentación de la propuesta. La fecha de expedición del certificado no podrá superar los 60 días calendario al cierre de la convocatoria. El programa de doctorado deberá contar con registro calificado vigente del Ministerio de Educación Nacional, al momento de radicada la propuesta en la plataforma SIGP, y se verificará en el portal del Sistema Nacional de Información de Educación Superior - SNIES. Se debe aportar el documento de cada persona postulante a la movilidad internacional (Este requisito SI es subsanable).
- 8.9 Los estudiantes de doctorado deben ser ciudadano colombiano. Adjuntar copia de la cédula de ciudadanía legible por ambas caras. Se debe aportar el documento de cada persona postulante a la movilidad internacional. (Este requisito SI es subsanable).
- 8.8 Ser ciudadano colombiano o extranjero residente en Colombia. Adjuntar copia de la cédula de ciudadanía o cédula de extranjería legible por ambas caras. Se debe aportar el documento de cada persona postulante a la movilidad internacional (Este requisito SI es subsanable).
- 8.7 La propuesta deberá ser postulada y cargada en la plataforma dispuesta por la entidad aliada. La propuesta de movilidad en el marco de un proyecto de I+D+i que no haya sido radicada en la contraparte europea, no cumple con este requisito. Para validar este requisito, se debe adjuntar en la plataforma del Sistema Integrado de gestión de Proyectos (SIGP) captura de pantalla de la presentación de la propuesta ante la contraparte correspondiente, en la que se evidencie que el registro se culminó de forma exitosa. (Este requisito SI es subsanable).

- 8.6 Propuesta de Movilidad en el marco de un proyecto de I+D+I. Adjuntar documento con el contenido de la propuesta en formato PDF con base en las orientaciones del numeral 11. Este documento será objeto de verificación de requisitos, sin embargo, la puntuación que se dé sobre la propuesta es objeto del proceso de evaluación. (Este requisito NO es subsanable).
- 8.5 Formato de Movilidad (Anexo 3). Debe estar completamente diligenciado con los datos de los investigadores postulantes que van a realizar la movilidad internacional, información respecto a los productos esperados donde se relacionen las actividades y resultados deseados en el marco de la propuesta de I+D+i, cronograma de actividades y fechas de las movilidades internacionales. (Este requisito NO es subsanable).
- 8.4 Experiencia Académica, Profesional e Investigativa (Anexo 2). Debe estar completamente diligenciado con la información de cada una de las personas que conforman el equipo de trabajo de la propuesta de I+D+i y firmada por el investigador principal. (Este requisito NO es subsanable).
- 8.3 Entidad del Sistema Nacional de Ciencia, Tecnología e Innovación (SNCTI). Las entidades del Sistema Nacional de Ciencia, Tecnología e Innovación- SNCTI que presenten las postulaciones, deben estar constituidas veinticuatro (24) meses antes del cierre de la convocatoria, información que se verificará en el Registro Único Empresarial y Social (RUES) y/o documento equivalente. Deberán adjuntar: certificado de existencia y representación legal expedido por las Cámaras de Comercio o por el Ministerio de Educación Superior para los casos que aplique, con una fecha de expedición no mayor a tres (3) meses a la fecha de cierre de la convocatoria. (Este requisito SI es subsanable).
- 8.2 Carta Unificada de Aval y de Compromiso Institucional (Anexo 1). Se debe adjuntar el Anexo 1 “Carta Unificada de Aval y de Compromiso Institucional” debidamente diligenciado y firmado por el representante legal o su apoderado de la Entidad del Sistema Nacional de Ciencia, Tecnología e Innovación. En el caso de apoderado, de manera obligatoria deberá anexar a cada una de las propuestas presentadas el/los documento(s) que lo acrediten y faculen para la representación legal de la entidad. La carta de aval debe especificar el aporte de contrapartida en dinero para el seguro de viaje internacional con cubrimiento de seguro médico y repatriación, y/o aporte en especie de al menos el 50%, sobre el valor que podría financiar el Ministerio de Ciencia, Tecnología e

Innovación y aliados (ECOS NORD o DAAD). En caso de requerir incluir otros gastos de movilidad, puede relacionarlo en la contrapartida como otros gastos de la movilidad internacional. (Este requisito SI es subsanable).

- 8.1. Dirigido a: Cumplir con lo establecido en el numeral 3. “DIRIGIDO A” de los presentes términos de referencia. La información registrada en el Sistema Integrado de Gestión de Proyectos del Ministerio de Ciencia, Tecnología e Innovación – SIGP, debe ser coherente con la información registrada en el Anexo 1. Carta Unificada de Aval y Compromiso Institucional. (Este requisito NO es subsanable).

- Roles Entidades
 - EJECUTOR
- Roles del Personal
 - INVESTIGADOR PRINCIPAL
 - PERSONAL DE APOYO
 - PROFESIONAL CON TÍTULO DE MAESTRÍA O ESTUDIANTE DE UN PROGRAMA DE DOCTORADO CON REGISTRO CALIFICADO VIGENTE DEL MINISTERIO DE EDUCACIÓN NACIONAL
- Temáticas de la Convocatoria
 - BIOECONOMÍA Y TERRITORIO
 - DERECHO HUMANO A LA ALIMENTACIÓN
 - TRANSICIÓN ENERGÉTICA
 - SOBERANÍA SANITARIA
 - CIENCIA PARA LA PAZ
- Líneas Temáticas de la Convocatoria
 - Temáticas Adicionales ECOS NORD (Francia)
 - Sublíneas Temáticas
 - Ciencia y Tecnologías Cuánticas
 - Ciencias de la Educación
 - Ciencias del Espacio
 - Ciudades Inteligentes
 - Inteligencia Artificial
 - Temáticas Adicionales DAAD (Alemania)

Sublineas Temáticas

- Ciencia y Tecnologías Cuánticas
- Ciencias de la Educación
- Ciencias del Espacio
- Inteligencia Artificial

- Información complementaria
 - 1 MODALIDAD
 - 2 Área Metropolitana por la que se postula:
 - 3 Departamento/Municipio (Modalidad Regional)
 - 4 Enfoque Diferencial Participantes
- Mensajes Personalizados

4. Pruebas funcionales

- Se realizan pruebas funcionales para verificar que la parametrización se haya realizado correctamente.
- Se identifican posibles ajustes o correcciones antes de notificar al usuario funcional.

5. Notificación al usuario funcional

- Se informa al usuario funcional que la parametrización ha finalizado.
- Se solicita que realicen pruebas para validar el correcto funcionamiento.

Mesa de Servicios Incidente

Software Builder, SB [Cerrar sesión](#)

Mostrar la actividad para Solicitud 155735

Número de Solicitud	Resumen de Solicitud	
155735	parametrización sigo	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
22/04/2025 19:48:12	22/04/2025 19:48:12	00:02:46
Descripción del usuario		
Notificación manual		
Buenas noches.		

Se finalizó la configuración de la convocatoria "CONVOCATORIA DE MOVILIDAD ACADÉMICA CON EUROPA 2025" de acuerdo con el formato de parametrización de convocatoria diligenciado por ustedes. En estos momentos se encuentra la convocatoria como invitación especial para poder realizar las pruebas, es decir, el acceso es por un link diferente al público y con credenciales de invitado para poder hacer el pre-registro.

Revisando la configuración, encontramos que ustedes solicitan registrar un monto máximo permitido por propuesta según la Movilidad ECOS NORD de Francia o DAAD - Servicio Alemán de Intercambio Académico, algo que actualmente el sistema no permite parametrizar, en cambio, permite configurar monto mínimo permitido por propuesta. En base a lo anterior, si se necesita la parametrización como ustedes quieren, sería un nuevo desarrollo a ser implementado en el sistema y podrían solicitarlo formalmente a la OTSI.

Los datos de acceso son los siguientes:

URL: <https://plataformasig.minciencias.gov.co:7003/FormularioProyectos/faces/proyecto/invitacionEspecial.xhtml>

MODALIDAD NACIONAL

Usuario: INV7582025

Clave: tUanH@eC

Convocatoria: "MOV-2025 CONVOCATORIA DE MOVILIDAD ACADÉMICA CON EUROPA 2025 MODALIDAD NACIONAL"

MODALIDAD REGIONAL

Usuario: INV7592025

Clave: hbsPFYSK

Convocatoria: "MOV-2025 CONVOCATORIA DE MOVILIDAD ACADÉMICA CON EUROPA 2025 MODALIDAD REGIONAL"

Quedo atentos a las pruebas realizadas por el usuario funcional.

Cordialmente,

Ing. Leonardo Suárez Baldovino
Software Builders Ltda.
lasb@softwarebuilders.co | www.softwarebuilders.co
Carrera 700 # 51 - 57

6. Pruebas y observaciones del usuario

- El usuario funcional realiza pruebas en el sistema.

7. Validación final del usuario

- Se recibe la aprobación del usuario funcional para confirmar que la convocatoria está parametrizada correctamente.

Re: Solicitud 155735 Notificación manual - Convocatoria MOVILIDAD ACADÉMICA CON EUROPA 2025 Disponible Pruebas

De: Carolina Múnera Camacho <cmunera@minciencias.gov.co>

Destinatario: mesadeservicios@minciencias.gov.co

Cc: Nidia Janneth León Cabeza <njleon@minciencias.gov.co>, Leonardo Suarez <lasb@softwarebuilders.co>, María Camila Amaris Mendoza <mcamaris@minciencias.gov.co>, Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>, Jean Paul Moreno Triviño <jpmoreno@minciencias.gov.co>, Virginia Hernández Caicedo <vhernand@minciencias.gov.co>

Fecha: 2025-04-23 18:04

[Resumen](#) [Cabeceras](#) [Solo texto](#) [Descargar todos los adjuntos](#)

M801PR01F03 Parametrización y Pruebas formulario de mecanismos V.01 (1).pdf (~93 KB) M801PR01F03 Parametrización y Pruebas formulario de mecanismos V.01 (1).xlsx (~187 KB)

Señores OTSI

Cordial saludo. Adjunto se remite el formato M801PR01F03 - Parametrización y Pruebas Formulario de Mecanismos V.01 posterior a la realización de las pruebas de parametrización de la Convocatoria No. 965 Movilidad Académica en Europa 2025, por favor liberar el formulario del ambiente de pruebas y dejarlo en producción al público hoy 23 de abril de 2025.

Cordialmente,

Carolina Múnera Camacho
Contratista
Equipo de Seguimiento Convenios y Mecanismos
cmunera@minciencias.gov.co
Teléfono: 601 6258480 Ext. 5306
Av. Calle 26 No. 57 - 83
Torre 8, pisos 2 al 6
Bogotá, Colombia
www.minciencias.gov.co

8. Publicación y puesta en producción

- Una vez recibido el aval del usuario, se procede a la publicación de la convocatoria en el sistema.
- Se garantiza que esté disponible para los usuarios finales.
- Se finaliza la parametrización de la Convocatoria 965-2025 Movilidades Académicas con Europa 2025 en el sistema SIGP.

Solicitud 155965

Asunto		Tiempo total de actividad	
Parametrización convocatoria JÓVENES INVESTIGADORES		00:28:57	
Descripción			
Cordial saludo. De manera atenta se solicita apoyo en la parametrización en SIGP de la convocatoria “JÓVENES INVESTIGADORES E INNOVADORES PARA FORTALECER EL CONOCIMIENTO DE FUENTES DE ENERGÍA PARA LA TRANSICIÓN ENERGÉTICA EN COLOMBIA: PROSPECTIVA DE GAS Y ALMACENAMIENTO GEOLÓGICO DE CO2”			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
07/04/2025 16:16:43	13/05/2025 17:39:34		

Acción Realizada 155965

1. Revisión de los términos de referencia

- Se analizan los términos de referencia de la **Convocatoria 967-2025 CONVOCATORIA JÓVENES INVESTIGADORES EN TRANSICIÓN ENERGÉTICA: GAS Y ALMACENAMIENTO GEOLÓGICO DE CO2** para comprender los requisitos y lineamientos.
- Se identifican las condiciones específicas que deben ser parametrizadas en el sistema.

2. Análisis e identificación de la información

- Se definen los datos que deben ser parametrizados en la convocatoria.
- Se establece la estructura del formulario de proyectos para capturar la información requerida.

