

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

PRIMER PAGO CONTRATO 237-2024

TABLA DE CONTENIDO

INFORME MANTENIMIENTO Y DESARROLLO ESPECIALIZADO SISTEMA DE INFORMACIÓN DE GESTIÓN DE PROYECTOS - SIGP	1
1. MÓDULO DE ADMINISTRACIÓN, (No. Obligación Contrato: 2.2.1, 2.2.5, 2,2,6).....	1
Aplicativo de Gestión	2
Creación pantalla de Administración de Dependencias	2
Creación pantalla de Administración de Tablas	57
Creación pantalla de Administración de Campos	102
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 MINICIENCIAS, (No. Obligación Contrato: 2.2.3, 2.2.4)	153
Introducción	153
Resumen de Actividades	153
Detalle de Casos Atendidos.....	153
Parametrización de Convocatorias	154
Generación de Reportes.....	193
Solución de Errores	224
Actualización de Datos	233
Requerimientos	236
Conclusiones.....	243

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

1. MÓDULO DE ADMINISTRACIÓN, (No. Obligación Contrato: 2.2.1, 2.2.5, 2,2,6)

Solicitud de requerimiento No. 154297

Descripción y alcance del Requerimiento

Este requerimiento tiene como objetivo el desarrollo e integración de tres nuevas funcionalidades dentro del módulo de administración del Sistema Integrado de Gestión de Proyectos (SIGP). Estas funcionalidades permitirán gestionar la configuración del sistema a través de las siguientes secciones:

1. **Configuración de Dependencias:** Administración de las dependencias de MinCiencias, estableciendo su jerarquía, relaciones y atributos principales.
2. **Configuración de Tablas:** Gestión de las tablas del sistema, facilitando su creación, modificación y eliminación.
3. **Configuración de Campos:** Administración de los campos que conforman cada tabla, definiendo su tipo de dato, longitud y restricciones.

El alcance del desarrollo incluye el diseño y desarrollo de las interfaces gráficas, la implementación de funcionalidades para la gestión de registros (creación, edición y eliminación), validaciones de integridad y reglas de negocio, y la integración con el módulo de administración del SIGP.

Desarrollo

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

- Creación de nueva pantalla de administración de Dependencias
- Creación de nueva pantalla de administración de Campos
- Creación de nueva pantalla de administración de Tablas

Aplicativo de Gestión

Creación pantalla de Administración de Dependencias

Función:

Esta pantalla permite la gestión de dependencias dentro del sistema. Se pueden registrar, modificar y eliminar dependencias, además de configurar sus relaciones con otras entidades del sistema.

Principales funcionalidades:

- **Listado de Dependencias:** Se visualizan todas las dependencias registradas en una tabla con opciones de filtrado.
- **Creación y Edición:** Formulario para ingresar los datos de una nueva dependencia o modificar una existente.
- **Eliminación:** Opción para eliminar dependencias que no tengan relaciones activas

Creación nuevo menú de Administración de Dependencias

```
-- funciones (menus)
insert into 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', 'GEST6180', 'Dependencias', 'F', null, null,
'/app/administracion/dependenciasSistema.xhtml', null, 'GEST6100', 'fa fa-building',
'SIGP_WEB', to_date('05-04-2024 13:37:59', 'dd-mm-yyyy hh24:mi:ss'), 'SIGP_WEB',
to_date('28-02-2025 12:34:04', 'dd-mm-yyyy hh24:mi:ss'));

-- derechos funciones (permisos)
insert into sb_derechos_funciones (DERECHO_FUNCION_ID, APLICACION, FUNCION, GRUPO,
ESTADO, USUARIO_CREADOR, FECHA_CREACION, USUARIO_MODIFICADOR, FECHA_MODIFICACION)
values (null, 'SIGP_WEB', 'GEST6180', 'SGPS_ADMINISTRADOR', 'H', 'SIGP_WEB',
to_date('05-04-2024 13:40:38', 'dd-mm-yyyy hh24:mi:ss'), 'SIGP_WEB', to_date('15-04-2024
15:07:26', 'dd-mm-yyyy hh24:mi:ss'));
```

Crear nueva entidad de la tabla ***SB_DEPENDENCIAS*** en la base de datos sigp

```
/*
 * 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.business.domain;

import com.sb.colciencias.gestion.business.domain.security.Usuario;
import com.sb.colciencias.gestion.business.valueObjects.AppSessionVO;
import java.io.Serializable;
import java.util.Date;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.Transient;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author dcangrejo
 */
@Entity
@Table(name = "SB_DEPENDENCIAS", schema = "HERCULES")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "DependenciaSistema.findAll", query = "SELECT d FROM DependenciaSistema d")
    , @NamedQuery(name = "DependenciaSistema.findById", query = "SELECT d FROM DependenciaSistema d WHERE d.dependenciaId = :dependenciaId")
    , @NamedQuery(name = "DependenciaSistema.findByAplicacion", query = "SELECT d FROM DependenciaSistema d WHERE d.aplicacion = :aplicacion")
    , @NamedQuery(name = "DependenciaSistema.findByNombreDependencia", query = "SELECT d FROM DependenciaSistema d WHERE d.nombreDependencia = :nombreDependencia")
    , @NamedQuery(name = "DependenciaSistema.findByUsuarioAdministrador", query = "SELECT d FROM DependenciaSistema d WHERE d.usuarioAdministrador = :usuarioAdministrador")
    , @NamedQuery(name = "DependenciaSistema.findByIdDependenciaPadreId", query = "SELECT d FROM DependenciaSistema d WHERE d.dependenciaPadreId = :dependenciaPadreId")
    , @NamedQuery(name = "DependenciaSistema.findByVigente", query = "SELECT d FROM DependenciaSistema d WHERE d.vigente = :vigente")
    , @NamedQuery(name = "DependenciaSistema.findByAtributo1", query = "SELECT d FROM DependenciaSistema d WHERE d.atributo1 = :atributo1")
    , @NamedQuery(name = "DependenciaSistema.findByAtributo2", query = "SELECT d FROM DependenciaSistema d WHERE d.atributo2 = :atributo2")
    , @NamedQuery(name = "DependenciaSistema.findByAtributo3", query = "SELECT d FROM DependenciaSistema d WHERE d.atributo3 = :atributo3")
    , @NamedQuery(name = "DependenciaSistema.findByAtributo4", query = "SELECT d FROM DependenciaSistema d WHERE d.atributo4 = :atributo4")
    , @NamedQuery(name = "DependenciaSistema.findByAtributo5", query = "SELECT d FROM DependenciaSistema d WHERE d.atributo5 = :atributo5")
    , @NamedQuery(name = "DependenciaSistema.findByAtributo6", query = "SELECT d FROM DependenciaSistema d WHERE d.atributo6 = :atributo6")
    , @NamedQuery(name = "DependenciaSistema.findByAtributo7", query = "SELECT d FROM DependenciaSistema d WHERE d.atributo7 = :atributo7")
})
```

```
, @NamedQuery(name = "DependenciaSistema.findByAtributo8", query = "SELECT d FROM
DependenciaSistema d WHERE d.atributo8 = :atributo8")
, @NamedQuery(name = "DependenciaSistema.findByAtributo9", query = "SELECT d FROM
DependenciaSistema d WHERE d.atributo9 = :atributo9")
, @NamedQuery(name = "DependenciaSistema.findByAtributo10", query = "SELECT d FROM
DependenciaSistema d WHERE d.atributo10 = :atributo10")
, @NamedQuery(name = "DependenciaSistema.findByUsuarioCreador", query = "SELECT d
FROM DependenciaSistema d WHERE d.usuarioCreador = :usuarioCreador")
, @NamedQuery(name = "DependenciaSistema.findByFechaCreacion", query = "SELECT d
FROM DependenciaSistema d WHERE d.fechaCreacion = :fechaCreacion")
, @NamedQuery(name = "DependenciaSistema.findByUsuarioModificador", query = "SELECT
d FROM DependenciaSistema d WHERE d.usuarioModificador = :usuarioModificador")
, @NamedQuery(name = "DependenciaSistema.findByFechaModificacion", query = "SELECT d
FROM DependenciaSistema d WHERE d.fechaModificacion = :fechaModificacion")
, @NamedQuery(name = "DependenciaSistema.findByTipoDependenciaId", query = "SELECT d
FROM DependenciaSistema d WHERE d.tipoDependenciaId = :tipoDependenciaId"))}
public class DependenciaSistema implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @Basic(optional = false)
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"SB_DEPENDENCIA_ID_SEQ")
    @SequenceGenerator(name = "SB_DEPENDENCIA_ID_SEQ", sequenceName =
"SB_DEPENDENCIA_ID_SEQ", allocationSize = 1)
    @NotNull
    @Column(name = "DEPENDENCIA_ID")
    private Long dependenciaId;

    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 10)
    @Column(name = "APLICACION")
    private String aplicacion;

    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 100)
    @Column(name = "NOMBRE_DEPENDENCIA")
    private String nombreDependencia;

    @Size(max = 1)
    @Column(name = "VIGENTE")
    private String vigente;

    @Size(max = 100)
    @Column(name = "ATRIBUTO1")
    private String atributo1;

    @Size(max = 100)
    @Column(name = "ATRIBUTO2")
    private String atributo2;

    @Size(max = 100)
    @Column(name = "ATRIBUTO3")
    private String atributo3;

    @Size(max = 100)
    @Column(name = "ATRIBUTO4")
    private String atributo4;

    @Size(max = 100)
```

```
@Column(name = "ATRIBUTO5")
private String atributo5;

@Size(max = 100)
@Column(name = "ATRIBUTO6")
private String atributo6;

@Size(max = 100)
@Column(name = "ATRIBUTO7")
private String atributo7;

@Size(max = 100)
@Column(name = "ATRIBUTO8")
private String atributo8;

@Size(max = 100)
@Column(name = "ATRIBUTO9")
private String atributo9;

@Size(max = 100)
@Column(name = "ATRIBUTO10")
private String atributo10;

@Size(max = 30)
@Column(name = "USUARIO_CREADOR")
private String usuarioCreador;

@Column(name = "FECHA_CREACION")
@Temporal(TemporalType.TIMESTAMP)
private Date fechaCreacion;

@Size(max = 30)
@Column(name = "USUARIO_MODIFICADOR")
private String usuarioModificador;

@Column(name = "FECHA_MODIFICACION")
@Temporal(TemporalType.TIMESTAMP)
private Date fechaModificacion;

//Joins
@JoinColumn(name = "DEPENDENCIA_PADRE_ID", referencedColumnName = "DEPENDENCIA_ID")
@ManyToOne(optional = false)
private DependenciaSistema dependenciaPadreId;

@JoinColumn(name = "USUARIO_ADMINISTRADOR", referencedColumnName = "USUARIO")
@ManyToOne(optional = false)
private Usuario usuarioAdministrador;

@JoinColumn(name = "TIPO_DEPENDENCIA_ID", referencedColumnName =
"VALOR_DEFINICION_ID")
@ManyToOne(optional = false)
private ValorDefinicion tipoDependenciaId;

//Auditoria
@Transient
private AppSessionVO appSessionVO;

public DependenciaSistema() {
}

public DependenciaSistema(Long dependenciaId) {
    this.dependenciaId = dependenciaId;
}
```

```
public DependenciaSistema(Long dependenciaId, String aplicacion, String
nombreDependencia) {
    this.dependenciaId = dependenciaId;
    this.aplicacion = aplicacion;
    this.nombreDependencia = nombreDependencia;
}

public Long getDependenciaId() {
    return dependenciaId;
}

public void setDependenciaId(Long dependenciaId) {
    this.dependenciaId = dependenciaId;
}

public String getAplicacion() {
    return aplicacion;
}

public void setAplicacion(String aplicacion) {
    this.aplicacion = aplicacion;
}

public String getNombreDependencia() {
    return nombreDependencia;
}

public void setNombreDependencia(String nombreDependencia) {
    this.nombreDependencia = nombreDependencia;
}

public Usuario getUsuarioAdministrador() {
    return usuarioAdministrador;
}

public void setUsuarioAdministrador(Usuario usuarioAdministrador) {
    this.usuarioAdministrador = usuarioAdministrador;
}

public DependenciaSistema getDependenciaPadreId() {
    return dependenciaPadreId;
}

public void setDependenciaPadreId(DependenciaSistema dependenciaPadreId) {
    this.dependenciaPadreId = dependenciaPadreId;
}

public String getVigente() {
    return vigente;
}

public void setVigente(String vigente) {
    this.vigente = vigente;
}

public String getAtributo1() {
    return atributo1;
}

public void setAtributo1(String atributo1) {
    this.atributo1 = atributo1;
}
```

```
public String getAtributo2() {  
    return atributo2;  
}  
  
public void setAtributo2(String atributo2) {  
    this.atributo2 = atributo2;  
}  
  
public String getAtributo3() {  
    return atributo3;  
}  
  
public void setAtributo3(String atributo3) {  
    this.atributo3 = atributo3;  
}  
  
public String getAtributo4() {  
    return atributo4;  
}  
  
public void setAtributo4(String atributo4) {  
    this.atributo4 = atributo4;  
}  
  
public String getAtributo5() {  
    return atributo5;  
}  
  
public void setAtributo5(String atributo5) {  
    this.atributo5 = atributo5;  
}  
  
public String getAtributo6() {  
    return atributo6;  
}  
  
public void setAtributo6(String atributo6) {  
    this.atributo6 = atributo6;  
}  
  
public String getAtributo7() {  
    return atributo7;  
}  
  
public void setAtributo7(String atributo7) {  
    this.atributo7 = atributo7;  
}  
  
public String getAtributo8() {  
    return atributo8;  
}  
  
public void setAtributo8(String atributo8) {  
    this.atributo8 = atributo8;  
}  
  
public String getAtributo9() {  
    return atributo9;  
}  
  
public void setAtributo9(String atributo9) {  
    this.atributo9 = atributo9;  
}
```

```
}

public String getAtributo10() {
    return atributo10;
}

public void setAtributo10(String atributo10) {
    this.atributo10 = atributo10;
}

public String getUsuarioCreador() {
    return usuarioCreador;
}

public void setUsuarioCreador(String usuarioCreador) {
    this.usuarioCreador = usuarioCreador;
}

public Date getFechaCreacion() {
    return fechaCreacion;
}

public void setFechaCreacion(Date fechaCreacion) {
    this.fechaCreacion = fechaCreacion;
}

public String getUsuarioModificador() {
    return usuarioModificador;
}

public void setUsuarioModificador(String usuarioModificador) {
    this.usuarioModificador = usuarioModificador;
}

public Date getFechaModificacion() {
    return fechaModificacion;
}

public void setFechaModificacion(Date fechaModificacion) {
    this.fechaModificacion = fechaModificacion;
}

public ValorDefinicion getTipoDependenciaId() {
    return tipoDependenciaId;
}

public void setTipoDependenciaId(ValorDefinicion tipoDependenciaId) {
    this.tipoDependenciaId = tipoDependenciaId;
}

public AppSessionVO getAppSessionVO() {
    return appSessionVO;
}

public void setAppSessionVO(AppSessionVO appSessionVO) {
    this.appSessionVO = appSessionVO;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (dependenciaId != null ? dependenciaId.hashCode() : 0);
    return hash;
}
```

```

    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof DependenciaSistema)) {
            return false;
        }
        DependenciaSistema other = (DependenciaSistema) object;
        if ((this.dependenciaId == null && other.dependenciaId != null) ||
            (this.dependenciaId != null && !this.dependenciaId.equals(other.dependenciaId))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "com.sb.colciencias.gestion.business.domain.DependenciaSistema["
            + "dependenciaId=" + dependenciaId + " ]";
    }
}

```

Crear nuevo DTO representativo de la entidad

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package com.sb.colciencias.gestion.business.transfer;

import com.sb.colciencias.gestion.business.domain.DependenciaSistema;
import com.sb.colciencias.gestion.business.domain.ValorDefinicion;
import com.sb.colciencias.gestion.business.domain.security.Usuario;
import com.sb.colciencias.gestion.business.util.annotation.Bind;
import com.sb.colciencias.gestion.business.valueObjects.AppSessionVO;
import java.io.Serializable;
import java.util.Date;

/**
 *
 * @author dcangrejo
 */
public class DependenciaSistemaDTO extends GenericDTO implements Serializable {

    @Bind(name = "dependenciaId")
    private Long dependenciaId;

    @Bind(name = "aplicacion")
    private String aplicacion;

    @Bind(name = "nombreDependencia")
    private String nombreDependencia;

    @Bind(name = "vigente")
    private String vigente;

    @Bind(name = "atributo1")

```

```
private String atributo1;

@Bind(name = "atributo2")
private String atributo2;

@Bind(name = "atributo3")
private String atributo3;

@Bind(name = "atributo4")
private String atributo4;

@Bind(name = "atributo5")
private String atributo5;

@Bind(name = "atributo6")
private String atributo6;

@Bind(name = "atributo7")
private String atributo7;

@Bind(name = "atributo8")
private String atributo8;

@Bind(name = "atributo9")
private String atributo9;

@Bind(name = "atributo10")
private String atributo10;

@Bind(name = "usuarioCreador")
private String usuarioCreador;

@Bind(name = "fechaCreacion")
private Date fechaCreacion;

@Bind(name = "usuarioModificador")
private String usuarioModificador;

@Bind(name = "fechaModificacion")
private Date fechaModificacion;

//Joins
@Bind(name = "dependenciaPadreId", returnType = DependenciaSistemaDTO.class, type =
DependenciaSistema.class)
private DependenciaSistemaDTO dependenciaPadreId;

@Bind(name = "usuarioAdministrador", returnType = UsuarioDTO.class, type =
Usuario.class)
private UsuarioDTO usuarioAdministrador;

@Bind(name = "tipoDependenciaId", returnType = ValorDefinicionDTO.class, type =
ValorDefinicion.class)
private ValorDefinicionDTO tipoDependenciaId;

//Auditoria
@Bind(name="appSessionVO")
private AppSessionVO appSessionVO;

public DependenciaSistemaDTO() {
}
```

```
public DependenciaSistemaDTO(Long dependenciaId) {
    this.dependenciaId = dependenciaId;
}

public DependenciaSistemaDTO(Long dependenciaId, String nombreDependencia) {
    this.dependenciaId = dependenciaId;
    this.nombreDependencia = nombreDependencia;
}

public Long getDependenciaId() {
    return dependenciaId;
}

public void setDependenciaId(Long dependenciaId) {
    this.dependenciaId = dependenciaId;
}

public String getAplicacion() {
    return aplicacion;
}

public void setAplicacion(String aplicacion) {
    this.aplicacion = aplicacion;
}

public String getNombreDependencia() {
    return nombreDependencia;
}

public void setNombreDependencia(String nombreDependencia) {
    this.nombreDependencia = nombreDependencia;
}

public UsuarioDTO getUsuarioAdministrador() {
    return usuarioAdministrador;
}

public void setUsuarioAdministrador(UsuarioDTO usuarioAdministrador) {
    this.usuarioAdministrador = usuarioAdministrador;
}

public String getVigente() {
    return vigente;
}

public void setVigente(String vigente) {
    this.vigente = vigente;
}

public String getAtributo1() {
    return atributo1;
}

public void setAtributo1(String atributo1) {
    this.atributo1 = atributo1;
}

public String getAtributo2() {
    return atributo2;
}

public void setAtributo2(String atributo2) {
    this.atributo2 = atributo2;
}
```

```
}

public String getAtributo3() {
    return atributo3;
}

public void setAtributo3(String atributo3) {
    this.atributo3 = atributo3;
}

public String getAtributo4() {
    return atributo4;
}

public void setAtributo4(String atributo4) {
    this.atributo4 = atributo4;
}

public String getAtributo5() {
    return atributo5;
}

public void setAtributo5(String atributo5) {
    this.atributo5 = atributo5;
}

public String getAtributo6() {
    return atributo6;
}

public void setAtributo6(String atributo6) {
    this.atributo6 = atributo6;
}

public String getAtributo7() {
    return atributo7;
}

public void setAtributo7(String atributo7) {
    this.atributo7 = atributo7;
}

public String getAtributo8() {
    return atributo8;
}

public void setAtributo8(String atributo8) {
    this.atributo8 = atributo8;
}

public String getAtributo9() {
    return atributo9;
}

public void setAtributo9(String atributo9) {
    this.atributo9 = atributo9;
}

public String getAtributo10() {
    return atributo10;
}

public void setAtributo10(String atributo10) {
```

```
this.atributo10 = atributo10;
}

public String getUsuarioCreador() {
    return usuarioCreador;
}

public void setUsuarioCreador(String usuarioCreador) {
    this.usuarioCreador = usuarioCreador;
}

public Date getFechaCreacion() {
    return fechaCreacion;
}

public void setFechaCreacion(Date fechaCreacion) {
    this.fechaCreacion = fechaCreacion;
}

public String getUsuarioModificador() {
    return usuarioModificador;
}

public void setUsuarioModificador(String usuarioModificador) {
    this.usuarioModificador = usuarioModificador;
}

public Date getFechaModificacion() {
    return fechaModificacion;
}

public void setFechaModificacion(Date fechaModificacion) {
    this.fechaModificacion = fechaModificacion;
}

public ValorDefinicionDTO getTipoDependenciaId() {
    return tipoDependenciaId;
}

public void setTipoDependenciaId(ValorDefinicionDTO tipoDependenciaId) {
    this.tipoDependenciaId = tipoDependenciaId;
}

public DependenciaSistemaDTO getDependenciaPadreId() {
    return dependenciaPadreId;
}

public void setDependenciaPadreId(DependenciaSistemaDTO dependenciaPadreId) {
    this.dependenciaPadreId = dependenciaPadreId;
}

public AppSessionVO getAppSessionVO() {
    return appSessionVO;
}

public void setAppSessionVO(AppSessionVO appSessionVO) {
    this.appSessionVO = appSessionVO;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (dependenciaId != null ? dependenciaId.hashCode() : 0);
}
```

```
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        if (!(object instanceof DependenciaSistemaDTO)) {
            return false;
        }
        DependenciaSistemaDTO other = (DependenciaSistemaDTO) object;
        if ((this.dependenciaId == null && other.dependenciaId != null) ||
            (this.dependenciaId != null && !this.dependenciaId.equals(other.dependenciaId))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "com.sb.colciencias.entidades.business.transfer.DependenciaSistemaDTO["
            + "dependenciaId=" + dependenciaId + " ]";
    }
}
```

La clase **DependenciaSistemaDTO** es un **Data Transfer Object (DTO)** que se utiliza para transportar datos relacionados con la entidad **DependenciaSistema** dentro de la aplicación. Hereda de **GenericDTO** e implementa **Serializable**, lo que permite que sus instancias puedan ser convertidas en una secuencia de bytes para su transmisión o almacenamiento.

Paquete e Importaciones

```
package com.sb.colciencias.gestion.business.transfer;
```

El código pertenece al paquete **com.sb.colciencias.gestion.business.transfer**, que parece estar destinado a la transferencia de datos en la capa de negocio de la aplicación.

Las clases importadas incluyen:

- **Entidades del dominio (DependenciaSistema, ValorDefinicion, Usuario):** Modelan los datos principales.
- **Anotación @Bind:** Se usa para mapear propiedades entre entidades y DTOs.
- **Objetos de sesión (AppSessionVO):** Probablemente contiene información de la sesión del usuario.
- **Serializable:** Para permitir la serialización de la clase.
- **Date:** Para manejar fechas.

Atributos

```
@Bind(name = "dependenciaId")  
private Long dependenciaId;
```

Cada atributo está anotado con `@Bind`, lo que sugiere que se están vinculando con propiedades de una entidad. Algunos atributos clave:

- **Identificación y datos principales:**
 - **dependenciaId:** Identificador único de la dependencia.
 - **aplicacion:** Nombre de la aplicación asociada.
 - **nombreDependencia:** Nombre de la dependencia.
 - **vigente:** Estado de la dependencia (activo/inactivo).
- **Atributos adicionales (atributo1 a atributo10):**
 - Campos genéricos que pueden representar diferentes características de la dependencia.
- **Auditoría:**
 - **usuarioCreador, fechaCreacion:** Identifican al usuario que creó la dependencia y la fecha de creación.
 - **usuarioModificador, fechaModificacion:** Información sobre modificaciones.
- **Relaciones con otras entidades:**

```
@Bind(name = "dependenciaPadreId", returnType = DependenciaSistemaDTO.class, type =  
DependenciaSistema.class)  
private DependenciaSistemaDTO dependenciaPadreId;
```

- **dependenciaPadreId:** Relación con una dependencia superior (jerarquía).
 - **usuarioAdministrador:** Relación con un usuario responsable.
 - **tipoDependenciaId:** Clasificación de la dependencia.
- **Objeto de sesión:**

```
@Bind(name="appSessionVO")  
private AppSessionVO appSessionVO;
```

- Permite capturar información de la sesión actual del usuario.

Constructores

```
public DependenciaSistemaDTO() { }
```

- **Constructor vacío:** Necesario para la serialización y uso en frameworks como JSF o JPA.

```
public DependenciaSistemaDTO(Long dependenciaId) {  
    this.dependenciaId = dependenciaId;  
}
```

- **Constructor con ID:** Facilita la creación del objeto con solo el identificador.

```
public DependenciaSistemaDTO(Long dependenciaId, String nombreDependencia) {  
    this.dependenciaId = dependenciaId;  
    this.nombreDependencia = nombreDependencia;  
}
```

- **Constructor con ID y Nombre:** Permite inicializar con valores básicos.

Métodos Getters y Setters

- Se incluyen métodos para acceder y modificar los atributos.
- Son necesarios para la serialización y uso en frameworks como JSF, JPA y JAXB.

Ejemplo:

```
public String getAplicacion() {  
    return aplicacion;  
}  
  
public void setAplicacion(String aplicacion) {  
    this.aplicacion = aplicacion;  
}
```

Métodos hashCode(), equals() y toString()

Estos métodos permiten:

- **Comparar objetos (equals()):**

```
@Override  
public boolean equals(Object object) {  
    if (!(object instanceof DependenciaSistemaDTO)) {  
        return false;  
    }
```

```
}
DependenciaSistemaDTO other = (DependenciaSistemaDTO) object;
return (this.dependenciaId != null && this.dependenciaId.equals(other.dependenciaId));
}
```

- Se considera que dos objetos son iguales si tienen el mismo dependenciaId.

- **Generar un identificador único (hashCode()):**

```
@Override
public int hashCode() {
    int hash = 0;
    hash += (dependenciaId != null ? dependenciaId.hashCode() : 0);
    return hash;
}
```

- **Conversión a String (toString()):**

```
@Override
public String toString() {
    return "com.sb.colciencias.entidades.business.transfer.DependenciaSistemaDTO["
    dependenciaId=" + dependenciaId + " ]";
}
```

- Devuelve una representación de la dependencia en formato de texto.

Conclusión

Este DTO:

- Permite transportar información de DependenciaSistema entre capas de la aplicación.
- Está diseñado para ser serializable y compatible con frameworks de persistencia y presentación.
- Usa anotaciones @Bind para mapear datos entre entidades y DTOs.
- Incluye información de auditoría y sesión.

-
- Adición del menú **"Dependencias"** en la plantilla del menú de Administración – 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 administración de dependencias del sistema.

```
<!-- Dependencias -->
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST6180')}">
    <li class="nav-item #{selectedPage eq 'dependenciasSistema' ? 'start active':''}">
```

```
<p:link outcome="/app/administracion/dependenciasSistema" styleClass="nav-link">
    <i class="fa fa-building"></i>
    #{msg['sb.menu.gestion.administracion.sistema.dependenciaSistema']}
</p:link>
</li>
</ui:fragment>
```

Control de Visibilidad con Seguridad

```
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST6180')}">
```

- Se utiliza **<ui:fragment>** para encapsular el elemento sin generar un nodo adicional en el DOM.
- El atributo **rendered** evalúa si el usuario tiene acceso a esta opción del menú.
- **SeguridadSessController.mapaMenu.containsKey('GEST6180')** verifica si el identificador 'GEST6180' está presente en el mapa de permisos, indicando que el usuario tiene acceso a la funcionalidad.

Estructura del Elemento del Menú

```
<li class="nav-item #{selectedPage eq 'dependenciasSistema' ? 'start active':''}">
```

- Define un elemento de lista **** con la clase **nav-item**, utilizada para el diseño del menú.
- La expresión **#{selectedPage eq 'dependenciasSistema' ? 'start active':''}** agrega clases CSS dinámicamente para resaltar el menú cuando la página actual corresponde a 'dependenciasSistema'.

Enlace de Navegación

```
<p:link outcome="/app/administracion/dependenciasSistema" styleClass="nav-link">
```

- **<p:link>** de **PrimeFaces** genera un enlace de navegación JSF.
- El atributo **outcome** define la ruta de la pantalla a la que se dirigirá el usuario al hacer clic en el botón (**/app/administracion/dependenciasSistema**).
- La clase **nav-link** proporciona el estilo adecuado para la apariencia del enlace.

Ícono y Etiqueta del Menú

```
<i class="fa fa-building"></i>
#{msg['sb.menu.gestion.administracion.sistema.dependenciaSistema']}
```

- **<i class="fa fa-building"></i>** inserta un ícono de edificio usando **FontAwesome** para representar visualmente la funcionalidad.
- **#{msg['sb.menu.gestion.administracion.sistema.dependenciaSistema']}** obtiene el texto del menú desde un archivo de internacionalización, permitiendo compatibilidad con múltiples idiomas.

El anterior código, agrega un botón en el menú que permite navegar a la pantalla de administración de dependencias. Su visibilidad depende de los permisos del usuario (**GEST6180**). Además, destaca la opción del menú cuando la página está activa y utiliza un ícono con texto dinámico para mejorar la experiencia del usuario.

Controlador de Vista DependenciaSistemaViewController

Creación Controlador de Vista **DependenciaSistemaViewController.java**, encargado de mostrar la información (listado de registros de Dependencias) de Minciencias. Este controlador gestiona la carga, filtrado y paginación de las dependencias.

Declaraciones de Clase

```
@Named(value = "dependenciaSistemaViewController")
@ViewScoped
public class DependenciaSistemaViewController extends
GenericSessionController<DependenciaSistemaDTO> implements Serializable {
```

- **@Named** permite que el controlador sea accesible en las vistas **JSF**.
- **@ViewScoped** mantiene el estado mientras la vista esté activa.
- **Extiende GenericSessionController<DependenciaSistemaDTO>** para heredar funcionalidades de sesión.

Inyección de Dependencias

```
@EJB
private AdministracionServiceFacade administracionServiceFacade;
@Inject
private ColcienciasSessController colcienciasSessController;
@Inject
private ConfigApplController configApplController;
```

- **administracionServiceFacade**: Proporciona acceso a la capa de negocio (EJB).
- **colcienciasSessController**: Controlador de sesión para manejar datos de usuario.
- **configAppController**: Controlador de configuración de la aplicación.

Variables de Estado

```
private LazyDataModel<DependenciaSistemaDTO> model;  
private List<DependenciaSistemaDTO> listDependenciaSistema;  
private List<UsuarioDTO> listUsuario;  
private boolean filtroActivo;  
private List<ValorDefinicionDTO> listTipoDependencia;  
private DependenciaSistemaDTO dependenciaSistemaFilterBusq;  
private DependenciaSistemaDTO dependenciaSistemaSelected;  
private List<DependenciaSistemaDTO> listaJerarquiaDependencia;
```

- **model**: Implementación de **LazyDataModel** para carga paginada en **PrimeFaces**.
- **listDependenciaSistema**: Lista de dependencias obtenidas de la base de datos.
- **listUsuario**: Lista de usuarios para asignaciones.
- **filtroActivo**: Indica si los filtros están activos.
- **listTipoDependencia**: Lista de tipos de dependencias disponibles.
- **dependenciaSistemaFilterBusq**: Filtros aplicados en la búsqueda.
- **dependenciaSistemaSelected**: Dependencia seleccionada por el usuario.
- **listaJerarquiaDependencia**: Lista de dependencias en estructura jerárquica.

Método cargarListas()

```
@PostConstruct  
public void cargarListas() {  
    cargarListasCampos();  
    this.model = new LazyDataModel<DependenciaSistemaDTO>() {
```

- **@PostConstruct**: Se ejecuta después de instanciar el **bean**.
- Llama a **cargarListasCampos()** para inicializar las listas.
- Crea un **LazyDataModel** para la carga de datos con paginación.

Implementación de LazyDataModel

```
@Override
public List<DependenciaSistemaDTO> 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());
            }
        }
    });

    try {
        filters.put("aplicacion", configApplController.getAplicacion().getAplicacion());
        if (dependenciaSistemaFilterBusq != null) {
            filters.put("dependenciaSistemaFilterBusq", dependenciaSistemaFilterBusq);
        }

        int count = administracionServiceFacade.dependenciaSistemaListCount(filters);
        if (count > 0) {
            list = administracionServiceFacade.dependenciaSistemaList(first, pageSize,
sortField, sortOrder.equals(SortOrder.ASCENDING) ? "A" : "D", filters);
        }
        model.setRowCount(count);
    } catch (Exception ex) {
        LOG.error("Error al cargar el listado de dependencias del sistema ", ex);
    }
    return list;
}
```

Explicación

- Obtiene los filtros aplicados en la tabla.
- Agrega la aplicación actual al filtro.
- Consulta el **conteo total de registros** en **administracionServiceFacade.dependenciaSistemaListCount(filters)**.
- Si hay registros, obtiene la lista con paginación (**dependenciaSistemaList**).
- Asigna el número total de registros al modelo **setRowCount(count)**.

Método getRowData()

```
@Override
public DependenciaSistemaDTO getRowData(String rowKey) {
    try {
        return administracionServiceFacade.queryDependenciaSistema(Long.parseLong(rowKey));
    } catch (Exception ex) {
        LOG.error("No se pudo cargar la dependencia del sistema ( id => " + rowKey + ")",
ex);
    }
    return null;
}
```

- Carga un **registro específico** desde la base de datos basado en su **rowKey (ID)**.

Método cargarListasCampos()

```
public void cargarListasCampos() {
    try {
        Map<String, Object> filters = new HashMap<>();
        filters.put("tipoDefinicion", "TIPDEPEN");
        listTipoDependencia = administracionServiceFacade.valorDefinicionList(-1, -1,
        "descripcionValorDefinicion", "A", filters);

        Map<String, Object> filters2 = new HashMap<>();
        listUsuario = administracionServiceFacade.usuariosList(-1, -1, "usuario", "A",
        filters2);

        Map<String, Object> filters3 = new HashMap<>();
        filters3.put("aplicacion", configApplController.getAplicacion().getAplicacion());
        listDependenciaSistema = administracionServiceFacade.dependenciaSistemaList(-1, -1,
        "nombreDependencia", "A", filters3);
    } catch (Exception ex) {
        LOG.error("Error al cargar los valores definiciones del tipo definición seleccionado
        ", ex);
    }
}
```

- Se cargan listas de:
 - Tipos de dependencia (**listTipoDependencia**).
 - Usuarios (**listUsuario**).
 - Dependencias del sistema (**listDependenciaSistema**).

El controlador maneja la carga eficiente de dependencias con filtros y paginación en **PrimeFaces**. El menú permite un acceso controlado basado en permisos.

Controlador de Conversación DependenciaSistemaConvController.java

El **DependenciaSistemaConvController** es un controlador en Java EE que gestiona las operaciones **CRUD** (Crear, Leer, Actualizar, Eliminar) sobre las dependencias del sistema en el contexto de la aplicación. Utiliza **JavaServer Faces (JSF)** junto con **PrimeFaces** para la interacción con la UI y **EJBs** para la capa de negocio.

Este controlador extiende **GenericConversationController<DependenciaSistemaDTO>**, lo que le proporciona funcionalidad base para la gestión de datos en una conversación de JSF.

Inyección de Dependencias

```
@EJB
private AdministracionServiceFacade administracionServiceFacade;
@Inject
private DependenciaSistemaViewController dependenciaSistemaViewController;
@Inject
private SeguridadSessController seguridadSessController;
@Inject
private ConfigApplController configApplController;
```

- **AdministracionServiceFacade:** Componente EJB que maneja la lógica de negocio relacionada con la administración de dependencias.
- **DependenciaSistemaViewController:** Controlador de vista que gestiona el estado y la interacción con la UI.
- **SeguridadSessController:** Controlador de sesión que proporciona información sobre el usuario autenticado.
- **ConfigApplController:** Proporciona configuraciones generales de la aplicación.

Variables Importantes

```
private final String programa = "GEST6180";

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

- **programa:** Código que identifica el módulo dentro del sistema (**GEST6180**).
- **LOG:** Logger de **Apache Log4j** para registrar eventos y errores.

Funcionalidades Implementadas

Guardar una Dependencia (guardar)

Permite registrar una nueva dependencia en el sistema.

```
@Override
public void guardar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
        DependenciaSistemaDTO dependenciaSistemaDTO =
dependenciaSistemaViewController.getSelected();

        dependenciaSistemaDTO.setUsuarioCreador(seguridadSessController.getSelected().getUsuario());
        dependenciaSistemaDTO.setFechaCreacion(new Date());
    }
}
```

```
dependenciaSistemaDTO.setAplicacion(configApplController.getAplicacion().getAplicacion(
));
//Auditoria
AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
appSessionVO.setPrograma(programa);
dependenciaSistemaDTO.setAppSessionVO(appSessionVO);
administracionServiceFacade.guardarDependenciaSistema(dependenciaSistemaDTO);

dependenciaSistemaViewController.setSelected(new DependenciaSistemaDTO());
configApplController.listasGenerales();
dependenciaSistemaViewController.cargarListas();
//se dehabilita el boton de editar / eliminar ya que no habra nada seleccionado
despues de la operación
this.onRowUnselect(null);
this.obtenerPestana = 0;
} catch (SystemException ex) {
    message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
} catch (Exception ex) {
    message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression("#{msg['sb.mensajes.transaccion.fallido']}"));
    LOG.error("Error al guardar el parametro de sistema", ex);
} finally {
    FacesContext.getCurrentInstance().addMessage(null, message);
}
}
```

Flujo:

1. Se obtiene la dependencia seleccionada desde **DependenciaSistemaViewController**.
2. Se establece el usuario creador y la fecha de creación.
3. Se configura la auditoría con información de sesión.
4. Se envía la información a **AdministracionServiceFacade.guardarDependenciaSistema()**.
5. Se limpia la selección y se recargan las listas.
6. Se muestra un mensaje de éxito o error en la UI.

Eliminar una Dependencia (eliminar)

Elimina una dependencia del sistema.

```
@Override
public void eliminar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
```

```

        DependenciaSistemaDTO dependenciaSistemaDTO =
dependenciaSistemaViewController.getSelected();
        //Auditoria
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        appSessionVO.setProgram(programa);
        dependenciaSistemaDTO.setAppSessionVO(appSessionVO);
        administracionServiceFacade.eliminarDependenciaSistema(dependenciaSistemaDTO);

        dependenciaSistemaViewController.setSelected(new DependenciaSistemaDTO());
        configApplController.listasGenerales();
        dependenciaSistemaViewController.cargarListas();
        this.onRowUnselect(null);
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        LOG.error("Error al eliminar el parametro de sistema ", ex);
    } finally {
        FacesContext.getCurrentInstance().addMessage(null, message);
        dependenciaSistemaViewController.cargarListas();
    }
}

```

Flujo:

1. Se obtiene la dependencia seleccionada.
2. Se establece la auditoría.
3. Se llama a **administracionServiceFacade.eliminarDependenciaSistema()**.
4. Se limpia la selección y se recargan las listas.
5. Se muestra un mensaje de éxito o error.

Editar una Dependencia (editar)

Permite modificar los datos de una dependencia existente.

```

@Override
public void editar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
        DependenciaSistemaDTO dependenciaSistemaDTO =
dependenciaSistemaViewController.getSelected();

        dependenciaSistemaDTO.setUsuarioModificador(seguridadSessController.getSelected().getUs
uario());
        dependenciaSistemaDTO.setFechaModificacion(new Date());
        //Auditoria
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        appSessionVO.setProgram(programa);
        dependenciaSistemaDTO.setAppSessionVO(appSessionVO);
    }
}

```

```

        administracionServiceFacade.editarDependenciaSistema(dependenciaSistemaDTO);

        dependenciaSistemaViewController.setSelected(new DependenciaSistemaDTO());
        configApplController.listasGenerales();
        dependenciaSistemaViewController.cargarListas();
        //se dehabilita el boton de editar / eliminar ya que no habra nada seleccionado
despues de la operación
        this.onRowUnselect(null);
        this.obtenerPestana = 0;
    } catch (SystemException ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
        resolveExpression(ex.getMessage()));
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
        resolveExpression("#{msg['sb.mensajes.transaccion.fallido']}"));
        LOG.error("Error al editar el parametro de sistema ", ex);
    } finally {
        FacesContext.getCurrentInstance().addMessage(null, message);
    }
}

```

Flujo:

1. Se obtiene la dependencia seleccionada.
2. Se establece el usuario modificador y la fecha de modificación.
3. Se configura la auditoría.
4. Se actualiza en la base de datos con
administracionServiceFacade.editarDependenciaSistema().
5. Se limpia la selección y se recargan las listas.
6. Se muestra un mensaje de éxito o error.