3. Diligenciamiento de la matriz de parametrización

- Se completa la **matriz de parametrización**, documentando cada campo, regla de validación y configuración necesaria en la convocatoria.
- Se parametriza lo siguiente:
 - Financiaciones
 - DIRECCIÓN DE VOCACIONES Y FORMACIÓN
 - Menús
 - PRY - Tipo Personal
 - PRY - Palabras Clave
 - PRY - Global Total
 - PRY - Función General Proyectos
 - PRY – Validaciones
 - PRY – Globales Entidad
 - PRY - Grupos
 - PRY - Generalidades Personal
 - PRY - Roles Entidad
 - PRY - Proyecto
 - PRY - Cambiar Clave
 - PRY - Requisitos
 - PRY - Presupuesto
 - PRY - Rubros
 - PRY - Información General Proyecto
 - PRY – Cronograma
 - PRY - Descripciones
 - PRY – Personal
 - PRY – Documentos
 - PRY - Información Adicional
 - PRY – Áreas Temáticas (Selección múltiple)
 - PRY - Productos
 - PRY - Productos - Desarrollo tecnológico e innovación

- PRY - Productos - Apropiación social del conocimiento
- PRY - Productos - Generación de nuevo conocimiento
- PRY - Productos – Formación Recurso Humano

- Rubros
 - BECA PASANTÍA JOVEN INVESTIGADOR E INNOVADOR ESTUDIANTES DE PREGRADO 1
 - BECA PASANTÍA JOVEN INVESTIGADOR E INNOVADOR ESTUDIANTES DE PREGRADO 2 MATERIALES
 - BECA PASANTÍA JOVEN INVESTIGADOR E INNOVADOR PROFESIONALES GRADUADOS 1 MATERIALES
 - BECA PASANTÍA JOVEN INVESTIGADOR E INNOVADOR PROFESIONALES GRADUADOS 2 MATERIALES
 - CONTRAPARTIDA
 - FINANCIACIÓN DEL PROYECTO JOVEN INVESTIGADOR E INNOVADOR ESTUDIANTE DE PREGRADO 1
 - FINANCIACIÓN DEL PROYECTO JOVEN INVESTIGADOR E INNOVADOR ESTUDIANTE DE PREGRADO 2
 - FINANCIACIÓN DEL PROYECTO JOVEN INVESTIGADOR E INNOVADOR PROFESIONAL GRADUADO 1
 - FINANCIACIÓN DEL PROYECTO JOVEN INVESTIGADOR E INNOVADOR PROFESIONAL GRADUADO 2
 - SEMILLERO DE INVESTIGACIÓN

- Descripciones
 - 1 RESUMEN EJECUTIVO
 - 2 IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA
 - 3 ANTECEDENTES
 - 4 JUSTIFICACIÓN
 - 5 MARCO CONCEPTUAL
 - 6 POBLACIÓN A IMPACTAR (INDICANDO SU ENFOQUE DIFERENCIAL O INTERSECCIONAL, SI APLICA)
 - 7 OBJETIVO GENERAL
 - 8 OBJETIVOS ESPECÍFICOS
 - 9 METODOLOGÍA
 - 10 SOSTENIBILIDAD
 - 11 RESULTADOS ESPERADOS

- 12 APROPIACIÓN SOCIAL DEL CONOCIMIENTO
- 13 IMPACTOS
- 14 RIESGOS
- 15 BIBLIOGRAFÍA
- Campos (Se ocultan)
 - CADENA PRODUCTIVA
 - CIUDAD DE FORMACION
 - CLASIFICACIÓN
 - CODIGO CIU
 - DURACIÓN PROYECTO HORAS
 - ESTADO EMPRESA
 - LINEA TEMATICA
 - NOMBRE CLÚSTER
 - PERMISOS
 - PROGRAMA NACIONAL DE CTOL PRINCIPAL
 - PROPÓSITO GENERAL
 - SECTOR
 - TEMÁTICAS ORIENTADORAS POR FOCO ESTRATÉGICO
 - TIPO DE FINANCIACIÓN
 - TIPO PERSONA
 - VALOR INSCRIPCION
- Financiadoras
 - MINCIENCIAS
- Requisitos
 - 6.13. Estado Académico de los Jóvenes Investigadores e Innovadores en calidad de graduados de programas de pregrado en Geología, Geociencias, Ingeniería Geológica, Ingeniería de Petróleos, Ingeniería Ambiental, ciencias de la Tierra o programas afines, cuyas disciplinas estén directamente relacionadas con estas áreas y con la investigación del conocimiento geocientífico en el sector de hidrocarburos. Deberán adjuntar uno de los siguientes documentos: (i) copia del diploma y/o acta de grado, o (ii) certificado donde se indique que ha culminado sus estudios y requisitos necesarios para la obtención del título y que solo se encuentra pendiente por ceremonia de grado. En caso de no cumplir con

este requisito, la propuesta será rechazado. (Este requisito SI es subsanable).

- 6.12. Estado Académico de los jóvenes investigadores e innovadores en calidad de estudiantes de pregrado en áreas de geociencias o ciencias de la tierra, e ingeniería de petróleos, cuyas disciplinas tengan relación directa con la generación de conocimiento geocientífico del subsuelo colombiano, deben haber cursado mínimo el 70% de los créditos académicos. Para acreditar este requisito, se debe adjuntar un único PDF con certificados emitidos por la institución de formación, indicando total de créditos del programa y los cursados al momento de postulación. Se aceptan documentos electrónicos oficiales; no se permiten capturas de pantalla. Si no cumple estas condiciones, será causal de rechazo. . (Este requisito SI es subsanable).
- 6.11. Estado académico de los integrantes del semillero de investigación. Los jóvenes integrantes del semillero deben ser estudiantes activos que se encuentren cursando cuarto semestre en adelante. Para acreditar este requisito se debe adjuntar en un único PDF los certificados de cada uno de los jóvenes, expedidos por la Institución de Educación Superior (IES) a la que están vinculados, en donde se indique el semestre académico que están cursando al momento de su postulación. (Este requisito SI es subsanable).
- 6.10. Adjuntar documento que acredite la fecha de constitución del Semillero de Investigación debidamente firmado por el representante legal de la entidad del Sistema Nacional de Ciencia, Tecnología e Innovación o quien haga sus veces (vicerrector académico, el decano de la facultad, el director de investigación o la instancia correspondiente), a la que esté vinculado el semillero. (Este requisito SI es subsanable).
- 6.9. Enfoque diferencial y/o interseccional (cuando aplique) Anexo 4: Este requisito es opcional y no es de carácter inhabilitante. Los integrantes del semillero de investigación, los jóvenes investigadores e innovadores estudiantes de pregrado, que se encuentren realizando su trabajo de grado o que cuenten con la aprobación para iniciar su ejecución y los jóvenes graduados que pertenezcan a población con enfoque diferencial y/o interseccional en alguna de las categorías que se especifica en la Tabla No. 1 Categorías de enfoque diferencial e interseccional para los Jóvenes Investigadores e innovadores”, del Anexo 4, deberán adjuntar en el SIGP en un solo PDF la documentación de

todos los jóvenes que cumplen con alguna de las categorías de enfoque diferencial y/o interseccional, de acuerdo con lo especificado en la columna “Descripción” de la Tabla No. 1, para poder optar por los puntos correspondientes al enfoque diferencial o enfoque interseccional. (Este requisito SI es subsanable).

- 6.8. Edad de los jóvenes investigadores e innovadores: Los integrantes del semillero de investigación, los jóvenes investigadores e innovadores estudiantes de pregrado, que se encuentren realizando su trabajo de grado o que cuenten con la aprobación para iniciar su ejecución y los jóvenes graduados deben ser colombianos y tener máximo 28 años de edad cumplidos al 31 de diciembre de 2025. Para acreditar este requisito se debe adjuntar en un único PDF las copias legibles de las cédulas de ciudadanía por ambas caras de todos los jóvenes investigadores e innovadores que hacen parte de la propuesta de investigación, desarrollo tecnológico y/o innovación. En caso de no cumplir con este requisito, la propuesta será rechazada. (Este requisito SI es subsanable).
- 6.7. El Tutor Principal debe contar con título en formación de alto nivel (maestría y/o doctorado) y debe tener su CvLAC registrado y/o actualizado, para realizar la inscripción en el Sistema Integrado de Gestión de Proyectos (SIGP). Para la verificación se debe adjuntar copia del diploma y/o acta de grado. (Este requisito SI es subsanable).
- 6.6. El Tutor Principal debe ser ciudadano colombiano. Para la verificación se debe adjuntar copia de la cédula de ciudadanía legible por ambas caras. (Este requisito SI es subsanable).
- 6.5. Anteproyectos de trabajo de grado de los estudiantes de pregrado: Adjuntar en un único PDF los anteproyectos de trabajo de grado de los estudiantes de pregrado a vincular en la propuesta, con el documento de aprobación firmado por el comité de investigación, comité de trabajos de grado, dirección de programa, facultad, decanatura o la instancia correspondiente (Este requisito SI es subsanable).
- 6.4. Carta de presentación de los jóvenes investigadores e innovadores y los tutores (Anexo No. 3) a vincular en la propuesta de investigación, desarrollo tecnológico y/o innovación, donde se refleje: i). Perfiles de jóvenes investigadores e innovadores y sus respectivos datos de contacto. ii). Justificación de la selección de cada Joven Investigador e Innovador para el desarrollo de la propuesta de investigación, desarrollo tecnológico y/o innovación. iii). El plan de actividades de investigación y

los resultados esperados de cada joven investigador e innovador en el marco de la propuesta. iv). Perfil del tutor, producción académica y científica, trayectoria en investigación y actividades para el acompañamiento de los jóvenes investigadores e innovadores. Se debe adjuntar el Anexo No. 3 Carta de presentación de los jóvenes investigadores e innovadores y los tutores firmada por el Representante Legal y el Tutor principal (Este requisito SI es subsanable sin modificar el contenido)

- 6.3. Entidad del Sistema Nacional de Ciencia, Tecnología e Innovación. La entidad del Sistema Nacional de Ciencia, Tecnología e Innovación (SNCTI), deberá estar legalmente constituida en Colombia con al menos doce (12) meses previos de la fecha de cierre de la convocatoria, información que se verificará con el Registro Único Empresarial (RUES). Las entidades que no estén registradas en el RUES deberán adjuntar: certificado de existencia y representación legal expedido por la autoridad competente o por el Ministerio de Educación Nacional, con fecha de expedición, no mayor a 3 meses al cierre de la convocatoria. (Este requisito SI es subsanable).
- 6.2. Carta unificada de aval y compromiso institucional (Anexo No. 1). Se debe adjuntar la Carta Unificada de Aval y Compromiso Institucional, debidamente diligenciada y firmada tanto por el representante legal o el apoderado de la entidad. En caso de que la firma corresponda a un apoderado de manera obligatoria se debe anexar a cada una de las propuestas presentados el documento que lo faculte para la representación legal de la entidad. (Este requisito SI es subsanable). Al diligenciar el Anexo No. 1 para el cumplimiento de este requisito el proponente indicará: Los compromisos de la Entidad ejecutora, la información del tutor principal con formación de alto nivel (maestría o doctorado) que acompañará a los jóvenes investigadores e innovadores y sus compromisos, así como también los compromisos adquiridos por los jóvenes investigadores e innovadores para el desarrollo de la propuesta.
- 6.1. Dirigido a: Cumplir con el “Dirigido a” de la presente convocatoria. La información registrada en el Sistema Integrado de Gestión de Proyectos del Ministerio de Ciencia, Tecnología e Innovación – SIGP, debe coincidir con la información registrada en el Anexo No. 1. Carta

Unificada de Aval y Compromiso Institucional. (Este requisito NO es subsanable).

- Roles Entidades
 - EJECUTOR
- Roles del Personal
 - JOVEN INVESTIGADOR E INNOVADOR (ESTUDIANTE DE PREGRADO)
 - JOVEN INVESTIGADOR E INNOVADOR (PROFESIONAL GRADUADO)
 - JOVEN INVESTIGADOR E INNOVADOR VINCULADO AL SEMILLERO DE INVESTIGACIÓN
 - OTROS TUTORES
 - TUTOR PRINCIPAL
- Temáticas
 - Caracterización geológica y/o evaluación de oportunidades exploratorias y/o de yacimientos de hidrocarburos, especialmente de gas, en cuencas colombianas.
 - Uso de nuevas tecnologías para el desarrollo de proyectos de exploración y desarrollo de hidrocarburos tales como la ciencia de datos, inteligencia artificial y/o Machine Learning.
 - Investigación, caracterización y evaluación de los procesos de captura, uso y almacenamiento geológico eficiente de CO₂ – CCUS.
 - Investigación en áreas de las geociencias y afines, cuyos productos incorporen nuevos paradigmas, investigación de punta y propuestas novedosas que contribuyan al aumento del factor de recobro en hidrocarburos.
- Mensajes Personalizados
- Información Complementaria
 - 1 LÍNEAS TEMÁTICAS Cuenca Sedimentaria (NMG) Si
Lista
 - 2 ENFOQUE DIFERENCIAL BENEFICIARIOS DE LA PROPUESTA

4. Pruebas funcionales

- Se realizan pruebas funcionales para verificar que la parametrización se haya realizado correctamente.

5. Notificación al usuario funcional

- Se informa al usuario funcional que la parametrización ha finalizado.
- Se solicita que realicen pruebas para validar el correcto funcionamiento.

Mostrar la actividad para Solicitud 155965		
Número de Solicitud	Resumen de Solicitud	
155965	Parametrización convocatoria JÓVENES INVESTIGADORES	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
02/05/2025 10:34:11	02/05/2025 10:34:11	00:04:01
Descripción del usuario		
Notificación manual		
Buenos días.		
Se finalizó la configuración de la convocatoria de JÓVENES INVESTIGADORES E INNOVADORES PARA FORTALECER EL CONOCIMIENTO DE FUENTES DE ENERGÍA PARA LA TRANSICIÓN ENERGÉTICA EN COLOMBIA: PROSPECTIVA DE GAS Y ALMACENAMIENTO GEOLÓGICO DE CO ₂ . En estos momentos se encuentra la convocatoria como invitación especial para poder realizar las pruebas, es decir, el acceso es por un link diferente al público y con credenciales de invitado para poder hacer el pre-registro.		
Los datos de acceso son los siguientes:		
Usuario: INV7642025		
Clave: invg8l8m		
URL: https://plataformasig.minciencias.gov.co:7003/FormularioProyectos/faces/proyecto/invitacionEspecial.xhtml		
* Seleccionar la convocatoria " COD-2025 CONV JÓVENES INVESTIGADORES EN TRANSICIÓN ENERGÉTICA: GAS Y ALMACENAMIENTO GEOLÓGICO DE CO ₂ ".		
Quedo atentos a las pruebas realizadas por el usuario funcional.		
...		
Cordialmente,		
Ing. Leonardo Suárez Baldovino		
Software Builders Ltda.		
✉ lasb@softwarebuilders.co www.softwarebuilders.co		
📍 Carrera 700 # 51 - 57		
Solicitud 155965 Notificación manual.		
Asignado a: Software Builder, SB		
Cliente: Múnera Camacho, Carolina		
Descripción: Cordial saludo. De manera atenta se solicita apoyo en la parametrización en SIGP de la convocatoria "JÓVENES INVESTIGADORES E INNOVADORES PARA FORTALECER EL CONOCIMIENTO DE FUENTES DE ENERGÍA PARA LA TRANSICIÓN ENERGÉTICA EN COLOMBIA: PROSPECTIVA DE GAS Y ALMACENAMIENTO GEOLÓGICO DE CO ₂ ".		

6. Pruebas y observaciones del usuario

- El usuario funcional realiza pruebas en el sistema.
- Se reciben observaciones o solicitudes de ajustes basadas en los resultados de las pruebas.

Observaciones Pruebas de parametrización SIGP Convocatoria ANH - Jóvenes Investigadores e Innovadores

De: Carolina Múnera Camacho <cmunera@ciencias.gov.co>
Destinatario: Leonardo Suarez <lasb@softwarebuilders.co>, Harold Alexander Mosler <hmosler@ciencias.gov.co>
Cc: Nidia Jarama Lario <nlar@ciencias.gov.co>, María Camila Amara Méndez <mcamara@ciencias.gov.co>, Juan Paul Moreno Trujillo <jpmoreno@ciencias.gov.co>
Fecha: 2023-05-06 13:59
Resumen | Conversar | Sólo texto

Cordial saludo,

Se informa que desde la Dirección de Vocaciones y Formación se realizaron las pruebas de parametrización en el SIGP de la convocatoria "Jóvenes investigadores e innovadores para fortalecer el conocimiento de fuentes de energía para la transición energética en Colombia: prospectiva de gas y almacenamiento geológico de CO2".

En términos generales, el formulario está acorde a lo solicitado y las observaciones y/o cambios solicitados, son en pro de mejorar la funcionalidad del formulario al momento de registro por parte de los proponentes.

A continuación se señalan las observaciones y/o ajustes requeridos:

PERSONAL
Las opciones de:
"Joven Investigador e Innovador (Pregrado)" Debe quedar: "Joven Investigador e Innovador (Estudiante de Pregrado)"
"Joven Investigador e Innovador (Profesional)" Debe quedar: "Joven Investigador e Innovador (Profesional Graduado)"

Tipo de personal *

Seleccionar...

Seleccionar...

JOVEN INVESTIGADOR E INNOVADOR (PREGRADO)
JOVEN INVESTIGADOR E INNOVADOR (PROFESIONAL)
JOVEN INVESTIGADOR E INNOVADOR VINCULADO AL SEMILLERO DE INVESTIGACIÓN
OTROS TUTORES
TUTOR PRINCIPAL

Personal - Generalidades Personas: Debe quedar parametrizado que se requiere validación del CVLac de TODO el personal. De no validar, no debe permitir continuar:

Presupuesto

Para todos los rubros dejar delimitado el valor máximo señalado en el formato de parametrización desde el momento de diligenciamiento del rubro y no al momento de "Validar y Enviar".

DESCRIPCIONES

Eliminar los campos de "Cronograma" y "Productos Esperados" en tanto hay secciones independientes para ello.

CONTRAPARTIDA

Debe quedar parametrizado que la contrapartida debe ser del 3% del valor total solicitado.

Cordialmente,

Carolina Múnera Camacho
Comunicación
Equipo de Seguimiento Convencios y Mecanismos
cmunera@ciencias.gov.co
Teléfono: 601 626480 Ext. 5306
Av. Calle 26 No. 97-83
Piso 9 - Torre A - 214

7. Ajustes y nueva notificación

- Se aplican los ajustes solicitados por el usuario funcional.
- Se notifica nuevamente al usuario para que realice una segunda revisión.

Mesa de Servicios Incidente Software Builder, SB Cerrar sesión

Archivo | Ver | Buscar | Ventana | Ayuda |  

Mostrar la actividad para Solicitud 155965

Número de Solicitud	Resumen de Solicitud
155965	Parametrización convocatoria JÓVENES INVESTIGADORES
Tipo de actividad	Analista
Notificación manual	Software Builder, SB
Marca de tiempo del sistema	Fecha de la actividad
12/05/2025 16:51:14	12/05/2025 16:51:14
Descripción del usuario	¿Interno?
Notificación manual	No
Buenas tardes Carolina Múnera,	Tiempo invertido
Cordial saludo,	00:10:42

En atención a las observaciones recibidas sobre la convocatoria "Jóvenes investigadores e innovadores para fortalecer el conocimiento de fuentes de energía para la transición energética en Colombia: prospectiva de gas y almacenamiento geológico de CO2", nos permitimos informar que se realizaron los ajustes solicitados en el formulario del SIGP, conforme al detalle recibido.

No obstante, es importante tener en cuenta lo siguiente:

- Validación de los rubros presupuestales: El sistema actualmente valida los topes máximos parametrizados únicamente en el momento de ejecutar la opción "Validar y Enviar", por lo cual no es posible realizar esta validación en tiempo real durante el diligenciamiento individual de cada rubro.
- Contrapartida del 3%: La validación automática de la contrapartida, como se encuentra actualmente implementada en la plataforma, se realiza comparando contra el valor total del proyecto, y no contra el valor total solicitado a Minciencias. Por tanto, si se requiere una validación distinta (por ejemplo, respecto al valor financiado), sería necesario un desarrollo adicional.

Quedamos atentos a cualquier otra observación o necesidad de ajuste adicional.

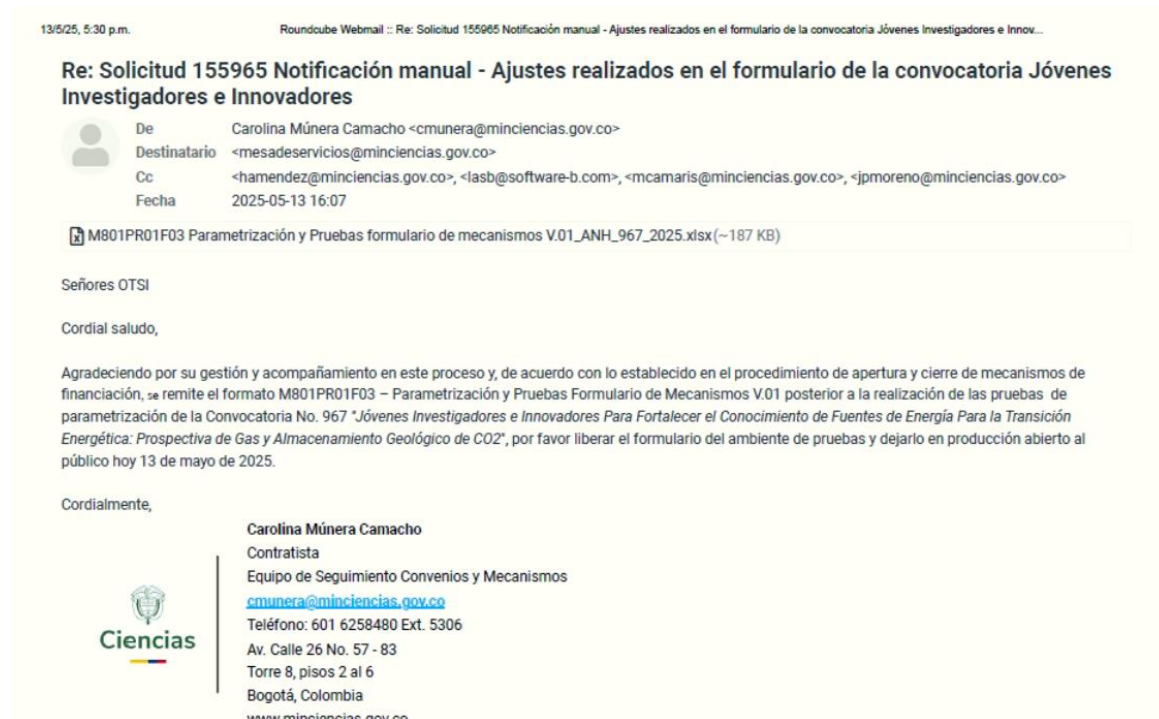
Cordialmente,

Ing. Leonardo Suárez Baldovino
Software Builders Ltda.
lasb@softwarebuilders.co | www.softwarebuilders.co
Carrera 700 # 51 - 57

Solicitud 155965 Notificación manual.
Asignado a: Software Builder, SB
Cliente: Múnera Camacho, Carolina
Descripción: Cordial saludo. De manera atenta se solicita apoyo en la parametrización en SIGP de la convocatoria "JÓVENES INVESTIGADORES E INNOVADORES PARA FORTALECER EL CONOCIMIENTO DE FUENTES DE ENERGÍA PARA LA TRANSICIÓN ENERGÉTICA EN COLOMBIA: PROSPECTIVA DE GAS Y ALMACENAMIENTO GEOLÓGICO DE CO2"

Validación final del usuario

- Se espera la aprobación del usuario funcional para confirmar que la convocatoria está parametrizada correctamente.



9. Publicación y puesta en producción

- Una vez recibido el aval del usuario, se procede a la publicación de la convocatoria en el sistema.
- Se garantiza que esté disponible para los usuarios finales.
- Se finaliza la parametrización de la Convocatoria “967-2025 CONVOCATORIA JÓVENES INVESTIGADORES EN TRANSICIÓN ENERGÉTICA: GAS Y ALMACENAMIENTO GEOLÓGICO DE CO2”

Parametrización de Rúbricas de Evaluación

Se atendieron las siguientes solicitudes relacionadas con la configuración y ajuste de convocatorias en la plataforma SIGP:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
156431	- Rubrica de Evaluación Convocatoria 953-2024 Comunidades Negras	Solucionado	Producción	[Ver detalle]
156652	- Rubrica de Evaluación Convocatoria 963-2025 Orquídeas	Solucionado	Producción	[Ver detalle]
157231	- Rubrica de Evaluación Convocatoria 953-2024 Comunidades Negras Ajustes 05_05_2025	Solucionado	Producción	[Ver detalle]

Solicitud 156431

Asunto		Tiempo total de actividad	
Procedimiento M801PR02		00:29:35	
Descripción			
De acuerdo con el procedimiento M801PR02 EVALUACIÓN DE PROGRAMAS, PROPUESTAS O PROYECTOS DE CTel, y en mi rol de líder de la convocatoria, comparto con ustedes el formato M801PR02MO8 RÚBRICA PARA EVALUACIÓN DE PROGRAMAS, PROPUESTAS O PROYECTOS DE CTel para avanzar en el paso de parametrizar el formulario de evaluación en la plataforma SIGP, junto con el cronograma aprobado.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
16/04/2025 11:06:44	12/05/2025 16:27:32	12/05/2025 16:08:27	12/05/2025 16:27:32

Acción Realizada 156431

- Se realiza la parametrización de la rúbrica de evaluación de la Convocatoria 953-2024 CONVOCATORIA PARA COMUNIDADES NEGRAS DE BOGOTÁ, CAQUETÁ, CÓRDOBA, MAGDALENA Y CHOCÓ con éxito.
- Se notifica al usuario funcional Alejandra Acosta Villamil de que la rúbrica se encuentra para pruebas y se le indica el proceso de como probarla

Solicitud 156652

Asunto		Tiempo total de actividad	
parametrizacion sigp		00:27:06	
Descripción			
Cordial saludo.			
De manera atenta, se solicita el apoyo de la OTSI en la parametrización en SIGP, de la rúbrica de evaluación de la Convocatoria No. 963 de 2025 "Orquídeas Mujeres en Inteligencia Artificial, Ciencia y Tecnologías Cuánticas".			
Dado el cronograma de la convocatoria, se requiere contar con la rúbrica parametrizada para el jueves 24 de abril, día en el cual se hacen las pruebas de parametrización para aprobación el 25 de abril.			
La rúbrica fue aprobada por la Directora de Vocaciones y Formación - Diana Rúa.			
Adjunto se remite rúbrica en formato excel definido para tal fin.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
23/04/2025 11:23:00	13/06/2025 14:35:05	12/06/2025 15:35:05	13/06/2025 14:35:05