Mostrar Lista de Dependencias (mostrar)

Limpia los filtros aplicados en la tabla y actualiza la UI.

```

@Override
public void mostrar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));
    //Obtengo la tabla+
    String modulo = dependenciaSistemaViewController.getModulo();
    DataTable dataTable = (DataTable)
    FacesContext.getCurrentInstance().getViewRoot().findComponent(modulo + "Form:" + modulo +
    "TabList:" + modulo + "Table");
    //limpio los filtros
    dataTable.setFilterBy(null);
}

```

```
//Se actualiza el formulario que contiene la tabla
PrimeFaces.current().ajax().update(modulo + "Form");
//Se muestra el growl con el mensaje
FacesContext.getCurrentInstance().addMessage(null, message);

dependenciaSistemaViewController.cargarListas();
this.onRowUnselect(null);
}
```

Flujo:

1. Se obtiene el identificador del módulo.
2. Se busca la tabla de datos en la vista y se limpian los filtros.
3. Se actualiza el formulario que contiene la tabla.
4. Se muestra un mensaje de éxito.

Construcción del Árbol Jerárquico de Dependencias (construirArbolDependencias)

Genera una estructura jerárquica de dependencias en un **TreeNode** de PrimeFaces.

```
public TreeNode construirArbolDependencias(DependenciaSistemaDTO dependenciaSeleccionada) {
    TreeNode root = new DefaultTreeNode("Root", null); // Crear el nodo raíz del árbol

    // Llamar al método recursivo para construir el árbol a partir de la dependencia
    seleccionada
    construirArbolRecursivo(dependenciaSeleccionada, root);

    return root;
}

private void construirArbolRecursivo(DependenciaSistemaDTO dependencia, TreeNode padre) {
    try {

        TreeNode nodo = new DefaultTreeNode(dependencia, padre);

        List<DependenciaSistemaDTO> hijos =
        administracionServiceFacade.dependenciaHijasList(dependencia);

        for (DependenciaSistemaDTO hijo : hijos) {
            construirArbolRecursivo(hijo, nodo);
        }

    } catch (Exception e) {
        LOG.error("Error al obtener la jerarquía de la dependencia", e);
    }
}
```

Flujo:

1. Se crea un nodo raíz.
2. Se llama recursivamente a **construirArbolRecursivo** para agregar nodos hijos.
3. Se obtiene la lista de dependencias hijas desde **administracionServiceFacade.dependenciaHijasList()**.
4. Se construye el árbol de dependencias dinámicamente.

Generación de Reporte (generarReporte)

Genera un informe de dependencias con un logotipo parametrizado.

```
/**
 * Metodo para generar el reporte de dependencias
 * @param dependenciaSistemaDTO
 */
public void generarReporte(DependenciaSistemaDTO dependenciaSistemaDTO) {

    FacesContext context = FacesContext.getCurrentInstance();
    HttpServletResponse response = (HttpServletResponse)
context.getExternalContext().getResponse();
    try {
        ReporteUtil reporteUtil = new ReporteUtil();
        HashMap hashMap = new HashMap();

        //Se setea la imagen del logo parametrizada, guardada en la base de datos
        String imagenLogoBase64 =
administracionServiceFacade.queryParametroSistema("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);

        //Se setea la imagen secundaria del logo parametrizada, guardada en la base de
datos
        String imagenLogoSecundarioBase64 =
administracionServiceFacade.queryParametroSistema("LOGIN_LOGO2").getDetalle();
        String imagenSecundaria = null;
        if (imagenLogoSecundarioBase64 != null) {
            String[] parts2 = imagenLogoSecundarioBase64.split(",");
            //      String part2 = parts2[0]; // data:image/jpeg;base64
            imagenSecundaria = parts2[1]; // IMAGEN
        }
        hashMap.put("IMG_SECUNDARIA_BASE64", imagenSecundaria);
        //
        hashMap.put("DEPENDENCIA_ID", dependenciaSistemaDTO.getDependenciaId());

        //Parametro usuario conectado
        hashMap.put("USUARIO_CONECTADO",
seguridadSessController.getSelected().getDescripcionUsuario() + "");
        //
        byte reporte[];
        reporteUtil.setNomCarpeta("ReporteDependenciaSistema");
        reporte = reporteUtil.generarReporte(hashMap, "reporteDependenciaSistema",
"PDF", context);
    }
}
```

```
response.setContentType("application/pdf");
response.setHeader("Content-Disposition", "attachment;
filename=\"ReporteDependenciaSistema.pdf\"");
response.setContentLength(reporte.length);
try (ServletOutputStream outputStream = response.getOutputStream()) {
    outputStream.write(reporte, 0, reporte.length);
    outputStream.flush();
}
} catch (Exception ex) {
    PrimeFaces.current().executeScript("toastr.error('" +
this.resolveExpression(ex.getMessage()) + "')");
    LOG.error("Error al generar el reporte de dependencias ", ex);
}
}
```

Flujo:

1. Se obtiene la imagen del logo almacenada en la base de datos.
2. Se convierte a formato **Base64**.
3. Se pasa como parámetro a ReporteUtil.
4. Se genera la salida en formato PDF o el requerido.

Consideraciones Técnicas

- **Auditoría:** Cada operación registra el usuario y la fecha en los campos correspondientes.
- **Mensajes UI:** Se gestionan con **FacesMessage** para proporcionar retroalimentación en la interfaz.
- **Manejo de Excepciones:** Se captura **SystemException** y **Exception** para registrar errores en **Log4j**.
- **Actualización de Listas:** Después de cada operación, se actualizan los datos en la UI.

*Se definen métodos de crud en Facade EJB (**AdministracionServiceFacade.java**) para funcionalidad de pantalla*

```
// _____
// DEPENDENCIA SISTEMA
/**
 * Consulta el listado de dependencias del sistema 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
* linea programatica/foco departamentos por region por mecanismo por
* convocatoria
* @return lista de linea programatica/foco departamentos por region
* mecanismo por convocatoria encontrados en el rango enviado y basado en
* los filtros enviados
* @throws Exception
*/
List<DependenciaSistemaDTO> dependenciaSistemaList(int first, int pageSize, String
sortField, String sortOrder, Map<String, Object> filters) throws Exception;

/**
* Cuenta la cantidad de parametro del sistema actualmente registrados en la
* base de datos
*
* @param filters
* @return
* @throws Exception
*/
int dependenciaSistemaListCount(Map<String, Object> filters) throws Exception;

/**
* Guarda el registro de parametro del sistema en la base de datos
*
* @param dependenciaSistemaDTO
* @throws Exception
*/
void guardarDependenciaSistema(DependenciaSistemaDTO dependenciaSistemaDTO) throws
Exception;

/**
* Edita el registro de parametro del sistema en la base de datos
*
* @param dependenciaSistemaDTO
* @throws Exception
*/
void editarDependenciaSistema(DependenciaSistemaDTO dependenciaSistemaDTO) throws Exception;

/**
* Elimina el registro de parametro del sistema en la base de datos
*
* @param dependenciaSistemaDTO
* @throws Exception
*/
void eliminarDependenciaSistema(DependenciaSistemaDTO dependenciaSistemaDTO) throws
Exception;

/**
* Buscar el registro de un parametro del sistema y lo retorna basado en el
* id (llave primaria)
*
* @param dependenciaSistemaId
* @return el parametro del sistema que coincide con el id enviado
* @throws Exception
*/
DependenciaSistemaDTO queryDependenciaSistema(Long dependenciaSistemaId) throws Exception;

List<TipoDependenciaDTO> obtenerTiposDependencias() throws Exception;

```

```
/**
 * Consulta las dependencias asociadas a una dependencia padre (enviado como
 * parametro)
 *
 * @param dependenciaSistemaDTO
 * @return dependencia que pertenecen a una dependencia padre
 * @throws Exception
 */
List<DependenciaSistemaDTO> dependenciaHijasList(DependenciaSistemaDTO
dependenciaSistemaDTO) throws Exception;
```

*Se implementan métodos crud en facade EJB
(AdministracionFacadeServiceBean.java) para funcionalidad de pantalla*

```
//----- Dependencia Sistema -----//
@Override
public List<DependenciaSistemaDTO> dependenciaSistemaList(int first, int pageSize, String
sortField, String sortOrder, Map<String, Object> filters) throws Exception {
    ObjectConverter<DependenciaSistemaDTO, DependenciaSistema> converter =
    objectConverterFactory.checkOut();
    List<DependenciaSistemaDTO> listDependenciaSistemaDTOS =
    converter.toTransferList(dependenciaSistemaService.dependenciaSistemaList(first, pageSize,
    sortField, sortOrder, filters), DependenciaSistemaDTO.class);
    objectConverterFactory.checkIn(converter);
    return listDependenciaSistemaDTOS;
}

@Override
public int dependenciaSistemaListCount(Map<String, Object> filters) throws Exception {
    DependenciaSistemaDTO dependenciaSistemaFilterDTO = (DependenciaSistemaDTO)
    filters.get("dependenciaSistemaFilterBusq");
    ObjectConverter<DependenciaSistemaDTO, DependenciaSistema> converter =
    objectConverterFactory.checkOut();
    DependenciaSistema dependenciaSistemaFilterBusq =
    converter.toEntity(dependenciaSistemaFilterDTO, DependenciaSistema.class);
    filters.put("dependenciaSistemaFilterBusq", dependenciaSistemaFilterBusq);
    return dependenciaSistemaService.dependenciaSistemaListCount(filters);
}

@Override
public void guardarDependenciaSistema(DependenciaSistemaDTO dependenciaSistemaDTO) throws
Exception {
    ObjectConverter<DependenciaSistemaDTO, DependenciaSistema> converter =
    objectConverterFactory.checkOut();
    dependenciaSistemaService.guardarDependenciaSistema(converter.toEntity(dependenciaSiste
maDTO, DependenciaSistema.class));
    objectConverterFactory.checkIn(converter);
}

@Override
public void editarDependenciaSistema(DependenciaSistemaDTO dependenciaSistemaDTO) throws
Exception {
    ObjectConverter<DependenciaSistemaDTO, DependenciaSistema> converter =
    objectConverterFactory.checkOut();
    dependenciaSistemaService.editarDependenciaSistema(converter.toEntity(dependenciaSistem
aDTO, DependenciaSistema.class));
}
```

```
        objectConverterFactory.checkIn(converter);
    }

    @Override
    public void eliminarDependenciaSistema(DependenciaSistemaDTO dependenciaSistemaDTO) throws
    Exception {
        ObjectConverter<DependenciaSistemaDTO, DependenciaSistema> converter =
        objectConverterFactory.checkOut();
        dependenciaSistemaService.eliminarDependenciaSistema(converter.toEntity(dependenciaSist
        emaDTO, DependenciaSistema.class));
        objectConverterFactory.checkIn(converter);
    }

    @Override
    public DependenciaSistemaDTO queryDependenciaSistema(Long dependenciaSistemaId) throws
    Exception {
        ObjectConverter<DependenciaSistemaDTO, DependenciaSistema> converter =
        objectConverterFactory.checkOut();
        DependenciaSistemaDTO dependenciaSistemaDTO =
        converter.toTransfer(dependenciaSistemaService.queryDependenciaSistema(dependenciaSistemaId)
        , DependenciaSistemaDTO.class);
        objectConverterFactory.checkIn(converter);
        return dependenciaSistemaDTO;
    }

    @Override
    public List<TipoDependenciaDTO> obtenerTiposDependencias() throws Exception {
        ObjectConverter<TipoDependenciaDTO, TipoDependencia> converter =
        objectConverterFactory.checkOut();
        List<TipoDependenciaDTO> lista =
        converter.toTransferList(dependenciaSistemaService.obtenerTiposDependencias(),
        TipoDependenciaDTO.class);
        objectConverterFactory.checkIn(converter);
        return lista;
    }

    @Override
    public List<DependenciaSistemaDTO> dependenciaHijasList(DependenciaSistemaDTO
    dependenciaSistemaDTO) throws Exception {
        ObjectConverter<DependenciaSistemaDTO, DependenciaSistema> converter =
        objectConverterFactory.checkOut();
        //Convertimos el DTO a Entity
        ObjectConverter<DependenciaSistemaDTO, DependenciaSistema> converter1 =
        objectConverterFactory.checkOut();
        DependenciaSistema dependencia = converter1.toEntity(dependenciaSistemaDTO,
        DependenciaSistema.class);
        objectConverterFactory.checkIn(converter1);
        //
        List<DependenciaSistemaDTO> listDependenciaSistemaDTOs =
        converter.toTransferList(dependenciaSistemaService.dependenciaHijasList(dependencia),
        DependenciaSistemaDTO.class);
        objectConverterFactory.checkIn(converter);
        return listDependenciaSistemaDTOs;
    }
}
```

*Se crea interfaz **DependenciaSistemaService.java** que define métodos de CRUD en Service EJB para funcionalidad de pantalla*

```
@Local
public interface DependenciaSistemaService {

    //----- DependenciaSistema -----/

    List<DependenciaSistema> dependenciaSistemaList(int first, int pageSize, String
sortField, String sortOrder, Map<String, Object> filters) throws SystemException;

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

    DependenciaSistema guardarDependenciaSistema(DependenciaSistema dependenciaSistema)
throws SystemException;

    void editarDependenciaSistema(DependenciaSistema dependenciaSistema) throws
SystemException;

    void eliminarDependenciaSistema(DependenciaSistema dependenciaSistema) throws
SystemException;

    DependenciaSistema queryDependenciaSistema(Long dependenciaSistemaId) throws
SystemException;

    List<TipoDependencia> obtenerTiposDependencias() throws SystemException;

    List<DependenciaSistema> dependenciaHijasList(DependenciaSistema dependenciaSistema)
throws SystemException;
}
```

*Se crea **DependenciaSistemaServiceBean.java** donde implementan métodos CRUD en Service EJB para funcionalidad de pantalla*

```
@Stateless
public class DependenciaSistemaServiceBean extends StoreManagerImpl implements
DependenciaSistemaService {

    //----- Parametro Sistema -----//
    @Override
    public List<DependenciaSistema> dependenciaSistemaList(int first, int pageSize, String
sortField, String sortOrder, Map<String, Object> filters) throws SystemException {
        return
this.getStoreManager(AdministracionStoreManager.class).dependenciaSistemaList(first,
pageSize, sortField, sortOrder, filters);
    }

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

    @Override
```

```

    public DependenciaSistema guardarDependenciaSistema(DependenciaSistema
dependenciaSistema) throws SystemException {
        return this.crear(dependenciaSistema, DependenciaSistema.class);
    }

    @Override
    public void editarDependenciaSistema(DependenciaSistema dependenciaSistema) throws
SystemException {
        this.modificar(dependenciaSistema, DependenciaSistema.class);
    }

    @Override
    public void eliminarDependenciaSistema(DependenciaSistema dependenciaSistema) throws
SystemException {
        this.eliminar(dependenciaSistema, DependenciaSistema.class);
    }

    @Override
    public DependenciaSistema queryDependenciaSistema(Long dependenciaSistemaId) throws
SystemException {
        return this.buscar(dependenciaSistemaId, DependenciaSistema.class);
    }

    @Override
    public List<TipoDependencia> obtenerTiposDependencias() throws SystemException {
        return this.obtenerTodos(TipoDependencia.class);
    }

    @Override
    public List<DependenciaSistema> dependenciaHijasList(DependenciaSistema
dependenciaSistema) throws SystemException {
        return this.buscarPor(DependenciaSistema.class, "dependenciaPadreId",
dependenciaSistema);
    }
}

```

Internacionalización en español e inglés

#ES	
sb.menu.gestion.administracion.dependenciaSistema.header	=
Dependencias	
sb.menu.gestion.administracion.dependenciaSistema.breadcrumb	=
Dependencias	
sb.menu.gestion.administracion.dependenciaSistema.busquedaAvanzada	= Búsqueda
Avanzada	
sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia	= Nombre
Dependencia	
sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia.placeholder	= Ingrese el
nombre de la dependencia	
sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia.required	= Debe
ingresar el nombre de la dependencia	
sb.menu.gestion.administracion.dependenciaSistema.usuarioAdministrador	= Usuario
Administrador	
sb.menu.gestion.administracion.dependenciaSistema.tipoDependencia	= Tipo
Dependencia	

sb.menu.gestion.administracion.dependenciaSistema.tipoDependencia.required	= Debe
seleccionar el tipo de dependencia	
sb.menu.gestion.administracion.dependenciaSistema.vigente	= Vigente
sb.menu.gestion.administracion.dependenciaSistema.vigente.required	= Debe
seleccionar vigente	
sb.menu.gestion.administracion.dependenciaSistema.dependenciaPadre	=
Dependencia Padre	
sb.menu.gestion.administracion.dependenciaSistema.jerarquia	= Ver
Jerarquía Dependencia	
sb.menu.gestion.administracion.dependenciaSistema.jerarquiaDependencia	= JERARQUÍA
DEPENDENCIA	
sb.menu.gestion.administracion.dependenciaSistema.reporte	= Descargar
Reporte Dependencia	
# EN	
sb.menu.gestion.administracion.dependenciaSistema.header	=
Dependencies	
sb.menu.gestion.administracion.dependenciaSistema.breadcrumb	=
Dependencies	
sb.menu.gestion.administracion.dependenciaSistema.busquedaAvanzada	= Advannced
search	
sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia	= Name
Dependency	
sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia.placeholder	= Enter the
name of the dependency	
sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia.required	= You must
enter the name of the dependency	
sb.menu.gestion.administracion.dependenciaSistema.usuarioAdministrador	=
Administator	
sb.menu.gestion.administracion.dependenciaSistema.tipoDependencia	= Dependency
Type	
sb.menu.gestion.administracion.dependenciaSistema.tipoDependencia.required	= You must
select the dependency type	
sb.menu.gestion.administracion.dependenciaSistema.vigente	= Current
sb.menu.gestion.administracion.dependenciaSistema.vigente.required	= You must
select current	
sb.menu.gestion.administracion.dependenciaSistema.dependenciaPadre	= Father
Dependency	
sb.menu.gestion.administracion.dependenciaSistema.jerarquia	= See
Dependency Hierarchy	
sb.menu.gestion.administracion.dependenciaSistema.jerarquiaDependencia	= DEPENDENCY
HIERACHYS	
sb.menu.gestion.administracion.dependenciaSistema.reporte	= Download
Dependency Report	

Pestaña de tabla de visualización de la información en la pantalla de Dependencias

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 **dependencias** dentro del sistema y cuenta con funcionalidades avanzadas como **lazy loading**, **filtros**, **ordenamiento**, **selección de filas**, **expansión de filas**, y **acciones personalizadas**.

```
<!-- TABLA -->
<div class="row">
  <div class="col-lg-12" style="overflow-x: scroll;">
    <p:dataTable id="#{dependenciaSistemaViewController.modulo}Table"
      var="dependenciaSistema"
      value="#{dependenciaSistemaViewController.modelo}"
      rowKey="#{dependenciaSistema.dependenciaId}"

      selection="#{dependenciaSistemaConvController.selected}"
      selectionMode="single"
      paginator="true"
      rows="10"
      lazy="true"
      sortBy="#{dependenciaSistema.dependenciaId}"
      sortOrder="ASCENDING"

      currentPageReportTemplate="#{msg['sb.tabla.totalRegistros']} {totalRecords}"
      paginatorTemplate="{CurrentPageReport}
      {FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
      {RowsPerPageDropdown}"
      rowsPerPageTemplate="10,20,30"
      paginatorPosition="bottom"
      emptyMessage="#{msg['sb.datatable.registro.empty']}"

      widgetVar="items#{dependenciaSistemaViewController.modulo}"
    >

      <f:facet name="header">
        <ui:decorate template="../../template/_tableHeader.xhtml">
          <ui:define name="comandos">
            <ui:param name="sessionController"
              value="#{dependenciaSistemaViewController}" />
            <ui:param name="conversationController"
              value="#{dependenciaSistemaConvController}" />
            <ui:param name="btnBusqueda" value="N" />
            <ui:param name="buscarVisible" value="S" />
          </ui:define>
        </ui:decorate>
      </f:facet>
      <p:ajax event="rowSelect"

        listener="#{dependenciaSistemaConvController.onRowSelect}"

        update="#{dependenciaSistemaViewController.modulo}Table"
        process="@this"/>
      <p:ajax event="rowUnselect"

        listener="#{dependenciaSistemaConvController.onRowUnselect}"

        update="#{dependenciaSistemaViewController.modulo}Table"
        process="@this"/>

      <p:column style="text-align: center; padding: 5px;"
        width="5%">
        <p:rowToggler/>
      </p:column>

      <p:column
        headerText="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia']}"
        filterBy="#{dependenciaSistema.nombreDependencia}"
        sortBy="#{dependenciaSistema.nombreDependencia}"
```

```

filterMatchMode="contains"
filterStyle="display:none;"
width="30%"
>
<h:outputText value="#{dependenciaSistema.nombreDependencia}"/>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.dependenciaSistema.tipoDependencia']}"
filterBy="#{dependenciaSistema.tipoDependenciaId.descripcionValorDefinicion}"
sortBy="#{dependenciaSistema.tipoDependenciaId.descripcionValorDefinicion}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="15%"
>
<h:outputText
value="#{dependenciaSistema.tipoDependenciaId.descripcionValorDefinicion}"/>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.dependenciaSistema.usuarioAdministrador']}"
filterBy="#{dependenciaSistema.usuarioAdministrador.usuario}"
sortBy="#{dependenciaSistema.usuarioAdministrador.usuario}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="15%"
>
<h:outputText
value="#{dependenciaSistema.usuarioAdministrador.usuario}"/>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.dependenciaSistema.vigente']}"
filterBy="#{dependenciaSistema.vigente}"
sortBy="#{dependenciaSistema.vigente}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="10%"
>
<h:outputText styleClass="badge #{dependenciaSistema.vigente eq
'S' ? 'badge-success' : 'badge-danger'}"
value="#{dependenciaSistema.vigente
!= null
?
msg['listaSINO.'concat(dependenciaSistema.vigente)]
:
msg['listaSINO.N']}"/>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.dependenciaSistema.dependenciaPadre']}"
filterBy="#{dependenciaSistema.dependenciaPadreId.nombreDependencia}"
sortBy="#{dependenciaSistema.dependenciaPadreId.nombreDependencia}"
filterMatchMode="contains"
filterStyle="display:none;"

```

```

                                style="text-align: center"
                                width="30%"
                                >
                                <h:outputText
value="#{dependenciaSistema.dependenciaPadreId.nombreDependencia}" />
                                </p:column>
                                <p:column headerText="#{msg['sb.component.boton.opciones']}"
                                style="text-align: center"
                                width="10%"
                                >
                                <p:commandLink id="btnJeraquiaDependencia"
                                style="color: #fff;"

title="#{msg['sb.menu.gestion.administracion.dependenciaSistema.jerarquia']}"
                                update="formJerarquia"
                                process="@this"
                                styleClass="btn btn-xs btn-primary

sbTooltip"

actionListener="#{dependenciaSistemaConvController.cargarListaJerarquia(dependenciaSistema)}"

oncomplete="$('#dlgJerarquia').modal('show');"

                                >
                                <i class="fa fa-sitemap"></i>
                                </p:commandLink>
                                <p:spacer />
                                <p:commandLink style="color: #fff;"
                                process="@this"

title="#{msg['sb.menu.gestion.administracion.dependenciaSistema.reporte']}"
                                styleClass="btn btn-xs btn-primary

sbTooltip"

actionListener="#{dependenciaSistemaConvController.generarReporte(dependenciaSistema)}"

                                immediate="true"
                                ajax="false"
                                >
                                <i class="fa fa-file-pdf-o"></i>
                                </p:commandLink>
                                </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['sb.menu.gestion.id']}" />
                                <h:outputText
value="#{dependenciaSistema.dependenciaId}" />

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

```

```

value="#{msg['sb.menu.gestion.fechaCreacion']}" />
value="#{dependenciaSistema.fechaCreacion}">
                                <f:convertDateTime
                                locale="es" pattern="dd/MM/yyyy"/>
                                </h:outputText>
                                <h:outputText
                                value="#{msg['sb.menu.gestion.usuarioModificador']}" />
                                <h:outputText
                                value="#{dependenciaSistema.usuarioModificador}" />
                                <h:outputText
                                value="#{msg['sb.menu.gestion.fechaModificacion']}" />
                                <h:outputText
                                value="#{dependenciaSistema.fechaModificacion}">
                                <f:convertDateTime
                                locale="es" pattern="dd/MM/yyyy"/>
                                </h:outputText>
                                </p:panelGrid>
                                </div>
                                </div>
                                <p:spacer/>
                                </div>
                                </p:rowExpansion>
                                </p:dataTable>
                                </div>
                                </div>

```

Estructura General

La tabla se encuentra dentro de un <div> con una clase de Bootstrap para asegurar que sea responsiva.

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

- **overflow-x: scroll;** permite el desplazamiento horizontal si la tabla es más ancha que el contenedor.

El componente base de **PrimeFaces** que define la tabla es **<p:dataTable>**.

Propiedades Principales de <p:dataTable>

```

<p:dataTable id="#{dependenciaSistemaViewController.modulo}Table"
var="dependenciaSistema"
value="#{dependenciaSistemaViewController.modelo}"
rowKey="#{dependenciaSistema.dependenciaId}"
selection="#{dependenciaSistemaConvController.selected}"
selectionMode="single"
paginator="true"
rows="10"
lazy="true"
sortBy="#{dependenciaSistema.dependenciaId}"

```

```
sortOrder="ASCENDING"
currentPageReportTemplate="#{msg['sb.tabla.totalRegistros']} {totalRecords}"
paginatorTemplate="{CurrentPageReport} {FirstPageLink} {PreviousPageLink}
{PageLinks} {NextPageLink} {LastPageLink} {RowsPerPageDropdown}"
rowsPerPageTemplate="10,20,30"
paginatorPosition="bottom"
emptyMessage="#{msg['sb.datatable.registro.empty']}"
widgetVar="items#{dependenciaSistemaViewController.modulo}">
```

Explicación de cada atributo:

Propiedad	Descripción
id	Identificador único de la tabla, dinámico según el módulo.
var	Variable de iteración que representa cada objeto de la lista de datos.
value	Fuente de datos (List<DependenciaSistema>) obtenida del controlador.
rowKey	Identificador único de cada fila basado en dependenciald.
selection	Almacena la fila seleccionada en dependenciaSistemaConvController.selected.
selectionMode="single"	Permite la selección de una única fila.
paginator="true"	Activa la paginación.
rows="10"	Define 10 filas por página por defecto.
lazy="true"	Habilita la carga diferida (usando LazyDataModel).
sortBy="#{dependenciaSistema.dependenciald}"	Define el ordenamiento inicial por dependenciald.
sortOrder="ASCENDING"	Orden ascendente por defecto.

Propiedad	Descripción
paginatorTemplate	Personaliza la barra de paginación.
rowsPerPageTemplate="10,20,30"	Permite seleccionar la cantidad de filas por página.
paginatorPosition="bottom"	Ubica la paginación en la parte inferior.
emptyMessage	Mensaje cuando no hay registros disponibles.
widgetVar	Permite manejar la tabla desde JavaScript usando PF('itemsXXX').

Eventos de Selección de Filas

```
<p:ajax event="rowSelect"
  listener="#{dependenciaSistemaConvController.onRowSelect}"
  update="#{dependenciaSistemaViewController.modulo}Table"
  process="@this"/>

<p:ajax event="rowUnselect"
  listener="#{dependenciaSistemaConvController.onRowUnselect}"
  update="#{dependenciaSistemaViewController.modulo}Table"
  process="@this"/>
```

- **rowSelect:** Llama a onRowSelect() en dependenciaSistemaConvController cuando el usuario selecciona una fila.
- **rowUnselect:** Llama a onRowUnselect() cuando el usuario deselectiona una fila.
- **update:** Refresca la tabla tras la selección.
- **process="@this":** Solo procesa el componente que disparó el evento.

Columnas de la Tabla

Cada <p:column> define una columna con datos de la entidad DependenciaSistema.

Ejemplo:

```
<p:column
  headerText="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia']}"
```

```
filterBy="#{dependenciaSistema.nombreDependencia}"
sortBy="#{dependenciaSistema.nombreDependencia}"
filterMatchMode="contains"
filterStyle="display:none;"
width="30%"
<h:outputText value="#{dependenciaSistema.nombreDependencia}"/>
</p:column>
```

Explicación de los atributos:

Propiedad	Descripción
headerText	Nombre de la columna obtenido desde un archivo de mensajes.
filterBy	Activa el filtrado basado en nombreDependencia.
sortBy	Activa el ordenamiento.
filterMatchMode="contains"	Filtrado con coincidencia parcial.
filterStyle="display:none;"	Oculto los filtros en la UI.
width="30%"	Define el ancho de la columna.

Se repite este patrón para otras columnas como **tipoDependenciaId.descripcionValorDefinicion**, **usuarioAdministrador.usuario**, vigente, etc.

Formato Condicional

```
<h:outputText styleClass="badge #{dependenciaSistema.vigente eq 'S' ? 'badge-success' :
'badge-danger'}"
value="#{dependenciaSistema.vigente != null
? msg['listaSINO.'.concat(dependenciaSistema.vigente)]
: msg['listaSINO.N']}" />
```

- **Asigna clases dinámicamente:**
 - 'badge-success' si vigente es 'S' (activo).
 - 'badge-danger' si vigente es 'N' (inactivo).
- Traduce el valor con `msg['listaSINO.X']`, donde X es S o N.

Acciones en las Filas

```
<p:commandLink id="btnJeraquiaDependencia"
    style="color: #fff;"
    title="#{msg['sb.menu.gestion.administracion.dependenciaSistema.jerarquia']}"
    update="formJerarquia"
    process="@this"
    styleClass="btn btn-xs btn-primary sbTooltip"

    actionListener="#{dependenciaSistemaConvController.cargarListaJerarquia(dependenciaSistema)}"
    "
    oncomplete="$('#dlgJerarquia').modal('show');">
    <i class="fa fa-sitemap"></i>
</p:commandLink>
```

- Botón para mostrar jerarquía (fa-sitemap):
 - Ejecuta **cargarListaJerarquia(dependenciaSistema)**.
 - Muestra un modal con **\$('#dlgJerarquia').modal('show');**.

Otro botón genera un **reporte en PDF** llamando a **generarReporte(dependenciaSistema)** con **ajax="false"**.

Expansión de Filas

```
<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['sb.menu.gestion.id']}" />
                    <h:outputText value="#{dependenciaSistema.dependenciaId}" />

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

- Permite **expandir filas** para ver detalles adicionales.
- Usa un **p:panelGrid** con fecha formateada.

El siguiente código implementa una tabla eficiente en **PrimeFaces** con paginación, ordenamiento, filtrado, selección de filas, expansión y acciones personalizadas.

- Botón de Filtro por Búsqueda Avanzada

```
<!-- FILTRO AVANZADO -->
<h:panelGroup layout="block" id="paramFiltro" class="row">
  <div class="col-lg-12">
    <ui:fragment>
      <button type="button" class="btn btn-white btn-bitbucket" data-
toggle="collapse" data-target="#demo"><i class="fa fa-filter"></i>
#{msg['sb.menu.gestion.administracion.dependenciaSistema.busquedaAvanzada']}</button>
    </ui:fragment>
    <div id="demo" class="collapse #{dependenciaSistemaViewController.filtroActivo
? 'in' : ''} well form-horizontal" style="padding: 20px">
      <div class="row">
        <!-- Nombre Dependencia -->
        <div class="col-lg-6">
          <div class="form-group">
            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia']}"
control-label"
styleClass="col-lg-6"
for="nombreDependenciaBusq"/>
            <div class="col-lg-6">
              <p:inputText id="nombreDependenciaBusq"
value="#{dependenciaSistemaViewController.dependenciaSistemaFilterBusq.nombreDependencia}"
placeholder="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia.plac
eholder']}"
styleClass="form-control"
/>
            </div>
          </div>
        <!-- Usuario Administrador -->
        <div class="col-lg-6">
          <div class="form-group">
            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.usuarioAdministrador']}"
control-label"
styleClass="col-lg-6"
for="usuarioAdministradorBusq-DS"/>
            <div class="col-lg-6">
              <p:selectOneMenu
id="usuarioAdministradorBusq-DS"
value="#{dependenciaSistemaViewController.dependenciaSistemaFilterBusq.usuarioAdministrador}
"
converter="usuarioConverter"
filter="true"
filterMatchMode="contains"
>
              <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</h:panelGroup>
```

```

value="#{dependenciaSistemaViewController.listUsuario}"
var="usuario"
itemLabel="#{usuario.usuario}"
itemValue="#{usuario}"/>
</p:selectOneMenu>
</div>
</div>
</div>
<div class="row">
<!-- Tipo Dependencia -->
<div class="col-lg-6">
<div class="form-group">
<p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.tipoDependencia']}"
control-label"
styleClass="col-lg-6"
for="tipoDependenciaBusq-dependenciaSistema"/>
<div class="col-lg-6">
<p:selectOneMenu
id="tipoDependenciaBusq-dependenciaSistema"
value="#{dependenciaSistemaViewController.dependenciaSistemaFilterBusq.tipoDependenciaId}"
converter="valorDefinicionConverter"
<f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
<f:selectItems
value="#{dependenciaSistemaViewController.listTipoDependencia}"
var="valorDef"
itemLabel="#{valorDef.descripcionValorDefinicion}"
itemValue="#{valorDef}"/>
</p:selectOneMenu>
</div>
</div>
<!-- Vigente-->
<div class="col-lg-6">
<div class="form-group">
<p:outputLabel value="Vigente"
control-label"
styleClass="col-lg-6"
for="vigenteBusq"/>
<div class="col-lg-6">
<p:selectOneMenu id="vigenteBusq"
value="#{dependenciaSistemaViewController.dependenciaSistemaFilterBusq.vigente}"
<f:selectItems
value="#{configApplController.listaSINO}"/>
</p:selectOneMenu>
</div>

```

```

        </div>
    </div>
</div>
<div class="row">
    <!-- Dependencia Padre -->
    <div class="col-lg-12">
        <div class="form-group">
            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.dependenciaPadre']}"
styleClass="col-lg-3
control-label"
for="dependenciaPadreBusq"/>
            <div class="col-lg-9">
                <p:selectOneMenu
id="dependenciaPadreBusq"
value="#{dependenciaSistemaViewController.dependenciaSistemaFilterBusq.dependenciaPadreId}"
converter="dependenciaSistemaConverter"
filter="true"
autoWidth="false"
effect="fade"
panelStyle="width:690px"
filterMatchMode="contains"
                <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
                <f:selectItems
value="#{dependenciaSistemaViewController.listDependenciaSistema}"
var="dependencia"
itemLabel="#{dependencia.nombreDependencia}"
itemValue="#{dependencia}"/>
            </p:selectOneMenu>
        </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="paramFiltro"
actionListener="#{dependenciaSistemaConvController.lazyLoad}"
update="#{dependenciaSistemaViewController.modulo}Table"
styleClass="btn btn btn-primary pull-left"

```

```

onclick="PF('items#{dependenciaSistemaViewController.modulo}').clearFilters()"
    >
    <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="paramFiltro,#{dependenciaSistemaViewController.modulo}Table"
process="@this"
actionListener="#{dependenciaSistemaConvController.limpiarFiltros}"
onclick="PF('items#{dependenciaSistemaViewController.modulo}').clearFilters()"
styleClass="btn btn btn-primary pull-right"
    >
    <i class="fa fa-times"></i>

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

```

Este código define una sección de filtro avanzado para la búsqueda de dependencias dentro del sistema de administración. Se implementa mediante componentes JSF y PrimeFaces, permitiendo a los usuarios realizar consultas refinadas con distintos criterios.

Estructura del Código

1. Botón de Desplegado del Filtro Avanzado

- Se utiliza un botón HTML **<button>** con la clase **btn btn-white btn-bitbucket**.
- El botón tiene un **data-toggle="collapse"** que controla la visibilidad del panel de filtros avanzados (**#demo**).
- El texto del botón se obtiene desde un archivo de mensajes **msg['sb.menu.gestion.administracion.dependenciaSistema.búsquedaAvanzada']**.

2. Panel de Filtros (<div id="demo" class="collapse ...">)

- La clase **collapse** permite ocultar/mostrar el panel de filtros.
- Se usa EL **#{dependenciaSistemaViewController.filtroActivo ? 'in' : ''}** para mantener el estado del panel abierto o cerrado según el estado de la variable **filtroActivo**.
- Se agrupan los filtros dentro de una estructura **well form-horizontal** para darle formato.

3. Filtros Disponibles

- **Nombre Dependencia:** Campo de texto (<p:inputText>) para ingresar el nombre de la dependencia a buscar.
- **Usuario Administrador:** Menú desplegable (<p:selectOneMenu>) con opciones filtrables.
- **Tipo Dependencia:** Menú desplegable para seleccionar el tipo de dependencia.
- **Vigencia:** Menú desplegable para filtrar por estado vigente.
- **Dependencia Padre:** Menú desplegable con filtro de búsqueda, permitiendo seleccionar la dependencia padre.

4. Botones de Acción

- **Botón de Buscar (<p:commandLink id="buscar">)**
 - Ejecuta el método **lazyLoad** en controlador **dependenciaSistemaConvController**.
 - Actualiza la tabla de resultados definida por **#{dependenciaSistemaViewController.modulo}Table**.
 - Limpia los filtros visuales con **PF('items#{dependenciaSistemaViewController.modulo}').clearFilters()**.
- **Botón de Limpiar (<p:commandLink id="limpiar">)**
 - Llama al método **limpiarFiltros** para restablecer los valores de búsqueda.

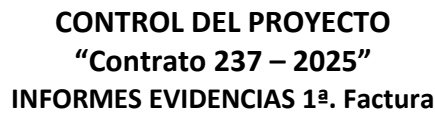
- Actualiza tanto el panel de filtros como la tabla de resultados.
- Limpia los filtros visuales.

Funcionamiento

1. El usuario presiona el botón de "Búsqueda Avanzada".
2. Se despliega el panel con los filtros.
3. Tras ingresar los criterios de búsqueda, el usuario puede:
 - **Buscar:** Filtrar la tabla según los criterios ingresados.
 - **Limpiar:** Restablecer los filtros a su estado inicial.
4. La tabla de resultados se actualiza automáticamente según los filtros aplicados.

Formulario de captura de datos para registro de Dependencias

```
<!--PESTAÑA GUARDAR/EDITAR -->
<p:tab title="#{msg['sb.component.tabview.guardarEditar']}">
  <p:panel id="formularioEditarCrear">
    <div class="ibox-content">
      <div class="form-horizontal">
        <div class="row">
          <div class="col-lg-12">
            <div class="form-group">
              <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia']}"
styleClass="col-lg-3 control-label"
align: right"
style="text-align: right;"
for="nombreDependencia"/>
            <div class="col-lg-9">
              <p:inputText
id="nombreDependencia"
value="#{dependenciaSistemaViewController.selected.nombreDependencia}"
required="true"
maxLength="100"
placeholder="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia.placeholder']}"
requiredMessage="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia.required']}"
              />
            </div>
          </div>
        </div>
      </div>
    </div>
  </p>
</p>
</tab>
```



```

styleClass="form-control">
</div>
</div>
</div>
<div class="row">
<div class="col-lg-6">
<div class="form-group">
<p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.tipoDependencia']}"
styleClass="col-lg-6 control-label"
align: right"
style="text-align: right;"
for="tipoDependencia-dependenciaSistema"/>
<div class="col-lg-6">
<p:selectOneMenu
id="tipoDependencia-dependenciaSistema"
value="#{dependenciaSistemaViewController.selected.tipoDependenciaId}"
required="true"
autoWidth="false"
effect="fade"
requiredMessage="#{msg['sb.menu.gestion.administracion.dependenciaSistema.tipoDependencia.required']}"
converter="valorDefinicionConverter"
>
<f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
<f:selectItems
value="#{dependenciaSistemaViewController.listTipoDependencia}"
var="tipoDep"
itemLabel="#{tipoDep.descripcionValorDefinicion}"
itemValue="#{tipoDep}"
/>
</p:selectOneMenu>
</div>
</div>
<div class="col-lg-6">
<div class="form-group">
<p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.usuarioAdministrador']}"
styleClass="col-lg-6 control-label"
align: right"
style="text-align: right;"

```

```

for="usuarioAdministrador-DS"/>
                                <div class="col-lg-6">
                                    <p:selectOneMenu
id="usuarioAdministrador-DS"
                                value="#{dependenciaSistemaViewController.selected.usuarioAdministrador}"
                                required="true"

requiredMessage="#{msg['sb.menu.gestion.administracion.dependenciaSistema.tipoDependencia.re
quired']}"

                                converter="usuarioConverter"
                                autoWidth="false"
                                effect="fade"
                                panelStyle="width:215px"
                                filter="true"
                                filterMatchMode="contains"
                                >
                                <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
                                <f:selectItems
value="#{dependenciaSistemaViewController.listUsuario}"
                                var="user"
                                itemLabel="#{user.usuario}"
                                itemValue="#{user}"
                                />
                                </p:selectOneMenu>
                                </div>
                                </div>
                                </div>
                                <div class="row">
                                    <div class="col-lg-12">
                                        <div class="form-group">
                                            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.dependenciaPadre']}"
styleClass="col-lg-3 control-label"
align: right"
style="text-align: right;"
for="dependenciaPadre"/>
                                <div class="col-lg-9">
                                    <p:selectOneMenu
id="dependenciaPadre"
                                value="#{dependenciaSistemaViewController.selected.dependenciaPadreId}"

```