Acción Realizada 156652

- Se realiza la parametrización de la rúbrica de evaluación de la Convocatoria 963-2025 Conv "Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 2025" con éxito
- Se notifica al usuario funcional Carolina Múnera de que la rúbrica se encuentra para pruebas y se le indica el proceso de como probarla

Solicitud 157231

Asunto		Tiempo total de actividad	
Convocatoria 953		00:06:29	
Descripción			
Una vez realizadas las pruebas de la rúbrica de la convocatoria 953, se solicita realizar los ajustes necesarios, para lo cual se solicita un espacio para poder socializar y realizar los ajustes en el menor tiempo posible.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
02/05/2025 18:14:43	08/05/2025 11:14:37	07/05/2025 12:14:37	08/05/2025 11:14:37

Acción Realizada 157231

- Se realizan los ajustes a la parametrización de la rúbrica de evaluación de la Convocatoria 953-2024 CONVOCATORIA PARA COMUNIDADES NEGRAS DE BOGOTÁ, CAQUETÁ, CÓRDOBA, MAGDALENA Y CHOCÓ con éxito
- Se notifica al usuario funcional Alejandra Acosta Villamil de que la rúbrica se encuentra nuevamente para pruebas y se le indica el proceso de como probarla

Generación de Reportes

Se gestionaron diversas solicitudes de generación de reportes, asegurando la disponibilidad de la información requerida:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
156776	- Reporte general de proyectos a evaluación Conv 953-2024 Comunidades Negras 25_04_2025	Solucionado		[Ver detalle]
156492	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 21_04_2025 10_00	Solucionado	-	[Ver detalle]
156658	- Reporte general de proyectos a evaluación Conv 963-2025 Orquídeas 23_04_2025	Solucionado	-	[Ver detalle]
156675	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 23_04_2025 15_00	Solucionado	-	[Ver detalle]
156763	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 25_04_2025 15_00	Solucionado	-	[Ver detalle]
156895	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 28_04_2025 10_00	Solucionado	-	[Ver detalle]
157006	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 29_04_2025 10_00	Solucionado	-	[Ver detalle]
157066	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 30_04_2025 11_00	Solucionado	-	[Ver detalle]
157021	- Reportes Cierre Convocatoria 037-2024 Macroproyectos Nacionales	Solucionado	-	[Ver detalle]
157144	- Reporte Conv 037-2024 Macroproyectos Nacional 30_04_2025	Solucionado	-	[Ver detalle]
157716	- Anonización Evaluadores Conv 953-2024 30_04_2025	Solucionado	-	[Ver detalle]

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
157164	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 02_05_2025 09_00	Solucionado	-	[Ver detalle]
157224	- Reporte Evaluación de la Conv 962-2024 SENAINNOVA 02_05_2025	Solucionado	-	[Ver detalle]
157235	- Reporte Proyectos banco elegibles con presupuesto consolidado Conv 958-2024	Solucionado	-	[Ver detalle]
157289	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 05_05_2025 11_00	Solucionado	-	[Ver detalle]
177391	- Reporte Proyectos Conv 038_040_042 del SGR 06_05_2025	Solucionado	-	[Ver detalle]
157443	- Reporte Evaluación de la Conv 962-2024 SENAINNOVA 06_05_2025	Solucionado	-	[Ver detalle]
157458	- Reporte general de proyectos Convocatoria 037-2024 Macroproyectos Nacionales 07_05_2025	Solucionado	-	[Ver detalle]
157495	- Anonimización Evaluadores Convocatoria 963-2025 Orquídeas	Solucionado	-	[Ver detalle]
157501	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 07_05_2025 12_00	Solucionado	-	[Ver detalle]
157516	- Reporte fecha de radicación proyectos 112926 y 112806	Solucionado	-	[Ver detalle]
157582	- Reporte Evaluación de la Conv 962-2024 SENAINNOVA 08_05_2025	Solucionado	-	[Ver detalle]
157588	- Reporte Auditoria Evaluación Proyectos 113501 y 113680	Solucionado	-	[Ver detalle]
157601	- Reporte Subsanación Requisitos Convocatoria 041 del SGR	Solucionado	-	[Ver detalle]
157654	- Reporte fecha de radicación proyectos 112926 y 112806	Solucionado	-	[Ver detalle]
157751	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 12_05_2025 12_00	Solucionado	-	[Ver detalle]

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
157834	- Anonimización Evaluadores Convocatoria BT 853-2019 Ventanilla Abierta	Solucionado	-	[Ver detalle]
157852	- Reporte Conv 964-2025 Formación Maestrías Pacifico Nariñense 13_05_2025 12_00	Solucionado	-	[Ver detalle]
157957	- Reporte Conv 964-2025 Formación Maestrías Pacifico Nariñense 14_05_2025 09_00	Solucionado	-	[Ver detalle]
157981	- Reporte Conv 964-2025 Formación Maestrías Pacifico Nariñense 14_05_2025 12_00	Solucionado	-	[Ver detalle]
157989	- Reportes Cierre Convocatoria 964-2025 Formación Maestrías Pacifico Nariñense	Solucionado	-	[Ver detalle]
157999	- Reportes Convocatorias Orquídeas 935-2023 y 948-2024	Solucionado	-	[Ver detalle]
158006	- Reporte Conv 964-2025 Formación Maestrías Pacifico Nariñense 14_05_2025 14_00	Solucionado	-	[Ver detalle]
158017	- Reporte Conv 964-2025 Formación Maestrías Pacifico Nariñense 14_05_2025 16_00	Solucionado	-	[Ver detalle]
158091	- Reporte Evaluación de la Conv 953-2024 Comunidades Negras	Solucionado	-	[Ver detalle]

Solicitud 156776

Asunto	Tiempo total de actividad
SOLICITUD TI Fwd: 953 Solicitud palabras clave e información	00:06:17
Descripción	

Con el fin de generar el reporte correspondiente y remitir el archivo por correo al Jefe de la OTSI, se ha creado este caso para dar seguimiento a las actividades de soporte realizadas por el proveedor Software Builders Ltda. (adjunto correo

Cordialmente,

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
24/04/2025 19:56:20	28/04/2025 10:41:37	25/04/2025 11:41:37	28/04/2025 10:41:37

Acción Realizada 156776

En atención a la solicitud recibida desde la OTSI, a través de la DDTI, referente al proceso de evaluación de la **Convocatoria 953-2024 Comunidades Negras**, se realizaron las siguientes acciones:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos registrados en la convocatoria, filtrando aquellos cuyo estado sea "REVISADO".
- La consulta incluyó los campos requeridos para el reporte general, contemplando además palabras claves y OCDE de conocimiento. * Gran área * Área * Disciplina

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente en la base de datos del aplicativo SIGP, aprovechando los permisos de acceso otorgados.
- Se generó un archivo Excel con los resultados obtenidos, estructurado según las necesidades del reporte.

3. Entrega del Reporte al Usuario Funcional

- Se notificó al usuario funcional Alejandra Acosta sobre la disponibilidad del reporte.
- El archivo fue enviado por correo electrónico para su respectiva revisión y uso.

Solicitud 156492

Asunto		Tiempo total de actividad	
Reporte Convocatoria 964		00:08:31	
Descripción			
<p>Buenos días.</p> <p>Solicito su amable colaboración para envío de un reporte de proyectos inscritos en la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 10:00am.</p> <p>Quedo atento a sus comentarios.</p> <p>Cordialmente,</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
21/04/2025 09:46:04	22/04/2025 10:00:03	21/04/2025 11:00:03	22/04/2025 10:00:03

Acción Realizada 156492

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 10:00 a.m. del 21/04/2025., se ejecutaron las siguientes actividades:

1. **Elaboración del Script de Consulta**

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 156658

Asunto	Tiempo total de actividad
Reportes SIGP	00:06:43
Descripción Cordial saludo. A partir de la entrega por parte de la DGR mediante memorando No. 20250002956M con la relación de propuestas que cumplen con los requisitos en la Convocatoria No. 963 de 2025 y pasan al proceso de evaluación, se requiere el reporte del SIGP de la totalidad de las propuestas postuladas identificando las que cumplen requisitos y las que no cumplen. Se requiere de la siguiente información registrada en la plataforma SIGP: Palabras claves de los programas, propuestas o proyectos que cumplen requisitos, así como la información relacionada con: gran área, área, disciplina, entidad, región, líneas temáticas, para definir el perfil de los evaluadores que se requiere e informar los	

filtros que deben hacerse para la selección anonimizada de los potenciales pares evaluadores.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
23/04/2025 11:39:43	24/04/2025 16:00:00	23/04/2025 18:05:26	24/04/2025 16:00:00

Acción Realizada 156658

En atención a la solicitud recibida desde la OTSI, a través de la DDTI, referente al proceso de evaluación de la **Convocatoria 963-2025 Orquideas**, se realizaron las siguientes acciones:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos registrados en la convocatoria, filtrando aquellos cuyo estado sea "REVISADO".
- La consulta incluyó los campos requeridos para el reporte general, contemplando además palabras claves y OCDE de conocimiento. * Gran área * Área * Disciplina

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente en la base de datos del aplicativo SIGP, aprovechando los permisos de acceso otorgados.
- Se generó un archivo Excel con los resultados obtenidos, estructurado según las necesidades del reporte.

3. Entrega del Reporte al Usuario Funcional

- Se notificó al usuario funcional Carolina Munera sobre la disponibilidad del reporte.
- El archivo fue enviado por correo electrónico para su respectiva revisión y uso.

Solicitud 156675

Asunto		Tiempo total de actividad	
convocatoria 964		00:06:55	
Descripción			
<p>Buenas tardes,</p> <p>Solicito su amable colaboración para generar un reporte de la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 3:00pm del día de hoy.</p> <p>Muchas gracias.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
23/04/2025 13:41:43	24/04/2025 16:00:00	23/04/2025 17:53:30	24/04/2025 16:00:00

Acción Realizada 156675

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 03:00 p.m. del 23/04/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 156763

Asunto		Tiempo total de actividad	
Reporte SIGP		00:05:18	
Descripción			
<p>Buenas tardes,</p> <p>Solicito su amable colaboración para generar un reporte de la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 3:00pm del día de hoy.</p> <p>Muchas gracias.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
24/04/2025 16:34:57	28/04/2025 14:03:16	25/04/2025 15:03:16	28/04/2025 14:03:16

Acción Realizada 156763

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 03:00 p.m. del 25/04/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 156895

Asunto		Tiempo total de actividad	
Reportes SIGP		00:03:56	
Descripción			
Buenos días, Solicito su amable colaboración para generar un reporte de la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 10:00am del día de hoy. Muchas gracias.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

28/04/2025 08:53:17	29/04/2025 09:13:18	28/04/2025 10:13:18	29/04/2025 09:13:18
------------------------	------------------------	------------------------	------------------------

Acción Realizada 156895

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 10:00 a.m. del 28/04/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157006

Asunto	Tiempo total de actividad
Reportes SIGP	00:06:22
Descripción	

<p>Buenos días,</p> <p>Solicito su amable colaboración para generar un reporte de la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 10:00am del día de hoy.</p> <p>Muchas gracias.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
29/04/2025 09:40:30	30/04/2025 10:53:04	29/04/2025 11:53:04	30/04/2025 10:53:04

Acción Realizada 157006

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 10:00 a.m. del 29/04/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157066

Asunto		Tiempo total de actividad	
Reporte convocatoria 964		00:05:22	
Descripción			
<p>Buenos días,</p> <p>Solicito su amable colaboración para generar un reporte de la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 9:00am del día de hoy.</p> <p>Muchas gracias.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
30/04/2025 08:39:54	01/05/2025 10:33:06	30/04/2025 11:33:06	01/05/2025 10:33:06

Acción Realizada 157066

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 11:00 a.m. del 30/04/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157021

Asunto		Tiempo total de actividad	
Reportes SIGP		00:03:41	
Descripción			
<p>Buen día estimados compañeros:</p> <p>Teniendo en cuenta que el día de mañana 30 de abril cierra la convocatoria 037 del SGR, agradezco que una vez cierre el mecanismo se genere y remita un reporte con todos los campos parametrizados.</p> <p>Quedo atenta.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
29/04/2025 11:34:00	01/05/2025 16:00:00	30/04/2025 17:50:23	01/05/2025 16:00:00

Acción Realizada 157021

En cumplimiento de la solicitud recibida para la generación de reportes al cierre de la **Convocatoria 037-2024 Convocatoria 37 “Macroproyectos Nacionales”**, cuyo cierre inicial fue el 30 de abril de 2025, se realizaron las siguientes acciones:

1. Elaboración de Scripts de Consulta

- Se diseñaron scripts SQL para extraer la información correspondiente a la convocatoria con ID 11721.
- Las consultas permitieron generar los siguientes reportes requeridos por los usuarios funcionales:
 - Reporte general de la convocatoria.
 - Reporte de presupuesto de los proponentes (monto solicitado y contrapartida).
 - Reporte de contrapartidas discriminadas por entidad participante.
 - Reporte de personal vinculado a las propuestas.
 - Reporte de Enfoque Diferencial

2. Ejecución de Consultas y Generación de Archivos Excel

- Se ejecutaron los scripts directamente sobre la base de datos SIGP.
- Se generaron archivos Excel con los datos solicitados, estructurados para facilitar su análisis y revisión.

3. Entrega de los Reportes al Usuario Funcional

- Se notificó a la funcionaria **Ginneth Cajamarca** sobre la disponibilidad de los reportes.
- Los documentos fueron enviados en formato Excel conforme a los lineamientos establecidos.

Solicitud 157144

Asunto		Tiempo total de actividad	
Reporte Convocaotroria No. 37		00:04:41	
Descripción			
Cordial saludo. Por medio de la presente me permito solicitar reporte de los proyectos inscritos en el SIGP y de los radicados de la Convocatoria No. 37. Por favor copiar a los siguientes correos: lfircaurte@minciencias.gov.co, secretaria_tec_ocad@minciencias.gov.co y illancheros@minciencias.gov.co.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
30/04/2025 16:22:26	01/05/2025 16:00:00	30/04/2025 17:35:46	01/05/2025 16:00:00

Acción Realizada 157144

En atención a la solicitud de generación de reporte para la **Convocatoria 037-2024 Convocatoria 37 "Macroproyectos Nacionales"**, con corte del día 30/04/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 037-2024.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.

- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte a los usuarios funcionales **Luis Fernando Ricaurte**, **Iván Lancheros** y **secretaria técnica OCAD** para su uso.

Solicitud 157716

Asunto	Tiempo total de actividad
Generar reporte 953	00:04:27
Descripción <p>Buenas día Director Harold</p> <p>Espero que esté bien.</p> <p>En el marco de la implementación del procedimiento de evaluación de la convocatoria 953, por medio del presente solicito su apoyo con la base de datos de las propuestas de la convocatoria 953 que cumplieron con la totalidad de requisitos para la convocatoria del asunto, y extraer:</p> <p>Palabras claves</p> <p>gran área</p> <p>área</p> <p>disciplina</p> <p>entidad</p> <p>región</p> <p>líneas temáticas</p> <p>Quedo atenta a cualquier requerimiento.</p>	

Saludo!

--

Alejandra Margarita Acosta Villamil
Contratista
Dirección de Capacidades y Apropiación del Conocimiento
amacosta@minciencias.gov.co
Teléfono: 601 6258480 Ext.
Av. Calle 26 No. 57 - 83
Torre 8, pisos 2 al 6
Bogotá, Colombia
www.minciencias.gov.co

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
12/05/2025 07:49:55	12/05/2025 16:27:08	12/05/2025 09:42:54	12/05/2025 16:27:08

Acción Realizada 157716

En cumplimiento de la solicitud relacionada con la **Convocatoria 953-2024 CONVOCATORIA PARA COMUNIDADES NEGRAS DE BOGOTÁ, CAQUETÁ, CÓRDOBA, MAGDALENA Y CHOCÓ**, se llevaron a cabo las siguientes actividades:

1. Generación del Reporte de Evaluadores

- Se construyó un reporte con el **listado de evaluadores** que cumplen con los criterios establecidos en cuanto a:
 - **Grandes áreas**
 - **Áreas**
 - **Disciplinas de conocimiento**
 - **Palabras clave asociadas**

2. Entrega del Reporte

- El documento resultante fue **generado y enviado** al jefe de la OTSI, Harold Méndez, como parte de la gestión solicitada.

Solicitud 157164

Asunto		Tiempo total de actividad	
Reporte convocatoria 964		00:04:04	
Descripción			
<p>Buenos días,</p> <p>Solicito su amable colaboración para generar un reporte de la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 9:00am del día de hoy.</p> <p>Muchas gracias.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
02/05/2025 06:59:44	05/05/2025 08:00:54	02/05/2025 09:00:54	05/05/2025 08:00:54

Acción Realizada 157164

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 09:00 a.m. del 02/05/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157224

Asunto		Tiempo total de actividad	
Generar la base de datos		00:13:29	
Descripción			
Solicito la atenta colaboración de OTSI para generar la base de datos del consolidado de evaluaciones por cada evaluador y por cada propuesta con la totalidad de los campos diligenciados y el puntaje de cada criterio y subcriterio, incluyendo las calificaciones del tercer evaluador y el puntaje total aplicando las reglas indicadas en el paso 38 del procedimiento de Evaluación de Programas, Propuestas o Proyectos de CTel - M801PR02 V8. Quedamos atentos a los dos reportes señalados en este paso del procedimiento.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

02/05/2025 16:20:39	05/05/2025 16:00:00	02/05/2025 18:00:47	05/05/2025 16:00:00
------------------------	------------------------	------------------------	------------------------

Acción Realizada 157224

En atención a la solicitud relacionada con la **Convocatoria 962-2024 SENAINNOVA PARA FOMENTAR ACCESO Y ADOPCIÓN DE TECNOLOGÍAS AVANZADAS EN SECTOR AGROALIMENTAR**, con corte al día 02 de mayo de 2025 se realizaron las siguientes actividades:

1. Generación del Reporte de Evaluaciones

- Se construyó un **informe consolidado** que contiene los resultados de las **evaluaciones por criterio y por evaluador** correspondientes al cierre de la convocatoria.

2. Entrega del Reporte

- El documento fue **notificado y enviado** al usuario funcional **Andrea Pachón Garzón** para su análisis y gestión correspondiente.

Solicitud 157235

Asunto	Tiempo total de actividad
Convocatoria 958	00:07:15
Descripción	
<p>Saludo cordial Ing. Jefe Harold,</p> <p>Por medio de la presente, me dirijo a ustedes con el fin de solicitar formalmente un reporte sobre los proyectos asociados a la Convocatoria 958 "Convocatoria transformación de la planeación minero energética para la convergencia territorial y</p>	

la transición energética justa."

Agradecería que el reporte contenga la información especificada a continuación para cada proyecto incluido:

Nombre del proyecto

Código Registro

Título

Entidad/Persona responsable

Línea Temática a la que pertenece

Rubro financiado

Contrapartida en efectivo aportada

Contrapartida en especie aportada

Valor total del proyecto.

Esta información es requerida para dar continuidad a procesos de publicación de banco de proyectos financiables.

Agradezco de antemano su valiosa colaboración y la celeridad con la que puedan atender esta solicitud.

Atentamente,

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
04/05/2025 09:09:34	06/05/2025 10:34:39	05/05/2025 11:24:39	06/05/2025 10:34:39

Acción Realizada 157235

En atención a la solicitud de generación de reporte para la **Convocatoria 958-2024 CONV TRANSFORMACIÓN DE LA PLANEACIÓN MINERO ENERGÉTICA PARA LA CONVERGENCIA TERRITORIAL**, se ejecutaron las siguientes actividades:

1. **Elaboración del Script de Consulta**

- Se diseñó un script SQL para obtener los proyectos en estado REVISADO y el consolidado de montos solicitados en la convocatoria 958-2024.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Oscar Daniel Sánchez Plazas** para su uso.

Solicitud 157289

Asunto		Tiempo total de actividad	
SIGP - (Convocatoria 964)		00:05:12	
Descripción			
Buenos días,			
Solicito su amable colaboración para generar un reporte de la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 11:00am del día de hoy.			
Muchas gracias.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

05/05/2025 10:29:34	06/05/2025 11:21:11	05/05/2025 12:21:11	06/05/2025 11:21:11
------------------------	------------------------	------------------------	------------------------

Acción Realizada 157289

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 11:00 a.m. del 05/05/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157391

Asunto	Tiempo total de actividad
MATRIZ PROYECTOS PRESENTADOS CONVOCATORIA ...	00:08:30
Descripción	

-[[SOLICITUD TI - MATRIZ PROYECTOS PRESENTADOS CONVOCATORIA 38]]-

Cordial saludo,

Desde la oficina Jurídica, de Secretaría, Técnica del OCAD, y de acuerdo al desarrollo de las convocatorias del SGR en curso, se solicita la siguiente información referente a las convocatorias 38, 40, y 42. la cual se detalla a continuación:

Identificación del proyecto ID

Nombre del proyecto

Entidad Proponente

Integrantes de la alianza o aliados

Valor de contrapartida

Valor de aporte de SGR

Valor total del proyecto

Datos de Contacto representante legal de la entidad proponente

Correo electrónico de la entidad proponente

Estado del Proyecto.

Agradezco de antemano su apoyo.

Cordialmente.

--

Teresa del Pilar Ávila Garzón

Contratista

Despacho del Ministro – Secretaria Técnica OCAD

tdpavila@minciencias.gov.co <tdpavila@minciencias.gov.co>

Teléfono: 601 6258480 Ext.

Av. Calle 26 No. 57 - 83

Torre 8, pisos 2 al 6

Bogotá, Colombia

www.minciencias.gov.co

--

No imprimas si no es necesario. Seamos responsables con la protección del

<p>medio ambiente.</p> <p>Declinación de Responsabilidades - Disclaimer <https://storage.googleapis.com/efor-static/CLCS/AVISO%20DE%20CONFIDENCIALIDAD%20-%20DISCLAIMER.pdf></p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
06/05/2025 11:19:05	08/05/2025 09:08:30	07/05/2025 10:08:30	08/05/2025 09:08:30

Acción Realizada 157391

En atención a la solicitud de generación de reporte para la **Convocatoria 38, 40 y 42 del SGR**, se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener los proyectos y el consolidado de montos solicitados en las convocatorias.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Teresa del Pilar Ávila** para su uso.

Solicitud 157443

Asunto		Tiempo total de actividad	
Reporte SIGP		00:05:59	
Descripción			
Cordial saludo. Solicito la atenta colaboración de OTSI para generar para la convocatoria 962-2024 SENAINNOVA, la base de datos del consolidado de evaluaciones por cada evaluador y por cada propuesta con la totalidad de los campos diligenciados y el puntaje de cada criterio y subcriterio, incluyendo las calificaciones del tercer evaluador y el puntaje total aplicando las reglas indicadas en el paso 38 del procedimiento de Evaluación de Programas, Propuestas o Proyectos de CTel - M801PR02 V8. Quedamos atentos a los dos reportes señalados en este paso del procedimiento. Muchas gracias			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
06/05/2025 16:02:43	08/05/2025 08:32:14	07/05/2025 09:32:14	08/05/2025 08:32:14

Acción Realizada 157443

En atención a la solicitud relacionada con la **Convocatoria 962-2024 SENAINNOVA PARA FOMENTAR ACCESO Y ADOPCIÓN DE TECNOLOGÍAS AVANZADAS EN SECTOR AGROALIMENTAR**, con corte al día 06 de mayo de 2025 se realizaron las siguientes actividades:

1. **Generación del Reporte de Evaluaciones**
 - Se construyó un **informe consolidado** que contiene los resultados de las **evaluaciones por criterio y por evaluador** correspondientes al cierre de la convocatoria.
2. **Entrega del Reporte**

- El documento fue **notificado y enviado** al usuario funcional **Andrea Pachón Garzón** para su análisis y gestión correspondiente.

Solicitud 157458

Asunto		Tiempo total de actividad	
reporte general de la convocatoria 37-SGR		00:04:29	
Descripción			
<p>Cordial saludo Ingeniero Leonardo Suárez,</p> <p>Agradecemos de su colaboración, para generar un reporte general de la convocatoria 37-SGR de los proyectos en estado "RADICADO", con los siguientes caracteres:</p> <p>PROYECTO_ID TITULO AREA_TEMATICA AREA_CIENCIA GRAN_AREA_OCDE AREA_OCDE ROL ENTIDAD (INDIVIDUAL) NOMBRE_CONTACTO CORREO_CONTACTO TELEFONO_CONTACTO RESUMEN_EJECUTIVO VALOR_TOTAL_PROPOSTA DEMANDA_TERRITORIAL MONTO_CONTRAPARTIDA_EFECTIVO MONTO_CONTRAPARTIDA_ESPECIE MONTO_SOLICITADO_SGR VALOR_SEGUIMIENTO ENTIDAD DIRECCION_ELECTRONICA TAMAÑO_EMPRESA REPRESENTANTE_LEGAL DIRECCION TELEFONO_PRIMER_CONTACTO CARGO_PRIMER_CONTACTO TELEFONO_PRIMER_CONTACTO CORREO_PRIMER_CONTACTO SEGUNDO_CONTACTO CARGO_SEGUNDO_CONTACTO TELEFONO_SEGUNDO_CONTACTO CORREO_SEGUNDO_CONTACTO INVESTIGADOR_PRINCIPAL IDENTIFICACION_INVPPAL SEXO_INVPPAL CORREO_INVPPAL PALABRAS_CLAVES GRUPOS CLASIFICACION LUGAR_EJECUCION_PROYECTO LINEA TEMATICA SUBLINEATEMATICA DEPART SOLICITAN RECURSOS</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

06/05/2025 17:01:05	08/05/2025 16:00:00	07/05/2025 18:14:00	08/05/2025 16:00:00
------------------------	------------------------	------------------------	------------------------

Acción Realizada 157458

En atención a la solicitud recibida desde la OTSI, a través de la DDTI, referente al proceso de evaluación de la **Convocatoria 037-2024 Macroproyectos Nacionales**, se realizaron las siguientes acciones:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos registrados en la convocatoria, filtrando aquellos cuyo estado sea "RADICADO".
- La consulta incluyó los campos requeridos para el reporte general, contemplando además palabras claves y OCDE de conocimiento. * Gran área * Área * Disciplina, montos solicitados, contrapartidas, información de la entidad, información del investigador principal

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente en la base de datos del aplicativo SIGP, aprovechando los permisos de acceso otorgados.
- Se generó un archivo Excel con los resultados obtenidos, estructurado según las necesidades del reporte.