```

        required="false"
        autoWidth="false"
        effect="fade"
        panelStyle="width:625px"
        converter="dependenciaSistemaConverter"
        filter="true"
        filterMatchMode="contains"
    >
itemLabel="#{msg['sb.selectOneMenu.registro']}">
value="#{dependenciaSistemaViewController.listDependenciaSistema}"
        var="dependencia"
        itemLabel="#{dependencia.nombreDependencia}"
        itemValue="#{dependencia}"
    />
</p:selectOneMenu>
</div>
</div>
</div>
<div class="row">
    <div class="col-lg-6">
        <div class="form-group">
            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.vigente']}"
styleClass="col-lg-6 control-label"
align: right"
for="vigente"/>
            <div class="col-lg-6">
                <p:selectOneMenu id="vigente"
value="#{dependenciaSistemaViewController.selected.vigente}"
required="true"
requiredMessage="#{msg['sb.menu.gestion.administracion.dependenciaSistema.vigente.required']}"
>
value="#{configApplController.listaSINO}"
    />
</p:selectOneMenu>

```

```

                </div>
            </div>
        </div>

        <div class="hr-line-dashed"></div>

        <ui:decorate template="../../template/_formButtons.xhtml">
            <ui:define name="comandos">
                <ui:param name="sessionController"
value="#{dependenciaSistemaViewController}" />
                <ui:param name="conversationController"
value="#{dependenciaSistemaConvController}" />
            </ui:define>
        </ui:decorate>
    </div>
</div>
</p:panel>
</p:tab>

```

Este código representa una pestaña dentro de un componente **p:tabView** en **PrimeFaces**, utilizada para capturar o editar información sobre dependencias del sistema en una aplicación JSF.

La interfaz incluye campos de entrada de texto y listas desplegables para ingresar los detalles de una dependencia. La información es manejada por **dependenciaSistemaViewController**, que actúa como el controlador de la vista.

Estructura del Código

Pestaña Principal

```
<p:tab title="#{msg['sb.component.tabview.guardarEditar']}">
```

Define la pestaña con un título que es obtenido del archivo de mensajes **msg**.

Panel Principal

```
<p:panel id="formularioEditarCrear">
```

El panel encapsula todos los campos y controles de la pestaña, asegurando una organización adecuada de los elementos.

Campos de Captura

Campo "Nombre de Dependencia"

```

<p:outputLabel
value="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia']}"
styleClass="col-lg-3 control-label"

```

```

        style="text-align: right"
        for="nombreDependencia"/>
<p:inputText id="nombreDependencia"
        value="#{dependenciaSistemaViewController.selected.nombreDependencia}"
        required="true"
        maxLength="100"

placeholder="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia.plac
eholder']}"

requiredMessage="#{msg['sb.menu.gestion.administracion.dependenciaSistema.nombreDependencia.
required']}"
        styleClass="form-control">
</p:inputText>

```

- **p:outputLabel:** Etiqueta del campo.
- **p:inputText:** Campo de entrada de texto.
 - Está vinculado a **dependenciaSistemaViewController.selected.nombreDependencia.**
 - Requiere un valor obligatorio (**required="true"**).
 - Se muestra un mensaje de error si el campo está vacío.

Campo "Tipo de Dependencia"

```

<p:selectOneMenu id="tipoDependencia-dependenciaSistema"
        value="#{dependenciaSistemaViewController.selected.tipoDependenciaId}"
        required="true"
        converter="valorDefinicionConverter">
    <f:selectItem itemLabel="#{msg['sb.selectOneMenu.registro']}" />
    <f:selectItems value="#{dependenciaSistemaViewController.listTipoDependencia}"
        var="tipoDep"
        itemLabel="#{tipoDep.descripcionValorDefinicion}"
        itemValue="#{tipoDep}" />
</p:selectOneMenu>

```

- **p:selectOneMenu:** Lista desplegable para seleccionar el tipo de dependencia.
- Se usa un convertidor **valorDefinicionConverter** para manejar la conversión de objetos.
- Se obtiene la lista de tipos de dependencia desde **dependenciaSistemaViewController.listTipoDependencia.**

Campo "Usuario Administrador"

```

<p:selectOneMenu id="usuarioAdministrador-DS"
        value="#{dependenciaSistemaViewController.selected.usuarioAdministrador}"
        required="true"
        converter="usuarioConverter"

```

```

        filter="true"
        filterMatchMode="contains">
<f:selectItem itemLabel="#{msg['sb.selectOneMenu.registro']}" />
<f:selectItems value="#{dependenciaSistemaViewController.listUsuario}"
        var="user"
        itemLabel="#{user.usuario}"
        itemValue="#{user}" />
</p:selectOneMenu>

```

- Lista desplegable que permite seleccionar un usuario administrador.
- Se usa el convertidor **usuarioConverter**.
- Tiene un filtro activado (**filter="true"**) para facilitar la búsqueda.

Campo "Dependencia Padre"

```

<p:selectOneMenu id="dependenciaPadre"
        value="#{dependenciaSistemaViewController.selected.dependenciaPadreId}"
        required="false"
        converter="dependenciaSistemaConverter"
        filter="true"
        filterMatchMode="contains">
<f:selectItem itemLabel="#{msg['sb.selectOneMenu.registro']}" />
<f:selectItems value="#{dependenciaSistemaViewController.listDependenciaSistema}"
        var="dependencia"
        itemLabel="#{dependencia.nombreDependencia}"
        itemValue="#{dependencia}" />
</p:selectOneMenu>

```

- Lista desplegable para seleccionar la dependencia padre.
- Se usa **dependenciaSistemaConverter** para conversión.
- Tiene filtro para búsqueda rápida.

Campo "Vigente"

```

<p:selectOneMenu id="vigente"
        value="#{dependenciaSistemaViewController.selected.vigente}"
        required="true">
<f:selectItems value="#{configApplController.listaSINO}" />
</p:selectOneMenu>

```

- Permite seleccionar si la dependencia está vigente.
- La lista de opciones proviene de **configApplController.listaSINO**.

Botones de Acción

```

<ui:decorate template="../../template/_formButtons.xhtml">
<ui:define name="comandos">

```

```
<ui:param name="sessionController" value="#{dependenciaSistemaViewController}" />
<ui:param name="conversationController" value="#{dependenciaSistemaConvController}"
/>
</ui:define>
</ui:decorate>
```

- Se incluyen los botones de guardar y cancelar desde una plantilla **_formButtons.xhtml**.
- Se pasan como parámetros los controladores **sessionController** y **conversationController**.

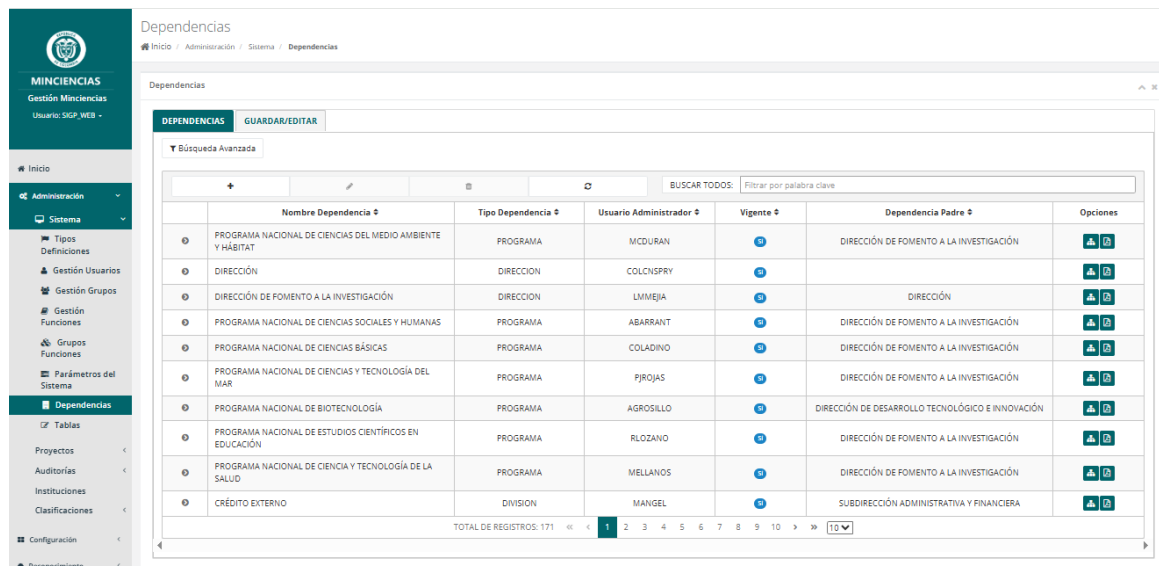
Resumen

Este formulario permite capturar y editar la información de una dependencia del sistema.

- Se compone de etiquetas, campos de entrada y listas desplegables.
- Utiliza **p:selectOneMenu** con filtros para facilitar la selección.
- Los datos son manejados por **dependenciaSistemaViewController**.
- Se incluyen validaciones y mensajes de error para los campos obligatorios.
- Se usan convertidores personalizados para transformar los objetos en la vista.

Esta estructura garantiza una interfaz intuitiva y validada para la gestión de dependencias del sistema.

Visualización de la pantalla



The screenshot shows the 'Dependencias' application interface. On the left is a sidebar menu with options like 'Inicio', 'Administración', 'Sistema', 'Tipos', 'Definiciones', 'Gestión Usuarios', 'Gestión Grupos', 'Gestión Funciones', 'Grupos', 'Funciones', 'Parámetros del Sistema', 'Dependencias', 'Tablas', 'Proyectos', 'Auditorías', 'Instituciones', 'Clasificaciones', and 'Configuración'. The main content area is titled 'Dependencias' and includes a search bar and a table of dependencies.

	Nombre Dependencia	Tipo Dependencia	Usuario Administrador	Vigente	Dependencia Padre	Opciones
0	PROGRAMA NACIONAL DE CIENCIAS DEL MEDIO AMBIENTE Y HÁBITAT	PROGRAMA	MCDURAN	01	DIRECCIÓN DE FOMENTO A LA INVESTIGACIÓN	 
0	DIRECCIÓN	DIRECCION	COLCHSPRY	01		 
0	DIRECCIÓN DE FOMENTO A LA INVESTIGACIÓN	DIRECCION	LMMEJIA	01	DIRECCIÓN	 
0	PROGRAMA NACIONAL DE CIENCIAS SOCIALES Y HUMANAS	PROGRAMA	ABARRANT	01	DIRECCIÓN DE FOMENTO A LA INVESTIGACIÓN	 
0	PROGRAMA NACIONAL DE CIENCIAS BÁSICAS	PROGRAMA	COLADINO	01	DIRECCIÓN DE FOMENTO A LA INVESTIGACIÓN	 
0	PROGRAMA NACIONAL DE CIENCIAS Y TECNOLOGÍA DEL MAR	PROGRAMA	PJROJAS	01	DIRECCIÓN DE FOMENTO A LA INVESTIGACIÓN	 
0	PROGRAMA NACIONAL DE BIOTECNOLOGÍA	PROGRAMA	AGROSILLO	01	DIRECCIÓN DE DESARROLLO TECNOLÓGICO E INNOVACIÓN	 
0	PROGRAMA NACIONAL DE ESTUDIOS CIENTÍFICOS EN EDUCACIÓN	PROGRAMA	RLOZANO	01	DIRECCIÓN DE FOMENTO A LA INVESTIGACIÓN	 
0	PROGRAMA NACIONAL DE CIENCIA Y TECNOLOGÍA DE LA SALUD	PROGRAMA	MELLANOS	01	DIRECCIÓN DE FOMENTO A LA INVESTIGACIÓN	 
0	CRÉDITO EXTERNO	DIVISION	MANGEL	01	SUBDIRECCIÓN ADMINISTRATIVA Y FINANCIERA	 

TOTAL DE REGISTROS: 171

Bienvenido Sistema de Gestión de Proyectos - SIGP - SIGP_WEB

Última conexión: 13-03-2025 18:03:15

Fecha/Hora

Cerrar sesión

MINCIENCIAS
Gestión Minciencias
Usuario: SIGP_WEB

Inicio

Administración

Sistema

Tipos Definiciones

Gestión Usuarios

Gestión Grupos

Gestión Funciones

Grupos Funciones

Parámetros del Sistema

Dependencias

Dependencias

DEPENDENCIAS GUARDAR/EDITAR

Nombre Dependencia * Ingrese el nombre de la dependencia

Tipo Dependencia * Seleccione...

Usuario Administrador * Seleccione...

Dependencia Padre Seleccione...

Vigente * Seleccione...

Guardar

Creación pantalla de Administración de Tablas

Función:

Gestiona las tablas utilizadas en el sistema, permitiendo su configuración y administración.

Principales funcionalidades:

- **Listado de Tablas:** Muestra todas las tablas registradas.
- **Creación y Edición:** Permite registrar nuevas tablas o modificar sus propiedades.
- **Relación con Dependencias:** Se pueden asociar tablas con dependencias específicas.

Creación nuevo menú de configuración de Tablas

```
-- funciones (menus)
insert into 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', 'GEST6181', 'Tablas', 'F', null, null,
'/app/administracion/tablas.xhtml', null, 'GEST6100', 'fa fa-edit', 'SIGP_WEB', to_date('15-
04-2024 15:06:55', 'dd-mm-yyyy hh24:mi:ss'), 'SIGP_WEB', to_date('18-04-2024 16:55:11', 'dd-
mm-yyyy hh24:mi:ss'));

-- derechos funciones (permisos)
insert into sb_derechos_funciones (DERECHO_FUNCION_ID, APLICACION, FUNCION, GRUPO, ESTADO,
USUARIO_CREADOR, FECHA_CREACION, USUARIO_MODIFICADOR, FECHA_MODIFICACION)
```

```
values (null, 'SIGP_WEB', 'GEST6181', 'SGPS_ADMINISTRADOR', 'H', 'SIGP_WEB', to_date('15-04-2024 15:07:20', 'dd-mm-yyyy hh24:mi:ss'), null, null);
```

*Mapear nueva entidad (Tabla.java) de la tabla **SB_TABLAS** en la base de datos sigp*

```
/*
 * 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.business.domain;

import com.sb.colciencias.gestion.business.domain.security.Aplicacion;
import com.sb.colciencias.gestion.business.valueObjects.AppSessionVO;
import java.io.Serializable;
import java.util.Date;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.Transient;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author dcangrejo
 */
@Entity
@Table(name = "SB_TABLAS", schema = "HERCULES")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Tabla.findAll", query = "SELECT s FROM Tabla s")
    , @NamedQuery(name = "Tabla.findByTablaId", query = "SELECT s FROM Tabla s WHERE s.tablaId = :tablaId")
    , @NamedQuery(name = "Tabla.findByAplicacion", query = "SELECT s FROM Tabla s WHERE s.aplicacion = :aplicacion")
    , @NamedQuery(name = "Tabla.findByTabla", query = "SELECT s FROM Tabla s WHERE s.tabla = :tabla")
    , @NamedQuery(name = "Tabla.findByDescripcionTabla", query = "SELECT s FROM Tabla s WHERE s.descripcionTabla = :descripcionTabla")
    , @NamedQuery(name = "Tabla.findBySecuencia", query = "SELECT s FROM Tabla s WHERE s.secuencia = :secuencia")
    , @NamedQuery(name = "Tabla.findByRegistraAuditoria", query = "SELECT s FROM Tabla s WHERE s.registraAuditoria = :registraAuditoria")
    , @NamedQuery(name = "Tabla.findByRegistroAntes", query = "SELECT s FROM Tabla s WHERE s.registroAntes = :registroAntes")
})
```

```
, @NamedQuery(name = "Tabla.findByRegistroDespues", query = "SELECT s FROM Tabla s WHERE
s.registroDespues = :registroDespues")
, @NamedQuery(name = "Tabla.findByCamposDescripcion", query = "SELECT s FROM Tabla s
WHERE s.camposDescripcion = :camposDescripcion")
, @NamedQuery(name = "Tabla.findByUsuarioCreador", query = "SELECT s FROM Tabla s WHERE
s.usuarioCreador = :usuarioCreador")
, @NamedQuery(name = "Tabla.findByFechaCreacion", query = "SELECT s FROM Tabla s WHERE
s.fechaCreacion = :fechaCreacion")
, @NamedQuery(name = "Tabla.findByUsuarioModificador", query = "SELECT s FROM Tabla s
WHERE s.usuarioModificador = :usuarioModificador")
, @NamedQuery(name = "Tabla.findByFechaModificacion", query = "SELECT s FROM Tabla s
WHERE s.fechaModificacion = :fechaModificacion"))}
public class Tabla implements Serializable {

    private static final long serialVersionUID = 1L;

    @Id
    @Basic(optional = false)
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "SB_TABLA_ID_SEQ")
    @SequenceGenerator(name = "SB_TABLA_ID_SEQ", sequenceName = "SB_TABLA_ID_SEQ",
allocationSize = 1)
    @Column(name = "TABLA_ID")
    @NotNull
    private Long tablaId;

    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 30)
    @Column(name = "TABLA")
    private String tabla;

    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 200)
    @Column(name = "DESCRIPCION_TABLA")
    private String descripcionTabla;

    @Size(max = 30)
    @Column(name = "SECUENCIA")
    private String secuencia;

    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 1)
    @Column(name = "REGISTRA_AUDITORIA")
    private String registraAuditoria;

    @Size(max = 1)
    @Column(name = "REGISTRO_ANTES")
    private String registroAntes;

    @Size(max = 1)
    @Column(name = "REGISTRO_DESPUES")
    private String registroDespues;

    @Size(max = 200)
    @Column(name = "CAMPOS_DESCRIPCION")
    private String camposDescripcion;

    @Size(max = 30)
    @Column(name = "USUARIO_CREADOR")
```

```
private String usuarioCreador;

@Column(name = "FECHA_CREACION")
@Temporal(TemporalType.TIMESTAMP)
private Date fechaCreacion;

@Size(max = 30)
@Column(name = "USUARIO_MODIFICADOR")
private String usuarioModificador;

@Column(name = "FECHA_MODIFICACION")
@Temporal(TemporalType.TIMESTAMP)
private Date fechaModificacion;

//Joins
@JoinColumn(name = "APLICACION", referencedColumnName = "APLICACION")
@ManyToOne(optional = false)
private Aplicacion aplicacion;

//Auditoria
@Transient
private AppSessionVO appSessionVO;

public Tabla() {
}

public Tabla(Long tablaId) {
    this.tablaId = tablaId;
}

public Tabla(Long tablaId, Aplicacion aplicacion, String tabla, String descripcionTabla,
String registraAuditoria) {
    this.tablaId = tablaId;
    this.aplicacion = aplicacion;
    this.tabla = tabla;
    this.descripcionTabla = descripcionTabla;
    this.registraAuditoria = registraAuditoria;
}

public Long getTablaId() {
    return tablaId;
}

public void setTablaId(Long tablaId) {
    this.tablaId = tablaId;
}

public Aplicacion getAplicacion() {
    return aplicacion;
}

public void setAplicacion(Aplicacion aplicacion) {
    this.aplicacion = aplicacion;
}

public String getTabla() {
    return tabla;
}

public void setTabla(String tabla) {
    this.tabla = tabla;
}
```

```
}

public String getDescripcionTabla() {
    return descripcionTabla;
}

public void setDescripcionTabla(String descripcionTabla) {
    this.descripcionTabla = descripcionTabla;
}

public String getSecuencia() {
    return secuencia;
}

public void setSecuencia(String secuencia) {
    this.secuencia = secuencia;
}

public String getRegistraAuditoria() {
    return registraAuditoria;
}

public void setRegistraAuditoria(String registraAuditoria) {
    this.registraAuditoria = registraAuditoria;
}

public String getRegistroAntes() {
    return registroAntes;
}

public void setRegistroAntes(String registroAntes) {
    this.registroAntes = registroAntes;
}

public String getRegistroDespues() {
    return registroDespues;
}

public void setRegistroDespues(String registroDespues) {
    this.registroDespues = registroDespues;
}

public String getCamposDescripcion() {
    return camposDescripcion;
}

public void setCamposDescripcion(String camposDescripcion) {
    this.camposDescripcion = camposDescripcion;
}

public String getUsuarioCreador() {
    return usuarioCreador;
}

public void setUsuarioCreador(String usuarioCreador) {
    this.usuarioCreador = usuarioCreador;
}

public Date getFechaCreacion() {
    return fechaCreacion;
}
}
```

```
public void setFechaCreacion(Date fechaCreacion) {
    this.fechaCreacion = fechaCreacion;
}

public String getUsuarioModificador() {
    return usuarioModificador;
}

public void setUsuarioModificador(String usuarioModificador) {
    this.usuarioModificador = usuarioModificador;
}

public Date getFechaModificacion() {
    return fechaModificacion;
}

public void setFechaModificacion(Date fechaModificacion) {
    this.fechaModificacion = fechaModificacion;
}

public AppSessionVO getAppSessionVO() {
    return appSessionVO;
}

public void setAppSessionVO(AppSessionVO appSessionVO) {
    this.appSessionVO = appSessionVO;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (tablaId != null ? tablaId.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    // TODO: Warning - this method won't work in the case the id fields are not set
    if (!(object instanceof Tabla)) {
        return false;
    }
    Tabla other = (Tabla) object;
    if ((this.tablaId == null && other.tablaId != null) || (this.tablaId != null &&
!this.tablaId.equals(other.tablaId))) {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "com.sb.colciencias.gestion.business.domain.Tabla[ tablaId=" + tablaId + "
]";
}
}
```

*Crear nuevo DTO (**TablaDTO.java**) representativo de la entidad*

El **TablaDTO** (Data Transfer Object) es una clase utilizada para la transferencia de datos entre capas de la aplicación. Se encarga de representar información relacionada con una tabla en la base de datos, incluyendo su identificador, descripción, configuración de auditoría y metadatos de creación y modificación.

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package com.sb.colciencias.gestion.business.transfer;

import com.sb.colciencias.gestion.business.domain.security.Aplicacion;
import com.sb.colciencias.gestion.business.util.annotation.Bind;
import com.sb.colciencias.gestion.business.valueObjects.AppSessionVO;
import java.io.Serializable;
import java.util.Date;

/**
 *
 * @author dcangrejo
 */
public class TablaDTO extends GenericDTO implements Serializable {

    @Bind(name = "tablaId")
    private Long tablaId;

    @Bind(name = "tabla")
    private String tabla;

    @Bind(name = "descripcionTabla")
    private String descripcionTabla;

    @Bind(name = "secuencia")
    private String secuencia;

    @Bind(name = "registraAuditoria")
    private String registraAuditoria;

    @Bind(name = "registroAntes")
    private String registroAntes;

    @Bind(name = "registroDespues")
    private String registroDespues;

    @Bind(name = "camposDescripcion")
    private String camposDescripcion;

    @Bind(name = "usuarioCreador")
    private String usuarioCreador;

    @Bind(name = "fechaCreacion")
    private Date fechaCreacion;

    @Bind(name = "usuarioModificador")
    private String usuarioModificador;
}
```

```
@Bind(name = "fechaModificacion")
private Date fechaModificacion;

//Joins
@Bind(name = "aplicacion", returnType = AplicacionDTO.class, type = Aplicacion.class)
private AplicacionDTO aplicacion;

//Campos checkBox
private Boolean registraAuditoriaBool;

//Auditoria
@Bind(name = "appSessionVO")
private AppSessionVO appSessionVO;

public TablaDTO() {
}

public Long getTablaId() {
    return tablaId;
}

public void setTablaId(Long tablaId) {
    this.tablaId = tablaId;
}

public String getTabla() {
    return tabla;
}

public void setTabla(String tabla) {
    this.tabla = tabla;
}

public String getDescripcionTabla() {
    return descripcionTabla;
}

public void setDescripcionTabla(String descripcionTabla) {
    this.descripcionTabla = descripcionTabla;
}

public String getSecuencia() {
    return secuencia;
}

public void setSecuencia(String secuencia) {
    this.secuencia = secuencia;
}

public String getRegistraAuditoria() {
    return registraAuditoria;
}

public void setRegistraAuditoria(String registraAuditoria) {
    if (registraAuditoria != null && registraAuditoria.equals("S")) {
        this.registraAuditoriaBool = true;
    }
    this.registraAuditoria = registraAuditoria;
}

public Boolean getRegistraAuditoriaBool() {
```

```
        return registraAuditoriaBool;
    }

    public void setRegistraAuditoriaBool(Boolean registraAuditoriaBool) {
        if (registraAuditoriaBool) {
            this.registraAuditoria = "S";
        } else {
            this.registraAuditoria = "N";
        }
        this.registraAuditoriaBool = registraAuditoriaBool;
    }

    public String getRegistroAntes() {
        return registroAntes;
    }

    public void setRegistroAntes(String registroAntes) {
        this.registroAntes = registroAntes;
    }

    public String getRegistroDespues() {
        return registroDespues;
    }

    public void setRegistroDespues(String registroDespues) {
        this.registroDespues = registroDespues;
    }

    public String getCamposDescripcion() {
        return camposDescripcion;
    }

    public void setCamposDescripcion(String camposDescripcion) {
        this.camposDescripcion = camposDescripcion;
    }

    public String getUsuarioCreador() {
        return usuarioCreador;
    }

    public void setUsuarioCreador(String usuarioCreador) {
        this.usuarioCreador = usuarioCreador;
    }

    public Date getFechaCreacion() {
        return fechaCreacion;
    }

    public void setFechaCreacion(Date fechaCreacion) {
        this.fechaCreacion = fechaCreacion;
    }

    public String getUsuarioModificador() {
        return usuarioModificador;
    }

    public void setUsuarioModificador(String usuarioModificador) {
        this.usuarioModificador = usuarioModificador;
    }

    public Date getFechaModificacion() {
```

```
        return fechaModificacion;
    }

    public void setFechaModificacion(Date fechaModificacion) {
        this.fechaModificacion = fechaModificacion;
    }

    public AplicacionDTO getAplicacion() {
        return aplicacion;
    }

    public void setAplicacion(AplicacionDTO aplicacion) {
        this.aplicacion = aplicacion;
    }

    public AppSessionVO getAppSessionVO() {
        return appSessionVO;
    }

    public void setAppSessionVO(AppSessionVO appSessionVO) {
        this.appSessionVO = appSessionVO;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (tablaId != null ? tablaId.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        if (!(object instanceof TablaDTO)) {
            return false;
        }
        TablaDTO other = (TablaDTO) object;
        if ((this.tablaId == null && other.tablaId != null) || (this.tablaId != null &&
!this.tablaId.equals(other.tablaId))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "com.sb.colciencias.entidades.business.transfer.TablaDTO[ tablaId=" + tablaId
+ " ]";
    }
}
```

Paquete

Ubicación: **com.sb.colciencias.gestion.business.transfer**

Este DTO pertenece al paquete de transferencia de negocio, lo que indica que su propósito es facilitar la comunicación de datos entre la capa de negocio y otras capas de la aplicación.

Implementaciones y Herencia

- **extends GenericDTO:** Hereda de una clase genérica **GenericDTO**, lo que sugiere que hereda funcionalidades comunes a todos los DTOs.
- **implements Serializable:** Permite que la clase sea serializable para ser transmitida en procesos de comunicación o persistencia temporal.

Atributos y Anotaciones

Los atributos de la clase están anotados con `@Bind`, lo que indica que pueden estar relacionados con una base de datos o ser utilizados en la vinculación de datos.

```
@Bind
private Long tablaId;
@Bind
private String tabla;
@Bind
private String descripcionTabla;
```

- **Identificadores y Descripción:**
 - **tablaId:** Identificador único de la tabla.
 - **tabla:** Nombre de la tabla.
 - **descripcionTabla:** Descripción de la tabla.

```
@Bind
private String registraAuditoria;
@Bind
private String registroAntes;
@Bind
private String registroDespues;
private Boolean registraAuditoriaBool;
```

- **Configuración de Auditoría:**
 - **registraAuditoria:** Indica si la tabla tiene habilitada la auditoría (valores "S" para sí, "N" para no).
 - **registroAntes:** Contiene los valores antes de una modificación.
 - **registroDespues:** Contiene los valores después de una modificación.
 - **registraAuditoriaBool:** Versión booleana del campo **registraAuditoria**, permitiendo una gestión más sencilla en la lógica de negocio.

```
@Bind
```

```
private String usuarioCreador;  
@Bind  
private Date fechaCreacion;  
@Bind  
private String usuarioModificador;  
@Bind  
private Date fechaModificacion;
```

- **Metadatos de Creación y Modificación:**

- **usuarioCreador:** Usuario que creó el registro.
- **fechaCreacion:** Fecha de creación del registro.
- **usuarioModificador:** Usuario que realizó la última modificación.
- **fechaModificacion:** Fecha de la última modificación.

```
@Bind  
private AplicacionDTO aplicacion;  
private AppSessionVO appSessionVO;
```

- **Relaciones con Otras Entidades:**

- **aplicacion:** Relación con **AplicacionDTO**, lo que indica que cada **TablaDTO** está asociada con una aplicación.
- **appSessionVO:** Objeto de sesión, posiblemente para gestionar permisos o auditoría en tiempo de ejecución.

Métodos

La clase incluye métodos **getter** y **setter** para cada atributo, asegurando encapsulamiento y manipulación controlada de los datos.

- **Conversión de Auditoría:**

```
public void setRegistraAuditoria(String registraAuditoria) {  
    this.registraAuditoria = registraAuditoria;  
    this.registraAuditoriaBool = "S".equals(registraAuditoria);  
}  
  
public void setRegistraAuditoriaBool(Boolean registraAuditoriaBool) {  
    this.registraAuditoriaBool = registraAuditoriaBool;  
    this.registraAuditoria = registraAuditoriaBool ? "S" : "N";  
}
```

- **setRegistraAuditoria(String registraAuditoria):** Convierte valores "S" o "N" a un booleano.

- **setRegistraAuditoriaBool(Boolean registraAuditoriaBool):** Convierte el booleano en "S" o "N".

- **Métodos Override:**

```
@Override
public int hashCode() {
    int hash = 7;
    hash = 31 * hash + (this.tablaId != null ? this.tablaId.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    if (this == object) {
        return true;
    }
    if (object == null || getClass() != object.getClass()) {
        return false;
    }
    final TablaDTO other = (TablaDTO) object;
    return this.tablaId != null && this.tablaId.equals(other.tablaId);
}

@Override
public String toString() {
    return "TablaDTO{" + "tablaId=" + tablaId + ", tabla=" + tabla + '}';
}
```

- **hashCode():** Genera un código hash basado en **tablaId**.
- **equals(Object object):** Compara dos instancias de **TablaDTO** para verificar si representan la misma entidad.
- **toString():** Devuelve una representación en cadena del objeto.

El **TablaDTO** facilita la manipulación de información sobre tablas y auditoría dentro de la aplicación. Su diseño sigue buenas prácticas de encapsulamiento, herencia y uso de anotaciones para la vinculación de datos.

*Adición del menú "**Tablas**" en la plantilla del menú del aplicativo de gestión*

```
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST6181')}">
    <li class="nav-item #{selectedPage eq 'tabla' ? 'start active':''}">
        <p:link outcome="/app/administracion/tablas" styleClass="nav-link">
            <i class="fa fa-edit"></i>
            #{msg['sb.menu.gestion.administracion.sistema.tablas']}
        </p:link>
    </li>
</ui:fragment>
```

Este fragmento de código representa un **ítem de menú** dentro de una barra de navegación, que se renderiza dinámicamente dependiendo de los permisos del usuario.

Restricción de Renderizado (ui:fragment)

```
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST6181')}">
```

- Usa `<ui:fragment>` para incluir el contenido solo si se cumple una condición.
- La condición se basa en el **controlador de sesión de seguridad** (SeguridadSessController).
- **mapaMenu.containsKey('GEST6181')**:
 - mapaMenu es un mapa que almacena las opciones del menú habilitadas para el usuario.
 - GEST6181 es el código que representa el permiso necesario para ver esta opción del menú.
 - Si el usuario tiene este permiso, el ítem de menú se renderiza; de lo contrario, no aparece.

Elemento del Menú (li.nav-item)

```
<li class="nav-item #{selectedPage eq 'tabla' ? 'start active':''}">
```

- `<li>` define un elemento de lista que representa una opción en el menú.
- La clase **CSS dinámica** se define usando `#{selectedPage eq 'tabla' ? 'start active':''}`:
 - selectedPage es una variable de la vista que indica la página activa.
 - Si el valor de selectedPage es **'tabla'**, se aplican las clases **'start'** y **'active'** para resaltar la opción del menú.

Enlace de Navegación (p:link)

```
<p:link outcome="/app/administracion/tablas" styleClass="nav-link">  
  <i class="fa fa-edit"></i>  
  #{msg['sb.menu.gestion.administracion.sistema.tablas']}  
</p:link>
```

- **p:link** (de PrimeFaces) genera un enlace de navegación.
- **outcome="/app/administracion/tablas"**:

- Indica que al hacer clic, se redirige a la página **/app/administracion/tablas.xhtml**.
- **styleClass="nav-link"**:
 - Aplica estilos CSS definidos en la plantilla del sistema.
- **Ícono (<i class="fa fa-edit"></i>)**:
 - Muestra un ícono de edición (de FontAwesome) antes del texto.
- **Etiqueta de menú ({msg['sb.menu.gestion.administracion.sistema.tablas']})**:
 - Se usa **msg** para obtener el texto desde un archivo de **internacionalización** (**.properties**).
 - **sb.menu.gestion.administracion.sistema.tablas** es la clave del mensaje, que puede traducirse según el idioma del usuario.

Resumen del Funcionamiento

1. **Verificación de permisos:**
 - El menú solo se muestra si el usuario tiene acceso (**mapaMenu.containsKey('GEST6181')**).
2. **Resaltado de la opción activa:**
 - Si **selectedPage** es 'tabla', la opción del menú se marca como activa.
3. **Navegación a la página de administración de tablas:**
 - Se usa **p:link** para redirigir a **/app/administracion/tablas**.
4. **Uso de un mensaje internacionalizado:**
 - Se obtiene el texto desde **msg['sb.menu.gestion.administracion.sistema.tablas']**, lo que permite soportar múltiples idiomas.

*Controlador de Vista **TablaViewController.java***

Este controlador es responsable de gestionar la vista de administración de tablas en la aplicación.
Se encarga de:

- Cargar la lista de tablas de la base de datos de forma paginada y con filtrado dinámico.
- Gestionar la secuencia de las tablas.
- Proporcionar el modelo de datos para su visualización en la interfaz de usuario.

Definición de la Clase

```
@Named(value = "tablaViewController")  
@ViewScoped  
public class TablaViewController extends GenericSessionController<TablaDTO> implements  
Serializable {
```

- **@Named(value = "tablaViewController"):**
 - Define este controlador como un **bean de JSF**, permitiendo su acceso en las páginas **XHTML** con **#{tablaViewController}**.
- **@ViewScoped:**
 - Mantiene el estado del controlador **mientras el usuario permanezca en la misma vista**.
- **extends GenericSessionController<TablaDTO>:**
 - Hereda funcionalidades de **GenericSessionController**, lo que sugiere que maneja sesiones de usuario y datos de tipo **TablaDTO**.
- **implements Serializable:**
 - Permite que el objeto pueda ser serializado, lo cual es importante para la gestión del ciclo de vida en JSF.

Inyección de Dependencias

```
@EJB  
private AdministracionServiceFacade administracionServiceFacade;  
  
@Inject  
private ConfigApplController configApplController;
```

- **@EJB AdministracionServiceFacade administracionServiceFacade;**
 - Inyecta un **EJB** (Enterprise JavaBean) que gestiona la lógica de negocio relacionada con la administración de tablas.

- **@Inject ConfigAppController configAppController;**
 - Inyecta el controlador de configuración de la aplicación.

Atributos de la Clase

```
private LazyDataModel<TablaDTO> model;  
private List<String> listaSecuencias;  
private static final org.apache.log4j.Logger LOG =  
org.apache.log4j.Logger.getLogger(TablaViewController.class);
```

- **LazyDataModel<TablaDTO> model;**
 - Representa un modelo de datos **perezoso**, optimizando la carga de registros en la vista.
- **List<String> listaSecuencias;**
 - Almacena una lista de secuencias disponibles para las tablas.
- **Logger LOG**
 - Se usa para registrar mensajes en el log del sistema.

Constructor de la Clase

```
public TablaViewController() {  
    super(TablaDTO.class);  
}
```

- Llama al constructor de la clase padre (GenericSessionController).
- Especifica que este controlador maneja objetos del tipo TablaDTO.

Inicialización con @PostConstruct

```
@PostConstruct  
public void cargarListas() {  
    cargarListaSecuencias();  
    this.model = new LazyDataModel<TablaDTO>() {
```

- **@PostConstruct:**
 - Se ejecuta **después de que se haya creado** la instancia del controlador.
 - Llama a cargarListaSecuencias() para obtener las secuencias de tablas.

Modelo de Datos con Carga Diferida (LazyDataModel)

Este bloque define cómo se cargan las tablas desde la base de datos de manera **perezosa (lazy loading)**:

```
@Override  
public List<TablaDTO> load(int first, int pageSize, String sortField, SortOrder sortOrder,  
Map<String, FilterMeta> filterBy) {
```

- **load(int first, int pageSize, String sortField, SortOrder sortOrder, Map<String, FilterMeta> filterBy)**
 - **first**: Índice del primer registro a cargar.
 - **pageSize**: Cantidad de registros por página.
 - **sortField**: Campo por el cual ordenar.
 - **sortOrder**: Dirección del ordenamiento (ascendente o descendente).
 - **filterBy**: Filtros aplicados a los datos.

Aplicación de Filtros

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

- Se filtran los datos utilizando globalFilter si el usuario ha ingresado un valor en el campo de búsqueda global.

Carga de Datos

```
filters.put("aplicacion", configApplController.getAplicacion().getAplicacion());  
int count = administracionServiceFacade.tablasListCount(filters);  
if (count > 0) {  
    list = administracionServiceFacade.tablasList(first, pageSize, sortField,  
sortOrder.equals(SortOrder.ASCENDING) ? "A" : "D", filters);  
}  
model.setRowCount(count);
```

- Se añade un filtro para que solo se carguen las tablas de la aplicación actual.

- Se obtiene el número total de registros con `tablasListCount(filters)`.
- Si hay registros disponibles, se cargan los datos paginados con `tablasList(...)`.

Manejo de Errores

```
} catch (Exception ex) {  
    LOG.error("Error al cargar el listado de tablas ", ex);  
}
```

- Captura errores en la consulta de tablas y los registra en el log.

Identificación de Registros en la Vista

```
@Override  
public Object getRowKey(TablaDTO tablaDTO) {  
    return tablaDTO.getTablaId();  
}
```

- Devuelve el **ID único** de cada tabla para su identificación en la vista.

```
@Override  
public TablaDTO getRowData(String rowKey) {  
    try {  
        return administracionServiceFacade.queryTabla(Long.parseLong(rowKey));  
    } catch (Exception ex) {  
        LOG.error("no se pudo cargar la tabla ( id => " + rowKey + ")", ex);  
    }  
    return null;  
}
```

- Cuando se selecciona una fila en la vista, se obtiene su información desde la base de datos.

Carga de Secuencias

```
private void cargarListaSecuencias() {  
    try {  
        Map<String, Object> filters = new HashMap<>();  
        filters.put("activo", "S");  
        listaSecuencias = administracionServiceFacade.obtenerSecuenciasTablas();  
    } catch (Exception ex) {  
        LOG.error("Error al cargar el listado de tipos definiciones para el campo tipo de  
validacion ", ex);  
    }  
}
```

- Obtiene las secuencias activas de la base de datos y las almacena en `listaSecuencias`.

Métodos Getters y Setters

```
public LazyDataModel<TablaDTO> getModel() { return model; }
```

```
public void setModel(LazyDataModel<TablaDTO> model) { this.model = model; }
```

- Permiten acceder y modificar el modelo de datos en la vista.

```
public ConfigApplController getConfigApplController() { return configApplController; }  
public void setConfigApplController(ConfigApplController configApplController) {  
    this.configApplController = configApplController; }
```

- Métodos para acceder al controlador de configuración.

```
public List<String> getListaSecuencias() { return listaSecuencias; }  
public void setListaSecuencias(List<String> listaSecuencias) { this.listaSecuencias =  
    listaSecuencias; }
```

- Métodos para acceder a la lista de secuencias.

Resumen del Funcionamiento

1. Carga las tablas de manera paginada y filtrada.
2. Usa un modelo de datos perezoso (**LazyDataModel**) para optimizar la carga.
3. Obtiene los datos desde **AdministracionServiceFacade**.
4. Filtra los datos por aplicación y búsqueda global.
5. Carga una lista de secuencias de la base de datos.
6. Maneja errores mediante LOG.error.
7. Identifica registros por su ID (**getRowKey**) y permite su selección en la UI (**getRowData**).

*Controlador de Conversación **TablaConvController.java***

El **TablaConvController** es un controlador de conversación en una aplicación Java EE con JSF y PrimeFaces. Su propósito es gestionar la creación, edición y eliminación de registros de tipo TablaDTO (CRUD), además de interactuar con servicios y controladores de sesión para manejar información y seguridad.

Anotaciones y Alcance

```
@Named(value = "tablaConvController")  
@ConversationScoped  
public class TablaConvController extends GenericConversationController<TablaDTO> implements  
    Serializable {
```

- **@Named(value = "tablaConvController"):** Define el nombre del bean para su uso en la vista JSF.
- **@ConversationScoped:** Establece el alcance del controlador a una conversación, permitiendo que la información persista entre peticiones dentro de la misma conversación.