3. Entrega del Reporte al Usuario Funcional

- Se notificó al usuario funcional Miguel Bernal sobre la disponibilidad del reporte.
- El archivo fue enviado por correo electrónico para su respectiva revisión y uso.

Solicitud 157495

Asunto	Tiempo total de actividad
Solicitud base datos evaluadores anonimizada	00:02:14
Descripción <p>Respetado Harold Alexander Méndez Jefe Oficina OTSI</p> <p>Reciba un cordial saludo.</p> <p>Con base en lo definido en el procedimiento EVALUACIÓN DE PROGRAMAS, PROPUESTAS O PROYECTOS DE CTeI (Código: M801PR02) en el numeral 16 "Solicitar al Jefe de la OTSI, a través de correo electrónico, la base de datos anonimizada de los potenciales evaluadores, enviando los criterios definidos en la actividad anterior". Es importante señalar que todos los evaluadores anonimizados deben tener formación doctoral.</p> <p>También hay que tener en cuenta los nombres del personal vinculado a cada propuesta, para que esta información se cruce con la de potenciales evaluadores, estos datos están registrados en los documentos del proponente, y en la plataforma SIGP. Se adjunta relación de palabras claves por Gran Area y Area del Conocimiento con base en el reporte entregado por la OTSI</p> <p>Agradezco su apoyo y amable gestión en éste proceso.</p> <p>Carolina Múnera Camacho Contratista Equipo de Seguimiento Convenios y Mecanismos cmunera@minciencias.gov.co Teléfono: 601 6258480 Ext. 5306</p>	

Av. Calle 26 No. 57 - 83 Torre 8, pisos 2 al 6 Bogotá, Colombia www.minciencias.gov.co			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
07/05/2025 11:00:17	09/05/2025 15:30:28	08/05/2025 16:30:28	09/05/2025 15:30:28

Acción Realizada 157495

En cumplimiento de la solicitud relacionada con la **Convocatoria 963-2025 Conv "Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 2025"**, se llevaron a cabo las siguientes actividades:

1. Generación del Reporte de Evaluadores

- Se construyó un reporte con el **listado de evaluadores** que cumplen con los criterios establecidos en cuanto a:
 - **Grandes áreas**
 - **Áreas**
 - **Palabras clave asociadas**
 - **Personal NO Asociado a los proyectos**
 - **Perfil del evaluador en cuanto al Nivel de Formación (Doctorado)**

2. Entrega del Reporte

- El documento resultante fue **generado y enviado al jefe de la OTSI, Harold Méndez**, como parte de la gestión solicitada.

Solicitud 157501

Asunto		Tiempo total de actividad	
reporte de la Convocatoria 964		00:03:18	
Descripción			
<p>Buenos días,</p> <p>Solicito su amable colaboración para generar un reporte de la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 12:00m del día de hoy.</p> <p>Muchas gracias.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
07/05/2025 11:25:30	08/05/2025 11:40:05	07/05/2025 12:40:05	08/05/2025 11:40:05

Acción Realizada 157501

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 12:00 p.m. del 07/05/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157516

Asunto	Tiempo total de actividad
CONVOCATORIA 41	00:05:17
Descripción Solicito amablemente fecha y hora de radicación en el SIGP de los proyectos 112926 y 112806, puesto que, se requiere para el consolidado final que se publicará mañana, para un criterio de desempate. Mil gracias y quedo atenta a su pronta respuesta. Atentamente, -- *Camila Meneses Prieto* Contratista Despacho Del Ministro - Secretaria Técnica OCAD *cmeneses@minciencias.gov.co <cmeneses@minciencias.gov.co>* Teléfono: 601 6258480 Ext. Av. Calle 26 No. 57 - 83 Torre 8, pisos 2 al 6 Bogotá, Colombia www.minciencias.gov.co	

<p>--</p> <p>No imprimas si no es necesario. Seamos responsables con la protección del medio ambiente.</p> <p>Declinación de Responsabilidades - Disclaimer <https://storage.googleapis.com/efor-static/CLCS/AVISO%20DE%20CONFIDENCIALIDAD%20-%20DISCLAIMER.pdf></p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
07/05/2025 15:05:30	08/05/2025 16:00:00	07/05/2025 18:04:12	08/05/2025 16:00:00

Acción Realizada 157516

En atención a la solicitud de generación de reporte para **los proyectos 112926 y 112806**, se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Camila Meneses Prieto** para su uso.

Solicitud 157582

Asunto		Tiempo total de actividad	
Reporte SIGP		00:02:47	
Descripción			
Cordial saludo. Solicito la atenta colaboración de OTSI para generar para la convocatoria 962-2024 SENAINNOVA, la base de datos del consolidado de evaluaciones por cada evaluador y por cada propuesta con la totalidad de los campos diligenciados y el puntaje de cada criterio y subcriterio, incluyendo las calificaciones del tercer evaluador y el puntaje total aplicando las reglas indicadas en el paso 40 del procedimiento de Evaluación de Programas, Propuestas o Proyectos de CTel - M801PR02 V8. Quedamos atentos a los dos reportes señalados en este paso del procedimiento. Muchas gracias			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
08/05/2025 15:24:29	09/05/2025 16:00:00	08/05/2025 17:08:25	09/05/2025 16:00:00

Acción Realizada 157582

En atención a la solicitud relacionada con la **Convocatoria 962-2024 SENAINNOVA PARA FOMENTAR ACCESO Y ADOPCIÓN DE TECNOLOGÍAS AVANZADAS EN SECTOR AGROALIMENTAR**, con corte al día 08 de mayo de 2025 se realizaron las siguientes actividades:

1. **Generación del Reporte de Evaluaciones**

- Se construyó un **informe consolidado** que contiene los resultados de las **evaluaciones por criterio y por evaluador** correspondientes al cierre de la convocatoria.

2. Entrega del Reporte

- El documento fue **notificado y enviado** al usuario funcional **Andrea Pachón Garzón** para su análisis y gestión correspondiente.

Solicitud 157588

Asunto		Tiempo total de actividad	
Reporte SIGP		00:06:27	
Descripción			
Solicitamos su atenta colaboración para que se genere un reporte de auditoría del acceso en SIGP a dos propuestas de la convocatoria 962-2024 SENAINNOVA. Lo anterior, en aras de validar las calificaciones emitidas y la línea de tiempo de acceso y modificación en el puntaje total y por criterio. Las propuestas son la 113501 y 113680. En dicho informe queremos validar fechas de acceso y modificación y si es posible el tipo de modificación realizada. Muchas gracias			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
08/05/2025 16:26:07	12/05/2025 16:00:00	09/05/2025 18:17:34	12/05/2025 16:00:00

Acción Realizada 157588

En atención a la solicitud de generación de reporte de auditoria para **las propuestas son la 113501 y 113680**, se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener la auditoria sobre las evaluaciones de los proyectos 113501 y 113680
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Andrea Pachón** para su uso.

Solicitud 157601

Asunto		Tiempo total de actividad	
Reporte convocatoria 41 del SGR		00:11:25	
Descripción			
buenas noches Ingeniero Leonardo amablemente solicito de su acostumbrada colaboracion para la generacion de un reprte de la convocatoria 41 del SGR, donde se indique la fecha y hora de subsanación de requisitos de las propuestas registradas, esto de manera urgente con el fin de avanzar con el proceso de evaluacion ya que se requiere para un criterio de desempate.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

08/05/2025 20:15:42	12/05/2025 09:58:08	09/05/2025 10:58:07	12/05/2025 09:58:08
------------------------	------------------------	------------------------	------------------------

Acción Realizada 157601

En atención a la solicitud de generación de reporte para la **Convocatoria 041-2024 Convocatoria Reto 2. Garantizar la Soberanía Alimentaria y el Derecho a la Alimentación**, se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y la subsanación de estos en la convocatoria 041-2024.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Lorena Monroy** para su uso.

Solicitud 157654

Asunto	Tiempo total de actividad
Proyectos pertenecientes a la convocatoria 41	00:03:53
Descripción	

-[[Solicitud TI]]-

----- Forwarded message -----

De: Camila Meneses Prieto <cmeneses@minciencias.gov.co>

Date: jue, 8 may 2025 a las 15:29

Subject: Solicitud información convocatoria 41.

To: Mesa de Servicios Minciencias <mesadeservicios@minciencias.gov.co>

Buen día,

Solicito amablemente fecha y hora de subsanación de los siguientes proyectos pertenecientes a la convocatoria 41:

- Proyecto 112926
- Proyecto 112806

Mil gracias.

Atentamente,

--

Camila Meneses Prieto

Contratista

Despacho Del Ministro - Secretaria Técnica OCAD

cmeneses@minciencias.gov.co <cmeneses@minciencias.gov.co>

Teléfono: 601 6258480 Ext.

Av. Calle 26 No. 57 - 83

Torre 8, pisos 2 al 6

Bogotá, Colombia

www.minciencias.gov.co

No imprimas si no es necesario. Seamos responsables con la protección del medio ambiente.

Declinación de Responsabilidades - Disclaimer

<https://storage.googleapis.com/efor-static/CLCS/AVISO%20DE%20CONFIDENCIALIDAD%20-%20DISCLAIMER.pdf>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
09/05/2025 12:23:26	12/05/2025 16:27:19	12/05/2025 09:21:11	12/05/2025 16:27:19

Acción Realizada 157654

En atención a la solicitud de generación de reporte para **los proyectos 112926 y 112806**, se indica que el caso fue atendido y se dio respuesta en la solicitud No. 157516 al usuario Camila Prieto Meneses el día 07/05/2025 17:45

Solicitud 157751

Asunto		Tiempo total de actividad	
SIGP - (Convocatoria 964)		00:06:06	
Descripción			
<p>Buenos días,</p> <p>Solicito su amable colaboración para generar un reporte de la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 12:00m del día de hoy.</p> <p>Muchas gracias.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

12/05/2025 10:48:06	12/05/2025 16:26:45	12/05/2025 14:59:03	12/05/2025 16:26:45
------------------------	------------------------	------------------------	------------------------

Acción Realizada 157751

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 12:00 p.m. del 12/05/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157834

Asunto	Tiempo total de actividad
---------------	----------------------------------

Solicitud de base anonimizada - 853		00:05:49	
Descripción			
<p>Estimado Director Harold,</p> <p>Mediante la presente solicito su colaboración para enviar una base de evaluadores anonimizados para evaluar dos propuestas de la convocatoria exención de IVA que tiene ventanilla abierta.</p> <p>Agradezco su colaboración.</p> <p>Ruben Darío Ortiz Morales Director Técnico Dirección de Desarrollo Tecnológico e Innovación rdortiz@minciencias.gov.co Teléfono: 601 6258480 Ext. Av. Calle 26 No. 57 - 83 Torre 8, pisos 2 al 6 Bogotá, Colombia www.minciencias.gov.co</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
13/05/2025 11:35:09	14/05/2025 13:46:06	13/05/2025 14:46:06	14/05/2025 13:46:06

Acción Realizada 157834

En cumplimiento de la solicitud relacionada con la **Convocatoria 853-2019 CONVOCATORIA REGISTRO DE PROPUESTAS QUE ACCEDERÁN A LA EXENCIÓN DE IVA (VENTANILLA ABIERTA)**, se llevaron a cabo las siguientes actividades:

1. Generación de los Reportes de Evaluadores para Proyectos 112134 y 112730

- Se construyó un reporte con el **listado de evaluadores** que cumplen con los criterios establecidos en cuanto a:
 - **Grandes áreas**
 - **Áreas**
 - **Disciplinas de conocimiento**
 - **Palabras clave asociadas**
 - **Entidades NO Asociadas a los proyectos**

2. Entrega del Reporte

- El documento resultante fue **generado y enviado** al **jefe de la OTSI, Harold Méndez**, como parte de la gestión solicitada.