Dependencias Inyectadas

```
@EJB
private AdministracionServiceFacade administracionServiceFacade;
@EJB
private BeneficiosServiceFacade beneficiosServiceFacade;
@Inject
private ColcienciasSessController colcienciasSessController;
@Inject
private TablaViewController tablaViewController;
@Inject
private SeguridadSessController seguridadSessController;
@Inject
private ConfigApplController configApplController;
```

- **AdministracionServiceFacade:** EJB para operaciones administrativas sobre tablas.
- **BeneficiosServiceFacade:** EJB relacionado con beneficios tributarios.
- **ColcienciasSessController:** Controlador de sesión del usuario.
- **TablaViewController:** Controlador de vista para la gestión de tablas.
- **SeguridadSessController:** Controlador de seguridad para autenticación y permisos.
- **ConfigApplController:** Controlador de configuración de la aplicación.

Atributos Principales

```
private final String programa = "GEST6181";

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

public TablaConvController() {
    this.selected = new TablaDTO();
}
```

- **programa:** Código del programa utilizado en auditorías.
- **LOG:** Logger de Log4J para registrar eventos y errores.

- selected: Objeto TablaDTO seleccionado, inicializado en el constructor.

Métodos Principales

guardar(ActionEvent ae)

```
@Override
public void guardar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
        TablaDTO tablaDTO = tablaViewController.getSelected();
        tablaDTO.setAplicacion(configApplController.getAplicacion());
        tablaDTO.setUsuarioCreador(seguridadSessController.getSelected().getUsuario());
        tablaDTO.setFechaCreacion(new Date());
        // tablaDTO.setTabla(tablaDTO.getTabla().trim());
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        appSessionVO.setPrograma(programa);
        tablaDTO.setAppSessionVO(appSessionVO);
        administracionServiceFacade.guardarTabla(tablaDTO);

        tablaViewController.setSelected(new TablaDTO());
        //Actualizar las listas en tiempo de Ejecución
        configApplController.listasGenerales();
        tablaViewController.cargarListas();
        //se dehabilita el boton de editar / eliminar ya que no habra nada seleccionado
    } catch (SystemException ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
            resolveExpression(ex.getMessage()));
        LOG.error("Error al guardar el Tabla ", ex);
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
            resolveExpression("#{msg['sb.mensajes.transaccion.fallido']}"));
        LOG.error("Error al guardar el Tabla ", ex);
    } finally {
        FacesContext.getCurrentInstance().addMessage(null, message);
    }
}
```

Guarda un nuevo registro de **TablaDTO**.

1. Obtiene el objeto seleccionado desde **tablaViewController**.
2. Asigna valores como aplicación, usuario creador y fecha de creación.
3. Configura la sesión de la aplicación (**AppSessionVO**).

4. Llama al servicio **administracionServiceFacade.guardarTabla(tablaDTO)**.
5. Reinicializa el objeto seleccionado y actualiza listas en **tablaViewController**.
6. Deshabilita botones de edición y elimina selección.
7. Actualiza la interfaz con **PrimeFaces.current().ajax().update(...)**.
8. Manejo de excepciones con **SystemException** y **Exception**, registrando logs.

eliminar(ActionEvent ae)

```
@Override
public void eliminar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
        TablaDTO tablaDTO = tablaViewController.getSelected();
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        appSessionVO.setProgram(programa);
        tablaDTO.setAppSessionVO(appSessionVO);

        //Eliminar Campos
        List<CampoDTO> camposHijos = new ArrayList<>();
        Map<String, Object> filters = new HashMap<>();
        filters.put("tabla", tablaDTO.getTabla());
        camposHijos = administracionServiceFacade.campoList(-1, -1, "posicion", "A",
filters);
        if (camposHijos != null && camposHijos.size() > 0) {
            for (CampoDTO campo : camposHijos) {
                campo.setAppSessionVO(appSessionVO);
                administracionServiceFacade.eliminarCampo(campo);
            }
        }

        administracionServiceFacade.eliminarTabla(tablaDTO);
        tablaViewController.setSelected(new TablaDTO());
        //Actualizar las listas en tiempo de Ejecución
        configApplController.listasGenerales();
        tablaViewController.cargarListas();
        this.onRowUnselect(null);
        PrimeFaces.current().ajax().update(tablaViewController.getModulo() + "Form");
    } catch (SystemException ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        LOG.error("Error al eliminar el Tabla ", ex);
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        LOG.error("Error al eliminar el Tabla ", ex);
    } finally {
        FacesContext.getCurrentInstance().addMessage(null, message);
    }
}
```

Elimina un registro de **TablaDTO**.

1. Obtiene el objeto seleccionado desde **tablaViewController**.
2. Configura la sesión de la aplicación (**AppSessionVO**).
3. Busca y elimina los **CampoDTO** asociados a la tabla.
4. Llama a **administracionServiceFacade.eliminarTabla(tablaDTO)**.
5. Reinicializa la selección y actualiza listas en **tablaViewController**.
6. Actualiza la interfaz con **PrimeFaces.current().ajax().update(...)**.
7. Manejo de excepciones similar a guardar.

editar(ActionEvent ae)

```
@Override
public void editar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {

        TablaDTO tablaDTO = tablaViewController.getSelected();
        TablaDTO tablaAntes = new TablaDTO();
        String auditoriaAntes = "";
        String auditoriaDespues = "";

        //Lista para verificar el estado del campo Auditoria de la tabla que se va a
editar
        List<TablaDTO> tablaAuditoria = new ArrayList<>();
        Map<String, Object> filters = new HashMap<>();
        filters.put("tabla", tablaDTO.getTabla());

        tablaAuditoria = administracionServiceFacade.tablasList(-1, -1, "tabla", "A",
filters);
        if (tablaAuditoria != null && tablaAuditoria.size() == 1) {
            tablaAntes = tablaAuditoria.get(0);
            auditoriaAntes = tablaAntes.getRegistraAuditoria();
        }
        auditoriaDespues =
tablaDTO.getRegistraAuditoriaBool().toString().equals("true") ? "S" : "N";

        if (!auditoriaAntes.equals(auditoriaDespues)) {
            if (auditoriaAntes.equals("N") && auditoriaDespues.equals("S")) {
                cargarValorCamposAuditoria("NoToYes");
            } else if (auditoriaAntes.equals("S") && auditoriaDespues.equals("N"))
{
                cargarValorCamposAuditoria("YesToNo");
            }
        }
    }
}
```

```

        tablaDTO.setUsuarioModificador(seguridadSessController.getSelected().getUsuario());
        tablaDTO.setFechaModificacion(new Date());
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        appSessionVO.setPrograma(programa);
        tablaDTO.setAppSessionVO(appSessionVO);
        administracionServiceFacade.editarTabla(tablaDTO);

        tablaViewController.setSelected(new TablaDTO());
        //Actualizar las listas en tiempo de Ejecución
        configApplController.listasGenerales();

        tablaViewController.cargarListas();
        //se dehabilita el boton de editar / eliminar ya que no habra nada seleccionado
despues de la operación
        this.onRowUnselect(null);
        this.obtenerPestana = 0;
        PrimeFaces.current().ajax().update(tablaViewController.getModulo() + "Form");
    } catch (SystemException ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        LOG.error("Error al editar el Tabla ", ex);
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression("#{msg['sb.mensajes.transaccion.fallido']}"));
        LOG.error("Error al editar el Tabla ", ex);
    } finally {
        FacesContext.getCurrentInstance().addMessage(null, message);
    }
}

```

Edita un registro de **TablaDTO**.

1. Obtiene el objeto seleccionado desde **tablaViewController**.
2. Recupera datos previos de auditoría.
3. Verifica y actualiza valores de auditoría si han cambiado.
4. Asigna valores como usuario modificador y fecha de modificación.
5. Llama a **administracionServiceFacade.editarTabla(tablaDTO)**.
6. Reinicializa selección y actualiza listas en **tablaViewController**.
7. Manejo de excepciones similar a guardar.

mostrar(ActionEvent ae)

```

@Override
public void mostrar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),

```

```

        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));
        //Obtengo la tabla
        String modulo = tablaViewController.getModulo();
        DataTable dataTable = (DataTable)
        FacesContext.getCurrentInstance().getViewRoot().findComponent(modulo + "Form:" + modulo +
        "TabList:" + modulo + "Table");
        //limpio los filtros
        dataTable.setFilterBy(null);
        //Se actualiza el formulario que contiene la tabla
        PrimeFaces.current().ajax().update(modulo + "Form");
        //Se muestra el growl con el mensaje
        FacesContext.getCurrentInstance().addMessage(null, message);

        tablaViewController.cargarListas();
        this.onRowUnselect(null);
    }

```

Prepara la interfaz para mostrar datos:

1. Obtiene el identificador del módulo desde **tablaViewController**.
2. Recupera el componente **DataTable** en la vista JSF.
3. Resetea filtros del **DataTable** para mostrar datos actualizados.

cargarTablas()

Este método se encarga de cargar las tablas en la aplicación a través de un servicio externo.

```

public void cargarTablas() {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        String mensaje = null;

        Map<String, Object> filters = new HashMap<>();
        filters.put("paquete",
        beneficiosServiceFacade.obtenerParametroSistema("PKGATABLAS").getCadena());
        filters.put("p_aplicacion", configApplController.getAplicacion().getAplicacion());
        filters.put("p_usuario", seguridadSessController.getSelected().getUsuario());
        filters.put("p_terminal", appSessionVO.getIpAddress());
        filters.put("p_job_accion",
        beneficiosServiceFacade.obtenerParametroSistema("PKGATABLAS2").getCadena());
        filters.put("accionPantalla", "C");

        mensaje = administracionServiceFacade.callSBCargarTablasCampos(filters);
        System.out.println("mensaje: " + mensaje);
        if (mensaje != null && !mensaje.equals("")) {
            message = new FacesMessage(FacesMessage.SEVERITY_INFO,
            resolveExpression("#{msg['sb.mensajes.titulo.exito']}"), mensaje);
            FacesContext.getCurrentInstance().addMessage(null, message);
        }
        tablaViewController.cargarListas();
    }
}

```

```
    } catch (SystemException ex) {  
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",  
resolveExpression(ex.getMessage()));  
        FacesContext.getCurrentInstance().addMessage(null, message);  
        LOG.error("Error al cargar tablas", ex);  
    } catch (Exception ex) {  
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",  
resolveExpression("#{msg['sb.mensajes.transaccion.fallido']}"));  
        FacesContext.getCurrentInstance().addMessage(null, message);  
        LOG.error("Error al cargar tablas", ex);  
    }  
}
```

cargarValorCamposAuditoria(String cambioEstado)

Este método cambia el estado de auditoría de los campos de una tabla.

```
public void cargarValorCamposAuditoria(String cambioEstado) {  
    try {  
        String mensaje = "";  
        Map<String, Object> filters = new HashMap<>();  
  
        if (cambioEstado != null) {  
            if (cambioEstado.equals("NoToYes")) {  
                filters.put("paquete",  
beneficiosServiceFacade.obtenerParametroSistema("OPENAUDIT").getCadena());  
                filters.put("accionPantalla", "Abrir");  
            } else if (cambioEstado.equals("YesToNo")) {  
                filters.put("paquete",  
beneficiosServiceFacade.obtenerParametroSistema("CLOSEAUDIT").getCadena());  
                filters.put("accionPantalla", "Cerrar");  
            }  
        }  
        filters.put("tabla", tablaViewController.getSelected().getTabla());  
        filters.put("p_usuario", seguridadSessController.getSelected().getUsuario());  
  
        mensaje = administracionServiceFacade.callSBcambiarEstadoAuditoria(filters);  
        System.out.println(mensaje);  
  
    } catch (SystemException ex) {  
        LOG.error("Error al cambiar el estado de la auditoria para los campos de una tabla",  
ex);  
    } catch (Exception ex) {  
        LOG.error("Error al cambiar el estado de la auditoria para los campos de una tabla",  
ex);  
    }  
}
```

Generación del Reporte

Para generar el reporte, se debe ejecutar el procedimiento almacenado correspondiente y exportar el resultado en formato RTF.

Este controlador maneja las operaciones CRUD de **TablaDTO**, asegurando que los datos sean consistentes con la configuración y seguridad de la aplicación. Utiliza **PrimeFaces** para actualizaciones dinámicas y logs para el seguimiento de errores y eventos importantes.

*Se definen métodos de CRUD en Facade EJB (**AdministracionServiceFacade.java**) para funcionalidad de pantalla*

```
//  
//  
//GESTIÓN TABLAS (SISTEMA)  
//  
/**  
 * Consulta el listado de tablas 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  
 * tablas  
 * @return lista de tablas encontrados en el rango enviado y basado en los  
 * filtros enviados  
 * @throws Exception  
 */  
List<TablaDTO> tablasList(int first, int pageSize, String sortField, String sortOrder,  
Map<String, Object> filters) throws Exception;  
  
/**  
 * Cuenta la cantidad de tablas actualmente registrados en la base de datos  
 *  
 * @param filters  
 * @return  
 * @throws Exception  
 */  
int tablasListCount(Map<String, Object> filters) throws Exception;  
  
/**  
 * Guarda el registro de tabla en la base de datos  
 *  
 * @param tablaDTO  
 * @throws Exception  
 */  
void guardarTabla(TablaDTO tablaDTO) throws Exception;  
  
/**  
 * Edita el registro de tabla en la base de datos  
 *  
 * @param tablaDTO  
 * @throws Exception  
 */  
void editarTabla(TablaDTO tablaDTO) throws Exception;
```

```

/**
 * Elimina el registro de tabla en la base de datos
 *
 * @param tablaDTO
 * @throws Exception
 */
void eliminarTabla(TablaDTO tablaDTO) throws Exception;

/**
 * Buscar el registro de un tabla y lo retorna basado en el id (llave
 * primaria)
 *
 * @param tablaId
 * @return el tabla que coincide con el id enviado
 * @throws Exception
 */
TablaDTO queryTabla(Long tablaId) throws Exception;

/**
 * Obtiene la lista de secuencias de la base de datos
 *
 * @return Lista de secuencias
 * @throws Exception
 */
List<String> obtenerSecuenciasTablas() throws Exception;

/**
 * Llama el procedimiento para cargar las tablas del sistema, o los campos
 * asociados a una tabla
 *
 * @param filters
 * @return Mensaje proporcionado por el procedimiento
 * @throws Exception
 */
String callSBCargarTablasCampos(Map<String, Object> filters) throws Exception;

/**
 * Llama el procedimiento para cambiar el valor de auditoria de los campos
 * de la tabla la cual se está editando
 *
 * @param filters
 * @return Mensaje proporcionado por el procedimiento
 * @throws Exception
 */
String callSBCambiarEstadoAuditoria(Map<String, Object> filters) throws Exception;

```

*Se implementan métodos CRUD en facade EJB
(AdministracionServiceFacadeBean.java) para funcionalidad de pantalla*

```

//----- Gestión Tablas (Sistema) -----//
@Override
public List<TablaDTO> tablasList(int first, int pageSize, String sortField, String
sortOrder, Map<String, Object> filters) throws Exception {
    ObjectConverter<TablaDTO, Tabla> converter = objectConverterFactory.checkOut();

```

```
List<TablaDTO> listTablaDTOS =
converter.toTransferList(tablaService.tablasList(first, pageSize, sortField, sortOrder,
filters), TablaDTO.class);
    objectConverterFactory.checkIn(converter);
    return listTablaDTOS;
}

@Override
public int tablasListCount(Map<String, Object> filters) throws Exception {
    return tablaService.tablasListCount(filters);
}

@Override
public void guardarTabla(TablaDTO tablaDTO) throws Exception {
    ObjectConverter<TablaDTO, Tabla> converter = objectConverterFactory.checkOut();
    tablaService.guardarTabla(converter.toEntity(tablaDTO, Tabla.class));
    objectConverterFactory.checkIn(converter);
}

@Override
public void editarTabla(TablaDTO tablaDTO) throws Exception {
    ObjectConverter<TablaDTO, Tabla> converter = objectConverterFactory.checkOut();
    tablaService.editarTabla(converter.toEntity(tablaDTO, Tabla.class));
    objectConverterFactory.checkIn(converter);
}

@Override
public void eliminarTabla(TablaDTO tablaDTO) throws Exception {
    ObjectConverter<TablaDTO, Tabla> converter = objectConverterFactory.checkOut();
    tablaService.eliminarTabla(converter.toEntity(tablaDTO, Tabla.class));
    objectConverterFactory.checkIn(converter);
}

@Override
public TablaDTO queryTabla(Long tablaId) throws Exception {
    ObjectConverter<TablaDTO, Tabla> converter = objectConverterFactory.checkOut();
    TablaDTO tablaDTO = converter.toTransfer(tablaService.queryTabla(tablaId),
TablaDTO.class);
    objectConverterFactory.checkIn(converter);
    return tablaDTO;
}

@Override
public List<String> obtenerSecuenciasTablas() throws Exception {
    List<String> lista = tablaService.obtenerSecuenciasTablas();
    return lista;
}

@Override
public String callSBCargarTablasCampos(Map<String, Object> filters) throws Exception {
    return tablaService.callSBCargarTablasCampos(filters);
}

@Override
public String callSBcambiarEstadoAuditoria(Map<String, Object> filters) throws Exception
{
    return tablaService.callSBcambiarEstadoAuditoria(filters);
}
```

*Se crea interfaz **TablaService.java** que define métodos de CRUD en Service EJB para funcionalidad de pantalla*

```
@Local
public interface TablaService {

    List<Tabla> tablasList(int first, int pageSize, String sortField, String sortOrder,
        Map<String, Object> filters) throws SystemException;

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

    void guardarTabla(Tabla tabla) throws SystemException;

    void editarTabla(Tabla tabla) throws SystemException;

    void eliminarTabla(Tabla tabla) throws SystemException;

    Tabla queryTabla(Long tablaId) throws SystemException;

    List<String> obtenerSecuenciasTablas() throws SystemException;

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

    String callSBCambiarEstadoAuditoria(Map<String, Object> filters) throws SystemException;
}
```

*Se crea **TablaServiceBean.java** donde implementan métodos CRUD en Service EJB para funcionalidad de pantalla*

```
@Stateless
public class TablaServiceBean extends StoreManagerImpl implements TablaService {

    //----- Gestión Tablas (Sistema) -----//
    @Override
    public List<Tabla> tablasList(int first, int pageSize, String sortField, String
        sortOrder, Map<String, Object> filters) throws SystemException {
        return this.getStoreManager(AdministracionStoreManager.class).tablasList(first,
            pageSize, sortField, sortOrder, filters);
    }

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

    @Override
    public void guardarTabla(Tabla tabla) throws SystemException {
        this.crear(tabla, Tabla.class);
    }

    @Override
    public void editarTabla(Tabla tabla) throws SystemException {
        this.modificar(tabla, Tabla.class);
    }
}
```

```
@Override
public void eliminarTabla(Tabla tabla) throws SystemException {
    this.eliminar(tabla, Tabla.class);
}

@Override
public Tabla queryTabla(Long tablaId) throws SystemException {
    return this.buscar(tablaId, Tabla.class);
}

@Override
public List<String> obtenerSecuenciasTablas() throws SystemException {
    return
this.getStoreManager(AdministracionStoreManager.class).obtenerSecuenciasTablas();
}

@Override
public String callSBCargarTablasCampos(Map<String, Object> filters) throws
SystemException {
    return
this.getStoreManager(AdministracionStoreManager.class).callSBCargarTablasCampos(filters);
}

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

Internacionalización en español e inglés

#ES	
sb.menu.gestion.administracion.tablas.header	= Tablas del
Sistema	
sb.menu.gestion.administracion.tablas.tabla.breadcrumb	= Gestión
Tablas	
sb.menu.gestion.administracion.tablas.tabla.title	= Gestión
Tablas	
sb.menu.gestion.administracion.tablas.tabla.header	= Tabla
sb.menu.gestion.administracion.tablas.descripcion.header	=
Descripción	
sb.menu.gestion.administracion.tablas.secuencia.header	= Secuencia
sb.menu.gestion.administracion.tablas.auditoria.header	= Auditoria
sb.menu.gestion.administracion.tablas.camposDescripcion.header	= Campos
Descripción	
sb.menu.gestion.administracion.tablas.campos.header	= Campos
sb.menu.gestion.administracion.tablas.tabla.label	= Tabla
sb.menu.gestion.administracion.tablas.tabla.placeholder	= Ingrese
Tabla	
sb.menu.gestion.administracion.tablas.tabla.required	= Debe
ingresar la tabla	
sb.menu.gestion.administracion.tablas.descripcion.label	=
Descripción	
sb.menu.gestion.administracion.tablas.descripcion.placeholder	= Ingrese
Descripción	

sb.menu.gestion.administracion.tablas.descripcion.required ingresar la descripción	= Debe
sb.menu.gestion.administracion.tablas.secuencia.label	= Secuencia
sb.menu.gestion.administracion.tablas.camposDescripcion.label Descripción	= Campos
sb.menu.gestion.administracion.tablas.camposDescripcion.placeholder Campos Descripción	= Ingrese
sb.menu.gestion.administracion.tablas.registraAuditoria.label Auditoría	= Registra
sb.menu.gestion.administracion.tablas.campos.button tabla	= Ver campos
sb.menu.gestion.administracion.tablas.cargarTablas.button Tablas	= Cargar
com.sb.persistence.constraint.ORA-00001HERCULES.TABLA_UK una tabla con el nombre indicado	= Ya existe
#EN	
sb.menu.gestion.administracion.tablas.header Tables	= System
sb.menu.gestion.administracion.tablas.tabla.breadcrumb Management	= Table
sb.menu.gestion.administracion.tablas.tabla.title Management	= Table
sb.menu.gestion.administracion.tablas.tabla.header	= Table
sb.menu.gestion.administracion.tablas.descripcion.header Description	=
sb.menu.gestion.administracion.tablas.secuencia.header	= Sequence
sb.menu.gestion.administracion.tablas.auditoria.header	= Audit
sb.menu.gestion.administracion.tablas.camposDescripcion.header Description	= Fields
sb.menu.gestion.administracion.tablas.campos.header	= Fields
sb.menu.gestion.administracion.tablas.tabla.label	= Table
sb.menu.gestion.administracion.tablas.tabla.placeholder Table	= Enter
sb.menu.gestion.administracion.tablas.tabla.required required	= Table is
sb.menu.gestion.administracion.tablas.descripcion.label Description	=
sb.menu.gestion.administracion.tablas.descripcion.placeholder Description	= Enter
sb.menu.gestion.administracion.tablas.descripcion.required Description is required	=
sb.menu.gestion.administracion.tablas.secuencia.label	= Sequence
sb.menu.gestion.administracion.tablas.camposDescripcion.label Description Fields	=
sb.menu.gestion.administracion.tablas.camposDescripcion.placeholder Description Fields	= Enter
sb.menu.gestion.administracion.tablas.registraAuditoria.label Audit	= Register
sb.menu.gestion.administracion.tablas.campos.button fields	= View table
sb.menu.gestion.administracion.tablas.cargarTablas.button Tablas	= Load
com.sb.persistence.constraint.ORA-00001HERCULES.TABLA_UK with the given name already exists	= A table

Pestaña de Tabla de visualización de la información en la pantalla de Tablas

```
<!--PESTAÑA TABLE -->
<p:tab title="#{msg['sb.menu.gestion.administracion.tablas.header']}">

    <!-- Boton - Cargar Tablas -->
    <div class="row">
        <div class="col-lg-12">
            <p:commandLink id="cargarTablas"

title="#{msg['sb.menu.gestion.administracion.tablas.cargarTablas.button']}"
styleClass="btn btn btn-primary pull-right
tooltips"
update="#{tablaViewController.modulo}Form"

actionListener="#{tablaConvController.cargarTablas()}"
>
                <i class="fa fa-refresh">
#{msg['sb.menu.gestion.administracion.tablas.cargarTablas.button']} </i>
            </p:commandLink>
        </div>
    </div>

    <br />

    <!-- TABLA -->
    <div class="row">
        <div class="col-lg-12" style="overflow-x: scroll;">
            <p:dataTable id="#{tablaViewController.modulo}Table"
                var="tabla"
                value="#{tablaViewController.model}"
                rowKey="#{tabla.tablaId}"
                selection="#{tablaConvController.selected}"
                selectionMode="single"
                paginator="true"
                rows="10"
                lazy="true"
                sortField="tabla"
                sortOrder="ASCENDING"

currentPageReportTemplate="#{msg['sb.tabla.totalRegistros']} {totalRecords}"
paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
                rowsPerPageTemplate="10,20,30"
                paginatorPosition="bottom"

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

widgetVar="items#{tablaViewController.modulo}"

                <f:facet name="header">
                    <ui:decorate
template="../../../template/_tableHeader.xhtml">
                        <ui:define name="comandos">
                            <ui:param name="sessionController"
                                value="#{tablaViewController}" />
                            <ui:param name="conversationController"
                                value="#{tablaConvController}" />
                            <ui:param name="btnBusqueda" value="N"
                        </ui:define>
                    </ui:decorate>
                </f:facet>
            </p:dataTable>
        </div>
    </div>
```

```

<ui:param name="buscarVisible"
value="S" />

</ui:define>
</ui:decorate>
</f:facet>

<p:ajax event="rowSelect"
listener="#{tablaConvController.onRowSelect}"
update="#{tablaViewController.modulo}Table"
process="@this"/>
<p:ajax event="rowUnselect"
listener="#{tablaConvController.onRowUnselect}"
update="#{tablaViewController.modulo}Table"
process="@this"/>

<p:column style="text-align: center; padding: 5px;"
width="5%">
<p:rowToggler/>
</p:column>

<p:column
headerText="#{msg['sb.menu.gestion.administracion.tablas.tabla.header']}"
sortBy="#{tabla.tabla}"
filterBy="#{tabla.tabla}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="30%"
>
<h:outputText value="#{tabla.tabla}"/>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.tablas.descripcion.header']}"
filterBy="#{tabla.descripcionTabla}"
sortBy="#{tabla.descripcionTabla}"
filterMatchMode="contains"
style="text-align: center"
filterStyle="display:none;"
width="25%"
>
<h:outputText value="#{tabla.descripcionTabla}"/>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.tablas.secuencia.header']}"
filterBy="#{tabla.secuencia}"
sortBy="#{tabla.secuencia}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="30%"
>
<h:outputText value="#{tabla.secuencia}"/>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.tablas.auditoria.header']}"
filterBy="#{tabla.registraAuditoria}"
sortBy="#{tabla.registraAuditoria}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="10%"

```

```

                                >
                                <h:outputText styleClass="badge
#{tabla.registraAuditoria eq 'S' ? 'badge-success' : 'badge-danger'}"
value="#{tabla.registraAuditoria != null
                                ?
msg['listaSINO.'.concat(tabla.registraAuditoria)]
                                :
msg['listaSINO.N']}"/>
                                </p:column>
                                <p:column
headerText="#{msg['sb.menu.gestion.administracion.tablas.campos.Descripcion.header']}"
                                filterBy="#{tabla.campos.Descripcion}"
                                sortBy="#{tabla.campos.Descripcion}"
                                filterMatchMode="contains"
                                filterStyle="display:none;"
                                style="text-align: center"
                                width="20%"
                                >
                                <h:outputText value="#{tabla.campos.Descripcion}"/>
                                </p:column>
                                <p:column
headerText="#{msg['sb.menu.gestion.administracion.tablas.campos.header']}"
                                style="text-align: center"
                                width="10%"
                                >
                                <p:commandLink style="color: #fff;"
process="@this"
title="#{msg['sb.menu.gestion.administracion.tablas.campos.button']}"
                                styleClass="btn btn-xs btn-
primary sbTooltip"
                                immediate="true"
                                disabled="#{tabla.tabla eq
'ACCESO AL SISTEMA'}"
actionListener="#{tablaConvController.irCamposTabla(tabla)}"
                                >
                                <i class="fa fa-cubes"></i>
                                </p:commandLink>
                                </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['sb.menu.gestion.id']}" />
                                <h:outputText
value="#{tabla.tablaId}" />
                                <h:outputText
value="#{msg['sb.menu.gestion.usuarioCreador']}" />
                                <h:outputText
value="#{tabla.usuarioCreador}" />

```

```

value="#{msg['sb.menu.gestion.fechaCreacion']}" />
value="#{tabla.fechaCreacion}">
    <f:convertDateTime locale="es" pattern="dd/MM/yyyy"/>
value="#{msg['sb.menu.gestion.usuarioModificador']}" />
value="#{tabla.usuarioModificador}" />
value="#{msg['sb.menu.gestion.fechaModificacion']}" />
value="#{tabla.fechaModificacion}">
    <f:convertDateTime locale="es" pattern="dd/MM/yyyy"/>
</h:outputText>
</h:outputText>
</h:outputText>
</h:outputText>
</h:outputText>
</h:outputText>
</p:panelGrid>
</div>
</div>
<p:spacer/>
</div>
</p:rowExpansion>
</p:dataTable>
</div>
</div>
</p:tab>

```

Este código corresponde a una pestaña dentro de una interfaz de usuario basada en JSF y PrimeFaces. Se encarga de gestionar la visualización y administración de tablas dentro de un módulo de administración.

La pestaña incluye:

- Un botón para cargar tablas.
- Una tabla de datos (**p:dataTable**) con paginación, filtrado y ordenamiento.
- Eventos de selección de fila.
- Expansión de filas para mostrar información detallada.

Secciones del Código

1. Definición de la Pestaña

```
<p:tab title="#{msg['sb.menu.gestion.administracion.tablas.header']}">
```

- Define una pestaña con un título obtenido desde un archivo de mensajes (msg).

2. Botón para Cargar Tablas

```
<p:commandLink id="cargarTablas"
    title="#{msg['sb.menu.gestion.administracion.tablas.cargarTablas.button']}"
    styleClass="btn btn-primary pull-right tooltips"
    update="#{tablaViewController.modulo}Form"
    actionListener="#{tablaConvController.cargarTablas()}">
    <i class="fa fa-refresh">
#{msg['sb.menu.gestion.administracion.tablas.cargarTablas.button']} </i>
</p:commandLink>
```

- Es un **p:commandLink** (botón con acción) que:
 - Llama al método **cargarTablas()** del **tablaConvController**.
 - Actualiza la interfaz (**update**) tras ejecutar la acción.
 - Muestra un icono de recarga (**fa fa-refresh**).

3. Tabla de Datos

```
<p:dataTable id="#{tablaViewController.modulo}Table"
    var="tabla"
    value="#{tablaViewController.model}"
    rowKey="#{tabla.tablaId}"
    selection="#{tablaConvController.selected}"
    selectionMode="single"
    paginator="true"
    rows="10"
    lazy="true"
    sortField="tabla"
    sortOrder="ASCENDING"
    currentPageReportTemplate="#{msg['sb.tabla.totalRegistros']} {totalRecords}"
    paginatorTemplate="{CurrentPageReport} {FirstPageLink} {PreviousPageLink}
{PageLinks} {NextPageLink} {LastPageLink} {RowsPerPageDropdown}"
    rowsPerPageTemplate="10,20,30"
    paginatorPosition="bottom"
    emptyMessage="#{msg['sb.datatable.registro.empty']}"
    widgetVar="items#{tablaViewController.modulo}">
```

- p:dataTable muestra los datos de tablaViewController.model.
- Configuraciones:
 - selectionMode="single": Permite seleccionar una fila a la vez.
 - paginator="true": Habilita la paginación.
 - lazy="true": Usa carga diferida de datos.

- sortField="tabla" sortOrder="ASCENDING": Ordena por defecto por la columna "tabla" en orden ascendente.
- emptyMessage: Mensaje cuando no hay registros.

4. Eventos de Selección de Fila

```
<p:ajax event="rowSelect"
  listener="#{tablaConvController.onRowSelect}"
  update="#{tablaViewController.modulo}Table"
  process="@this"/>
<p:ajax event="rowUnselect"
  listener="#{tablaConvController.onRowUnselect}"
  update="#{tablaViewController.modulo}Table"
  process="@this"/>
```

- Capturan eventos de selección (rowSelect) y deselección (rowUnselect) de filas.
- Invocan métodos correspondientes en tablaConvController.

5. Columnas de la Tabla

Ejemplo de una columna con ordenamiento y filtrado:

```
<p:column headerText="#{msg['sb.menu.gestion.administracion.tablas.descripcion.header']}"
  filterBy="#{tabla.descripcionTabla}"
  sortBy="#{tabla.descripcionTabla}"
  filterMatchMode="contains"
  style="text-align: center"
  width="25%">
  <h:outputText value="#{tabla.descripcionTabla}"/>
</p:column>
```

- headerText: Nombre de la columna.
- filterBy: Permite filtrar registros.
- sortBy: Habilita ordenamiento.
- filterMatchMode="contains": Filtra registros que contengan el texto ingresado.
- h:outputText: Muestra el valor de tabla.descripcionTabla.

6. Expansión de Fila

```
<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['sb.menu.gestion.id']}" />
<h:outputText value="#{tabla.tablaId}" />

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

<h:outputText value="#{msg['sb.menu.gestion.fechaCreacion']}" />
<h:outputText value="#{tabla.fechaCreacion}">
    <f:convertDateTime locale="es" pattern="dd/MM/yyyy"/>
</h:outputText>
</p:panelGrid>
</div>
</div>
<p:spacer/>
</div>
</p:rowExpansion>
```

- **p:rowExpansion**: Define contenido expandible dentro de cada fila.
- **p:panelGrid**: Organiza los datos en una tabla con dos columnas.
- **f:convertDateTime**: Formatea fechas en formato dd/MM/yyyy.

7. Acción sobre Campos de una Tabla

```
<p:commandLink title="#{msg['sb.menu.gestion.administracion.tablas.campos.button']}"
    styleClass="btn btn-xs btn-primary sbTooltip"
    immediate="true"
    disabled="#{tabla.tabla eq 'ACCESO AL SISTEMA'}"
    actionListener="#{tablaConvController.irCamposTabla(tabla)}">
    <i class="fa fa-cubes"></i>
</p:commandLink>
```

- **p:commandLink**: Botón de acción para gestionar los campos de una tabla.
- **disabled**: Deshabilita el botón para la tabla "ACCESO AL SISTEMA".
- **actionListener**: Invoca **irCamposTabla(tabla)** al hacer clic.

Este código proporciona una interfaz rica en funcionalidades para administrar tablas en una aplicación JSF con PrimeFaces. Permite la carga dinámica de datos, filtrado, ordenamiento, selección de filas y visualización detallada de registros, optimizando la gestión de información en la aplicación.

Pestaña de Captura de Datos para Registro de Tablas

```
<!--PESTAÑA GUARDAR/EDITAR -->
<p:tab title="#{msg['sb.component.tabview.guardarEditar']}">
    <p:panel id="formularioEditarCrear">
```

```

<div class="ibox-content">

    <br/>

    <!-- Fila tabla, descripciónTabla -->
    <div class="row">
        <div class="col-lg-12">
            <div class="col-lg-6">
                <div class="form-group">
                    <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.tablas.tabla.label']}"
styleClass="col-lg-3 control-label"
align: right"
for="tabla"/>

                    <div class="col-lg-8">
                        <p:inputText id="tabla"

onblur="value = value.toUpperCase()"
value="#{tablaViewController.selected.tabla}"
required="true"
maxlength="30"
placeholder="#{msg['sb.menu.gestion.administracion.tablas.tabla.placeholder']}"
requiredMessage="#{msg['sb.menu.gestion.administracion.tablas.tabla.required']}"
styleClass="form-control">

                        </p:inputText>
                    </div>
                </div>
            </div>
            <div class="col-lg-6">
                <div class="form-group">
                    <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.tablas.descripcion.label']}"
styleClass="col-lg-4 control-label"
align: right"
for="descripcion"/>

                    <div class="col-lg-8">
                        <p:inputText id="descripcion"

value="#{tablaViewController.selected.descripcionTabla}"
required="true"
maxlength="200"
placeholder="#{msg['sb.menu.gestion.administracion.tablas.descripcion.placeholder']}"
requiredMessage="#{msg['sb.menu.gestion.administracion.tablas.descripcion.required']}"
styleClass="form-control">

```

```

</p:inputText>
</div>
</div>
</div>
</div>
<br/>
<br/>
<!-- Fila secuencia, camposDescripción -->
<div class="row">
  <div class="col-lg-12">
    <div class="col-lg-6">
      <div class="form-group">
        <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.tablas.secuencia.label']}"
styleClass="col-lg-3 control-label"
align: justify"
for="secuencia"/>
        <div class="col-lg-8">
          <p:selectOneMenu id="secuencia"
value="#{tablaViewController.selected.secuencia}"
style="width:100%"
panelStyle=""
required="false"
>
          <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
          <f:selectItems
value="#{tablaViewController.listaSecuencias}"/>
        </p:selectOneMenu>
      </div>
    </div>
    <div class="col-lg-6">
      <div class="form-group">
        <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.tablas.camposDescripcion.label']}"
styleClass="col-lg-4 control-label"
align: right"
for="camposDescripción"/>
        <div class="col-lg-8">
          <p:inputText
id="camposDescripción"
value="#{tablaViewController.selected.camposDescripcion}"
required="false"

```

```

maxlength="200"

placeholder="#{msg['sb.menu.gestion.administracion.tablas.camposDescripcion.placeholder']}"

styleClass="form-control">
    </p:inputText>
</div>
</div>
</div>
</div>
<br/>
<br/>
<!-- Fila registraAuditoria -->
<div class="row">
    <div class="col-lg-12">
        <div class="col-lg-6">
            <div class="form-group">
                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.tablas.registraAuditoria.label']}"
styleClass="col-lg-4 control-label"
for="registraAuditoria"/>
                <div class="col-lg-4">
                    <h:selectBooleanCheckbox
id="registraAuditoria"
value="#{tablaViewController.selected.registraAuditoriaBool}"
required="false"
styleClass="js-switch"
>
                    </h:selectBooleanCheckbox>
                </div>
            </div>
        </div>
    </div>
</div>
<br/>
<br/>
<div class="hr-line-dashed"></div>
<ui:decorate template="../../template/_formButtons.xhtml">
    <ui:define name="comandos">
        <ui:param name="sessionController"
value="#{tablaViewController}" />
        <ui:param name="conversationController"
value="#{tablaConvController}" />
    </ui:define>
</ui:decorate>
</div>
</p:panel>
</p:tab>

```

Este fragmento de código en JSF y PrimeFaces define una pestaña dentro de un componente **p:tabView** utilizada para la captura y edición de datos en un formulario de administración de tablas.

1. Estructura General

El código define una pestaña (**p:tab**) con el título obtenido desde un archivo de mensajes **msg['sb.component.tabview.guardarEditar']**. Dentro de esta pestaña, se encuentra un **p:panel** (**formularioEditarCrear**) que contiene el formulario de captura de datos estructurado en diferentes filas (**div.row**).

2. Campos del Formulario

El formulario está dividido en distintas secciones con etiquetas (**p:outputLabel**) y componentes de entrada de datos (**p:inputText**, **p:selectOneMenu**, **h:selectBooleanCheckbox**).

2.1. Campos: Nombre de la Tabla y Descripción

- **p:outputLabel**: Etiqueta para el campo de entrada.
- **p:inputText** (ID: tabla): Campo donde el usuario ingresa el nombre de la tabla.
 - Convierte el texto a mayúsculas al perder el foco (**onblur="value = value.toUpperCase()"**).
 - Tiene un límite de 30 caracteres (**maxlength="30"**).
 - Mensajes de validación definidos en el archivo de mensajes (**requiredMessage**).
- **p:inputText** (ID: descripcion): Campo para la descripción de la tabla.
 - Límite de 200 caracteres (**maxlength="200"**).

2.2. Campos: Secuencia y Descripción de Campos

- **p:selectOneMenu** (ID: secuencia): Menú desplegable para seleccionar una secuencia de la lista **tablaViewController.listaSecuencias**.
- **p:inputText** (ID: camposDescripcion): Campo opcional para la descripción de los campos.

2.3. Checkbox: Registro de Auditoría

- **h:selectBooleanCheckbox** (ID: registraAuditoria): Casilla de verificación para activar o desactivar la auditoría en la tabla.
 - Enlazado a `tableViewController.selected.registraAuditoriaBool`.

3. Botones y Acciones

- Se utiliza `ui:decorate` para incluir un `template _formButtons.xhtml` que contiene los botones de acción.
- Se pasan los controladores `tableViewController` y `tablaConvController` mediante `ui:param`.

4. Características Adicionales

- **Clases de estilo (styleClass):** Se utilizan clases de Bootstrap (`col-lg-*`, `form-control`, `control-label`) para una mejor distribución del formulario.
- **Mensajes de validación y placeholders:** Se definen con variables del archivo de mensajes (`msg[]`), lo que facilita la internacionalización.
- **Divisores (hr-line-dashed):** Mejoran la organización visual del formulario.

Visualización de la pantalla

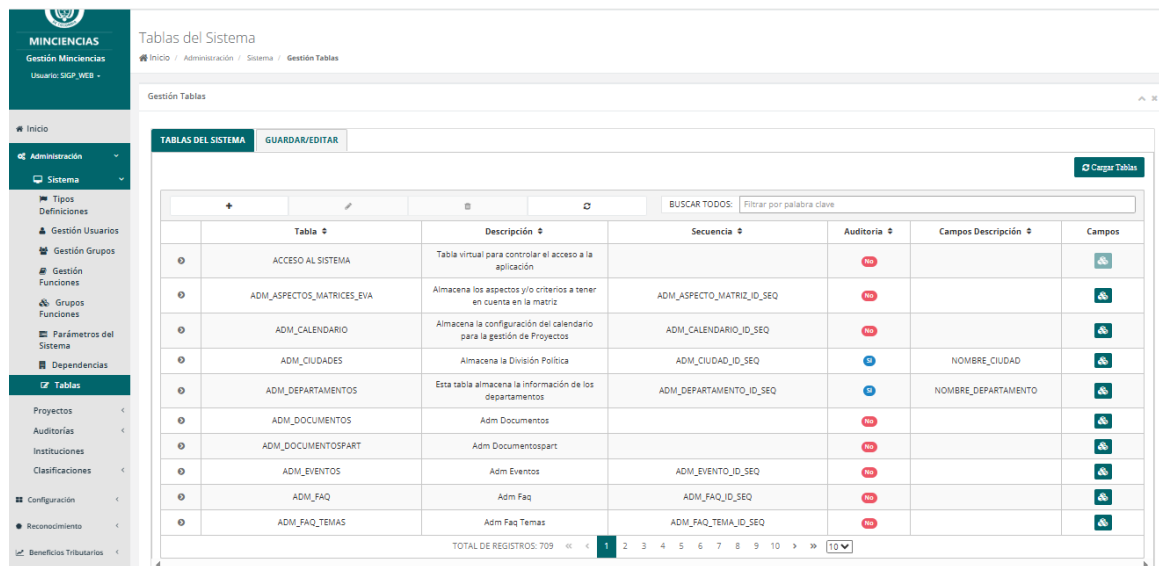
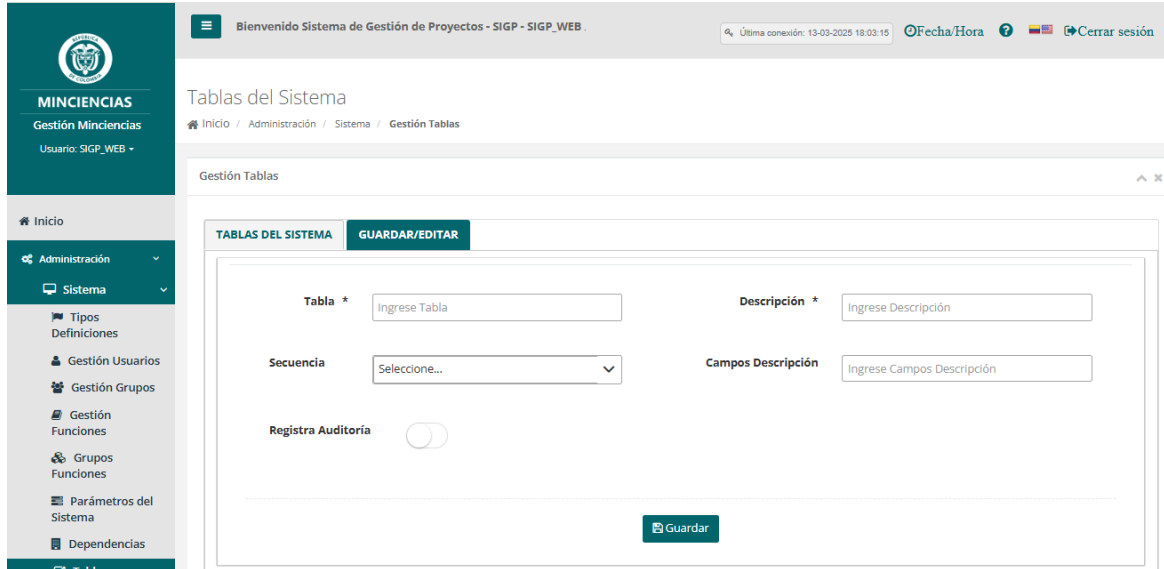


Tabla	Descripción	Secuencia	Auditoria	Campos Descripción	Campos
ACCESO AL SISTEMA	Tabla virtual para controlar el acceso a la aplicación		Yes		
ADM_ASPECTOS_MATRICES_EVA	Almacena los aspectos y/o criterios a tener en cuenta en la matriz	ADM_ASPECTO_MATRIZ_ID_SEQ	Yes		
ADM_CALENDARIO	Almacena la configuración del calendario para la gestión de Proyectos	ADM_CALENDARIO_ID_SEQ	Yes		
ADM_CIUDADES	Almacena la División Política	ADM_CIUADAD_ID_SEQ	No	NOMBRE_CIUADAD	
ADM_DEPARTAMENTOS	Esta tabla almacena la información de los departamentos	ADM_DEPARTAMENTO_ID_SEQ	No	NOMBRE_DEPARTAMENTO	
ADM_DOCUMENTOS	Adm Documentos		Yes		
ADM_DOCUMENTOSPART	Adm Documentospart		Yes		
ADM_EVENTOS	Adm Eventos	ADM_EVENTO_ID_SEQ	Yes		
ADM_FAQ	Adm Faq	ADM_FAQ_ID_SEQ	Yes		
ADM_FAQ_TEMAS	Adm Faq Temas	ADM_FAQ_TEMA_ID_SEQ	Yes		



Creación pantalla de Administración de Campos

Función:

Permite la gestión de los campos de cada tabla en el sistema, definiendo sus características y restricciones.

Principales funcionalidades:

- **Listado de Campos:** Se muestran los campos configurados por tabla.
- **Configuración de Propiedades:** Se definen atributos como nombre, tipo de dato, longitud, restricciones y relaciones con otras tablas.
- **Validaciones:** Se establecen reglas de obligatoriedad, claves primarias y auditoría.

Creación nuevo menú de configuración de Campos por Tablas

```
-- funciones (menus)
insert into 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', 'GEST6182', 'Campos', 'F', null, null,
'/app/administracion/campos.xhtml', null, 'GEST6100', null, 'SIGP_WEB', to_date('18-04-2024
16:55:11', 'dd-mm-yyyy hh24:mi:ss'), null, null);

-- derechos funciones (permisos)
insert into sb_derechos_funciones (DERECHO_FUNCION_ID, APLICACION, FUNCION, GRUPO, ESTADO,
USUARIO_CREADOR, FECHA_CREACION, USUARIO_MODIFICADOR, FECHA_MODIFICACION)
values (null, 'SIGP_WEB', 'GEST6182', 'SGPS_ADMINISTRADOR', 'H', 'SIGP_WEB', to_date('28-02-
2025 12:31:49', 'dd-mm-yyyy hh24:mi:ss'), null, null);
```

*Crear nueva entidad (**Campo.java**) de la tabla **SB_CAMPOS** en la base de datos sigp*

```
/*
 * 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.business.domain;

import com.sb.colciencias.gestion.business.domain.security.Aplicacion;
import com.sb.colciencias.gestion.business.valueObjects.AppSessionVO;
import java.io.Serializable;
import java.util.Date;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;
import javax.persistence.TemporalType;
import javax.persistence.Transient;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author dcangrejo
 */
@Entity
@Table(name = "SB_CAMPOS", schema = "HERCULES")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "Campo.findAll", query = "SELECT s FROM Campo s")
    , @NamedQuery(name = "Campo.findById", query = "SELECT s FROM Campo s WHERE
s.campoId = :campoId")
    , @NamedQuery(name = "Campo.findByAplicacion", query = "SELECT s FROM Campo s WHERE
s.aplicacion = :aplicacion")
    , @NamedQuery(name = "Campo.findByTabla", query = "SELECT s FROM Campo s WHERE
s.tabla.tabla = :tabla")
```

```

        , @NamedQuery(name = "Campo.findByCampo", query = "SELECT s FROM Campo s WHERE s.campo = :campo")
        , @NamedQuery(name = "Campo.findByDescripcionCampo", query = "SELECT s FROM Campo s WHERE s.descripcionCampo = :descripcionCampo")
        , @NamedQuery(name = "Campo.findByPosicion", query = "SELECT s FROM Campo s WHERE s.posicion = :posicion")
        , @NamedQuery(name = "Campo.findByObligatorio", query = "SELECT s FROM Campo s WHERE s.obligatorio = :obligatorio")
        , @NamedQuery(name = "Campo.findByLlavePrimaria", query = "SELECT s FROM Campo s WHERE s.llavePrimaria = :llavePrimaria")
        , @NamedQuery(name = "Campo.findByTipoDato", query = "SELECT s FROM Campo s WHERE s.tipoDato = :tipoDato")
        , @NamedQuery(name = "Campo.findByLongitud", query = "SELECT s FROM Campo s WHERE s.longitud = :longitud")
        , @NamedQuery(name = "Campo.findByDecimales", query = "SELECT s FROM Campo s WHERE s.decimales = :decimales")
        , @NamedQuery(name = "Campo.findByRegistraAuditoria", query = "SELECT s FROM Campo s WHERE s.registraAuditoria = :registraAuditoria")
        , @NamedQuery(name = "Campo.findByTablaFk", query = "SELECT s FROM Campo s WHERE s.tablaFk = :tablaFk")
        , @NamedQuery(name = "Campo.findByCampoFk", query = "SELECT s FROM Campo s WHERE s.campoFk = :campoFk")
        , @NamedQuery(name = "Campo.findByChequeo", query = "SELECT s FROM Campo s WHERE s.chequeo = :chequeo")
        , @NamedQuery(name = "Campo.findByComentario", query = "SELECT s FROM Campo s WHERE s.comentario = :comentario")
        , @NamedQuery(name = "Campo.findByUsuarioCreador", query = "SELECT s FROM Campo s WHERE s.usuarioCreador = :usuarioCreador")
        , @NamedQuery(name = "Campo.findByFechaCreacion", query = "SELECT s FROM Campo s WHERE s.fechaCreacion = :fechaCreacion")
        , @NamedQuery(name = "Campo.findByUsuarioModificador", query = "SELECT s FROM Campo s WHERE s.usuarioModificador = :usuarioModificador")
        , @NamedQuery(name = "Campo.findByFechaModificacion", query = "SELECT s FROM Campo s WHERE s.fechaModificacion = :fechaModificacion"))
    public class Campo implements Serializable {

        private static final long serialVersionUID = 1L;
        @Id
        @Basic(optional = false)
        @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "SB_CAMPO_ID_SEQ")
        @SequenceGenerator(name = "SB_CAMPO_ID_SEQ", sequenceName = "SB_CAMPO_ID_SEQ", allocationSize = 1)
        @NotNull
        @Column(name = "CAMPO_ID")
        private Long campoId;

        @Basic(optional = false)
        @NotNull
        @Size(min = 1, max = 10)
        @Column(name = "APLICACION")
        private String aplicacion;

        @Basic(optional = false)
        @NotNull
        @Size(min = 1, max = 100)
        @Column(name = "CAMPO")
        private String campo;

        @Basic(optional = false)
        @NotNull
        @Size(min = 1, max = 200)

```

```
@Column(name = "DESCRIPCION_CAMPO")
private String descripcionCampo;

@Basic(optional = false)
@NotNull
@Column(name = "POSICION")
private Long posicion;

@Basic(optional = false)
@NotNull
@Size(min = 1, max = 1)
@Column(name = "OBLIGATORIO")
private String obligatorio;

@Size(max = 1)
@Column(name = "LLAVE_PRIMARIA")
private String llavePrimaria;

@Size(max = 100)
@Column(name = "TIPO_DATO")
private String tipoDato;

@Column(name = "LONGITUD")
private Long longitud;

@Column(name = "DECIMALES")
private Short decimales;

@Size(max = 1)
@Column(name = "REGISTRA_AUDITORIA")
private String registraAuditoria;

@Size(max = 30)
@Column(name = "TABLA_FK")
private String tablaFk;

@Size(max = 100)
@Column(name = "CAMPO_FK")
private String campoFk;

@Size(max = 4000)
@Column(name = "CHEQUEO")
private String chequeo;

@Size(max = 4000)
@Column(name = "COMENTARIO")
private String comentario;

@Size(max = 30)
@Column(name = "USUARIO_CREADOR")
private String usuarioCreador;

@Column(name = "FECHA_CREACION")
@Temporal(TemporalType.TIMESTAMP)
private Date fechaCreacion;

@Size(max = 30)
@Column(name = "USUARIO_MODIFICADOR")
private String usuarioModificador;

@Column(name = "FECHA_MODIFICACION")
```

```
@Temporal(TemporalType.TIMESTAMP)
private Date fechaModificacion;

@Size(max = 1)
@Column(name = "LLAVE_UNICA")
private String llaveUnica;

//Joins
@JoinColumn(name = "TABLA", referencedColumnName = "TABLA")
@ManyToOne(optional = false)
private Tabla tabla;

@JoinColumn(name = "APLICACION_ID", referencedColumnName = "APLICACION_ID")
@ManyToOne(optional = false)
private Aplicacion aplicacionId;

//Auditoria
@Transient
private AppSessionVO appSessionVO;

public Campo() {
}

public Campo(Long campoId) {
    this.campoId = campoId;
}

public Campo(Long campoId, String aplicacion, Tabla tabla, String campo, String
descripcionCampo, Long posicion, String obligatorio) {
    this.campoId = campoId;
    this.aplicacion = aplicacion;
    this.tabla = tabla;
    this.campo = campo;
    this.descripcionCampo = descripcionCampo;
    this.posicion = posicion;
    this.obligatorio = obligatorio;
}

public Long getCampoId() {
    return campoId;
}

public void setCampoId(Long campoId) {
    this.campoId = campoId;
}

public String getAplicacion() {
    return aplicacion;
}

public void setAplicacion(String aplicacion) {
    this.aplicacion = aplicacion;
}

public Tabla getTabla() {
    return tabla;
}

public void setTabla(Tabla tabla) {
    this.tabla = tabla;
}
```

```
public String getCampo() {
    return campo;
}

public void setCampo(String campo) {
    this.campo = campo;
}

public String getDescripcionCampo() {
    return descripcionCampo;
}

public void setDescripcionCampo(String descripcionCampo) {
    this.descripcionCampo = descripcionCampo;
}

public Long getPosicion() {
    return posicion;
}

public void setPosicion(Long posicion) {
    this.posicion = posicion;
}

public String getObligatorio() {
    return obligatorio;
}

public void setObligatorio(String obligatorio) {
    this.obligatorio = obligatorio;
}

public String getLlavePrimaria() {
    return llavePrimaria;
}

public void setLlavePrimaria(String llavePrimaria) {
    this.llavePrimaria = llavePrimaria;
}

public String getTipoDato() {
    return tipoDato;
}

public void setTipoDato(String tipoDato) {
    this.tipoDato = tipoDato;
}

public Long getLongitud() {
    return longitud;
}

public void setLongitud(Long longitud) {
    this.longitud = longitud;
}

public Short getDecimales() {
    return decimales;
}
```

```
public void setDecimales(Short decimales) {
    this.decimales = decimales;
}

public String getRegistraAuditoria() {
    return registraAuditoria;
}

public void setRegistraAuditoria(String registraAuditoria) {
    this.registraAuditoria = registraAuditoria;
}

public String getTablaFk() {
    return tablaFk;
}

public void setTablaFk(String tablaFk) {
    this.tablaFk = tablaFk;
}

public String getCampoFk() {
    return campoFk;
}

public void setCampoFk(String campoFk) {
    this.campoFk = campoFk;
}

public String getChequeo() {
    return chequeo;
}

public void setChequeo(String chequeo) {
    this.chequeo = chequeo;
}

public String getComentario() {
    return comentario;
}

public void setComentario(String comentario) {
    this.comentario = comentario;
}

public String getUsuarioCreador() {
    return usuarioCreador;
}

public void setUsuarioCreador(String usuarioCreador) {
    this.usuarioCreador = usuarioCreador;
}

public Date getFechaCreacion() {
    return fechaCreacion;
}

public void setFechaCreacion(Date fechaCreacion) {
    this.fechaCreacion = fechaCreacion;
}

public String getUsuarioModificador() {
```

```
        return usuarioModificador;
    }

    public void setUsuarioModificador(String usuarioModificador) {
        this.usuarioModificador = usuarioModificador;
    }

    public Date getFechaModificacion() {
        return fechaModificacion;
    }

    public void setFechaModificacion(Date fechaModificacion) {
        this.fechaModificacion = fechaModificacion;
    }

    public String getLlaveUnica() {
        return llaveUnica;
    }

    public void setLlaveUnica(String llaveUnica) {
        this.llaveUnica = llaveUnica;
    }

    public Aplicacion getAplicacionId() {
        return aplicacionId;
    }

    public void setAplicacionId(Aplicacion aplicacionId) {
        this.aplicacionId = aplicacionId;
    }

    public AppSessionVO getAppSessionVO() {
        return appSessionVO;
    }

    public void setAppSessionVO(AppSessionVO appSessionVO) {
        this.appSessionVO = appSessionVO;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (campoId != null ? campoId.hashCode() : 0);
        return hash;
    }

    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof Campo)) {
            return false;
        }
        Campo other = (Campo) object;
        if ((this.campoId == null && other.campoId != null) || (this.campoId != null &&
!this.campoId.equals(other.campoId))) {
            return false;
        }
        return true;
    }

    @Override
```

```
        public String toString() {  
            return "com.sb.colciencias.gestion.business.domain.Campo[ campoId=" + campoId + "  
        ]";  
        }  
    }  
}
```

*Crear nuevo DTO (**CampoDTO.java**) representativo de la entidad*

```
/*  
 * To change this template, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package com.sb.colciencias.gestion.business.transfer;  
  
import com.sb.colciencias.gestion.business.domain.Tabla;  
import com.sb.colciencias.gestion.business.domain.security.Aplicacion;  
import com.sb.colciencias.gestion.business.util.annotation.Bind;  
import com.sb.colciencias.gestion.business.valueObjects.AppSessionVO;  
import java.io.Serializable;  
import java.util.Date;  
  
/**  
 *  
 * @author dcangrejo  
 */  
public class CampoDTO extends GenericDTO implements Serializable {  
  
    @Bind(name = "campoId")  
    private Long campoId;  
  
    @Bind(name = "aplicacion")  
    private String aplicacion;  
  
    @Bind(name = "campo")  
    private String campo;  
  
    @Bind(name = "descripcionCampo")  
    private String descripcionCampo;  
  
    @Bind(name = "posicion")  
    private Long posicion;  
  
    @Bind(name = "obligatorio")  
    private String obligatorio;  
  
    @Bind(name = "llavePrimaria")  
    private String llavePrimaria;  
  
    @Bind(name = "tipoDato")  
    private String tipoDato;  
  
    @Bind(name = "longitud")  
    private Long longitud;  
  
    @Bind(name = "decimales")  
    private Short decimales;  
}
```

```
@Bind(name = "registraAuditoria")
private String registraAuditoria;

@Bind(name = "tablaFk")
private String tablaFk;

@Bind(name = "campoFk")
private String campoFk;

@Bind(name = "chequeo")
private String chequeo;

@Bind(name = "comentario")
private String comentario;

@Bind(name = "usuarioCreador")
private String usuarioCreador;

@Bind(name = "fechaCreacion")
private Date fechaCreacion;

@Bind(name = "usuarioModificador")
private String usuarioModificador;

@Bind(name = "fechaModificacion")
private Date fechaModificacion;

@Bind(name = "llaveUnica")
private String llaveUnica;

//Joins
@Bind(name = "tabla", returnType = TablaDTO.class, type = Tabla.class)
private TablaDTO tabla;
@Bind(name = "aplicacionId", returnType = AplicacionDTO.class, type = Aplicacion.class)
private AplicacionDTO aplicacionId;

//Campos checkBox
private Boolean registraAuditoriaBool;
private Boolean obligatorioBool;
private Boolean llavePrimariaBool;

//Auditoria
@Bind(name = "appSessionVO")
private AppSessionVO appSessionVO;

public CampoDTO() {

}

public Long getCampoId() {
    return campoId;
}

public void setCampoId(Long campoId) {
    this.campoId = campoId;
}

public String getAplicacion() {
    return aplicacion;
}
```

```
public void setAplicacion(String aplicacion) {
    this.aplicacion = aplicacion;
}

public String getCampo() {
    return campo;
}

public void setCampo(String campo) {
    this.campo = campo;
}

public String getDescripcionCampo() {
    return descripcionCampo;
}

public void setDescripcionCampo(String descripcionCampo) {
    this.descripcionCampo = descripcionCampo;
}

public Long getPosicion() {
    return posicion;
}

public void setPosicion(Long posicion) {
    this.posicion = posicion;
}

public String getObligatorio() {
    return obligatorio;
}

public void setObligatorio(String obligatorio) {
    if (obligatorio != null && obligatorio.equals("S")) {
        this.obligatorioBool = true;
    }
    this.obligatorio = obligatorio;
}

public Boolean getObligatorioBool() {
    return obligatorioBool;
}

public void setObligatorioBool(Boolean obligatorioBool) {
    if (obligatorioBool) {
        this.obligatorio = "S";
    } else {
        this.obligatorio = "N";
    }
    this.obligatorioBool = obligatorioBool;
}

public String getLlavePrimaria() {
    return llavePrimaria;
}

public void setLlavePrimaria(String llavePrimaria) {
    if (llavePrimaria != null && llavePrimaria.equals("S")) {
        this.llavePrimariaBool = true;
    }
    this.llavePrimaria = llavePrimaria;
}
```

```
}

public Boolean getLlavePrimariaBool() {
    return llavePrimariaBool;
}

public void setLlavePrimariaBool(Boolean llavePrimariaBool) {
    if (llavePrimariaBool) {
        this.llavePrimaria = "S";
    } else {
        this.llavePrimaria = "N";
    }
    this.llavePrimariaBool = llavePrimariaBool;
}

public String getTipoDato() {
    return tipoDato;
}

public void setTipoDato(String tipoDato) {
    this.tipoDato = tipoDato;
}

public Long getLongitud() {
    return longitud;
}

public void setLongitud(Long longitud) {
    this.longitud = longitud;
}

public Short getDecimales() {
    return decimales;
}

public void setDecimales(Short decimales) {
    this.decimales = decimales;
}

public String getRegistraAuditoria() {
    return registraAuditoria;
}

public void setRegistraAuditoria(String registraAuditoria) {
    if (registraAuditoria != null && registraAuditoria.equals("S")) {
        this.registraAuditoriaBool = true;
    }
    this.registraAuditoria = registraAuditoria;
}

public Boolean getRegistraAuditoriaBool() {
    return registraAuditoriaBool;
}

public void setRegistraAuditoriaBool(Boolean registraAuditoriaBool) {
    if (registraAuditoriaBool) {
        this.registraAuditoria = "S";
    } else {
        this.registraAuditoria = "N";
    }
    this.registraAuditoriaBool = registraAuditoriaBool;
}
```

```
}

public String getTablaFk() {
    return tablaFk;
}

public void setTablaFk(String tablaFk) {
    this.tablaFk = tablaFk;
}

public String getCampoFk() {
    return campoFk;
}

public void setCampoFk(String campoFk) {
    this.campoFk = campoFk;
}

public String getChequeo() {
    return chequeo;
}

public void setChequeo(String chequeo) {
    this.chequeo = chequeo;
}

public String getComentario() {
    return comentario;
}

public void setComentario(String comentario) {
    this.comentario = comentario;
}

public String getUsuarioCreador() {
    return usuarioCreador;
}

public void setUsuarioCreador(String usuarioCreador) {
    this.usuarioCreador = usuarioCreador;
}

public Date getFechaCreacion() {
    return fechaCreacion;
}

public void setFechaCreacion(Date fechaCreacion) {
    this.fechaCreacion = fechaCreacion;
}

public String getUsuarioModificador() {
    return usuarioModificador;
}

public void setUsuarioModificador(String usuarioModificador) {
    this.usuarioModificador = usuarioModificador;
}

public Date getFechaModificacion() {
    return fechaModificacion;
}
```

```

public void setFechaModificacion(Date fechaModificacion) {
    this.fechaModificacion = fechaModificacion;
}

public String getLlaveUnica() {
    return llaveUnica;
}

public void setLlaveUnica(String llaveUnica) {
    this.llaveUnica = llaveUnica;
}

public TablaDTO getTabla() {
    return tabla;
}

public void setTabla(TablaDTO tabla) {
    this.tabla = tabla;
}

public AppSessionVO getAppSessionVO() {
    return appSessionVO;
}

public void setAppSessionVO(AppSessionVO appSessionVO) {
    this.appSessionVO = appSessionVO;
}

public String getUserAuth() {
    return userAuth;
}

public void setUserAuth(String userAuth) {
    this.userAuth = userAuth;
}

public AplicacionDTO getAplicacionId() {
    return aplicacionId;
}

public void setAplicacionId(AplicacionDTO aplicacionId) {
    this.aplicacionId = aplicacionId;
}

@Override
public int hashCode() {
    int hash = 0;
    hash += (campoId != null ? campoId.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    if (!(object instanceof CampoDTO)) {
        return false;
    }
    CampoDTO other = (CampoDTO) object;
    if ((this.campoId == null && other.campoId != null) || (this.campoId != null &&
!this.campoId.equals(other.campoId))) {
        return false;
    }
}

```

```
    }  
    return true;  
}  
  
@Override  
public String toString() {  
    return "com.sb.colciencias.entidades.business.transfer.CampoDTO[ campoId= " +  
    campoId + " ]";  
}  
}
```

La clase CampoDTO es un objeto de transferencia de datos (DTO) utilizado para manejar información sobre los campos de una aplicación en un entorno de gestión de bases de datos. Sirve como puente entre la capa de persistencia y la capa de presentación, asegurando una manipulación eficiente de la información.

1. Importaciones Clave

- Serializable: Permite la serialización del objeto para su transmisión.
- Date: Manejo de fechas.
- Bind: Anotación personalizada para asignar nombres de atributos.
- TablaDTO y AplicacionDTO: Relaciones con otras entidades.

2. Atributos

Los atributos de la clase representan las propiedades del campo y están anotados con @Bind para facilitar su mapeo a la base de datos.

- **Datos principales:**
 - campold: Identificador único del campo.
 - aplicacion: Nombre de la aplicación a la que pertenece el campo.
 - campo: Nombre del campo.
 - descripcionCampo: Descripción del campo.
 - posicion: Ubicación del campo en la tabla.
 - tipoDato: Tipo de dato del campo (VARCHAR, INT, etc.).
 - longitud: Longitud del campo.

- decimales: Número de decimales en caso de ser un campo numérico.
- **Restricciones y claves:**
 - obligatorio: Indica si el campo es obligatorio ('S' para sí, 'N' para no).
 - llavePrimaria: Indica si el campo es clave primaria.
 - llaveUnica: Indica si el campo es único.
- **Auditoría:**
 - registraAuditoria: Indica si se audita el campo.
 - usuarioCreador y usuarioModificador: Usuarios que crean y modifican el campo.
 - fechaCreacion y fechaModificacion: Fechas de creación y modificación.
- **Relaciones:**
 - tablaFk y campoFk: Referencias a otra tabla y campo relacionado.
 - tabla: Objeto TablaDTO asociado.
 - aplicacionId: Objeto AplicacionDTO asociado.
- **Campos auxiliares:**
 - registraAuditoriaBool, obligatorioBool, llavePrimariaBool: Variables booleanas para facilitar la manipulación de los valores de auditoría, obligatoriedad y clave primaria.

3. Métodos Getter y Setter

Cada atributo tiene sus respectivos métodos get y set para acceder y modificar los valores.

- **Conversión de booleanos:**
 - Cuando obligatorio, llavePrimaria o registraAuditoria son establecidos con 'S' o 'N', también se actualizan sus respectivas variables booleanas.
 - Inversamente, cuando se establece una variable booleana, se actualiza el valor del atributo correspondiente con 'S' o 'N'.

4. Métodos de Utilidad

- hashCode() y equals(): Se implementan para comparar objetos CampoDTO correctamente según campold.
- toString(): Devuelve una representación en texto del objeto, incluyendo su campold.

Controlador de Vista CampoViewController.java

El **CampoViewController** es un controlador de vista dentro del módulo de administración que gestiona la visualización y manipulación de campos (**CampoDTO**) asociados a una tabla específica (**TablaDTO**). Implementa carga diferida (**LazyDataModel**) para optimizar la gestión de datos en grandes volúmenes, asegurando un rendimiento eficiente.

```
/*
 * 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.administracion.view;

import
com.sb.colciencias.gestion.business.facade.administracion.AdministracionServiceFacade;
import com.sb.colciencias.gestion.business.transfer.TablaDTO;
import com.sb.colciencias.gestion.business.transfer.CampoDTO;
import com.sb.colciencias.gestion.web.controller.application.ConfigApplController;
import com.sb.colciencias.gestion.web.session.ColcienciasSessController;
import com.sb.standard.web.controller.session.GenericSessionController;
import java.io.Serializable;
import java.util.List;
import java.util.Map;
import javax.ejb.EJB;
import javax.inject.Inject;
import javax.inject.Named;
import javax.annotation.PostConstruct;
import org.omnifaces.cdi.ViewScoped;
import org.primefaces.model.LazyDataModel;
import org.primefaces.model.SortOrder;
import org.primefaces.model.FilterMeta;
import java.util.TreeMap;

/**
 *
 * @author kasuarez
 */
@Named(value = "campoViewController")
@ViewScoped
public class CampoViewController extends GenericSessionController<CampoDTO> implements
Serializable {

    @EJB
    private AdministracionServiceFacade administracionServiceFacade;
    @Inject
```

```
private ColcienciasSessController colcienciasSessController;
@Inject
private ConfigApplController configApplController;

private LazyDataModel<CampoDTO> model;

private TablaDTO tablaSelected;

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

public CampoViewController() {
    super(CampoDTO.class);
}

@PostConstruct
public void cargarListas() {

    Map<String, Object> mapReferencias = (Map<String, Object>)
colcienciasSessController.getUtilObject();
    tablaSelected = (TablaDTO) mapReferencias.get("tablaSelected");

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

        @Override
        public List<CampoDTO> 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());
                    }
                }
            });

            try {
                filters.put("tabla", tablaSelected.getTabla());
                int count = administracionServiceFacade.campoListCount(filters);
                if (count > 0) {
                    list = administracionServiceFacade.campoList(first, pageSize,
sortField, sortOrder.equals(SortOrder.ASCENDING) ? "A" : "D", filters);
                }
                model.setRowCount(count);
            } catch (Exception ex) {
                LOG.error("Error al cargar el listado de campos ", ex);
            }
            return list;
        }

        @Override
        public Object getRowKey(CampoDTO campoDTO) {
            return campoDTO.getCampoId();
        }

        @Override
        public CampoDTO getRowData(String rowKey) {
```

```
        try {
            //Se consulta en la base de datos para mantener independiente el objeto
            que trae en la lista principal y no lo actualice por referencia
            CampoDTO campoDTO =
            administracionServiceFacade.queryCampo(Long.parseLong(rowKey));
            return campoDTO;
        } catch (Exception ex) {
            LOG.error("no se pudo cargar el campo ( id => " + rowKey + ")", ex);
        }
        return null;
    }
};

public LazyDataModel<CampoDTO> getModel() {
    return model;
}

public void setModel(LazyDataModel<CampoDTO> model) {
    this.model = model;
}

public ConfigApplController getConfigApplController() {
    return configApplController;
}

public void setConfigApplController(ConfigApplController configApplController) {
    this.configApplController = configApplController;
}

public TablaDTO getTablaSelected() {
    return tablaSelected;
}

public void setTablaSelected(TablaDTO tablaSelected) {
    this.tablaSelected = tablaSelected;
}
}
```

El **CampoViewController** proporciona una implementación robusta para la administración de campos en la aplicación, asegurando una carga eficiente con **LazyDataModel** y garantizando la integridad de los datos con consultas independientes.

Dependencias e Inyección de Dependencias

- @EJB AdministracionServiceFacade → Servicio de negocio que proporciona operaciones sobre los campos y tablas.
- @Inject ColcienciasSessController → Controlador de sesión que maneja la persistencia de datos de sesión.
- @Inject ConfigApplController → Controlador de configuración de la aplicación.

Atributos

- LazyDataModel<CampoDTO> model → Modelo de datos diferidos utilizado en la interfaz de usuario para la carga paginada de registros.
- TablaDTO tablaSelected → Representa la tabla actualmente seleccionada, cuyos campos serán gestionados.

Ciclo de Vida

@PostConstruct cargarListas()

Método de inicialización que:

1. Recupera la tabla seleccionada desde la sesión.
2. Inicializa el modelo LazyDataModel para cargar los CampoDTO asociados a la tabla seleccionada.

Carga Diferida (LazyDataModel)

- **load(int first, int pageSize, String sortField, SortOrder sortOrder, Map<String, FilterMeta> filterBy)**
 - Construye un conjunto de filtros a partir de los valores proporcionados por la interfaz.
 - Llama a campoListCount(filters) para obtener el total de registros que cumplen con los filtros.
 - Recupera los datos paginados con campoList(first, pageSize, sortField, sortOrder, filters).
 - Maneja errores de carga mediante el log.
- **getRowKey(CampoDTO campoDTO)**
 - Retorna el identificador único del campo.
- **getRowData(String rowKey)**
 - Recupera un campo específico desde la base de datos usando su ID.

- Evita que la referencia de objetos en la lista principal se actualice automáticamente.
- Registra errores en caso de fallo en la consulta.

Métodos Getters y Setters

Incluye métodos estándar para la manipulación de model, tablaSelected y configAppController.

Registro de Errores

Se emplea **org.apache.log4j.Logger** para registrar errores en la carga de datos y en la recuperación de campos individuales, garantizando trazabilidad y depuración eficiente.

*Controlador de Conversación **CampoConvController.java***

El CampoConvController es un controlador de conversación en una aplicación web basada en Java EE con PrimeFaces. Maneja la gestión de entidades CampoDTO, proporcionando funcionalidad para crear, editar, eliminar y mostrar registros en la interfaz de usuario.

Paquete y Dependencias

Paquete: com.sb.colciencias.gestion.web.controller.administracion.conversation

Dependencias Principales:

```
@EJB
private AdministracionServiceFacade administracionServiceFacade;
@EJB
private BeneficiosServiceFacade beneficiosServiceFacade;
@Inject
private CampoViewController campoViewController;
@Inject
private SeguridadSessController seguridadSessController;
@Inject
private ColcienciasSessController colcienciasSessController;
@Inject
private ConfigAppController configAppController;
```

- AdministracionServiceFacade: Proporciona servicios de administración para la gestión de campos.
- BeneficiosServiceFacade: Facilita el acceso a servicios de beneficios tributarios.
- CampoViewController: Controlador de vista para la gestión de CampoDTO.

- SeguridadSessController: Controlador de sesión para gestión de usuarios y seguridad.
- ColcienciasSessController: Controlador de sesión para objetos de uso general.
- ConfigApplController: Controlador de configuración de la aplicación.

Atributos

```
private final String programa = "GEST6182";

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

public CampoConvController() {
    this.selected = new CampoDTO();
}
```

- programa: Identificador del programa asociado a la gestión de campos.
- LOG: Logger para registrar eventos y errores.
- selected: Instancia de CampoDTO para la selección y manipulación de datos.

Métodos Principales

guardar(ActionEvent ae)

```
@Override
public void guardar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
        CampoDTO campoDTO = campoViewController.getSelected();
        campoDTO.setTabla(campoViewController.getTablaSelected());
        campoDTO.setUsuarioCreador(seguridadSessController.getSelected().getUsuario());
        campoDTO.setFechaCreacion(new Date());
        //Auditoria
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        appSessionVO.setPrograma(programa);
        campoDTO.setAppSessionVO(appSessionVO);
        campoDTO.setAplicacion(configApplController.getAplicacion().getAplicacion());
        administracionServiceFacade.guardarCampo(campoDTO);

        campoViewController.setSelected(new CampoDTO());
        //Actualizar las listas en tiempo de Ejecución
        configApplController.listasGenerales();

        campoViewController.cargarListas();
        //se deshabilita el boton de editar / eliminar ya que no habra nada seleccionado
        despues de la operación
        this.onRowUnselect(null);
        this.obtenerPestana = 0;
    } catch (SystemException ex) {
```

```

        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
        resolveExpression(ex.getMessage()));
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
        resolveExpression("#{msg['sb.mensajes.transaccion.fallido']}"));
        LOG.error("Error al guardar el Valor definicion ", ex);
    } finally {
        FacesContext.getCurrentInstance().addMessage(null, message);
    }
}

```

Guarda un nuevo campo en la base de datos.

- Asigna valores como usuario creador y fecha de creación.
- Llama al método **guardarCampo()** de **AdministracionServiceFacade**.
- Actualiza las listas de la aplicación.

eliminar(ActionEvent ae)

```

@Override
public void eliminar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
        CampoDTO campoDTO = campoViewController.getSelected();
        //Auditoria
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        appSessionVO.setPrograma(programa);
        campoDTO.setAppSessionVO(appSessionVO);
        administracionServiceFacade.eliminarCampo(campoDTO);

        campoViewController.setSelected(new CampoDTO());
        //Actualizar las listas en tiempo de Ejecución
        configApplController.listasGenerales();
        campoViewController.cargarListas();
        this.onRowUnselect(null);
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
        resolveExpression(ex.getMessage()));
        LOG.error("Error al eliminar el Valor definicion ", ex);
    } finally {
        FacesContext.getCurrentInstance().addMessage(null, message);
    }
}

```

Elimina un campo seleccionado.

- Asigna valores de auditoría.
- Llama a **eliminarCampo()** de **AdministracionServiceFacade**.

- Refresca la lista de campos en la UI.

editar(ActionEvent ae)

```
@Override
public void editar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
        CampoDTO campoDTO = campoViewController.getSelected();

        campoDTO.setUsuarioModificador(seguridadSessController.getSelected().getUsuario());
        campoDTO.setFechaModificacion(new Date());
        //Auditoria
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        appSessionVO.setPrograma(programa);
        campoDTO.setAppSessionVO(appSessionVO);
        administracionServiceFacade.editarCampo(campoDTO);

        campoViewController.setSelected(new CampoDTO());
        //Actualizar las listas en tiempo de Ejecución
        configApplController.listasGenerales();
        campoViewController.cargarListas();
        //se deshabilita el boton de editar / eliminar ya que no habra nada seleccionado
        despues de la operación
        this.onRowUnselect(null);
        this.obtenerPestana = 0;
    } catch (SystemException ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
            resolveExpression(ex.getMessage()));
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
            resolveExpression("#{msg['sb.mensajes.transaccion.fallido']}"));
        LOG.error("Error al editar el Tipo definicion ", ex);
    } finally {
        FacesContext.getCurrentInstance().addMessage(null, message);
    }
}
```

Edita un campo existente.

- Modifica los valores de usuario y fecha de modificación.
- Llama a editarCampo() de AdministracionServiceFacade.
- Actualiza las listas y deselecciona el registro modificado.

mostrar(ActionEvent ae)

```
@Override
public void mostrar(ActionEvent ae) {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
```

```

        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));
        //Obtengo la tabla+
        String modulo = campoViewController.getModulo();
        DataTable dataTable = (DataTable)
        FacesContext.getCurrentInstance().getViewRoot().findComponent(modulo + "Form:" + modulo +
        "TabList:" + modulo + "Table");
        //limpio los filtros
        dataTable.setFilterBy(null);
        //Se actualiza el formulario que contiene la tabla
        PrimeFaces.current().ajax().update(modulo + "Form");
        //Se muestra el growl con el mensaje
        FacesContext.getCurrentInstance().addMessage(null, message);

        campoViewController.cargarListas();
        this.onRowUnselect(null);
    }

```

Limpia filtros y actualiza la tabla de datos en la UI.

goNuevo(ActionEvent ae)

```

@Override
public void goNuevo(ActionEvent ae) {
    campoViewController.setSelected(new CampoDTO());
    campoViewController.modaGuardar();
    this.obtenerPestana = 1;
}

```

Prepara el formulario para la creación de un nuevo campo.

goEditar(ActionEvent ae)

```

@Override
public void goEditar(ActionEvent ae) {
    campoViewController.modaEditar();
    this.obtenerPestana = 1;
}

```

Habilita el modo edición en la vista.

onRowSelect(SelectEvent event)

```

@Override
public void onRowSelect(SelectEvent event) {
    super.onRowSelect(event);
    campoViewController.setSelected(this.selected);
    campoViewController.setActivaEditar(true);
    campoViewController.modaEditar();
}

```

Maneja la selección de una fila en la tabla.

onRowUnselect(UnselectEvent event)

```
@Override
public void onRowUnselect(UnselectEvent event) {
    super.onRowUnselect(event);
    campoViewController.setSelected(new CampoDTO());
    campoViewController.setActivaEditar(false);
    campoViewController.modaGuardar();
    selected = null;
}
```

Maneja la deselección de una fila.

volverTabla()

```
public void volverTabla() {
    Map<String, Object> mapReferencias = new HashMap<>();
    colcienciasSessController.setUtilObject(mapReferencias);

    this.redirectPage("app/administracion/tablas.xhtml");
}
```

Redirige a la pantalla principal de gestión de tablas.

cargarCampos()

```
public void cargarCampos() {

    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));

    try {
        AppSessionVO appSessionVO = seguridadSessController.getAppSessionVO();
        String mensaje = null;

        Map<String, Object> filters = new HashMap<>();
        filters.put("paquete",
beneficiosServiceFacade.obtenerParametroSistema("PKGTABLAS").getCadena());
        filters.put("p_aplicacion",
configApplController.getAplicacion().getAplicacion());
        filters.put("p_usuario", seguridadSessController.getSelected().getUsuario());
        filters.put("p_terminal", appSessionVO.getIpAddress());
        filters.put("p_job_accion",
beneficiosServiceFacade.obtenerParametroSistema("PKGCAMPOS").getCadena().replace("?1",
campoViewController.getTablaSelected().getTabla());
        filters.put("tabla", campoViewController.getTablaSelected().getTabla());
        filters.put("accionPantalla", "E");

        mensaje = administracionServiceFacade.callSBCargarTablasCampos(filters);
        System.out.println("mensaje: " + mensaje);
        if (mensaje != null && !mensaje.equals("")) {
            message = new FacesMessage(FacesMessage.SEVERITY_INFO,
resolveExpression("#{msg['sb.mensajes.titulo.exito']}"), mensaje);
            FacesContext.getCurrentInstance().addMessage(null, message);
        }
        campoViewController.cargarListas();