Solicitud 157852

Asunto		Tiempo total de actividad	
SIGP - (Convocatoria 964)		00:03:21	
Descripción			
<p>Buenas tardes,</p> <p>Solicito su amable colaboración para el envío de un reporte de proyectos inscritos a la convocatoria 964 con corte a las 2:00pm del día de hoy.</p> <p>Muchas gracias</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

13/05/2025 14:04:04	14/05/2025 13:38:54	13/05/2025 14:38:54	14/05/2025 13:38:54
------------------------	------------------------	------------------------	------------------------

Acción Realizada 157852

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 12:00 p.m. del 13/05/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157957

Asunto	Tiempo total de actividad
SIGP - (Convocatoria 964)	00:04:24
Descripción	

Cordial para solicitar el reporte de la convocatoria 964			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
14/05/2025 09:09:23	14/05/2025 17:36:02	14/05/2025 10:16:14	14/05/2025 17:36:02

Acción Realizada 157957

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 09:00 a.m. del 14/05/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157981

Asunto	Tiempo total de actividad

SIGP - (Convocatoria 964)			00:01:37
Descripción			
<p>Cordial saludo,</p> <p>Solicito su amable colaboración para el envío de un reporte de proyectos inscritos a la convocatoria 964 con corte a las 12:00m del día de hoy.</p> <p>Muchas gracias</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
14/05/2025 11:00:54	14/05/2025 17:38:09	14/05/2025 13:12:06	14/05/2025 17:38:09

Acción Realizada 157981

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 12:00 p.m. del 14/05/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 157989

Asunto		Tiempo total de actividad	
Reporte convocatoria 964		00:04:44	
Descripción			
<p>Buen día:</p> <p>Teniendo en cuenta que la convocatoria 964 cierra el día de hoy 14 de mayo del 2025, atentamente solicito su apoyo en la generación de un reporte con todos los campos parametrizados, adicionalmente agradezco incluir un campo relacionado con que la persona participante cuenta con hoja de vida en CvLAC.</p> <p>Quedo atenta.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
14/05/2025 11:57:11	15/05/2025 16:00:00	14/05/2025 18:14:19	15/05/2025 16:00:00

Acción Realizada 157989

En cumplimiento de la solicitud recibida para la generación de reportes al cierre de la **Convocatoria “964-2025 CONVOCATORIA FORMACIÓN EN MAESTRÍAS PACÍFICO NARIÑENSE”** cuyo cierre fue el 14 de mayo de 2025, se realizaron las siguientes acciones:

1. Elaboración de Scripts de Consulta

- Se diseñaron scripts SQL para extraer la información correspondiente a la convocatoria con ID 11715.
- Las consultas permitieron generar los siguientes reportes requeridos por los usuarios funcionales:
 - Reporte general de la convocatoria.
 - Reporte de presupuesto de los proponentes (monto solicitado y contrapartida).
 - Reporte de contrapartidas discriminadas por entidad participante.
 - Reporte de personal vinculado a las propuestas.

2. Ejecución de Consultas y Generación de Archivos Excel

- Se ejecutaron los scripts directamente sobre la base de datos SIGP.
- Se generaron archivos Excel con los datos solicitados, estructurados para facilitar su análisis y revisión.

3. Entrega de los Reportes al Usuario Funcional

- Se notificó a la funcionaria **Ginneth Cajamarca** sobre la disponibilidad de los reportes.
- Los documentos fueron enviados en formato Excel conforme a los lineamientos establecidos.

Solicitud 157999

Asunto	Tiempo total de actividad
SIGP - (REPORTES)	00:09:05

Descripción			
<p>Buenas tardes, esperamos se encuentren bien.</p> <p>Mediante el presente solicitamos amablemente su valioso apoyo para la entrega de información relacionada con la convocatoria No.935 de 2023 y la convocatoria No.948 de 2024 de Orquídeas. En este sentido, me permito adjuntar el esquema de la BD en la cual agregar la información requerida, para la hoja 2023 a partir de la columna G "titulo_propuesta" hasta la columna BH "valor_total" y para la hoja 2024 a partir de la columna G "titulo_propuesta" hasta la columna BL "valor total". Lo anterior con el fin de dar respuesta a diversos requerimientos que están llegando a la Dirección de Vocaciones y Formación.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
14/05/2025 13:48:47	19/05/2025 16:00:00	17/05/2025 21:02:31	19/05/2025 16:00:00

Acción Realizada 157999

En cumplimiento de la solicitud recibida para la generación de reportes al cierre de la **Convocatorias No. 935 de 2023 y No. 948 de 2024 del programa Orquídeas**, se realizaron las siguientes acciones:

1. Elaboración de Scripts de Consulta

- Se diseñaron scripts SQL para extraer la información correspondiente a las convocatorias No. 935 de 2023 y No. 948 de 2024 del programa Orquídeas.
- Las consultas permitieron generar reporte de proyectos elegibles por cada convocatoria

2. Ejecución de Consultas y Generación de Archivos Excel

- Se ejecutaron los scripts directamente sobre la base de datos SIGP.
- Se generaron archivos Excel con los datos solicitados, estructurados para facilitar su análisis y revisión.

3. Entrega de los Reportes al Usuario Funcional

- Se notificó a la funcionaria **Daniela Jimenez** sobre la disponibilidad de los reportes.
- Los documentos fueron enviados en formato Excel conforme a los lineamientos establecidos.

Solicitud 158006

Asunto		Tiempo total de actividad	
Reporte convocatoria 964		00:03:46	
Descripción			
<p>Cordial saludo,</p> <p>Solicito su colaboración para generar un reporte de proyectos inscritos en la CV 964 con corte a las 2:00pm.</p> <p>Muchas gracias.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
14/05/2025 14:29:57	14/05/2025 17:34:32	14/05/2025 14:57:07	14/05/2025 17:34:32

Acción Realizada 158006

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 02:00 p.m. del 14/05/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 158017

Asunto		Tiempo total de actividad	
Reporte convocatoria 964 FINAL		00:06:29	
Descripción			
<p>Buenas tardes,</p> <p>Solicito su amable colaboración para el envío del reporte final de proyectos inscritos de la Convocatoria 964 - Formación en Maestrías Pacífico Nariñense, que tiene como fecha de cierre el día de hoy a las 4:00pm.</p> <p>Cordialmente,</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

14/05/2025 15:25:33	14/05/2025 17:34:20	14/05/2025 16:41:07	14/05/2025 17:34:20
------------------------	------------------------	------------------------	------------------------

Acción Realizada 158017

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 04:00 p.m. del 14/05/2025., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.
- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 158091

Asunto	Tiempo total de actividad
SIGP - (Convocatoria 953)	00:06:46
Descripción	

De acuerdo con correo remitido al jefe de la otsi el día 14 de mayo de 2025, Buenas tardes Jefe Harold, y de acuerdo con el procedimiento M801PR02 - paso 34, solicito amablemente su colaboración con el envío de un reporte de la plataforma SIGP que muestre el estado de la evaluación de cada propuesta de la convocatoria 953.

De antemano agradezco su apoyo.

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
15/05/2025 11:46:39	16/05/2025 14:25:20	15/05/2025 15:25:20	16/05/2025 14:25:20

Acción Realizada 158091

En atención a la solicitud relacionada con la **Convocatoria 953-2024 CONVOCATORIA PARA COMUNIDADES NEGRAS DE BOGOTÁ, CAQUETÁ, CÓRDOBA, MAGDALENA Y CHOCÓ**, con corte al día 15 de mayo de 2025 se realizaron las siguientes actividades:

1. Generación del Reporte de Evaluaciones

- Se construyó un **informe consolidado** que contiene los resultados de las **evaluaciones por criterio y por evaluador** correspondientes al cierre de la convocatoria.

2. Entrega del Reporte

- El documento fue **notificado y enviado** al usuario funcional **Alejandra Acosta Villamil** para su análisis y gestión correspondiente.

Solución de Errores

Se solucionaron errores reportados por los usuarios en la plataforma:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
157772	- Atención solicitud PQRDS 20250013875R Convocatoria 963-2025 Orquídeas	Solucionado	-	[Ver detalle]

Solicitud 157772

<p>Descripción</p> <p>Jefe Harold Méndez</p> <p>Reciba un cordial saludo. De manera atenta informo que se encuentra pendiente la respuesta a la PQR 20250013875R, sobre la cual llega nuevamente consulta con radicado 20250018527R y se da traslado a la OTSI, al estar relacionada con una posible falla en el SIGP en la radicación de una propuesta en la convocatoria No. 963 de 2025 Orquídeas.</p> <p>En el radicado PQR No. 20250013875R enviaron soporte de potencial falla en el SIGP del 19 de marzo a las 4:00 p.m. Sin embargo, se observa que los campos no estaban diligenciados en su totalidad. Al tratarse de una falla en la plataforma se dio traslado a la OTSI para la validación.</p> <p>image.png</p> <p>Cordialmente,</p> <p>Carolina Múnera Camacho Contratista Equipo de Seguimiento Convenios y Mecanismos cmunera@minciencias.gov.co Teléfono: 601 6258480 Ext. 5306 Av. Calle 26 No. 57 - 83 Torre 8, pisos 2 al 6 Bogotá, Colombia www.minciencias.gov.co</p>

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
12/05/2025 11:57:20	12/05/2025 16:26:35	12/05/2025 15:58:34	12/05/2025 16:26:35

Acción Realizada 157772

En atención a la PQR con radicado 20250013875R, reiterada mediante el radicado 20250018527R y trasladada a esta oficina por estar relacionada con el funcionamiento de la plataforma SIGP, nos permitimos informar lo siguiente:

El usuario identificado con el proyecto ID 114226 intentó postular su propuesta a la convocatoria "963-2025 Convocatoria Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 2025". Durante el proceso de envío, el sistema arrojó una validación indicando que hacía falta el registro en la categoría "Género" del enfoque diferencial.

La opción "Validar y Enviar" del formulario verifica que se haya completado toda la información mínima obligatoria. En cuanto al enfoque diferencial, es requisito registrar las cinco categorías definidas, independientemente de si aplican o no:

- Población víctima del conflicto armado
- Población en condición de discapacidad
- Género
- Grupo étnico
- Grupos etarios

El usuario únicamente registró una de estas categorías, por lo que el sistema le notificó la información faltante justo al momento del cierre de la convocatoria, el 19 de marzo de 2025 a las 4:00 p. m. Posteriormente, el formulario no permitió el envío por haber expirado el plazo, como es habitual en el mecanismo de cierre automático de la plataforma.

Es importante señalar que entre las 3:50 p. m. y las 4:00 p. m. se radicaron 28 proyectos sin inconvenientes (se adjunta soporte), siendo el último a las 3:58:27 p. m., lo cual evidencia que la plataforma funcionó correctamente.

En consecuencia, no se trata de un fallo técnico, sino de una validación propia del sistema que busca asegurar el cumplimiento de los requisitos establecidos en la convocatoria.

Actualización de Datos

Se realizó la siguiente actualización en la plataforma SIGP:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
157135	- Actualización Razón Social Entidad SISTEMAS RENOVABLES DE COLOMBIA SAS BIC	Solucionado	-	[Ver detalle]
157139	- Actualizar Fecha Cierre Convocatoria 964-2025 Formación Maestrías Pacífico Nariñense	Solucionado	-	[Ver detalle]
157142	- Actualizar Fecha Cierre Convocatoria 037-2024 Macroproyectos Nacionales	Solucionado	-	[Ver detalle]
157719	- Información Estado Proyecto 112134 de Beneficios Tributarios	Solucionado	-	[Ver detalle]
157805	- Eliminación de evaluación de proyectos ID 113501 y 113680	Solucionado	-	[Ver detalle]
157868	- Error Sumatoria Puntaje Evaluaciones Proyectos Convocatoria	Solucionado	-	[Ver detalle]

Solicitud 157135

Asunto	Tiempo total de actividad
Solicitud de actualización razón social - ENT1358442019	00:18:20
Descripción	
Buenas tardes ing. Leonardo:	

Amablemente se solicita realizar la actualización de la razón social de la entidad para atender la solicitud no. 20250018481R.

Anexo anexo cámara de comercio

Cordialmente,

Nidia Janneth León Cabeza

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
30/04/2025 15:22:07	01/05/2025 15:08:34	30/04/2025 16:08:34	01/05/2025 15:08:34

Acción Realizada 157135

- Se actualizó con éxito el nombre de la entidad de "TOBO Y COMPAÑIA SAS" a "SISTEMAS RENOVABLES DE COLOMBIA SAS BIC" en el sistema.
- La actualización se realizó conforme al certificado de existencia y representación legal o de inscripción de documentos presentado.
- Se verificó y confirmó la correcta modificación en la base de datos.

Solicitud 157139

Asunto	Tiempo total de actividad
SOLICITUD DE SERVICIO OTIC	00:03:29
Descripción	

Buenas tardes,

Solicitamos su atenta colaboración para realizar las siguientes modificaciones en el formulario de SIGP correspondiente a la Convocatoria 964, Formación en Maestrías Pacífico Nariñense.