    } catch (SystemException ex) {
```

```

        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        FacesContext.getCurrentInstance().addMessage(null, message);
        LOG.error("Error al cargar campos", ex);
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression("#{msg['sb.mensajes.transaccion.fallido']}"));
        FacesContext.getCurrentInstance().addMessage(null, message);
        LOG.error("Error al cargar campos", ex);
    }
}

```

Carga los campos configurados llamando a un procedimiento almacenado en la base de datos.

Auditoría y Seguridad

- Usa **AppSessionVO** para registrar información del usuario y del programa.
- Interactúa con **SeguridadSessController** para manejar credenciales y permisos.

Manejo de Errores

- Captura excepciones de tipo **SystemException** y **Exception**.
- Registra errores con **LOG.error()**.
- Muestra mensajes en la interfaz de usuario con **FacesMessage**.

Este controlador facilita la gestión de campos en la aplicación web, asegurando operaciones seguras y auditadas dentro de la plataforma.

*Se definen métodos de CRUD en Facade EJB (**AdministracionServiceFacade.java**) para funcionalidad de pantalla*

```

//
//CAMPOS
/**
 * Consulta el listado de campos 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
 * tipos definiciones
 * @return lista de tipos definiciones encontrados en el rango enviado y
 * basado en los filtros enviados
 * @throws Exception
 */

```

```
List<CampoDTO> campoList(int first, int pageSize, String sortField, String sortOrder,
Map<String, Object> filters) throws Exception;

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

/**
 * Guarda el registro del campo en la base de datos
 *
 * @param campoDTO
 * @throws Exception
 */
void guardarCampo(CampoDTO campoDTO) throws Exception;

/**
 * Edita el registro del campo en la base de datos
 *
 * @param campoDTO
 * @throws Exception
 */
void editarCampo(CampoDTO campoDTO) throws Exception;

/**
 * Elimina el registro del campo en la base de datos
 *
 * @param campoDTO
 * @throws Exception
 */
void eliminarCampo(CampoDTO campoDTO) throws Exception;

/**
 * Buscar el registro de un campo y lo retorna basado en el id (llave
 * primaria)
 *
 * @param campoId
 * @return el campo que coincide con el id enviado
 * @throws Exception
 */
CampoDTO queryCampo(Long campoId) throws Exception;

/**
 * Metodo para obtener el listado de valores de un campo de una tabla
 *
 * @param tabla
 * @param campo
 * @param query
 * @return
 * @throws Exception
 */
List<Object> obtenerValoresCampo(String tabla, String campo, String query) throws
Exception;

/**
 * Metodo para obtener el campo Llave primaria de una tabla
 *
 */
```

```
* @param tabla
* @param campo
* @return
* @throws Exception
*/
String obtenerCampoPKTabla(String tabla, String campo) throws Exception;
```

*Se implementan métodos CRUD en facade EJB
(AdministracionServiceFacadeBean.java) para funcionalidad de pantalla*

```
//----- Campos -----//
@Override
public List<CampoDTO> campoList(int first, int pageSize, String sortField, String
sortOrder, Map<String, Object> filters) throws Exception {
    ObjectConverter<CampoDTO, Campo> converter = objectConverterFactory.checkOut();
    List<CampoDTO> listCampoDTOs =
converter.toTransferList(campoService.campoList(first, pageSize, sortField, sortOrder,
filters), CampoDTO.class);
    objectConverterFactory.checkIn(converter);
    return listCampoDTOs;
}

@Override
public int campoListCount(Map<String, Object> filters) throws Exception {
    return campoService.campoListCount(filters);
}

@Override
public void guardarCampo(CampoDTO campoDTO) throws Exception {
    ObjectConverter<CampoDTO, Campo> converter = objectConverterFactory.checkOut();
    campoService.guardarCampo(converter.toEntity(campoDTO, Campo.class));
    objectConverterFactory.checkIn(converter);
}

@Override
public void editarCampo(CampoDTO campoDTO) throws Exception {
    ObjectConverter<CampoDTO, Campo> converter = objectConverterFactory.checkOut();
    campoService.editarCampo(converter.toEntity(campoDTO, Campo.class));
    objectConverterFactory.checkIn(converter);
}

@Override
public void eliminarCampo(CampoDTO campoDTO) throws Exception {
    ObjectConverter<CampoDTO, Campo> converter = objectConverterFactory.checkOut();
    campoService.eliminarCampo(converter.toEntity(campoDTO, Campo.class));
    objectConverterFactory.checkIn(converter);
}

@Override
public CampoDTO queryCampo(Long campoId) throws Exception {
    ObjectConverter<CampoDTO, Campo> converter = objectConverterFactory.checkOut();
    CampoDTO campoDTO = converter.toTransfer(campoService.queryCampo(campoId),
CampoDTO.class);
    objectConverterFactory.checkIn(converter);
    return campoDTO;
}
```

```
@Override
public List<Object> obtenerValoresCampo(String tabla, String campo, String query) throws
Exception {
    return campoService.obtenerValoresCampo(tabla, campo, query);
}
@Override
public String obtenerCampoPKTabla(String tabla, String campo) throws Exception {
    return campoService.obtenerCampoPKTabla(tabla, campo);
}
```

*Se crea interfaz **CampoService.java** que define métodos de CRUD en Service EJB para funcionalidad de pantalla*

```
@Local
public interface CampoService {

    //----- Campos -----/
    List<Campo> campoList(int first, int pageSize, String sortField, String sortOrder,
    Map<String, Object> filters) throws SystemException;

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

    Campo guardarCampo(Campo campo) throws SystemException;

    void editarCampo(Campo campo) throws SystemException;

    void eliminarCampo(Campo campo) throws SystemException;

    Campo queryCampo(Long campoId) throws SystemException;

    List<Object> obtenerValoresCampo(String tabla, String campo, String query) throws
    SystemException;

    String obtenerCampoPKTabla(String tabla, String campo) throws SystemException;

}
```

*Se crea **CampoServiceBean.java** donde implementan métodos CRUD en Service EJB para funcionalidad de pantalla*

```
@Stateless
public class CampoServiceBean extends StoreManagerImpl implements CampoService {

    //----- Valores Definiciones -----//
    @Override
    public List<Campo> campoList(int first, int pageSize, String sortField, String
    sortOrder, Map<String, Object> filters) throws SystemException {
        return this.getStoreManager(AdministracionStoreManager.class).campoList(first,
        pageSize, sortField, sortOrder, filters);
    }
}
```

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

@Override
public Campo guardarCampo(Campo campo) throws SystemException {
    return this.crear(campo, Campo.class);
}

@Override
public void editarCampo(Campo campo) throws SystemException {
    this.modificar(campo, Campo.class);
}

@Override
public void eliminarCampo(Campo campo) throws SystemException {
    this.eliminar(campo, Campo.class);
}

@Override
public Campo queryCampo(Long campoId) throws SystemException {
    return this.buscar(campoId, Campo.class);
}

@Override
public List<Object> obtenerValoresCampo(String tabla, String campo, String query) throws
SystemException {
    return
this.getStoreManager(AdministracionStoreManager.class).obtenerValoresCampo(tabla, campo,
query);
}

@Override
public String obtenerCampoPKTabla(String tabla, String campo) throws SystemException {
    return
this.getStoreManager(AdministracionStoreManager.class).obtenerCampoPKTabla(tabla, campo);
}
}
```

- Internacionalización en español e inglés

# ES	
sb.menu.gestion.administracion.campos.header	= Campos
sb.menu.gestion.administracion.campos.campo.breadcrumb	= Campos
sb.menu.gestion.administracion.campos.campo.header	= Campo
sb.menu.gestion.administracion.campo.volverTablas.button	= Volver al
listado de Tablas	
sb.menu.gestion.administracion.campos.posicion.header	= Posición
sb.menu.gestion.administracion.campos.descripcion.header	=
Descripción	
sb.menu.gestion.administracion.campos.obligatorio.header	=
Obligatorio	
sb.menu.gestion.administracion.campos.llavePrimaria.header	= Llave
Primaria	

sb.menu.gestion.administracion.campos.tipoDato.header	= Tipo Dato
sb.menu.gestion.administracion.campos.longitud.header	= Longitud
sb.menu.gestion.administracion.campos.decimales.header	= Decimales
sb.menu.gestion.administracion.campos.registraAuditoria.header	= Auditoría
sb.menu.gestion.administracion.campos.posicion.label	= Posición
sb.menu.gestion.administracion.campos.posicion.placeholder	= Ingrese la
Posición	
sb.menu.gestion.administracion.campos.posicion.required	= Debe
ingresar la Posición	
sb.menu.gestion.administracion.campos.descripcion.label	=
Descripción	
sb.menu.gestion.administracion.campos.campo.label	= Campo
sb.menu.gestion.administracion.campos.campo.placeholder	= Ingrese el
Campo	
sb.menu.gestion.administracion.campos.campo.required	= Debe
ingresar el Campo	
sb.menu.gestion.administracion.campos.descripcion.placeholder	= Ingrese la
Descripción	
sb.menu.gestion.administracion.campos.descripcion.required	= Debe
ingresar la Descripción	
sb.menu.gestion.administracion.campos.obligatorio.label	=
Obligatorio	
sb.menu.gestion.administracion.campos.llavePrimaria.label	= Llave
Primaria	
sb.menu.gestion.administracion.campos.registraAuditoria.label	= Auditoría
sb.menu.gestion.administracion.campos.tipoDato.label	= Tipo Dato
sb.menu.gestion.administracion.campos.tipoDato.placeholder	= Ingrese el
Tipo de Dato	
sb.menu.gestion.administracion.campos.longitud.label	= Longitud
sb.menu.gestion.administracion.campos.longitud.placeholder	= Ingrese la
Longitud	
sb.menu.gestion.administracion.campos.decimales.label	= Decimales
sb.menu.gestion.administracion.campos.decimales.placeholder	= Ingrese
los Decimales	
sb.menu.gestion.administracion.campos.tablaForanea.label	= Tabla
Foránea	
sb.menu.gestion.administracion.campos.tablaForanea.placeholder	= Ingrese la
Tabla Foránea	
sb.menu.gestion.administracion.campos.campoForaneo.label	= Campo
Foráneo	
sb.menu.gestion.administracion.campos.campoForaneo.placeholder	= Ingrese el
Campo Foráneo	
sb.menu.gestion.administracion.campos.chequeo.label	= Chequeo
sb.menu.gestion.administracion.campos.chequeo.placeholder	= Ingrese el
Chequeo	
sb.menu.gestion.administracion.campos.comentario.label	= Comentario
sb.menu.gestion.administracion.campos.comentario.placeholder	= Ingrese el
Comentario	
sb.menu.gestion.administracion.campos.cargarCampos.button	= Cargar
Campos	
com.sb.persistence.constraint.ORA-00001HERCULES.CAMPO_UK	= Ya existe
un registro con el nombre del campo ingresado para la tabla seleccionada	
# Campos	
sb.menu.gestion.administracion.campos.header	= Fields
sb.menu.gestion.administracion.campos.campo.breadcrumb	= Fields
sb.menu.gestion.administracion.campos.campo.header	= Field
sb.menu.gestion.administracion.campo.volverTablas.button	= Return to
the list of Tables	
sb.menu.gestion.administracion.campos.posicion.header	= Position

sb.menu.gestion.administracion.campos.descripcion.header	=
Description	
sb.menu.gestion.administracion.campos.obligatorio.header	= Required
sb.menu.gestion.administracion.campos.llavePrimaria.header	= Primary
Key	
sb.menu.gestion.administracion.campos.tipoDato.header	= Data Type
sb.menu.gestion.administracion.campos.longitud.header	= Length
sb.menu.gestion.administracion.campos.decimales.header	= Decimals
sb.menu.gestion.administracion.campos.registraAuditoria.header	= Audit
sb.menu.gestion.administracion.campos.posicion.label	= Position
sb.menu.gestion.administracion.campos.posicion.placeholder	= Enter
Position	
sb.menu.gestion.administracion.campos.posicion.required	= Position
is required	
sb.menu.gestion.administracion.campos.descripcion.label	=
Description	
sb.menu.gestion.administracion.campos.descripcion.placeholder	= Enter
Description	
sb.menu.gestion.administracion.campos.descripcion.required	=
Description is required	
sb.menu.gestion.administracion.campos.campo.label	= Field
sb.menu.gestion.administracion.campos.campo.placeholder	= Enter
Field	
sb.menu.gestion.administracion.campos.campo.required	= Field is
required	
sb.menu.gestion.administracion.campos.obligatorio.label	= Required
sb.menu.gestion.administracion.campos.llavePrimaria.label	= Primary
Key	
sb.menu.gestion.administracion.campos.registraAuditoria.label	= Audit
sb.menu.gestion.administracion.campos.tipoDato.label	= Data Type
sb.menu.gestion.administracion.campos.tipoDato.placeholder	= Enter Data
Type	
sb.menu.gestion.administracion.campos.longitud.label	= Length
sb.menu.gestion.administracion.campos.longitud.placeholder	= Enter
Length	
sb.menu.gestion.administracion.campos.decimales.label	= Decimals
sb.menu.gestion.administracion.campos.decimales.placeholder	= Enter
Decimals	
sb.menu.gestion.administracion.campos.tablaForanea.label	= Foreign
Table	
sb.menu.gestion.administracion.campos.tablaForanea.placeholder	= Enter
Foreign Table	
sb.menu.gestion.administracion.campos.campoForaneo.label	= Foreign
Field	
sb.menu.gestion.administracion.campos.campoForaneo.placeholder	= Enter
Foreign Field	
sb.menu.gestion.administracion.campos.chequeo.label	= Check
sb.menu.gestion.administracion.campos.chequeo.placeholder	= Enter
Check	
sb.menu.gestion.administracion.campos.comentario.label	= Comment
sb.menu.gestion.administracion.campos.comentario.placeholder	= Enter
Comment	
sb.menu.gestion.administracion.campos.cargarCampos.button	= Load
Fields	
com.sb.persistence.constraint.ORA-00001HERCULES.CAMPO_UK	= A record
with the entered field name already exists for the selected table	

Pestaña de Tabla de Visualización de la información en la pantalla de Campos

La pestaña "**Campos**" en la interfaz de usuario permite a los administradores visualizar y gestionar los campos de una determinada tabla. Esta pestaña presenta una tabla de datos (**p:dataTable**) con soporte para filtrado, ordenamiento, paginación y carga diferida (lazy loading). Además, se incluyen opciones de interacción como la selección de filas y la expansión de filas para mostrar información adicional.

```
<!--PESTAÑA TABLE -->
<p:tab title="#{msg['sb.menu.gestion.administracion.campos.header']}">

    <!-- Boton - Cargar Tablas -->
    <div class="row">
        <div class="col-lg-12">
            <p:commandLink id="cargarCampos"

title="#{msg['sb.menu.gestion.administracion.campos.cargarCampos.button']}"
styleClass="btn btn btn-primary pull-right
tooltips"
update="#{campoViewController.modulo}Form"

actionListener="#{campoConvController.cargarCampos()}"
>
                <i class="fa fa-refresh">
#{msg['sb.menu.gestion.administracion.campos.cargarCampos.button']} </i>
            </p:commandLink>
        </div>
    </div>

    <!-- TABLA -->
    <div class="row">
        <div class="col-lg-12" style="overflow-x: scroll;">
            <p:dataTable id="#{campoViewController.modulo}Table"
var="campo"
value="#{campoViewController.modelo}"
rowKey="#{campo.campoId}"
selection="#{campoConvController.selected}"
selectionMode="single"
paginator="true"
rows="10"
lazy="true"
reflow="true"
sortField="posicion"
sortOrder="ASCENDING"
currentPageReportTemplate="Total de registros:
{totalRecords}"
paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
rowsPerPageTemplate="10,20,30"
paginatorPosition="bottom"

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

widgetVar="items#{campoViewController.modulo}">

                <f:facet name="header">
```

```

<ui:decorate
template="../../../template/_tableHeader.xhtml">
    <ui:define name="comandos">
        <ui:param name="sessionController"
value="#{campoViewController}" />
        <ui:param name="conversationController"
value="#{campoConvController}" />
        <ui:param name="btnBusqueda" value="N"
/>
        <ui:param name="buscarVisible"
value="S" />
    </ui:define>
</ui:decorate>
</f:facet>

<p:ajax event="rowSelect"
listener="#{campoConvController.onRowSelect}"
update="#{campoViewController.modulo}Table"
process="@this"/>
<p:ajax event="rowUnselect"
listener="#{campoConvController.onRowUnselect}"
update="#{campoViewController.modulo}Table"
process="@this"/>

<p:column style="text-align: center; padding: 5px;"
width="5%">
    <p:rowToggler/>
</p:column>

<p:column
headerText="#{msg['sb.menu.gestion.administracion.campos.posicion.header']}"
filterBy="#{campo.posicion}"
sortBy="#{campo.posicion}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="10%"
>
    <h:outputText value="#{campo.posicion}"/>
</p:column>

<p:column
headerText="#{msg['sb.menu.gestion.administracion.campos.campo.header']}"
filterBy="#{campo.campo}"
sortBy="#{campo.campo}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="25%"
>
    <h:outputText value="#{campo.campo}"/>
</p:column>

<p:column
headerText="#{msg['sb.menu.gestion.administracion.campos.descripcion.header']}"
filterBy="#{campo.descripcionCampo}"
sortBy="#{campo.descripcionCampo}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="15%"

```

```

        <h:outputText value="#{campo.descripcionCampo}"/>
    </p:column>

    <p:column
headerText="#{msg['sb.menu.gestion.administracion.campos.obligatorio.header']}"
        filterBy="#{campo.obligatorio}"
        sortBy="#{campo.obligatorio}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        style="text-align: center"
        width="15%"
    >
        <h:outputText value="#{campo.obligatorio eq 'S' ? 'Sí'
: 'No'}"/>
    </p:column>

    <p:column
headerText="#{msg['sb.menu.gestion.administracion.campos.llavePrimaria.header']}"
        filterBy="#{campo.llavePrimaria}"
        sortBy="#{campo.llavePrimaria}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        style="text-align: center"
        width="15%"
    >
        <h:outputText value="#{campo.llavePrimaria eq 'S' ?
'Sí' : 'No'}"/>
    </p:column>

    <p:column
headerText="#{msg['sb.menu.gestion.administracion.campos.tipoDato.header']}"
        filterBy="#{campo.tipoDato}"
        sortBy="#{campo.tipoDato}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        style="text-align: center"
        width="15%"
    >
        <h:outputText value="#{campo.tipoDato}"/>
    </p:column>

    <p:column
headerText="#{msg['sb.menu.gestion.administracion.campos.longitud.header']}"
        filterBy="#{campo.longitud}"
        sortBy="#{campo.longitud}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        style="text-align: center"
        width="15%"
    >
        <h:outputText value="#{campo.longitud}"/>
    </p:column>

    <p:column
headerText="#{msg['sb.menu.gestion.administracion.campos.decimales.header']}"
        filterBy="#{campo.decimales}"
        sortBy="#{campo.decimales}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        style="text-align: center"
    >

```

```

width="15%"
>
<h:outputText value="#{campo.decimales}"/>
</p:column>

<p:column
headerText="#{msg['sb.menu.gestion.administracion.campos.registraAuditoria.header']}"
filterBy="#{campo.registraAuditoria}"
sortBy="#{campo.registraAuditoria}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
width="15%"
>
<h:outputText value="#{campo.registraAuditoria eq 'S' ?
'Sí' : 'No'}"/>
</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"

>

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

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

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

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

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

</div>
</p:panelGrid>
</div>
</div>

```

```

                <p:spacer/>
            </div>
        </p:rowExpansion>
    </p:dataTable>
</div>
</div>
</p:tab>

```

Componentes Principales

1. Botón "Cargar Tablas"

Este botón permite actualizar la lista de campos cargados en la tabla. Se implementa con un `p:commandLink` que ejecuta la acción `cargarCampos()` del `campoConvController`, recargando el formulario de la pestaña.

- **ID:** `cargarCampos`
- **Acción:** `campoConvController.cargarCampos()`
- **Actualiza:** `#{campoViewController.modulo}Form`
- **Estilo:** Botón tipo `btn-primary` con icono `fa-refresh`

2. Tabla de Datos (`p:dataTable`)

La tabla muestra la lista de campos y permite interacciones como filtrado, ordenamiento y selección de filas.

- **ID:** `#{campoViewController.modulo}Table`
- **Fuente de Datos:** `campoViewController.model`
- **Carga Diferida:** `lazy="true"`
- **Paginación:** `rows="10"`
- **Selección:** `single`, almacenada en `campoConvController.selected`
- **Ordenación Predeterminada:** Campo `posicion` en orden `ASCENDING`
- **Mensajes Personalizados:**
 - Si no hay registros: `emptyMessage="#{msg['sb.datatable.registro.empty']}"`

- Plantilla de paginación: {CurrentPageReport} {FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink} {RowsPerPageDropdown}

2.1 Columnas de la Tabla

Cada columna muestra un atributo del objeto campo con filtrado (filterBy), ordenamiento (sortBy) y formatos personalizados según el tipo de dato.

Campo	Descripción	Filtrable	Ordenable	Formato
posicion	Posición del campo en la estructura de datos	✓	✓	Número
campo	Nombre del campo	✓	✓	Texto
descripcionCampo	Descripción del campo	✓	✓	Texto
obligatorio	Indica si el campo es obligatorio (Sí/No)	✓	✓	Conversión 'S' → 'Sí', 'N' → 'No'
llavePrimaria	Indica si el campo es clave primaria (Sí/No)	✓	✓	Conversión 'S' → 'Sí', 'N' → 'No'
tipoDato	Tipo de dato del campo (Ej. VARCHAR, INT)	✓	✓	Texto
longitud	Longitud del campo (si aplica)	✓	✓	Número
decimales	Número de decimales permitidos (si aplica)	✓	✓	Número
registraAuditoria	Indica si el campo tiene auditoría (Sí/No)	✓	✓	Conversión 'S' → 'Sí', 'N' → 'No'

2.2 Expansión de Filas (p:rowExpansion)

Cuando se expande una fila, se muestra información adicional del campo en un p:panelGrid con los siguientes datos:

- **ID del campo**
- **Usuario creador**
- **Fecha de creación** (dd/MM/yyyy)
- **Usuario modificador**
- **Fecha de modificación** (dd/MM/yyyy)

3. Eventos de Selección de Fila

Se implementan dos eventos p:ajax en la tabla:

- **rowSelect:** Al seleccionar una fila, se ejecuta campoConvController.onRowSelect, actualizando la tabla.
- **rowUnselect:** Al deseleccionar una fila, se ejecuta campoConvController.onRowUnselect, actualizando la tabla.

Consideraciones Técnicas

- Se utiliza **carga diferida** (lazy="true") para optimizar el rendimiento.
- El **scroll horizontal** está habilitado (overflow-x: scroll;) para evitar desbordes cuando hay muchas columnas.
- Se usa widgetVar="items#{campoViewController.modulo}" para acceder a la tabla mediante JavaScript si es necesario.

Recomendaciones para Mejoras

1. **Optimizar la carga de datos:** Implementar LazyDataModel para mejorar la eficiencia en consultas grandes.
2. **Habilitar filtros visibles:** Actualmente, los filtros están ocultos (filterStyle="display:none;"). Se pueden mostrar para mejorar la usabilidad.
3. **Agregar acciones en cada fila:** Botones para editar o eliminar registros directamente desde la tabla.
4. **Mejorar el diseño responsivo:** Usar p:scrollableDataTable o PrimeFlex para optimizar la visualización en dispositivos móviles.

Formulario de captura de datos para Registro de Campos

Este formulario permite la creación y edición de campos en la configuración del sistema. Incluye validaciones y controles para capturar información como la posición del campo, descripción, tipo de dato, longitud, llaves primarias y relaciones con otras tablas.

```
<!--PESTAÑA FORMULARIO GUARDAR/EDITAR -->
<p:tab title="#{msg['sb.component.tabview.guardarEditar']}">

    <br />

    <p:panel id="formularioEditarCrear">
        <div class="ibox-content">

            <br/>

            <!-- Fila Posicion, Campo -->
            <div class="row">
                <div class="col-lg-6">
                    <div class="form-group">
                        <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.posicion.label']}"
control-label"
right"
                        styleClass="col-lg-4
                        style="text-align:
                        for="posicion"/>
                    <div class="col-lg-6">
                        <p:inputNumber id="posicion"
value="#{campoViewController.selected.posicion}"
required="true"
decimalPlaces="0"
maxValue="99999"
maxLength="5"
minValue="1"
placeholder="#{msg['sb.menu.gestion.administracion.campos.posicion.placeholder']}"
requiredMessage="#{msg['sb.menu.gestion.administracion.campos.posicion.required']}"
                        >
                    </p:inputNumber>
                </div>
            </div>
            <div class="col-lg-6">
                <div class="form-group">
                    <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.campo.label']}"
control-label"
                    styleClass="col-lg-2
```

```

right"
style="text-align:
for="campo"/>
<div class="col-lg-8">
  <p:inputText id="campo"
value="#{campoViewController.selected.campo}"
required="true"
maxlength="100"
placeholder="#{msg['sb.menu.gestion.administracion.campos.campo.placeholder']}"
requiredMessage="#{msg['sb.menu.gestion.administracion.campos.campo.required']}"
styleClass="form-control">
    </p:inputText>
  </div>
</div>
</div>
<br/>
<br/>
<!-- Fila Descripción -->
<div class="row">
  <div class="col-lg-12">
    <div class="form-group">
      <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.descripcion.label']}"
styleClass="col-lg-2
control-label"
style="text-align:
right"
for="descripcionCampo"/>
    <div class="col-lg-9">
      <p:inputTextarea id="descripcionCampo"
value="#{campoViewController.selected.descripcionCampo}"
required="true"
maxlength="200"
rows="2"
placeholder="#{msg['sb.menu.gestion.administracion.campos.descripcion.placeholder']}"
requiredMessage="#{msg['sb.menu.gestion.administracion.campos.descripcion.required']}"
styleClass="form-control">
    </p:inputTextarea>
  </div>
</div>
</div>
<br/>

```

```

<br/>

<!-- Fila obligatorio, llavePrimaria, Auditoria -->
<div class="row">
  <div class="col-lg-4">
    <div class="form-group">
      <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.obligatorio.label']}"
styleClass="col-lg-6
control-label"
style="text-align:
right"
for="obligatorio"/>
      <div class="col-lg-6">
        <h:selectBooleanCheckbox
id="obligatorio"

        value="#{campoViewController.selected.obligatorioBool}"
        required="false"
        styleClass="js-switch"
      >
      </h:selectBooleanCheckbox>
    </div>
  </div>
  <div class="col-lg-3">
    <div class="form-group">
      <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.llavePrimaria.label']}"
styleClass="col-lg-9
control-label"
style="text-align:
right"
for="llavePrimaria"/>
      <div class="col-lg-3">
        <h:selectBooleanCheckbox
id="llavePrimaria"

        value="#{campoViewController.selected.llavePrimariaBool}"
        required="false"
        styleClass="js-switch"
      >
      </h:selectBooleanCheckbox>
    </div>
  </div>
  <div class="col-lg-4">
    <div class="form-group">
      <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.registraAuditoria.label']}"
styleClass="col-lg-9
control-label"
style="text-align:
right"

```

```

for="registraAuditoria"/>
                                <div class="col-lg-3">
                                    <h:selectBooleanCheckbox
id="registraAuditoria"
                                value="#{campoViewController.selected.registraAuditoriaBool}"
                                required="false"
                                styleClass="js-switch"
                                >
                                    </h:selectBooleanCheckbox>
                                </div>
                            </div>
                        </div>
                    <br/>
                    <br/>
                    <!-- Fila Tipo Dato, Longitud, Decimales -->
                    <div class="row">
                        <div class="col-lg-4">
                            <div class="form-group">
                                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.tipoDato.label']}"
                                styleClass="col-lg-6
control-label"
                                style="text-align:
right"
                                for="tipoDato"/>
                                <div class="col-lg-6">
                                    <p:inputText id="tipoDato"
value="#{campoViewController.selected.tipoDato}"
                                    required="false"
                                    maxLength="100"
                                    placeholder="#{msg['sb.menu.gestion.administracion.campos.tipoDato.placeholder']}"
                                    styleClass="form-control">
                                        </p:inputText>
                                    </div>
                                </div>
                                <div class="col-lg-4">
                                    <div class="form-group">
                                        <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.longitud.label']}"
                                        styleClass="col-lg-5
control-label"
                                        style="text-align:
right"
                                        for="longitud"/>
                                        <div class="col-lg-7">
                                            <p:inputNumber id="longitud"

```

```

value="#{campoViewController.selected.longitud}"
required="false"
decimalPlaces="0"
maxLength="5"
maxValue="99999"
minValue="1"

placeholder="#{msg['sb.menu.gestion.administracion.campos.longitud.placeholder']}"
    </p:inputNumber>
    </div>
    </div>
    </div>
    <div class="col-lg-4">
        <div class="form-group">
            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.decimales.label']}"
styleClass="col-lg-5
control-label"
style="text-align:
right"
for="decimales"/>
            <div class="col-lg-7">
                <p:inputNumber id="decimales"

value="#{campoViewController.selected.decimales}"
required="false"
decimalPlaces="0"
maxLength="2"
maxValue="99"
minValue="1"

placeholder="#{msg['sb.menu.gestion.administracion.campos.decimales.placeholder']}"
    </p:inputNumber>
    </div>
    </div>
    </div>
    <br/>
    <br/>

    <!-- Fila tablaForenea, CampoForaneo -->
    <div class="row">
        <div class="col-lg-6">
            <div class="form-group">
                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.tablaForenea.label']}"
styleClass="col-lg-4
control-label"

```

```

right"
style="text-align:
for="tablaForanea"/>
<div class="col-lg-7">
  <p:inputText id="tablaForanea"
value="#{campoViewController.selected.tablaFk}"
required="false"
maxlength="30"
placeholder="#{msg['sb.menu.gestion.administracion.campos.tablaForanea.placeholder']}"
styleClass="form-control">
  </p:inputText>
</div>
</div>
<div class="col-lg-6">
  <div class="form-group">
    <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.campoForaneo.label']}"
styleClass="col-lg-3
control-label"
style="text-align:
right"
for="campoForaneo"/>
    <div class="col-lg-7">
      <p:inputText id="campoForaneo"
value="#{campoViewController.selected.campoFk}"
required="false"
maxlength="30"
placeholder="#{msg['sb.menu.gestion.administracion.campos.campoForaneo.placeholder']}"
styleClass="form-control">
      </p:inputText>
    </div>
  </div>
</div>
<br/>
<br/>
<!-- Fila chequeo -->
<div class="row">
  <div class="col-lg-12">
    <div class="form-group">
      <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.chequeo.label']}"
styleClass="col-lg-2
control-label"
style="text-align:
right"
for="chequeo"/>
    <div class="col-lg-9">
      <p:inputText id="chequeo"

```

```

value="#{campoViewController.selected.chequeo}"
required="false"
maxlength="4000"
placeholder="#{msg['sb.menu.gestion.administracion.campos.chequeo.placeholder']}"
styleClass="form-control">
    </p:inputText>
    </div>
    </div>
    </div>
    </div>
    <br/>
    <br/>
    <!-- Fila comentario -->
    <div class="row">
        <div class="col-lg-12">
            <div class="form-group">
                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.campos.comentario.label']}"
control-label"
right"
                styleClass="col-lg-2"
                style="text-align:
                for="comentario"/>
                <div class="col-lg-9">
                    <p:inputText id="comentario"

value="#{campoViewController.selected.comentario}"
required="false"
maxlength="4000"
placeholder="#{msg['sb.menu.gestion.administracion.campos.comentario.placeholder']}"
styleClass="form-control">
    </p:inputText>
    </div>
    </div>
    </div>
    <br/>
    <div class="hr-line-dashed"></div>
    <ui:decorate template="../../template/_formButtons.xhtml">
        <ui:define name="comandos">
            <ui:param name="sessionController"
value="#{campoViewController}" />
            <ui:param name="conversationController"
value="#{campoConvController}" />
        </ui:define>
    </ui:decorate>
    </div>
</p:panel>

```

</p:tab>

Estructura del Formulario

El formulario se encuentra dentro de una pestaña (p:tab) y está estructurado en varias secciones organizadas en filas (div.row) para mejorar la legibilidad y la alineación de los campos.

1. Identificación del Campo

- **Posición:** Número entero obligatorio que indica el orden del campo.
 - **Componente:** <p:inputNumber>
 - **Validaciones:**
 - Debe ser un número entero sin decimales.
 - Máximo de 5 dígitos (maxlength="5").
 - Rango permitido: 1 a 99999.
 - Mensaje de error si está vacío.
- **Nombre del Campo:** Cadena de texto obligatoria con un máximo de 100 caracteres.
 - **Componente:** <p:inputText>
 - **Validaciones:**
 - No debe estar vacío.
 - Máximo de 100 caracteres.

2. Descripción del Campo

- **Descripción detallada:** Texto de hasta 200 caracteres.
 - **Componente:** <p:inputTextarea>
 - **Validaciones:**
 - Máximo de 200 caracteres.
 - Campo obligatorio.

3. Configuración de Propiedades

- **Obligatorio:** Define si el campo debe ser obligatorio.
 - **Componente:** <h:selectBooleanCheckbox>
 - **Valor booleano (true/false).**
- **Llave Primaria:** Indica si el campo es clave primaria.
 - **Componente:** <h:selectBooleanCheckbox>
 - **Valor booleano.**
- **Registra Auditoría:** Indica si los cambios en el campo deben ser registrados.
 - **Componente:** <h:selectBooleanCheckbox>
 - **Valor booleano.**

4. Tipo de Dato y Restricciones

- **Tipo de Dato:** Indica el tipo de dato (ej. VARCHAR, INT, DATE).
 - **Componente:** <p:inputText>
 - **Máximo 100 caracteres.**
- **Longitud:** Especifica la cantidad máxima de caracteres o dígitos.
 - **Componente:** <p:inputNumber>
 - **Valores permitidos:** 1 a 99999.
- **Decimales:** Cantidad de decimales para los valores numéricos.
 - **Componente:** <p:inputNumber>
 - **Valores permitidos:** 1 a 99.

5. Relaciones con Otras Tablas

- **Tabla Foránea:** Nombre de la tabla con la que se relaciona el campo.
 - **Componente:** <p:inputText>

- **Máximo 30 caracteres.**
- **Campo Foráneo:** Nombre del campo en la tabla foránea.
 - **Componente:** <p:inputText>
 - **Máximo 30 caracteres.**

Comportamiento y Validaciones

El formulario incluye validaciones de longitud, obligatoriedad y tipos de datos. Los campos numéricos usan <p:inputNumber> para restringir los valores, y los booleanos usan <h:selectBooleanCheckbox>.

Mensajes de Error

Cada campo obligatorio tiene un mensaje de error (requiredMessage) definido en el archivo de mensajes.

Consideraciones Técnicas

- Se recomienda utilizar un ManagedBean (campoViewController) con la anotación @ViewScoped para mantener el estado del formulario en la vista.
- Los mensajes de validación deben estar configurados en messages.properties para facilitar la internacionalización.
- Se pueden agregar validaciones adicionales en el backend para garantizar la integridad de los datos.

Visualización de la pantalla

MINICIENCIAS
Gestión Minciencias
Usuario: SIGP_WEB

Campos
Inicio / Administración / Sistema / Gestión Tablas / Campos

Tabla: ADM_CIUDES
Descripción: Almacena la División Política

CAMPOS GUARDAR/EDITAR

Buscar Todos: Filtrar por palabra clave

	Posición	Campo	Descripción	Obligatorio	Llave Primaria	Tipo Dato	Longitud	Decimales	Auditoría
1	1	CIUDAD_ID	Ciudad Id	Sí	Sí	NUMBER	15	0	Sí
2	2	NOMBRE_CIUDES	Nombre Ciudad	Sí	No	VARCHAR2	100	0	Sí
3	3	PAIS_ID	País Id	Sí	No	NUMBER	15	0	Sí
4	4	TIPO_LUGAR_ID	Tipo Lugar Id	No	No	NUMBER	15	0	Sí
5	5	INDICATIVO	Indicativo	No	No	NUMBER	5	0	Sí
6	6	REGION_ID	Región Id	No	No	NUMBER	15	0	Sí
7	7	ATRIBUTO1	Atributo1	No	No	VARCHAR2	100	0	Sí
8	8	ATRIBUTO2	Atributo2	No	No	VARCHAR2	100	0	Sí
9	9	ATRIBUTO3	Atributo3	No	No	VARCHAR2	100	0	Sí
10	10	ATRIBUTO4	Atributo4	No	No	VARCHAR2	100	0	Sí

TOTAL DE REGISTROS: 15

Campos GUARDAR/EDITAR

Tabla: ADM_CIUDES
Descripción: Almacena la División Política

Posición * Ingrese la Posición

Campo * Ingrese el Campo

Descripción * Ingrese la Descripción

Obligatorio ☐ Llave Primaria ☐ Auditoría ☐

Tipo Dato Ingrese el Tipo de Dato Longitud Ingrese la Longitud Decimales Ingrese los Decimales

Tabla Foránea Ingrese la Tabla Foránea Campo Foráneo Ingrese el Campo Foráneo

Chequeo Ingrese el Chequeo

Comentario Ingrese el Comentario

Guardar

El cambio fue desplegado en pruebas el 13/03/2025

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: 17/Febrero/2025 – 14/03/2025

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	6
Generación de Reportes	14
Solución de Errores	4
Actualización de Datos	1
Requerimientos	2
Total	27

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
151729	Parametrización: <i>CONVOCATORIA FORMACIÓN EN MAESTRÍAS PACÍFICO NARIÑENSE</i>	Pendiente Usuario	Pruebas	[Ver detalle]
152706	Parametrización: <i>963-2025 Conv "Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 2025"</i>	Solucionado	Producción	[Ver detalle]
153242	Parametrización: <i>953-2024 CONVOCATORIA PARA COMUNIDADES NEGRAS DE BOGOTÁ, CAQUETÁ, CÓRDOBA, MAGDALENA Y CHOCÓ</i>	Solucionado	Producción	[Ver detalle]
153303	Parametrización: <i>CONVOCATORIA PARA PROYECTOS DE I+D+i EN RECOBRO MEJORADO Y FORTALECIMIENTO EN IES – 2025 (Modalidad 1 y Modalidad 2)</i>	Pendiente Usuario	Pruebas	[Ver detalle]
153373	Parametrización: <i>CONVOCATORIA PARA EL FORTALECIMIENTO Y CONSOLIDACIÓN DE LOS CENTROS DE INNOVACIÓN DE LA RED ECONOVA</i>	Pendiente Usuario	Pruebas	[Ver detalle]
154026	Parametrización: <i>NUEVA CONVOCATORIA EN OCÉANOS, CLIMA Y RECURSOS MARINOS COSTEROS</i>	Pendiente Usuario	Pruebas	[Ver detalle]

Solicitud 151729

Asunto	Tiempo total de actividad
Referencia en SIGP	00:23:49

Descripción
<p>Solicito amablemente su colaboración para solicitar la estimación de tiempo y parametrización de los Términos de Referencia en SIGP, junto con sus anexos, esto haciendo referencia al punto 3 del *"PROCEDIMIENTO APERTURA Y CIERRE DE CONVOCATORIAS - CÓDIGO M801PR01"*</p> <p>Adjunto el borrador de los términos de referencia, así como los correspondientes anexos.</p> <p>Quedo atento.</p> <p>Cordialmente,</p> <p>--</p> <p>*Deiver Andrey Acosta Linares*</p> <p>Contratista</p> <p>Dirección de Ciencia</p> <p>*daacosta@minciencias.gov.co <daacosta@minciencias.gov.co>*</p> <p>Teléfono: 601 6258480 Ext.</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
21/01/2025 18:11:11	14/03/2025 11:33:52		

Acción Realizada 151729

1. Revisión de los términos de referencia

- Se analizan los términos de referencia de la **Convocatoria de Maestrías en Nariño** 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 GESTIÓN DE RECURSOS PARA LA (CTeI)
 - Financiaciones
 - Recuperación Contingente
 - Menús
 - PRY - Tipo Personal
 - PRY - Palabras Clave

- PRY - Función General Proyectos
 - PRY - Validaciones
 - PRY - Generalidades Personal
 - PRY - Roles Entidad
 - PRY - Proyecto
 - PRY - Cambiar Clave
 - PRY - Requisitos
 - PRY – Presupuesto
 - PRY - Rubros
 - PRY - Información General Proyecto
 - PRY - Cronograma
 - PRY - Gran Área OCDE, Área OCDE y Disciplina OCDE
 - PRY - Descripciones
 - PRY - Personal
 - PRY – Documentos
- Rubros
- APOYO AL PROYECTO DE INVESTIGACIÓN
 - SOSTENIMIENTO
 - COSTO DE INSCRIPCIÓN
 - MATRÍCULA
- Se configuran Montos Roles Personal por cada Rubro
- Descripciones
- 1 RESUMEN EJECUTIVO
 - 2 IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA
 - 3 JUSTIFICACIÓN
 - 4 ARTICULACIÓN DE LA PROPUESTA CON LAS MISIONES DE LA POLÍTICA DE INVESTIGACIÓN E INNOVACIÓN DEL MINISTERIO
 - 5 MARCO TEÓRICO
 - 6 POBLACIÓN A IMPACTAR
 - 7 IMPACTO EN LA COMUNIDAD EDUCATIVA Y SU ENTORNO
 - 8 OBJETIVO GENERAL
 - 9 OBJETIVOS ESPECÍFICOS
 - 10 METODOLOGÍA

- 11 ACTIVIDADES PARA LA ARTICULACIÓN DE ACTORES DE CIENCIA, TECNOLOGÍA E INNOVACIÓN PARA EL DESARROLLO DE LA PROPUESTA EN LA COMUNIDAD EDUCATIVA Y SU ENTORNO
- 12 RESULTADOS ESPERADOS Y PRODUCTOS I+D+i
- 13 BIBLIOGRAFÍA
- Campos (Se ocultan)
 - CADENA PRODUCTIVA
 - CIUDAD DE FORMACION
 - CLASIFICACIÓN
 - CODIGO CIU
 - DURACIÓN PROYECTO HORAS
 - EJECUCIÓN CRONOGRAMA
 - ESTADO EMPRESA
 - FOCO ESTRATÉGICO - RETOS - TEMATICAS ORIENTADORAS
 - LINEA TEMATICA
 - LUGAR DE EJECUCIÓN
 - MÚLTIPLES FOCOS Y MÚLTIPLES LÍNEAS TEMÁTICAS
 - NIVEL MADUREZ
 - NOMBRE CLÚSTER
 - PERMISOS
 - PROGRAMA NACIONAL DE CTEL PRINCIPAL
 - PROPÓSITO GENERAL
 - SECTOR
 - SUBLINEA TEMATICA PRINCIPAL
 - SUBLINEAS TEMATICAS SECUNDARIAS
 - TEMÁTICAS ORIENTADORAS POR FOCO ESTRATÉGICO
 - TIPO DE FINANCIACIÓN
 - TIPO PERSONA
 - VALOR INSCRIPCION
 - VALOR SOLICITADO
- Financiadoras
 - MINCIENCIAS
- Requisitos
 - 7.1.4 Ser ciudadano(a) colombiano(a).