Cierre de la convocatoria: 14 de mayo de 2025, hasta 4:00pm (Hora colombiana)

Periodo de revisión de requisitos: Del 15 al 23 de mayo de 2025

Período de subsanación de requisitos: Del 26 al 30 de mayo de 2025, hasta 4:00pm (Hora colombiana)

Publicación de banco preliminar de candidatos elegibles: 16 de julio de 2025

Período de solicitud de aclaraciones al banco preliminar de candidatos elegibles: Del 17 al 18 de julio de 2025, hasta 4:00pm (Hora colombiana)

Respuesta a solicitud de aclaraciones: Del 21 al 29 de Julio de 2025

Publicación del banco definitivo de candidato elegibles: 14 de agosto de 2025

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
30/04/2025 15:43:18	01/05/2025 16:00:00	30/04/2025 17:57:56	01/05/2025 16:00:00

Acción Realizada 157139

- Se actualizó con éxito la fecha de cierre de la **convocatoria 964-2025 Formación Maestrías Pacífico Nariñense** en el sistema hasta el día 14 de mayo de 2025, hasta 4:00pm.
- Se verificó y confirmó la correcta modificación en la base de datos.

Solicitud 157142

Asunto	Tiempo total de actividad

Ampliación de fechas			00:04:19
Descripción			
Nos permitimos solicitar ampliar la fecha de cierre de la Convocatoria No. 37 hasta el día 30 de mayo del presente año, en la plataforma SIGP, en atención a la modificación No. 1 de la convocatoria (Adjunta), la cual fue aprobada y publicada hoy por el OCAD,			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
30/04/2025 16:19:19	01/05/2025 16:00:00	30/04/2025 18:07:20	01/05/2025 16:00:00

Acción Realizada 157142

- Se actualizó con éxito la fecha de cierre de la **convocatoria 037-2024 Macroproyectos Nacionales** en el sistema hasta el día 30 de mayo de 2025, hasta 4:00pm.
- Se verificó y confirmó la correcta modificación en la base de datos.

Solicitud 157719

Asunto	Tiempo total de actividad
SIGP - (Convocatoria)	00:09:23
Descripción	
-[[SOLICITUD TI]]- Buen día, Por medio de la presente, solicito muy amablemente su colaboración para que	

me orienten respecto a cómo realizar la consulta, o con qué área debo comunicarme, para conocer el estado del proyecto 112134 de la Universidad Nacional de Colombia, presentado a la *Convocatoria para el registro de propuestas que accederán a la exención del IVA (ventanilla abierta)* del programa de Beneficios Tributarios

Lo anterior con el fin de dar respuesta a una PQRS presentada por un ciudadano, en la que se requiere información sobre el estado actual del proyecto. Cabe señalar que la propuesta fue radicada en el mes de octubre, y según lo indicado en la Guía Técnica para la presentación y calificación de propuestas que accederán a la exención del IVA por importación de equipos y elementos, publicada en la página web, se establece en el numeral 4 "Cronograma ventanilla interna" que:

*"Una vez enviada la propuesta a Colciencias se procederá a realizar la revisión de requisitos y su evaluación en los casos que aplique. El tiempo máximo para el trámite de una exención será de 90 días hábiles." *

Agradezco de antemano su atención y quedo atenta a su orientación.

Cordialmente

--

Andrea Camila Gómez Vargas

Contratista

Equipo Beneficios Tributarios

acgomez@minciencias.gov.co <acgomez@minciencias.gov.co>

Teléfono: 601 6258480 Ext.

Av. Calle 26 No. 57 - 83

Torre 8, pisos 2 al 6

Bogotá, Colombia

www.minciencias.gov.co

--

No imprimas si no es necesario. Seamos responsables con la protección del medio ambiente.

Declinación de Responsabilidades - Disclaimer < https://storage.googleapis.com/efor-static/CLCS/AVISO%20DE%20CONFIDENCIALIDAD%20-%20DISCLAIMER.pdf >			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
12/05/2025 08:01:31	12/05/2025 16:26:54	12/05/2025 10:22:22	12/05/2025 16:26:54

Acción Realizada 157719

En atención a su solicitud, nos permitimos informarle que el proyecto identificado con el ID 112134, titulado "Ampliación del uso de la mecánica cuántica desde el punto de vista experimental y su relación con la teoría generando desarrollos en tecnologías cuánticas útiles para metrología y computación cuántica a nivel nacional", presentado por la UNIVERSIDAD NACIONAL DE COLOMBIA - SEDE BOGOTÁ (OFICIAL) en el marco de la Convocatoria "853-2019 CONVOCATORIA REGISTRO DE PROPUESTAS QUE ACCEDERÁN A LA EXENCIÓN DE IVA (VENTANILLA ABIERTA)", se encuentra actualmente en estado "REVISADO".

Esto significa que la propuesta ya superó la fase de verificación de requisitos y fue remitida a evaluación, conforme al procedimiento establecido en la guía técnica de la convocatoria. El cambio de estado fue realizado el día 28 de octubre de 2024 a las 09:01 a. m. por el funcionario Miguel Camilo Bernal Bolaños, quien dejó constancia de que el proyecto cumple con los requisitos establecidos.

Solicitud 157805

Asunto	Tiempo total de actividad
SIGP - FUNCIONALIDAD	00:10:57

Descripción			
<p>[[solicitud TI: eliminar registro concepto de evaluador en SIGP - convocatoria 962]]-</p> <p>Cordial saludo,</p> <p>Atentamente se solicita dar de baja en SIGP los conceptos de evaluación registrados por la evaluadora Irene Nieto Escribano en las propuestas con ID 113501 y 113680 de la convocatoria 962-2024, atendiendo lo definido en la nota 3 de la actividad 37 del procedimiento de evaluación M801PR02, considerando los resultados entregados por OTSI a las auditorías de dichos proyectos en el ticket 157588. La descripción ampliada de este caso se encuentra soportada en documento adjunto.</p> <p>Muchas gracias,</p> <p>*Andrea Carolina Pachón Garzón*</p> <p>Profesional Especializado</p> <p>Dirección de Desarrollo Tecnológico e Innovación</p> <p>*acpachon@minciencias.gov.co <acpachon@minciencias.gov.co>*</p> <p>Teléfono: 601 6258480 Ext. 4802</p> <p>Av. Calle 26 No. 57 - 83</p> <p>Torre 8, pisos 2 al 6</p> <p>Bogotá, Colombia</p> <p>www.minciencias.gov.co</p> <p>--</p> <p>No imprimas si no es necesario. Seamos responsables con la protección del medio ambiente.</p> <p>Declinación de Responsabilidades - Disclaimer</p> <p><https://storage.googleapis.com/efor-static/CLCS/AVISO%20DE%20CONFIDENCIALIDAD%20-%20DISCLAIMER.pdf></p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

12/05/2025 15:39:54	23/05/2025 09:35:26	22/05/2025 10:35:26	23/05/2025 09:35:26
------------------------	------------------------	---------------------	------------------------

Acción Realizada 157805

- Según el requerimiento dar de baja en SIGP los conceptos de evaluación registrados por la evaluadora Irene Nieto Escribano en las propuestas con ID 113501 y 113680 de la convocatoria 962-2024. el proceso fue realizado con éxito.
- Se verificó y confirmó la correcta eliminación en la base de datos.

Solicitud 157868

Asunto	Tiempo total de actividad
Revisión convocatoria 962 reportes de eval...	00:05:03
Descripción <p> -[[solicitud TI: revisión convocatoria 962 reportes de evaluación]]- Cordial saludo, Considerando que ya se han adelantado las actividades de asignación de propuestas y emisión de conceptos de evaluación por parte de los evaluadores de la convocatoria 962-2024 SENAINNOVA, se ha procedido a generar de la herramienta SIGP los reportes con el consolidado de la evaluación por pares. Al verificar los documentos descargados de "Reporte de Calificación y Concepto Argumentado Ponderado" de las propuestas enlistadas enseguida, se evidencia que no coincide el valor que figura al final del reporte con la suma de cada criterio de evaluación: • 113608 - Puntaje total del proyecto correcto es 82,00 puntos (suma de </p>	

todos los puntajes), sin embargo, el reporte consolidado de calificación presenta un error ya que se indica un total de 83,50 puntos.

- 113721 - Puntaje total del proyecto correcto es 78,50 puntos (suma de todos los puntajes), sin embargo, el reporte consolidado de calificación presenta un error ya que se indica un total de 78,00 puntos.
- 113769 –Puntaje total del proyecto correcto es 75,50 puntos (suma de todos los puntajes), sin embargo, el reporte consolidado de calificación presenta un error ya que se indica un total de 74,50 puntos.
- 113728 - Puntaje total del proyecto correcto es 49,50 puntos (suma de todos los puntajes), sin embargo, el reporte consolidado de calificación presenta un error ya que se indica un total de 48 puntos.

Así mismo, esta situación se repite en los documentos descargados de estas mismas propuestas denominado "Reporte Calificación".

Agradecemos realizar la verificación correspondiente y ajustar.

Atentamente,

Andrea Carolina Pachón Garzón

Profesional Especializado

Dirección de Desarrollo Tecnológico e Innovación

acpachon@minciencias.gov.co <acpachon@minciencias.gov.co>

Teléfono: 601 6258480 Ext. 4802

Av. Calle 26 No. 57 - 83

Torre 8, pisos 2 al 6

Bogotá, Colombia

www.minciencias.gov.co

--

No imprimas si no es necesario. Seamos responsables con la protección del medio ambiente.

Declinación de Responsabilidades - Disclaimer

<https://storage.googleapis.com/efor-static/CLCS/AVISO%20DE%20CONFIDENCIALIDAD%20-%20DISCLAIMER.pdf>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
13/05/2025 15:01:30	14/05/2025 17:37:54	14/05/2025 15:03:12	14/05/2025 17:37:54

Acción Realizada 157868

En atención a la solicitud relacionada con las inconsistencias encontradas en los reportes de calificación y concepto argumentado ponderado de las propuestas de la convocatoria 962-2024 SENAINNOVA, nos permitimos informar que se realizaron los ajustes correspondientes y se resolvió la situación reportada.

Los puntajes totales ahora coinciden correctamente con la suma de los criterios de evaluación para cada propuesta mencionada, tanto en el Reporte de Calificación y Concepto Argumentado Ponderado como en el Reporte de Calificación

Requerimientos

Se recibieron solicitudes de implementación de nuevas funcionalidades en el sistema:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
157596	- Implementar Registro múltiples líneas y sublíneas temáticas por proyecto	Pendiente de Usuario	Producción	[Ver detalle]

Solicitud 157596

Asunto		Tiempo total de actividad	
Parametrización Convocatoria 37		00:52:06	
Descripción			
En el marco de la SOLICITUD DE REQUERIMIENTO PARA CAMBIOS A SISTEMAS DE INFORMACIÓN - Convocatoria 37, solicitamos su apoyo para el proceso de parametrización de la adenda de esta convocatoria.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
08/05/2025 17:50:01	03/06/2025 14:58:28		

Acción Realizada 157596

1. Estimación de esfuerzos

- Se elabora y diligencia el **formato de estimación de esfuerzos**, donde se detallan los recursos necesarios, tiempos estimados y complejidad del desarrollo.
- Se revisa y ajusta la estimación en función de los requerimientos del usuario y las capacidades del equipo de desarrollo.

2. Inicio del desarrollo

- Se implementan las funcionalidades solicitadas de acuerdo con los requerimientos técnicos y funcionales.
- Se realizan pruebas unitarias para verificar el correcto funcionamiento del desarrollo antes de su despliegue.

3. Despliegue en ambiente de pruebas

- Se despliega el requerimiento en **ambiente de pruebas** para su validación.
- Se ejecutan pruebas funcionales y de integración para detectar posibles errores o mejoras.
- Se documentan los resultados de las pruebas y se hacen los ajustes necesarios.

4. Aprobación y despliegue en producción

- Se presenta el desarrollo para aprobación por parte del usuario funcional o equipo de calidad.
- Una vez aprobado, se coordina el **despliegue en ambiente de producción** asegurando que los cambios no afecten la estabilidad del sistema.
- Se realiza una validación post-despliegue para confirmar el correcto funcionamiento en producción.

5. Pendientes

- La solicitud aún no ha sido cerrada debido a que no se cuenta aun con el formato de pruebas de aceptación firmado y cargado en la plataforma por el líder funcional del SIGP.

Estado Actual:

- ✓ **Estimación de esfuerzos completada.**
- ✓ **Desarrollo implementado.**
- ✓ **Despliegue en pruebas realizado.**
- ✓ **Pendiente de aprobación para producción.**
- ✓ **Despliegue final en producción.**

Conclusiones

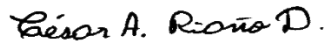
Durante este período se brindó soporte técnico en la plataforma SIGP, permitiendo la parametrización de convocatorias, parametrización de rubricas de evaluación, la generación de reportes, la solución de errores y la gestión de requerimientos.

Se continúa con el seguimiento de los casos que permanecen en estado **Pendiente Usuario** y **Pendiente Proveedor**, a la espera de la validación o documentación requerida.

Para garantizar una mejor eficiencia en la resolución de solicitudes, se recomienda:

- Mantener una comunicación constante con los usuarios para agilizar las aclaraciones necesarias.
- Optimizar los tiempos de respuesta en los casos que requieren intervención de terceros.
- Seguir documentando los cambios realizados para futuras referencias.

Firma:



Ing. César Augusto Riaño Delgado
Gerente Proyecto