- 7.1.5 Carta respaldo al docente y/o directivo docente del establecimiento educativo (Anexo No. 1).
 - 7.1.6 Acreditación del establecimiento educativo.
 - 7.1.7 Propuesta de investigación a desarrollar en el marco de la maestría (Anexo No. 2).
 - 7.1.8 Estado Académico.
 - 7.1.9 Perfil del candidato que asumirá el desarrollo de la Propuesta (Anexo No. 3).
 - 7.2.1 Promedio académico.
 - 7.2.2 Enfoque diferencial y/o interseccional (cuando aplique).
- Roles Entidades
 - PROPONENTE
- Roles del Personal
 - ESTUDIANTE DE MAESTRÍA EN DIRECCIÓN EMPRESARIAL
 - ESTUDIANTE DE MAESTRÍA EN EDUCACIÓN
 - ESTUDIANTE DE MAESTRÍA EN EDUCACIÓN AMBIENTAL Y DESARROLLO SOSTENIBLE
 - ESTUDIANTE DE MAESTRÍA EN EDUCACIÓN MODALIDAD VIRTUAL
- Focos de la Convocatoria
 - Valor simbólico, identidad cultural y generación de contenidos
 - Educación para las industrias creativas y culturales
 - Circulación de contenidos, articulación con industria, integración de cadenas de valor
 - Regulación, financiación, información
 - Infraestructura y tecnología
- 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.

Mostrar la actividad para Solicitud 151729		
Número de Solicitud	Resumen de Solicitud	
151729	Referencia en SIGP	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
20/02/2025 10:04:41	20/02/2025 10:04:41	00:07:02
Descripción del usuario		
Notificación manual		
Buenos días.		
Se finalizó la configuración de la convocatoria de Maestrías Nariño 2025. 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: INV7422025		
Clave: TNbv7SME		
URL: https://plataformasigp.minciencias.gov.co:7003/FormularioProyectos/faces/proyecto/invitacionEspecial.xhtml		
* Seleccionar la convocatoria "YYYY-2025 CONVOCATORIA PARA LA FORMACIÓN EN MAESTRÍAS DEPARTAMENTO NARIÑO".		
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 70D # 51 - 57		

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.

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.

Mostrar la actividad para Solicitud 151729

Número de Solicitud	Resumen de Solicitud	
151729	Referencia en SIGP	
Tipo de actividad	Analista	¿Interno?
Registrar comentario	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
13/03/2025 15:18:04	13/03/2025 15:18:04	00:02:07

Descripción del usuario

Buenas tardes.

Los ajustes solicitados fueron realizados y desplegados para su verificación. Quedamos atentos a cualquier ajuste pendiente y/o aprobación de las pruebas de parametrización exitosas en la convocatorias.

Quedo atento

--

Cordialmente,

Ing. Leonardo Suárez Baldovino
Software Builders Ltda.

✉ lasb@softwarebuilders.co | 🌐 www.softwarebuilders.co

📍 Carrera 70D # 51 - 57

8. Validación final del usuario

- Se espera la aprobación del usuario funcional para confirmar que la convocatoria está parametrizada correctamente.

Solicitud 152706

Asunto		Tiempo total de actividad	
Gestión de Proyectos - SIGP		00:46:39	
Descripción			
De manera atenta se solicita la parametrización en el Sistema Integrado de Gestión de Proyectos - SIGP - de la convocatoria "Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 2025"., para lo cual se adjuntan los términos de referencia y anexos.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

10/02/2025 12:14:43	20/02/2025 14:56:14	19/02/2025 15:56:13	20/02/2025 14:56:14
------------------------	------------------------	------------------------	------------------------

Acción Realizada 152706

1. Revisión de los términos de referencia

- Se analizan los términos de referencia de la **Convocatoria Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 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
 - DGR EQUIPOS MECANISMOS
 - Financiaciones
 - Recuperación Contingente
 - Menús
 - PRY - Tipo Personal
 - PRY - Palabras Clave
 - PRY - Función General Proyectos
 - PRY – Validaciones
 - PRY - Productos - Desarrollo tecnológico e innovación

- PRY - Productos - Apropiación social del conocimiento
- PRY - Generalidades Personal
- PRY - Productos - Generación de nuevo conocimiento
- PRY - Roles Entidad
- PRY - Proyecto
- PRY - Cambiar Clave
- PRY - Productos - Formación de recurso humano
- PRY - Documentos
- PRY - Requisitos
- PRY - Presupuesto
- PRY - Rubros
- PRY - Información General Proyecto
- PRY - Cronograma
- PRY - Gran Área OCDE y Área OCDE
- PRY - Información Adicional
- PRY - Descripciones
- PRY - Productos
- PRY - Personal
- PRY - Análisis de Riesgos
- PRY - Enfoque Diferencial
- Rubros
 - APOYO AL SOSTENIMIENTO MENSUAL DE LA DOCTORA
 - BIBLIOGRAFÍA
 - CAPACITACIÓN
 - EQUIPOS
 - EXPERTOS
 - GASTOS DE OPERACIÓN
 - MATERIALES E INSUMOS
 - MOVILIDAD INVESTIGADORAS
 - PARTICIPACIÓN EN EVENTOS ACADÉMICOS
 - PUBLICACIONES
 - SALIDAS DE CAMPO
 - SERVICIOS TÉCNICOS
 - VINCULACIÓN DE LA JOVEN INVESTIGADORA (ESTUDIANTE)
 - VINCULACIÓN DE LA JOVEN INVESTIGADORA (GRADUADA)

Se configuran Montos Roles Personal por cada Rubro

- Descripciones
 - 1 RESUMEN EJECUTIVO
 - 2 ALINEACIÓN CON LA POLÍTICA PÚBLICA Y/O LÍNEAS DE INVESTIGACIÓN DE LA ENTIDAD DEL SISTEMA NACIONAL DE CIENCIA, TECNOLOGÍA E INNOVACIÓN QUE INTEGRA LA ALIANZA
 - 3 IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA
 - 4 ANTECEDENTES
 - 5 JUSTIFICACIÓN
 - 6 ARTICULACIÓN DE LA PROPUESTA DE PROYECTO CON LA RUTA DE INVESTIGACIÓN DE LA POLÍTICA DE INVESTIGACIÓN E INNOVACIÓN ORIENTADA POR MISIONES ELEGIDA
 - 7 MARCO CONCEPTUAL
 - 8 POBLACIÓN A IMPACTAR (INDICANDO SU ENFOQUE DIFERENCIAL O INTERSECCIONAL, SI APLICA)
 - 9 OBJETIVOS
 - 10 METODOLOGÍA
 - 11 SOSTENIBILIDAD DE LA PROPUESTA DE PROYECTO
 - 12 RESULTADOS E IMPACTOS ESPERADOS A NIVEL NACIONAL Y/O REGIONAL
 - 13 ESTRATEGIA DE APROPIACIÓN SOCIAL DEL CONOCIMIENTO Y DIVULGACIÓN PÚBLICA DE LA CIENCIA
 - 14 INDICADORES DE GESTIÓN
 - 15 IDONEIDAD Y TRAYECTORIA DE LA ENTIDAD DEL SISTEMA NACIONAL DE CIENCIA, TECNOLOGÍA E INNOVACIÓN QUE INTEGRA LA ALIANZA
 - 16 BIBLIOGRAFÍA
- Campos (Se ocultan)
 - CADENA PRODUCTIVA
 - CIUDAD DE FORMACION
 - CLASIFICACIÓN
 - CODIGO CIU
 - DURACIÓN PROYECTO HORAS
 - EJECUCIÓN CRONOGRAMA
 - ESTADO EMPRESA

- FOCO ESTRATÉGICO - RETOS - TEMATICAS ORIENTADORAS
 - LINEA TEMATICA
 - LUGAR DE EJECUCIÓN
 - MÚLTIPLES FOCOS Y MÚLTIPLES LÍNEAS TEMÁTICAS
 - NIVEL MADUREZ
 - NOMBRE CLÚSTER
 - PERMISOS
 - PROGRAMA NACIONAL DE CTEL PRINCIPAL
 - PROPÓSITO GENERAL
 - SECTOR
 - SUBLINEA TEMATICA PRINCIPAL
 - SUBLINEAS TEMATICAS SECUNDARIAS
 - TEMÁTICAS ORIENTADORAS POR FOCO ESTRATÉGICO
 - TIPO DE FINANCIACIÓN
 - TIPO PERSONA
 - VALOR INSCRIPCION
 - VALOR SOLICITADO
- Financiadoras
 - MINCIENCIAS
 - Requisitos
 - 5.2. Carta Unificada de Aval y Compromiso Institucional (Anexo 1).
 - 5.3 Comité de ética y/o de Bioética.
 - 5.4 Perfil de las Mujeres Científicas a Vincular en el Proyecto (Anexo 3).
 - 5.5 Entidad del Sistema Nacional de Ciencia, Tecnología e Innovación que integra la alianza.
 - 5.6. Experiencia e idoneidad de la entidad del Sistema Nacional de Ciencia, Tecnología e Innovación que integra la alianza (Anexo 4).
 - 5.7 Ser ciudadana colombiana y que en su documento de identidad se registre sexo femenino a la fecha de apertura de la convocatoria (Doctora).
 - 5.8 Contar con título de doctorado (Doctora).
 - 5.9. Enfoque diferencial y/o interseccional (Doctora).
 - 5.10 Ser ciudadana colombiana y que en su documento de identidad se registre sexo femenino a la fecha de apertura de la convocatoria y tener

máximo 28 años de edad cumplidos al 31 de diciembre de 2025 (Joven Investigadora e Innovadora).

- 5.11 Estado académico (Joven Investigadora e Innovadora).
- 5.12 Enfoque diferencial y/o interseccional (Joven Investigadora e Innovadora)
- Roles Entidades
 - EJECUTOR
- Roles del Personal
 - DOCTORA
 - JOVEN INVESTIGADORA
- Focos de la Convocatoria
 - Bioeconomía y territorio
 - Derecho humano a la alimentación
 - Transición energética
 - Soberanía sanitaria
 - Ciencia para la paz
 - Otro
- Mensajes Personalizados
- Información Complementaria

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.
- Se notifica de inconvenientes para realizar pruebas debido a falta de configuración de UVT para la vigencia 2024

Mostrar la actividad para Solicitud 152706

Número de Solicitud	Resumen de Solicitud	
152706	Gestión de Proyectos - SIGP	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
17/02/2025 16:29:38	17/02/2025 16:29:38	00:14:04

Descripción del usuario

Notificación manual
Buenas tardes.

Se esta terminando de hacer la configuración de la convocatoria de Orquideas, pero nos encontramos con un problema al momento de que una entidad del sector empresarial se postule a una convocatoria, debido a que hace falta ser configurados los valores de UVT de la vigencia 2024, los cuales son necesarios para determinar de forma calculada el tamaño de una entidad.

La anterior configuración de UVT es realizada por personal del area de beneficios tributarios ya que estos son los que poseen la información referente a UVT.

Quedo atento a sus comentarios.

--

Cordialmente,

Leonardo Suarez Baldovino
Ingeniero de Sistemas
lasb@softwarebuilders.co

Descripción del sistema

Notificación manual

Notificar 'cmunera@minciencias.gov.co' 'drrua@minciencias.gov.co' 'drtarquino@minciencias.gov.co' 'njlleon@minciencias.gov.co' 'cfgomez@minciencias.gov.co' 'lasb@softwarebuilders.co'

Título del mensaje: Solicitud 152706 Notificación manual - Falta Configuración UVT

- Se realiza la configuración de UVT 2024 a partir de la configuración de la vigencia 2023

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 152706

Número de Solicitud	Resumen de Solicitud	
152706	Gestión de Proyectos - SIGP	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
17/02/2025 19:41:16	17/02/2025 19:41:16	00:11:21

Descripción del usuario

Notificación manual
Buenas noches.

Se finalizó la configuración de la convocatoria de Orquídeas 2025. 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: INV7392025

Clave: 7foIGU3V

URL: <https://plataformasigp.minciencias.gov.co:7003/FormularioProyectos/faces/proyecto/invitacionEspecialLxhtml>

* Seleccionar la convocatoria "XXX-2025 CONVOCATORIA PROGRAMA ORQUÍDEAS".

Quedo atentos a las pruebas realizadas por el usuario funcional.

--

Cordialmente,

Leonardo Suarez Baldovino
Ingeniero de Sistemas
lasb@softwarebuilders.co

Descripción del sistema

Notificación manual

Notificar 'cmunera@minciencias.gov.co' 'drrua@minciencias.gov.co' 'drtarquino@minciencias.gov.co' 'njleone@minciencias.gov.co' 'cfgomez@minciencias.gov.co' 'lasb@softwarebuilders.co'

Título del mensaje: Solicitud 152706 Notificación manual - Convocatoria Orquídeas Disponible Pruebas

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.

Mostrar la actividad para Solicitud 152706

Número de Solicitud	Resumen de Solicitud	
152706	Gestión de Proyectos - SIGP	
Tipo de actividad	Analista	¿Interno?
Notificación manual	León Cabeza, Nidia Janneth	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
18/02/2025 16:37:53	18/02/2025 16:37:53	00:02:49

Descripción del usuario

Notificación manual
Buenas tardes Ing. Leonardo:

Se recibe comunicación (adjunto correo) para generar ajustes al formulario de la convocatoria de Orquídeas:

Asunto: Re: Solicitud 152706 Notificación manual - Convocatoria
Orquídeas Disponible Pruebas
Fecha: 2025-02-18 12:44
De: Carolina Múnera Camacho <cmunera@minciencias.gov.co>
Destinatario: mesadeservicios@minciencias.gov.co
Cc: dirue@minciencias.gov.co, njleen@minciencias.gov.co, cfigomez@minciencias.gov.co, lasb@softwarebuilders.co, María Camila Amaris Mendoza <mcamaris@minciencias.gov.co>

Cordial saludo

Informó que una vez realizadas las pruebas de parametrización de la Convocatoria "Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 2025", se tienen las siguientes observaciones, con base en la aprobación en sesión de Comité de Gestión de Recursos de la CTeI del 18 de febrero de 2025:

1. Ajustar el nombre de la convocatoria, el cual quedó aprobado de la siguiente manera: Convocatoria "Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 2025".
2. Ajustar la fecha de apertura, quedando el 18 de febrero de 2025.
3. Ajustar la fecha de cierre con base en el cronograma propuesto en la sesión de Comité de Gestión de Recursos. Se encuentra 21 de abril de 2025 y la nueva fecha es 19 de marzo de 2025.

4. se deben ajustar las Misiones señaladas en el formulario, se señalan tres (3) Misiones que corresponden a la Convocatoria No. 948 de 2025 y en esta nueva Convocatoria se trabajará con las cinco (5) Misiones que se establecen en las Políticas de Investigación e Innovación Orientadas por Misiones (PIIOM) que se exponen a continuación: 1. Bioeconomía y territorio, 2. Derecho humano a la alimentación, 3. Transición energética, 4. Soberanía sanitaria y 5. Ciencia para la paz:

5. Con base en la reunión del 17 de febrero, no se evidencia el registro de la información del interlocutor de la alianza.

6. En Identificación y Descripción del problema, se señalan 3 misiones y deben señalarse 5 (...). Descripción y justificación del problema que permita identificar las necesidades u oportunidades planteadas en torno a la línea temática seleccionada. Los proyectos de investigación, desarrollo tecnológico, y/o innovación, para el desarrollo de la estancia posdoctoral deberán estar relacionadas con al menos una de las rutas de innovación establecidas en las Misiones: "Bioeconomía y Territorio, "Derechos Humanos a la Alimentación y Ciencia para la Paz" la Misión "Ciencia para la Paz".(...)

7. El Sistema no guarda información si se hace en dos momentos (tiempos diferentes)

8. En cronograma, no permite el registro de información en inicio, fin y ejecución.

9. El campo de información complementario, donde se encuentre el desplegable del eje en el cual se registra (Nacional o Pacífico) debe estar al inicio del registro.

10. Eliminar el interlocutor del campo de personal y adicional a la entidad Alianza

11. Cambiar la redacción de DOCTOR a DOCTORA - JOVEN INVESTIGADOR a JOVEN INVESTIGADORA

12. Se debe ajustar la nota aclaratoria del monto máximo solicitado. Se encuentra por valor de \$60.000.000 y los montos máximo solicitados son:

- Apoyo al sostenimiento mensual de la doctora: \$138.864.000 por 12 meses
- Vinculación de la joven investigadora e innovadora en calidad de estudiante pregrado: \$21.468.000 por 12 meses
- Vinculación de Joven Investigadora e Innovadora en calidad de recién graduada: \$34.092.000 por 12 meses
- Apoyo para el desarrollo del proyecto en el marco de la propuesta presentada para el desarrollo de la Estancia: Hasta \$60.000.000 por un período de doce (12) meses

13. Hace falta el registro de lugar de nacimiento registrado en la cédula para la Doctora y la Jóvenes Investigadora e Innovadora.

Cordialmente,

Nidia Janneth León Cabeza
Oficina de Tecnologías y Sistemas de Información
Esta notificación se ha enviado con los archivos adjuntos siguientes:

Correo de Minciencias Fwd_ Re_ Solicitud 152706 Notifica

Descripción del sistema

Notificación manual
Notificar 'Software Builder, SB ' León Cabeza, Nidia Janneth ' Gómez Vega, César Fabián '
Título del mensaje: Solicitud 152706 Notificación manual - Ajustes a convocatoria Orquídeas

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.

Mostrar la actividad para Solicitud 152706

Número de Solicitud	Resumen de Solicitud	
152706	Gestión de Proyectos - SIGP	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
18/02/2025 18:15:32	18/02/2025 18:15:32	00:11:08

Descripción del usuario

Notificación manual

Buenas tardes.

Los ajustes solicitados posterior a las pruebas realizadas por el usuario funcional ya fueron ajustados en el formulario de proyectos. Por favor realizar nuevamente las pruebas de verificación del formulario.

NOTA 1: El punto 8 donde se indica que el cronograma, no permite el registro de información en inicio, fin y ejecución, cabe mencionar que los valores de inicio y fin son numéricos y no permite letras, el campo de "ejecución" es calculado y depende de la selección realizada en el campo "Ejecución del cronograma" de la pantalla de Información General.

NOTA 2: Los campos relacionados con el eje de la alianza (Nacional o Pacífico), el interlocutor y el Lugar de nacimiento registrado en la cédula para la DOCTORA y JOVEN INVESTIGADORA, fueron configurados para ser registrados como información complementaria en el formulario

Quedo atento.

...

Cordialmente,

Ing. Leonardo Suárez Baldovino
Software Builders Ltda
lasb@softwarebuilders.co | www.softwarebuilders.co

Carrera 700 # 51 - 57

Descripción del sistema

Notificación manual

Notificar 'csgomez@minciencias.gov.co' 'cmunera@minciencias.gov.co' 'njleon@minciencias.gov.co'

Título del mensaje: Solicitud 152706 Notificación manual - Ajustes de Observaciones Realizados

Validación final del usuario

- Se espera la aprobación del usuario funcional para confirmar que la convocatoria está parametrizada correctamente.

Mostrar la actividad para Solicitud 152706		
Número de Solicitud	Resumen de Solicitud	
152706	Gestión de Proyectos - SIGP	
Tipo de actividad	Analista	¿Interno?
Registrar comentario	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
19/02/2025 15:48:16	19/02/2025 11:14:16	00:04:54
Descripción del usuario		
Buenas tardes		
Los ajustes solicitados posterior a las pruebas realizadas por el usuario funcional se realizaron exitosamente.		
El usuario de Aval para publicar la convocatoria de Orquídeas en ambiente de producción y se procede a su publicación. Se adjunta trazabilidad de correos de aprobación con el usuario Carolina Munera		
Quedo atento.		
--		
Cordialmente,		
Ing. Leonardo Suárez Baldovino		
Software Builders Ltda		
✉ lasb@softwarebuilders.co 🌐 www.softwarebuilders.co		
📍 Carrera 700 # 51 - 57		
Descripción del sistema		
registrar comentario de usuario		

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 "Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 2025" en el sistema SIGP.

Solicitud 153242

Asunto	Tiempo total de actividad
Solicitud TI	00:10:38
Descripción	
<p>Estimados compañeros/as buenas tardes.</p> <p>Agradecemos su apoyo para poder hacer el ajuste en la parametrización de la convocatoria 953, toda vez que se realizó la adenda 1 que modifica la fecha de cierre de postulación de proyectos para el 21 de marzo a las 5:00pm.</p> <p>Adicionalmente, agradecemos que se haga pública la convocatoria, pues la parametrización está correcta.</p>	

Adjunto la adenda.

Saludos.

--

Raúl Ivan Clavijo

Contratista

Dirección de Capacidades y Apropiación del Conocimiento

riclavijo@minciencias.gov.co <riclavijo@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
18/02/2025 13:04:26	20/02/2025 16:00:00	19/02/2025 17:49:16	20/02/2025 16:00:00

Acción Realizada 153242

- Se realiza actualización en la fecha de cierre de la convocatoria y publicar la convocatoria
- Se notifica al usuario de la publicación de la convocatoria en ambiente de producción

Mostrar la actividad para Solicitud 153242

Número de Solicitud	Resumen de Solicitud	
153242	Solicitud TI	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
18/02/2025 15:46:34	18/02/2025 15:46:34	00:08:24
Descripción del usuario		
Notificación manual		
Buenas tardes.		
La convocatoria 953-2024 CONVOCATORIA PARA COMUNIDADES NEGRAS DE BOGOTÁ, CAQUETÁ, CÓRDOBA, MAGDALENA Y CHOCÓ se encuentra publicada en ambiente de producción de Minciencias. Adicionalmente se modificó la fecha de cierre de la convocatoria por la indicada (21 de marzo a las 5:00pm)		
El acceso al formulario de proyectos se hace a través de la url https://plataformasigp.minciencias.gov.co:7003/FormularioProyectos/		
Quedo atento.		
--		
Cordialmente,		
Ing. Leonardo Suárez Baldovino		
Software Builders Ltda.		
✉ lasb@softwarebuilders.co www.softwarebuilders.co		
📍 Carrera 700 # 51 - 57		
Descripción del sistema		
Notificación manual.		
Notificar 'nclavijo@minciencias.gov.co' 'nclavijo@minciencias.gov.co'		
Título del mensaje: Solicitud 153242 Notificación manual - Convocatoria Publicada		

- Se da solución y cierre a la solicitud

Mostrar la actividad para Solicitud 153242

Número de Solicitud	Resumen de Solicitud	
153242	Solicitud TI	
Tipo de actividad	Analista	¿Interno?
Resuelto	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
19/02/2025 17:47:34	19/02/2025 17:47:34	00:01:42
Descripción del usuario		
Buenas tardes.		
Se finaliza la parametrización y publicación de la convocatoria 953.		
--		
Cordialmente,		
Ing. Leonardo Suárez Baldovino		
Software Builders Ltda.		
✉ lasb@softwarebuilders.co www.softwarebuilders.co		
📍 Carrera 700 # 51 - 57		
Descripción del sistema		
Estado de Ejecución modificado a Solucionado		

Solicitud 153303

Asunto	Tiempo total de actividad
Parametrizacion	00:18:09
Descripción	
<p>Estimados equipo de Mesa de Servicios,</p> <p>Buenos días.</p> <p>Por medio del presente, solicito su apoyo para realizar la parametrización del formulario SIGP. Para ello, adjunto el formato diligenciado, Parametrización</p>	

convocatoria 2025 - Recobro mejorado en formato Excel que había sido previamente compartido por Oscar Daniel Álvarez Mora a Lili Tatiana Vega Clavijo mediante la Solicitud 148326 - Notificación manual.

Agradezco su colaboración en la gestión de esta solicitud y quedo atenta a cualquier información adicional que requieran.

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
19/02/2025 10:53:56	03/03/2025 09:53:12		

Acción Realizada 153303

1. Revisión de los términos de referencia

- Se analizan los términos de referencia de la **CONVOCATORIA PARA PROYECTOS DE I+D+i EN RECOBRO MEJORADO Y FORTALECIMIENTO EN IES – 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.
- Se deben configurar dos convocatorias por cada modalidad (1 y 2)

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
 - PROGRAMA NACIONAL DE CIENCIA, TECNOLOGÍA E INNOVACIÓN EN ENERGÍA Y MINERÍA
- Financiaciones
 - Recuperación Contingente
- Menús
 - PRY - Tipo Personal
 - PRY - Palabras Clave
 - PRY - Global Total
 - PRY - Función General Proyectos
 - PRY - Validaciones
 - PRY - Productos - Desarrollo tecnológico e innovación
 - PRY - Productos - Apropiación social del conocimiento
 - PRY - Globales Entidad
 - PRY - Grupos
 - PRY - Generalidades Personal
 - PRY - Productos - Generación de nuevo conocimiento
 - PRY - Roles Entidad
 - PRY - Proyecto
 - PRY - Cambiar Clave
 - PRY - Productos - Formación de recurso humano
 - PRY - Requisitos
 - PRY - Presupuesto
 - PRY - Rubros
 - PRY - Información General Proyecto
 - PRY - Cronograma
 - PRY - Descripciones
 - PRY - Productos
 - PRY - Personal
 - PRY - Documentos
 - PRY - Enfoque Diferencial
 - PRY - Gran Área OCDE, Área OCDE y Disciplina OCDE
- Rubros

- ADECUACION DE INFRAESTRUCTURA
- ADQUISICIÓN DE EQUIPOS
- BIBLIOGRAFÍA
- CONSULTORÍA ESPECIALIZADA
- EVENTOS ACADÉMICOS
- GASTOS DE OPERACIÓN
- GASTOS DE PROPIEDAD INTELECTUAL
- MATERIALES E INSUMOS
- MOVILIDAD DE INVESTIGACIÓN
- PERSONAL CIENTÍFICO
- PUBLICACIONES
- SALIDAS DE CAMPO
- SERVICIOS TÉCNICOS
- SOFTWARE

- Descripciones

- 1 CONFORMACIÓN DEL EQUIPO DE INVESTIGACIÓN
- 2 NIVEL DE TRL INICIAL
- 3 NIVEL DE TRL ESPERADO AL FINAL DEL PROYECTO
- 4 RESUMEN EJECUTIVO
- 5 ANTECEDENTES Y RESULTADOS PREVIOS DEL EQUIPO DE INVESTIGACIÓN
- 6 PLANTEAMIENTO DEL PROBLEMA Y DE LA PREGUNTA DE INVESTIGACIÓN
- 7 ESTADO DEL ARTE
- 8 OBJETIVO GENERAL
- 9 OBJETIVOS ESPECÍFICOS
- 10 METODOLOGÍA
- 11 RESULTADOS ESPERADOS DE LA INVESTIGACIÓN
- 12 PLAN DE DIFUSIÓN DE LOS RESULTADOS
- 13 CARÁCTER NOVEDOSO DEL PROYECTO
- 14 ESTRATEGIA DE TRANSFERENCIA DE LOS RESULTADOS A LOS SECTORES PRODUCTIVOS E INSTITUCIONAL (ANH)
- 15 APLICACIÓN POTENCIAL DE LOS RESULTADOS
- 16 IMPACTO AMBIENTAL
- 17 BIBLIOGRAFÍA

- Campos (Se ocultan)
 - CADENA PRODUCTIVA
 - CIUDAD DE FORMACION
 - CLASIFICACIÓN
 - CODIGO CIU
 - DURACIÓN PROYECTO HORAS
 - ESTADO EMPRESA
 - LINEA TEMATICA
 - NIVEL MADUREZ
 - NOMBRE CLÚSTER
 - PERMISOS
 - PROPÓSITO GENERAL
 - SECTOR
 - TEMÁTICAS ORIENTADORAS POR FOCO ESTRATÉGICO
 - TIPO PERSONA
 - VALOR INSCRIPCION
- Financiadoras
 - MINCIENCIAS
- Requisitos
 - Carta unificada de aval y compromiso institucional
 - Carta de Aval del Comité de Ética/Bioética nacional
 - Acto administrativo que soporta la conformación del Comité de ética que lo expide
 - Hoja de vida de los Investigadores empresa/internacionales
 - Experiencia específica del personal
 - Experiencia en proyectos de recobro mejorado de hidrocarburos
 - Certificado de Existencia y Representación Legal empresa co-ejecutora
 - Certificado de Existencia y Representación Legal aliados
 - Registro del personal en el SIGP
 - Certificado de Capacidad Financiera y Organizacional
 - Permisos o licencias ambientales
- Roles Entidades
 - EJECUTOR
 - COEJECUTOR

- ALIADO
- Roles del Personal
 - ESTUDIANTE MAESTRIA
 - ESTUDIANTE POST-DOCTORAL
 - INVESTIGADOR PRINCIPAL O DIRECTOR DEL PROYECTO
 - JOVEN INVESTIGADOR E INNOVADOR (PREGRADO)
- Contrapartidas por Tamaño de empresa
- Mensajes Personalizados
- Información Complementaria

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

Mostrar la actividad para Solicitud 153303		
Número de Solicitud	Resumen de Solicitud	
153303	Parametrización	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
25/02/2025 18:19:13	25/02/2025 18:19:13	00:10:34
Descripción del usuario		
Notificación manual		
Buenos días,		
Se finalizó la configuración de la convocatoria 2025 - Recobro mejorado. 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:		
URL: https://plataformasigp.minciencias.gov.co:7003/FormularioProyectos/faces/proyecto/invitacionEspecial.xhtml		
***** MODALIDAD 1 *****		
Usuario: INV7462025		
Clave: J9T4UzC@		
* Seleccionar la convocatoria Modalidad 1 "XXXX-2025 CONVOCATORIA PARA PROYECTOS DE I+D+i EN RECOBRO MEJORADO Y FORTALECIMIENTO EN IES - MOD 1".		
***** MODALIDAD 2 *****		
Usuario: INV7472025		
Clave: aARQ5Vv		
* Seleccionar la convocatoria Modalidad 2 "XXXX-2025 CONVOCATORIA PARA PROYECTOS DE I+D+i EN RECOBRO MEJORADO Y FORTALECIMIENTO EN IES - MOD 2".		
Quedo atento 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		
Descripción del sistema		
Notificación manual		
Notificar: 'hjaon@minciencias.gov.co' 'lmerchan@minciencias.gov.co' 'lasb@softwarebuilders.co'		
Título del mensaje: Solicitud 153303 Notificación manual - Convocatoria 2025 Recobro Mejorada Disponible Pruebas		

- Se notifica por segunda vez al usuario para que realice pruebas funcionales de la parametrización de la convocatoria

Mostrar la actividad para Solicitud 153303		
Número de Solicitud	Resumen de Solicitud	
153303	Parametrización	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder: SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo Invertido
03/03/2025 09:47:32	03/03/2025 09:47:32	00:05:40
Descripción del usuario		
Notificación manual		
Buenos días,		
El pasado 24 de febrero de 2025, se envió una notificación informando que la configuración de la convocatoria 2025 - Recobro Mejorado había sido finalizada y estaba disponible para pruebas de verificación. Sin embargo, a la fecha no hemos recibido confirmación sobre la ejecución de dichas pruebas ni observaciones al respecto.		
Agradecemos su retroalimentación sobre el estado de la validación realizada o, en caso de no haber iniciado las pruebas, su confirmación de cuándo se podrán llevar a cabo. Es importante contar con esta validación para garantizar el correcto funcionamiento de la convocatoria antes de su publicación definitiva.		
Los datos de acceso son los siguientes:		
🔗 URL: https://plataformasgi.minciencias.gov.co/7003/FormularioProyectos/faces/proyecto/invitacionEspecial.xhtml		
MODALIDAD 1		
👤 Usuario: INV7462025		
🔑 Clave: J9T6UzCg		
📄 Convocatoria: "000V-2025 CONVOCATORIA PARA PROYECTOS DE I+D+I EN RECOBRO MEJORADO Y FORTALECIMIENTO EN IES - MOD 1"		
MODALIDAD 2		
👤 Usuario: INV7472025		
🔑 Clave: aAKQ5Vv		
📄 Convocatoria: "000V-2025 CONVOCATORIA PARA PROYECTOS DE I+D+I EN RECOBRO MEJORADO Y FORTALECIMIENTO EN IES - MOD 2"		
Quedamos atentos a su confirmación o cualquier inquietud que pueda surgir.		
Saludos,		
--		
Cordialmente,		
Ing. Leonardo Suárez Baldovino		
Software Builders Ltda		
📧 info@softwarebuilders.co 🌐 www.softwarebuilders.co		
📍 Carrera 700 # 51 - 57		

6. Pruebas y observaciones del usuario

- PENDIENTE de Retroalimentación de pruebas por parte del usuario funcional.

Solicitud 153373

Asunto	Tiempo total de actividad
Parametrización convocatoria 2025 ECONOVA.xlsx	00:42:36
Descripción	
<p>Estimados equipo de Mesa de Servicios,</p> <p>Saludo cordial</p> <p>Por medio del presente, solicito su apoyo para realizar la parametrización del formulario SIGP. Para ello, adjunto el archivo (Parametrización convocatoria 2025 ECONOVA.xlsx) en formato Excel.</p>	

Agradezco su colaboración en la gestión de esta solicitud y quedo atento a cualquier información adicional que se requiera.

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
19/02/2025 19:20:33	06/03/2025 16:56:22		

Acción Realizada 153373

1. Revisión de los términos de referencia

- Se analizan los términos de referencia de la **CONVOCATORIA PARA EL FORTALECIMIENTO Y CONSOLIDACIÓN DE LOS CENTROS DE INNOVACIÓN DE LA RED ECONOVA** 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 CIENCIA
 - Financiaciones
 - Recuperación Contingente

- Menús

- PRY - Función General Proyectos
- PRY - Cambiar Clave
- PRY - Información General Proyecto
- PRY - Gran Área OCDE, Área OCDE y Disciplina OCDE
- PRY - Palabras Clave
- PRY - Descripciones
- PRY - Cronograma
- PRY - Roles Entidad
- PRY - Proyecto
- PRY - Personal
- PRY - Tipo Personal
- PRY - Generalidades Personal
- PRY - Presupuesto
- PRY - Rubros
- PRY - Globales Entidad
- PRY - Global Total
- PRY - Documentos
- PRY - Requisitos
- PRY - Productos
- PRY - Productos - Generación de nuevo conocimiento
- PRY - Productos - Apropiación social del conocimiento
- PRY - Productos - Desarrollo tecnológico e innovación
- PRY - Productos - Formación de recurso humano
- PRY - Validaciones
- PRY - Enfoque Diferencial

- Rubros

- ACTIVIDADES DE DIFUSIÓN
- ASESORÍA ESPECIALIZADA
- COSTO DE REGISTRO Y CERTIFICACIONES
- EQUIPO DE TRABAJO
- EQUIPOS
- GASTOS DE ADMINISTRACIÓN
- GASTOS DE DESPLAZAMIENTO
- GASTOS DE PROPIEDAD INTELECTUAL
- MADURACIÓN DE SOLUCIONES TECNOLÓGICAS

- MATERIAL BIBLIOGRÁFICO
- MATERIALES E INSUMOS
- PERSONAL DE APOYO
- PUBLICACIONES Y DIFUSIÓN DE RESULTADOS
- SALIDAS DE CAMPO
- SERVICIOS TÉCNICOS Y TECNOLÓGICOS
- SOFTWARE ESPECIALIZADO

- Descripciones
 - 1 EQUIPO DE TRABAJO
 - 2 RESUMEN EJECUTIVO
 - 3 OBJETIVO GENERAL
 - 4 OBJETIVOS ESPECÍFICOS
 - 5 METODOLOGÍA
 - 6 RESULTADOS ESPERADOS
 - 7 IMPACTOS POTENCIALES
 - 8 BIBLIOGRAFÍA

- Campos (Se ocultan)
 - CADENA PRODUCTIVA
 - CIUDAD DE FORMACION
 - CLASIFICACIÓN
 - CODIGO CIU
 - DURACIÓN PROYECTO HORAS
 - ESTADO EMPRESA
 - LINEA TEMATICA
 - NIVEL MADUREZ
 - NOMBRE CLÚSTER
 - PERMISOS
 - PROPÓSITO GENERAL
 - SECTOR
 - TEMÁTICAS ORIENTADORAS POR FOCO ESTRATÉGICO
 - TIPO PERSONA
 - VALOR INSCRIPCION

- Financiadoras
 - MINCIENCIAS

- Requisitos
 - Anexo 3. Carta Unificada de aval y compromiso institucional
 - Anexo 4. Experiencia de la Alianza
 - Soportes de anexo de experiencia de la alianza
 - Soportes de anexo de experiencia calificable
 - Anexo 5. Experiencia equipo de trabajoRequisitos de Documentos
 - Soportes de anexo de experiencia mínima habilitante equipo de trabajo
 - Soportes de anexo de experiencia calificable equipo de trabajo
 - Anexo 6. Cálculo Indicadores financieros
 - Estados financieros a 2024
- Roles Entidades
 - EJECUTOR
 - COEJECUTOR
- Roles del Personal
 - ASESOR EN PROPIEDAD INTELECTUAL
 - COINVESTIGADOR
 - DIRECTOR DE LA PROPUESTA
 - ESTUDIANTE DE DOCTORADO
 - ESTUDIANTE DE MAESTRÍA
 - ESTUDIANTE DE PREGRADO
 - GESTOR DE PROYECTO
 - JOVEN INVESTIGADOR
 - LÍDER ADMINISTRATIVO Y FINANCIERO
 - LÍDER DE CAPACITACIONES
 - LÍDER DE INNOVACIÓN Y TRANSFERENCIA TECNOLÓGICA
 - PERSONAL DE APOYO
 - PROFESIONAL ADMINISTRATIVO - FINANCIERO
- Mensajes Personalizados

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.

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 de Observaciones y dudas al usuario funcional

- Se envía una notificación manual y documento adjunto al usuario sobre Consultas y observaciones sobre la parametrización de la convocatoria 2025 ECONOVA

Mostrar la actividad para Solicitud 153373		
Número de Solicitud	Resumen de Solicitud	
153373	Parametrización convocatoria 2025 ECONOVA.xlsx	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builders: 59	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo Invertido
27/02/2025 12:04:05	27/02/2025 12:04:05	00:16:59
Descripción del usuario		
Notificación manual.		
Buen día Oscar Sánchez,		
Espero que se encuentre bien.		
Tras revisar la parametrización de la convocatoria 2025 ECONOVA, me surgen algunas dudas y observaciones que agradecería su apoyo en aclarar:		
1. Descripción de la convocatoria: ¿Existe una descripción que indique en qué consiste y/o de qué trata la convocatoria?		
2. Programa de la convocatoria: ¿Podrían indicar a qué programa está asociada esta convocatoria?		
3. Etapa de subsección: ¿Se ha definido una fecha y hora de inicio y fin para esta etapa?		
4. Dirigido a:		
* ¿El rol "Otros coejecutores" es diferente al rol "Co-ejecutor"?		
* En caso de que sea el mismo, ¿es obligatorio que al menos una entidad esté registrada con el rol "Co-ejecutor"?		
5. Equipo de trabajo:		
* ¿Cuáles roles requieren que el personal esté registrado en CUIAC?		
* ¿Existe un límite mínimo o máximo de personal que se puede registrar por cada rol?		
6. Descripción de equipo de trabajo: La descripción en la sección de observaciones parece estar incompleta.		
7. Componente presupuestal: No observo restricciones asociadas a rubros específicos. La única restricción general identificada es el monto máximo de financiamiento de \$9.274.009.854. ¿Podrían confirmar si existen otras restricciones?		
8. Mensajes informativos: ¿Se tiene previsto mostrar algún mensaje informativo para el usuario en el formulario? En caso afirmativo, ¿en qué secciones específicas se mostrarían y cuál sería su contenido?		
Agradezco su orientación sobre estos puntos y quedo atento/a a su respuesta.		
...		
Cordialmente,		
Ing. Leonardo Suárez Baldovino		
Software Builders Ltda		
✉ lasb@softwarebuilders.co 🌐 www.softwarebuilders.co		
📞 Carrera 700 # 51 - 57		

- Se notifica por segunda vez al usuario sobre las observaciones enviadas y que no han tenido respuesta alguna

Mostrar la actividad para Solicitud 153373

Número de Solicitud	Resumen de Solicitud	
153373	Parametrización convocatoria 2025 ECONOVA.xlsx	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
03/03/2025 09:58:23	03/03/2025 09:58:23	00:17:54
Descripción del usuario		
Notificación manual		
Buen día,		

Espero que se encuentren bien.

El pasado 27 de febrero de 2025, envié un correo con algunas consultas y observaciones sobre la parametrización de la convocatoria 2025 ECONOVA. Hasta el momento, no hemos recibido respuesta.

Agradecemos su retroalimentación sobre los puntos planteados, ya que esta información es clave para continuar con el proceso de parametrización y validación de la convocatoria.

En caso de requerir más tiempo para la revisión, le agradeceríamos nos indique una fecha estimada en la que podríamos recibir sus comentarios.

Quedo atento a su pronta respuesta.

Saludos,

—

Cordialmente,

Ing. Leonardo Suárez Baldovino
Software Builders Ltda
✉ las@softwarebuilders.co | www.softwarebuilders.co
📍 Carrera 700 # 51 - 57

Solicitud 153373 Notificación manual.

Asignado a: Software Builder, SB

Cliente: Sánchez Plazas, Oscar Daniel

Descripción: Estimados equipo de Mesa de Servicios,

Saludo cordial.

Por medio del presente, solicito su apoyo para realizar la parametrización del formulario SIGR. Para ello, adjunto el archivo (Parametrización convocatoria 2025 ECONOVA.xlsx) en formato Excel.

Agradecemos su colaboración en la gestión de esta solicitud y quedo atento a cualquier información adicional que se requiera.

Haga clic en la URL siguiente para visualizar:
<https://soporte.minciencias.gov.co:8443/Caiad/pdmweb.exe?OP=SEARCH+FACTOR=cr+SKIPLIST=1+QBEEQ=593167>

- Se recibe retroalimentación del usuario y se procede a ajustar la parametrización de la convocatoria según lo indicado

Estimada Mesa de Servicios,

Reciba un cordial saludo. En atención a sus recientes interrogantes, me permito ofrecer las siguientes respuestas. Es importante destacar que varias de las respuestas a sus preguntas se encuentran detalladas en los términos y condiciones y anexos, a los cuales puede acceder a través del siguiente enlace: (https://drive.google.com/drive/folders/1v9qD1Jt3jevT_QOfuKb0Ctpn2kP14P4K?usp=drive_link). Le recomiendo revisar estos documentos para obtener información completa y detallada.

Con respecto a la pregunta número 8, agradezco su comprensión mientras nos tomamos el tiempo necesario para discutirlo minuciosamente con el equipo correspondiente. Le proporcionaré una respuesta detallada a la brevedad posible. Mientras tanto, me permito responder a las demás preguntas formuladas.

Quedo a su entera disposición para aclarar cualquier duda adicional que pueda surgir. No dude en ponerse en contacto si requiere información adicional o asistencia.

1. Descripción de la convocatoria: ¿Existe una descripción que indique en qué consiste y/o de qué trata la convocatoria?
 - a. Se anexan los términos de referencia a este correo, sin embargo el objetivo principal es conformar un banco de propuestas elegibles enfocadas al fortalecimiento de capacidades de la red de centros de innovación ECONOVA de Ecopetrol y al desarrollo de comunidades de innovación en territorios asociados a los centros. Este se encuentra en el numeral 2 de los términos de referencia.
2. Programa de la convocatoria: ¿Podrían indicar a qué programa está asociada esta convocatoria?
 - a. Esto hace parte del Convenio Especial de Cooperación 656-2022.
3. Etapa de subsanación: ¿Se ha definido una fecha y hora de inicio y fin para esta etapa?
 - a. Sí, se encuentran en los términos aquí anexados, en el numeral 15, en donde se muestra que el periodo de subsanación de requisitos es del 3 al 4 de junio de 2025 hasta las 4:00 pm.

4. Dirigido a:

* ¿El rol "Otros coejecutores" es diferente al rol "Co-ejecutor"?

* En caso de que sea el mismo, ¿es obligatorio que al menos una entidad esté registrada con el rol "Co-ejecutor"? , Sí, para conformar la alianza.

5. Equipo de trabajo:

* ¿Cuáles roles requieren que el personal esté registrado en CvLAC?

¿Existe un límite mínimo o máximo de personal que se puede registrar por cada rol? Mínimo:

- Líder de innovación y transferencia tecnológica
- Líder de capacitaciones
- Asesor en propiedad intelectual
- Líder administrativo y financiero
- Profesional administrativo y financiero
- Gestores de proyecto (Mínimo uno por centro para un total de siete)

6. Descripción de equipo de trabajo: La descripción en la sección de observaciones parece estar incompleta. Respuesta: Ya se encuentra detallada en los términos de referencia y en el anexo 4

7. Componente presupuestal: No observo restricciones asociadas a rubros específicos. La única restricción general identificada es el monto máximo de financiamiento de \$9.274.009.854. ¿Podrían confirmar si existen otras restricciones? Se encuentran estipuladas en los términos de referencia y en el anexo 2, numeral 2.2. COMPONENTE PRESUPUESTAL.

8. Mensajes informativos: ¿Se tiene previsto mostrar algún mensaje informativo para el usuario en el formulario? En caso afirmativo, ¿en qué secciones específicas se mostrarían y cuál sería su contenido?

6. 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 153373		
Número de Solicitud	Resumen de Solicitud	
153373	Parametrización convocatoria 2025 ECONOVIA	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
06/03/2025 16:53:13	06/03/2025 16:53:13	00:03:09
Descripción del usuario		
Notificación manual		
Buenas tardes.		
Se finalizó la configuración de la convocatoria 2025 - ECONOVIA. 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:		
URL: https://plataformas.gp.minciencias.gov.co:7003/FormularioProyectos/faces/proyecto/invitacionEspecial.xhtml		
Usuario: INV7492025		
Clave: vby92n9		
* Seleccionar la convocatoria "CONVOCATORIA PARA EL FORTALECIMIENTO Y CONSOLIDACIÓN DE LOS CENTROS DE INNOVACIÓN DE LA RED ECONOVIA".		
Quedo atentos a las pruebas realizadas por el usuario funcional.		
--		
Cordialmente,		
Ing. Leonardo Suárez Beldovino		
Software Builders Ltda.		
✉ lasb@softwarebuilders.co www.softwarebuilders.co		
📍 Carrera 700 # 51 - 57		

7. Pruebas y observaciones del usuario

- El usuario funcional realiza pruebas en el sistema.
- PENDIENTES a pruebas realizadas por el usuario funcional.

Solicitud 154026

Asunto		Tiempo total de actividad	
Convocatoria Océanos, Clima y Recursos Marino CosterosBuenos días. una vez abierto y asignado el ticket a Leonardo, envío los archivos requeridos.		00:28:02	
Descripción			
Convocatoria Océanos, Clima y Recursos Marino Costeros			
Buenos días. una vez abierto y asignado el ticket a Leonardo, envío los archivos requeridos.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
04/03/2025 09:48:27	13/03/2025 12:45:19		

Acción Realizada 153373

1. Revisión de los términos de referencia

- Se analizan los términos de referencia de la **NUEVA CONVOCATORIA EN OCÉANOS, CLIMA Y RECURSOS MARINOS COSTEROS** 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 CIENCIA
 - Financiaciones
 - Recuperación Contingente
 - Menús
 - PRY - Función General Proyectos
 - PRY - Cambiar Clave
 - PRY - Información General Proyecto
 - PRY - Gran Área OCDE, Área OCDE y Disciplina OCDE
 - PRY - Palabras Clave
 - PRY - Descripciones
 - PRY - Cronograma
 - PRY - Roles Entidad
 - PRY - Proyecto

- PRY - Personal
- PRY - Tipo Personal
- PRY - Generalidades Personal
- PRY - Presupuesto
- PRY - Rubros
- PRY - Globales Entidad
- PRY - Global Total
- PRY - Documentos
- PRY - Requisitos
- PRY - Productos
- PRY - Productos - Generación de nuevo conocimiento
- PRY - Productos - Apropiación social del conocimiento
- PRY - Productos - Desarrollo tecnológico e innovación
- PRY - Productos - Formación de recurso humano
- PRY - Validaciones
- PRY - Enfoque Diferencial
- Rubros
 - ACTIVIDADES DE DIFUSIÓN
 - ASESORÍA ESPECIALIZADA
 - COSTO DE REGISTRO Y CERTIFICACIONES
 - EQUIPO DE TRABAJO
 - EQUIPOS
 - GASTOS DE ADMINISTRACIÓN
 - GASTOS DE DESPLAZAMIENTO
 - GASTOS DE PROPIEDAD INTELECTUAL
 - MADURACIÓN DE SOLUCIONES TECNOLÓGICAS
 - MATERIAL BIBLIOGRÁFICO
 - MATERIALES E INSUMOS
 - PERSONAL DE APOYO
 - PUBLICACIONES Y DIFUSIÓN DE RESULTADOS
 - SALIDAS DE CAMPO
 - SERVICIOS TÉCNICOS Y TECNOLÓGICOS
 - SOFTWARE ESPECIALIZADO
- Descripciones
 - 1 EQUIPO DE TRABAJO

- 2 RESUMEN EJECUTIVO
- 3 OBJETIVO GENERAL
- 4 OBJETIVOS ESPECÍFICOS
- 5 METODOLOGÍA
- 6 RESULTADOS ESPERADOS
- 7 IMPACTOS POTENCIALES
- 8 BIBLIOGRAFÍA
- Campos (Se ocultan)
 - CADENA PRODUCTIVA
 - CIUDAD DE FORMACION
 - CLASIFICACIÓN
 - CODIGO CIU
 - DURACIÓN PROYECTO HORAS
 - ESTADO EMPRESA
 - LINEA TEMATICA
 - NIVEL MADUREZ
 - NOMBRE CLÚSTER
 - PERMISOS
 - PROPÓSITO GENERAL
 - SECTOR
 - TEMÁTICAS ORIENTADORAS POR FOCO ESTRATÉGICO
 - TIPO PERSONA
 - VALOR INSCRIPCION
- Financiadoras
 - MINCIENCIAS
- Requisitos
 - Anexo 3. Carta Unificada de aval y compromiso institucional
 - Anexo 4. Experiencia de la Alianza
 - Soportes de anexo de experiencia de la alianza
 - Soportes de anexo de experiencia calificable
 - Anexo 5. Experiencia equipo de trabajoRequisitos de Documentos
 - Soportes de anexo de experiencia mínima habilitante equipo de trabajo
 - Soportes de anexo de experiencia calificable equipo de trabajo
 - Anexo 6. Cálculo Indicadores financieros

- Estados financieros a 2024
- Roles Entidades
 - EJECUTOR
 - COEJECUTOR
- Roles del Personal
 - ASESOR EN PROPIEDAD INTELECTUAL
 - COINVESTIGADOR
 - DIRECTOR DE LA PROPUESTA
 - ESTUDIANTE DE DOCTORADO
 - ESTUDIANTE DE MAESTRÍA
 - ESTUDIANTE DE PREGRADO
 - GESTOR DE PROYECTO
 - JOVEN INVESTIGADOR
 - LÍDER ADMINISTRATIVO Y FINANCIERO
 - LÍDER DE CAPACITACIONES
 - LÍDER DE INNOVACIÓN Y TRANSFERENCIA TECNOLÓGICA
 - PERSONAL DE APOYO
 - PROFESIONAL ADMINISTRATIVO - FINANCIERO
- 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.
- Se envía una notificación manual al usuario de Observaciones e Información Adicional

Mostrar la actividad para Solicitud 154026

Número de Solicitud	154026	Resumen de Solicitud	Convocatoria Océanos, Clima y Recursos Marino Costeros Buenos días, una vez abierto y asignado el ticket a Leonardo, envió los archivos requeridos.
Tipo de actividad	Notificación manual	Analista	¿Interno?
Notificación manual	Software Builder, 58		No
Marca de tiempo del sistema	07/03/2025 13:00:46	Fecha de la actividad	Tiempo Invertido
Descripción del usuario	Notificación manual		00:04:40

Buenos días.

Observando la matriz de parametrización que ustedes diligenciaron de la convocatoria en océanos, clima y recursos marinos - costero, se tienen las siguientes observaciones:

1. No se indicó la Hora de cierre de la convocatoria ni tampoco la hora del cierre de la etapa de subsanación
2. Cuando se incluyen más entidades a la alianza, que rol deben llevar esas entidades (Ej: ALIADO, COOPERANTE, etc.)
3. En la sección del Documento Técnico o contenido de la propuesta, NO incluyeron las descripciones respectivas de cada ítem, dicha información es importante ser registrada para que permita ser parametrizada en el sistema.
4. En el equipo investigador, no hay claridad sobre los roles asignados al personal:

* Personal que se asocie al proyecto
* Personal con actividades de CTeI debe tener el registro en el CVLAC

Es decir, si es personal de apoyo, personal científico, personal administrativo, etc. Así mismo, tampoco hay claridad sobre la cantidad mínima y máxima de personas que pueden registrarse por cada rol.

5. No se puede limitar a que la entidad ejecutora que presenta la propuesta pueda ser solamente de los tipos:

- Instituciones de Educación Superior (IES), públicas y/o privadas.
- Centros o institutos de investigación (públicos o dependientes) con reconocimiento vigente al cierre de la convocatoria.

Para lo anterior, o la entidad ejecutora es de un tipo o del otro o en su defecto queda abierto para cualquier tipo de entidad y ya se haría la validación respectiva internamente en cada área si cumple o no los TDR.

Quedo atento

--

Cordialmente,

Ing. Leonardo Suárez Baldovino
Software Builders Ltda
[i] lasb@softwarebuilders.co [t] www.softwarebuilders.co
[l] Carrera 700 # 51 - 57

- Se recibe retroalimentación del usuario y se procede a ajustar la parametrización de la convocatoria según lo indicado

Re: Solicitud 154026 Notificación manual - Observaciones y/o Información Adicional

De: Mónica Carolina Soto Tamayo <mcsoto@minciencias.gov.co>
Destinatario: mesadeservicios@minciencias.gov.co
Cc: lcovalle@minciencias.gov.co, nleon@minciencias.gov.co, lasb@software-b.com, Iván Rodrigo Luna Castro <irluna@minciencias.gov.co>, Ana Lucía Caicedo Laurido <alcaicedo@minciencias.gov.co>, Shilene Yalime Ortiz Rolón <eyortiz@minciencias.gov.co>, Andrés Felipe Chilma Arias <afchilma@minciencias.gov.co>, Paula Judith Rojas Higuera <projas@minciencias.gov.co>
Fecha: 2025-03-07 15:10
[i] Resumen [c] Cabezeras [t] Solo texto

ANEXO 4. DESCRIPCIÓN DE CONTENIDOS DE LA PROPUESTA.docx (~164 KB)

Estimado Ing. Leonardo,

De manera atenta, remitimos las aclaraciones a las observaciones de matriz de parametrización de la convocatoria en océanos, clima y recursos marinos - costeros.

1. Cierre de la convocatoria: 30 de Mayo de 2025 hora 4:00 p.m.
Cierre de la etapa de subsanación: Del 9 al 11 de junio de 2025 hora 4:00 p.m.
2. Cuando se incluyen más entidades a la alianza, el rol que deben llevar esas entidades es de ALIADO
3. Se adjunta documentos anexo 4. Descripción de contenidos de la propuesta
4. El presupuesto para el pago de personal científico y de apoyo no podrá exceder el 60% del presupuesto total solicitado al Ministerio.
- Se permitirá la inclusión de costos de administración de hasta 15% del valor del proyecto, esto incluye el personal administrativo
- Joven Investigador Tener 28 años o menos al 31 de marzo de 2025. Para acreditar este requisito debe adjuntar la fotocopia de la cédula de ciudadanía por ambas caras ampliada al 150%. este tendrá una asignación presupuestal de TREINTA Y CUATRO MILLONES NOVENTA Y DOS MIL PESOS (\$ 34.092.000) MCTE., con pagos mensuales iguales por un valor de DOS MILLONES CUATROCIENTOS OCHENTA Y UN MIL PESOS (\$2.481.000) MCTE., por doce (12) meses.

Quedamos atentos a sus comentarios.

Mónica Carolina Soto Tamayo
Contratista
Dirección de Ciencia
mcsoto@minciencias.gov.co
Teléfono: 601 6258480 Ext. 4502
Av. Calle 26 No. 57- 83
Torre 8, pisos 2 al 6
Bogotá, Colombia

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 154026		
Número de Solicitud	Resumen de Solicitud	
154026	Convocatoria Océanos, Clima y Recursos Marino Costeros Buenos días, una vez abierto y asignado el ticket a Leonardo, envío los archivos requeridos.	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo Invertido
10/03/2025 10:36:00	10/03/2025 10:36:00	00:10:26
Descripción del usuario		
Notificación manual		
Buenos días Mónica Soto.		
Se finalizó la configuración y parametrización de la Convocatoria en océanos, clima y recursos marinos - costeros. 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: INV7512025		
Clave: Bu+mknRk		
URL: https://plataformas/gp.minciencias.gov.co/7003/FormularioProyectos/faces/proyecto/invitacionEspecial.xhtml		
* Seleccionar la convocatoria "ZZZZ-2025 NUEVA CONVOCATORIA EN OCEANOS, CLIMA Y RECURSOS MARINOS COSTEROS".		
Quedo atento a las pruebas realizadas por el usuario funcional.		
...		
Cordialmente,		
Ing. Leonardo Suárez Beldovino		
Software Builders Ltda		
✉ last@softwarebuilders.co 🌐 www.softwarebuilders.co		
📍 Carrera 700 # 51 - 57		

7. Ajustes y nueva notificación

- PENDIENTES a pruebas realizadas por el usuario funcional.

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
152950	Reporte Convocatoria No 40	Pendiente Usuario	Aclaraciones	[Ver detalle]
153379	Informe Auditoría Proyecto 109715	Solucionado	-	[Ver detalle]
153656	Reporte 962-2024 SENAINNOVA	Solucionado	-	[Ver detalle]
153657	Reporte Solicitudes Actores	Solucionado	-	[Ver detalle]
153701	Reporte evaluación Convocatoria SGR No 39 de 2024	Solucionado	-	[Ver detalle]
153944	Reporte 953-2024 Comunidades negras	Solucionado	-	[Ver detalle]
154158	Reportes Evaluación Convocatorias 013-033 SGR	Solucionado	-	[Ver detalle]
154183	Convocatoria 962 solicitud reporte propuestas	Solucionado	-	[Ver detalle]
154255	Reporte general y el reporte de presupuesto Convocatoria 962-2024 SENAINNOVA	Solucionado	-	[Ver detalle]
154276	Convocatoria 962 solicitud reporte propuestas cierre de convocatoria	Solucionado	-	[Ver detalle]

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
154339	Reporte General Convocatoria 962-2024 SENAINNOVA	Solucionado	-	[Ver detalle]
154351	Reporte Convocatoria 953-2024	Solucionado	-	[Ver detalle]
154464	Reporte Proyectos Convocatoria 963-2025 Orquídeas	Solucionado	-	[Ver detalle]
154607	Reporte Convocatoria 953-2024 14_03_2025	Solucionado	-	[Ver detalle]

Solicitud 152950

Asunto		Tiempo total de actividad	
Reporte convocatoria No. 40		00:33:17	
Descripción			
<p>Cordial saludo,</p> <p>En virtud de la nueva asignación que he recibido como líder de la Convocatoria No. 40 por parte del Director de Ciencia, me permito solicitar amablemente el Reporte final de Evaluación de la CONV 040-2024.</p> <p>Este documento es fundamental para llevar a cabo la revisión y verificación correspondiente, con el fin de gestionar y responder adecuadamente a las PQRS recibidas en el marco de esta convocatoria.</p> <p>Agradezco su apoyo y quedo atenta su pronta respuesta.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
13/02/2025 11:54:43	14/03/2025 11:43:40		

Acción Realizada 152950

Para atender la solicitud de generación de reportes, se realizó lo siguiente:

- Se notifica al usuario Solbrigitte Merchán Vaca, para que brinde la información de manera precisa, le agradecería si pudiera especificar con mayor detalle el tipo de reporte que requiere.

Mostrar la actividad para Solicitud 152950		
Número de Solicitud	Resumen de Solicitud	
152950	Reporte convocatoria No. 40	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
12/03/2025 11:01:04	12/03/2025 11:01:04	00:17:23
Descripción del usuario		
Notificación manual		
Cordial saludo,		
Agradezco su solicitud respecto al Reporte Final de Evaluación de la CONV 040-2024. Sin embargo, para poder brindarle la información de manera precisa, le agradecería si pudiera especificar con mayor detalle el tipo de reporte que requiere.		
¿Se refiere a un consolidado general de todas las evaluaciones realizadas, un informe con detalles específicos de cada evaluación, o algún otro documento en particular?		
Cabe mencionar que los reportes de evaluación NO son cargados a la mesa de servicios, sino que son entregados directamente por el jefe de la OTSI Harold Méndez al usuario y/o jefe de área encargada debido a la confidencialidad de la información.		
Quedo atento a su confirmación para proceder con su solicitud.		
--		
Cordialmente,		
Ing. Leonardo Suárez Baldovino		
Software Builders Ltda.		
✉ lasb@softwarebuilders.co 🌐 www.softwarebuilders.co		
📍 Carrera 700 # 51 - 57		
Solicitud 152950 Notificación manual.		
Asignado a: Software Builder, SB		
Cliente: Merchán Vaca, Solbrigitte		
Descripción: Cordial saludo,		
En virtud de la nueva asignación que he recibido como líder de la Convocatoria No. 40 por parte del Director de Ciencias, me permito solicitar amablemente el Reporte Final de Evaluación de la CONV 040-2024.		
Este documento es fundamental para llevar a cabo la revisión y verificación correspondiente, con el fin de gestionar y responder adecuadamente a las PORS recibidas en el marco de esta convocatoria.		
Agradezco su apoyo y quedo atento a su pronta respuesta.		
Haga clic en la URL siguiente para visualizar:		
https://soporte.minciencias.gov.co:8443/CA/sol/pdmweb.exe?OP=SEARCH+FACTORV+cr+SKPLIST+1+QBEEQ?id=592744		
Descripción del sistema		
Notificación manual		
Notificar Merchán Vaca, Solbrigitte Luna Castro, Iván Rodrigo León Cabeza, Nidia Janneth Software Builder, SB Méndez Guevara, Harold Alexander Ovalle Acosta, Luis Carlos		
Título del mensaje: Solicitud 152950 Notificación manual - Aclaración sobre Reporte Final de Evaluación - CONV 040-2024		

- Pendiente de retroalimentación del usuario funcional

Solicitud 153379

Asunto	Tiempo total de actividad
SOLICITUD TI Solicitud de informe de auditoría de trazabilidad del formulario de la propuesta con código SIGP 109715	00:09:47
Descripción	
<p>Estimado ing. Leonardo:</p> <p>Para gestionar la auditoría de este caso: "Solicitud de informe de auditoría de trazabilidad del formulario de la propuesta con código SIGP 109715" (adjunto memorando)</p>	

Cordialmente,

Nidia Janneth León Cabeza

Contratista

Oficina de Tecnologías y Sistemas de Información

njleon@minciencias.gov.co <njleon@minciencias.gov.co>

Teléfono: 601 6258480 Ext. 3504

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
20/02/2025 08:20:20	26/02/2025 10:01:29	25/02/2025 11:01:29	26/02/2025 10:01:29

Acción Realizada 153379

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos de la convocatoria con **ID 109715**
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solicitud 153656

Asunto	Tiempo total de actividad
SOLICITUD TI Fwd: Solicitud OTSI: reporte fallas SIGP convocatoria 962 SENAINNOVA	00:22:21
Descripción	
<p>Buenas tardes Ing. Leonardo:</p> <p>Por favor me apoya revisando la situación que escalan en este correo.</p> <p>Gracias.</p>	

Cordialmente,

Nidia Janneth León Cabeza

Contratista

Oficina de Tecnologías y Sistemas de Información

njleon@minciencias.gov.co <njleon@minciencias.gov.co>

Teléfono: 601 6258480 Ext. 3504

Av. Calle 26 No. 57 - 83

Torre 8, pisos 2 al 6

Bogotá, Colombia

www.minciencias.gov.co

----- Forwarded message -----

De: Beatriz Helena Robayo Castro <bhrobayo@minciencias.gov.co>

Date: mar, 25 feb 2025 a las 11:41

Subject: Solicitud OTSI: reporte fallas SIGP convocatoria 962 SENAINNOVA

To: César Fabián Gómez Vega <cfgomez@minciencias.gov.co>

Cc: Nidia Janneth León Cabeza <njleon@minciencias.gov.co>, Andrea Carolina Pachón Garzon <acpachon@minciencias.gov.co>

Apreciado Director César,

En el marco de la oferta institucional de mecanismos del Ministerio, actualmente se encuentra abierta la convocatoria 962-2024 SENAINNOVA y se dispuso en los términos de referencia diseñados por la DDTI que las propuestas se presenten a través del SIGP. Así mismo, en el procedimiento de inscripción se indicó a los proponentes que en caso de que se presenten errores en el aplicativo SIGP durante el proceso de inscripción y envío del proyecto o del cargue de la documentación soporte, se remitiera una PQRS con las evidencias y el asunto "*Falla carga SIGP – Convocatoria SENAINNOVA 2024".*

Considerando que durante el inicio de año se reportaron algunas fallas e intermitencia en el acceso a SIGP, tanto para el registro de entidades como de propuestas, solicitamos su apoyo para conocer cuántas PQRS han recibido y que reporten fallas en SIGP para la convocatoria 962. Así mismo, si se han identificado factores de riesgo tecnológico en el proceso de inscripción a través de SIGP o situaciones particulares que requieran de desarrollos adicionales en la plataforma o casos que aún no se han podido resolver frente a lo informado por los posibles proponentes.

Aunado a lo anterior y considerando que desde la DDTI no se tiene autorizado el acceso en SIGP a esta convocatoria, solicitamos su apoyo para conocer cuántas propuestas han sido radicadas y cuántas se encuentran en diligenciamiento.

Agradecemos su apoyo para remitir esta información a la mayor brevedad posible para identificar posibles acciones a tomar en el seguimiento del mecanismo.

Atentamente,

--

Beatriz Helena Robayo Castro

Gestora en Ciencia y Tecnología

Dirección de Ciencia

bhrobayo@minciencias.gov.co <bhrobayo@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

<p><https://storage.googleapis.com/efor-static/CLCS/AVISO%20DE%20CONFIDENCIALIDAD%20-%20DISCLAIMER.pdf></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
25/02/2025 12:02:40	27/02/2025 10:28:29	26/02/2025 11:28:29	27/02/2025 10:28:29

Acción Realizada 153656

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

5. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos de la convocatoria con **ID 11709**, filtrando aquellos cuyo estado sea **"DILIGENCIANDO"** o **"RADICADO"**.
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

6. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

7. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

8. Entrega del Reporte al Usuario Funcional

- Se notificó la disponibilidad del reporte al usuario funcional.
- El archivo fue enviado a través del **Jefe de la OTSI**, asegurando la correcta entrega y comunicación.

Solicitud 153657

Asunto	Tiempo total de actividad
Solicitud de datos SIGP	00:17:54
Descripción <p>Por el presente solicito las bases de datos (en formato Excel) de los registros de las solicitudes de actores en la plataforma de SIGP.</p> <p>Quedo atento a sus comentarios.</p> <p>Cordialmente,</p> <p>*Alexandra Orjuela Ruiz* Contratista Dirección de Ciencia *aorjuela@minciencias.gov.co <aorjuela@minciencias.gov.co> Teléfono: 601 6258480 Ext. 4516 Av. Calle 26 No. 57 - 83</p>	

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
25/02/2025 12:03:25	28/02/2025 10:33:20	27/02/2025 11:33:20	28/02/2025 10:33:20

Acción Realizada 153657

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer la información solicitada de los registros a día de 27/02/2025 de las solicitudes de actores (dependientes, independientes y unidades) en la plataforma de SIGP .
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.

- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solicitud 153701

Asunto		Tiempo total de actividad	
Reporte Final SIGP		00:23:33	
Descripción			
Reciba un cordial saludo. Debido a que nos encontramos en la consolidación de información del proceso de evaluación de la Convocatoria SGR No. 39 de 2024, para la entrega a la Secretaría Técnica del OCAD para la elaboración del Banco Final de Propuestas Elegibles, requerimos del apoyo de la OTSI en la generación de un reporte final del proceso en SIGP.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
25/02/2025 19:27:39	27/02/2025 14:01:54	26/02/2025 15:01:54	27/02/2025 14:01:54

Acción Realizada 153701

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñan dos scripts SQL que permite extraer las evaluaciones de proyectos de la convocatoria SGR No. 39 de 2024.
 - Reporte de evaluación por CRITERIO de una convocatoria.
 - Reporte de evaluación por EVALUADOR de una convocatoria.
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó la disponibilidad del reporte al usuario funcional.
- El archivo fue enviado a través del **Jefe de la OTSI**, asegurando la correcta entrega y comunicación.

Solicitud 153944

Asunto	Tiempo total de actividad

SOLICITUD TI Fwd: 953 Solicitud de información - registros OTSI de la convocatoria.	00:10:27
Descripción	
<p>Para generar reporte.</p> <p>Cordialmente,</p> <p>*Nidia Janneth León Cabeza*</p> <p>Contratista</p> <p>Oficina de Tecnologías y Sistemas de Información</p> <p>*njleon@minciencias.gov.co <njleon@minciencias.gov.co>*</p> <p>Teléfono: 601 6258480 Ext. 3504</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>----- Forwarded message -----</p> <p>De: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co></p> <p>Date: vie, 28 feb 2025 a las 16:30</p> <p>Subject: Fwd: 953 Solicitud de información - registros OTSI de la convocatoria.</p> <p>To: Nidia Janneth León Cabeza <njleon@minciencias.gov.co>, <lasb@softwarebuilders.co></p> <p>Buenas tardes apreciados ingenieros atentamente solicito su colaboración para la Dirección de Capacidades y Apropiación del Conocimiento elaborar la respuesta solicitada y verificar si la convocatoria está asociada a la Dirección de Capacidades y Apropiación del Conocimiento</p> <p>Cordial Saludo</p>	

----- Forwarded message -----

De: Ángela Vanessa Melo Gámez <avmelo@minciencias.gov.co>

Date: vie, 28 feb 2025 a las 12:07

Subject: 953 | Solicitud de información - registros OTSI de la convocatoria.

To: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>

Cc: Alejandra Margarita Acosta Villamil <amacosta@minciencias.gov.co>

Estimado Harold, recibe un caluroso saludo

Agradezco su apoyo y buenos oficios, con la emisión de un reporte de la plataforma del SIGP, que nos dé información sobre cuántas organizaciones (entidades) se han pre-registrado hasta la fecha de la convocatoria 953 para el fortalecimiento de capacidades científicas en formas organizativas de comunidades negras en Caquetá, Chocó, Magdalena, Córdoba y Bogotá D.C. E igualmente cuántas han terminado el proceso de inscripción del proyecto; lo anterior, para el debido seguimiento por parte de esta Dirección.

Muchas gracias, quedo atenta

Cordialmente,

Ángela Vanessa Melo Gámez

Directora Técnica

Dirección de Capacidades y Apropiación del Conocimiento

avmelo@minciencias.gov.co <avmelo@minciencias.gov.co>

Teléfono: 601 6258480 Ext. Sin Información

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>>

--

Harold Alexander Méndez Guevara

Jefe Oficina de Tecnologías y Sistemas de Información

Oficina de Tecnologías y Sistemas de Información

hamendez@minciencias.gov.co <hamendez@minciencias.gov.co>

Teléfono: 601 6258480 Ext. 3500

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>>

--

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**

03/03/2025 08:48:33	04/03/2025 16:48:01	04/03/2025 08:48:01	04/03/2025 16:48:01
------------------------	------------------------	---------------------	------------------------

Acción Realizada 153944

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos de la convocatoria 953 para el fortalecimiento de capacidades científicas en formas organizativas de comunidades negras en Caquetá, Chocó, Magdalena, Córdoba y Bogotá D.C. E igualmente cuántas han terminado el proceso de inscripción del proyecto
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solicitud 154158

Asunto		Tiempo total de actividad	
SOLICITUD DE SERVICIO OTIC		00:04:19	
Descripción			
<p>Para generar reporte SGP. (adjunto correo)</p> <p>Se solicita reporte de evaluación por criterio para las convocatorias del SGR desde la convocatoria 013 a la 033, dicho reporte debe ser enviado en formato Excel en un archivo independiente por convocatoria, cada reporte debe tener los siguientes campos:</p> <p>Cordialmente,</p> <p>Nidia Janneth León Cabeza Contratista Oficina de Tecnologías y Sistemas de Información</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
05/03/2025 15:35:01	07/03/2025 16:00:00	06/03/2025 17:17:07	07/03/2025 16:00:00

Acción Realizada 154158

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer la evaluación de proyectos por criterio para las convocatorias del SGR desde la convocatoria 013 a la 033
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó la disponibilidad del reporte al usuario funcional.
- El archivo fue enviado al **Jefe de la OTSI**, asegurando la correcta entrega y comunicación.

Solicitud 154183

Asunto	Tiempo total de actividad
Convocatoria 962 solicitud reporte propuestas	00:13:40
Descripción	
Cordial saludo,	

Atentamente solicitamos desde la DDTI la generación de un reporte del número de propuestas radicadas y en diligenciamiento de la convocatoria 962-2024 SENAINNOVA.

Agradecemos remitir los archivos soporte de los cuales se genera el dato para cada estado de la propuesta.

Muchas gracias,

Profesional Especializado

Andrea Carolina Pachón Garzón

acpachon@minciencias.gov.co

Teléfono: 6016258480 Ext. 7307

AV. Calle 26 No. 57-83 Torre 8. Piso 2 al 6

Bogotá, Colombia

www.minciencias.gov.co <<http://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
06/03/2025 10:03:28	07/03/2025 16:00:00	06/03/2025 17:58:04	07/03/2025 16:00:00

Acción Realizada 154183

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos de la convocatoria con **ID 11709**, filtrando aquellos cuyo estado sea **"DILIGENCIANDO"** o **"RADICADO"**.
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solicitud 154255

Asunto	Tiempo total de actividad
Reportes SIGP	00:07:47
Descripción	
<p>Asunto: Solicitud de reporte general – Convocatoria 962</p> <p>Cordial saludo, equipo OTSI,</p> <p>Agradecemos su gestión y colaboración para generar el reporte general de la</p>	

convocatoria 962 en la plataforma SIGP, con fecha de cierre hoy, viernes, a las 4:00 p. m. Agradecemos contar con este insumo el día de hoy para iniciar su proceso en la Dirección de Gestión de Recursos.

Solicitamos el reporte general y el reporte de presupuesto de las entidades participantes.

Muchas gracias.

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
07/03/2025 12:29:57	07/03/2025 18:40:06	07/03/2025 17:07:28	07/03/2025 17:52:55

Acción Realizada 154255

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los presupuestos generales y de contrapartidas por rol entidad de los proyectos por convocatoria (**ID 11709**)
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.

- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solicitud 154276

Asunto	Tiempo total de actividad
convocatoria 962 solicitud reporte propuestas al cierre	00:18:25
Descripción Cordial saludo, Atentamente solicitamos desde la DDTI la generación de un reporte del número de propuestas radicadas y en diligenciamiento de la convocatoria 962-2024 SENAINNOVA con corte a las 4 pm del 7 de marzo de 2025. Agradecemos remitir los archivos soporte de los cuales se genera el dato para cada estado de la propuesta. Muchas gracias, Profesional Especializado *Andrea Carolina Pachón Garzón* *acpachon@minciencias.gov.co* Teléfono: 6016258480 Ext. 7307 AV. Calle 26 No. 57-83 Torre 8. Piso 2 al 6 Bogotá, Colombia *www.minciencias.gov.co < http://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
07/03/2025 15:43:00	11/03/2025 14:49:28	10/03/2025 15:49:28	11/03/2025 14:49:28

Acción Realizada 154276

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

5. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos de la convocatoria con **ID 11709**, filtrando aquellos cuyo estado sea **"DILIGENCIANDO"** o **"RADICADO"**.
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

6. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

7. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

8. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solicitud 154339

Asunto	Tiempo total de actividad
Reporte general de la convocatoria 962	00:20:43
Descripción	
<p>Cordial saludo, Leonardo Suárez</p> <p>Agradezco de su colaboración para generar un reporte general de la convocatoria 962 del aplicativo SIGP, los reportes emitidos solo hacen referencia a la contrapartida, (de carácter urgente)</p> <p>En el reporte general se debe incluir las siguientes descripciones:</p> <p>C_CONVOCATORIA_ID C_FUENTE_RECURSOS C_MECANISMO C_NOMBRE C_CODIGO C_AÑO C_FECHA_APERTURA C_FECHA_CIERRE C_ESTADO_CONVOCATORIA C_DEPENDENCIA_MINCIENCIAS C_PROGRAMA_MINCIENCIAS P_PROYECTO_ID P_CODIGO_PROYECTO P_TITULO P_TIPO_PROPOSTA</p>	

P_TIPO_FINANCIACION
P_APROBADO
P_OBSERVACIONES_REGISTRO
P_CUMPLE_REQUISITOS
P_TIPO_PROYECTO
P_ESTADO_PROYECTO
P_AREA ESTRATEGICA
P_DESAFIO
P_RETO
P_RESUMEN_ANTEPROYECTO
P_AREA_TEMATICA
P_AREA_CIENCIA
P_LINEA_TEMATICA
P_SUBLINEA_TEMATICA
P_ANTEPROYECTO
P_TIPOS_SERVICIOS
P_APUESTAS_PRODUCTIVAS
P_DEPTOS_SOLICITAN_RECURSOS
P_LUGAR_EJECUCION_PROYECTO
P_DURACION_PROYECTO_MESES
P_FECHA_CREACION
P_FECHA_RADICACION
P_FECHA_REVISION
P_FECHA_APROBACION
P_AÑO_APROBACION
P_FECHA_FINALIZACION
P_PALABRAS_CLAVES
P_GRUPOS_CLASIFICACION
P_MONTO_SOLICITADO
P_MONTO_CONTRAPARTIDA_EFEC
P_MONTO_CONTRAPARTIDA_ESP
P_CONTRAPARTIDA_TOTAL
P_MONTO_TOTAL
P_MONTO_FINANCIADO_APROB
P_MNT_CONTRAPARTIDA_EFEC_APROB
P_MNT_CONTRAPARTIDA_ESP_APROB

P_CONTRAPARTIDA_TOTAL_APROB
P_MONTO_TOTAL_APROB
E_NOMBRE_ENTIDAD
E_NIT
E_DIRECCION_ELECTRONICA
E_TAMANO_EMPRESA
E_SECTOR_ACONOMICO
E_TIPO_ENTIDAD
E_PAIS
E_DEPARTAMENTO
E_CIUDDAD
E_NOMBRE_CIUDDAD
E_REPRESENTANTE_LEGAL
E_DIRECCION
E_TELEFONO
E_PRIMER_CONTACTO
E_CARGO_PRIMER_CONTACTO
E_TELEFONO_PRIMER_CONTACTO
E_CORREO_PRIMER_CONTACTO
E_SEGUNDO_CONTACTO
E_CARGO_SEGUNDO_CONTACTO
E_TELEFONO_SEGUNDO_CONTACTO
E_CORREO_SEGUNDO_CONTACTO
I_COD_RH_INV_PPAL
I_INVESTIGADOR_PRINCIPAL
I_IDENTIFICACION_INVPPAL
I_TIPO_IDENTIFICACION
I_CORREO_INVPPAL
I_CATEGORIA
I_CARGO
CATEGORIA

**Fecha/hora de
apertura**

**Última
modificación**

**Fecha/hora de
resolución**

**Fecha/hora de
cierre**

10/03/2025 10:15:15	11/03/2025 14:48:27	10/03/2025 15:48:26	11/03/2025 14:48:27
------------------------	------------------------	------------------------	------------------------

Acción Realizada 154339

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos de la convocatoria con **ID 11709**, filtrando aquellos cuyo estado **"RADICADO"**.
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solicitud 154351

Asunto		Tiempo total de actividad	
convocatoria 953		00:10:52	
Descripción			
emisión de un reporte de la plataforma del SIGP, que nos dé información sobre cuántas organizaciones (entidades) se han pre-registrado hasta la fecha de la convocatoria 953 para el fortalecimiento de capacidades científicas en formas organizativas de comunidades negras en Caquetá, Chocó, Magdalena, Córdoba y Bogotá D.C. E igualmente cuántas han terminado el proceso de inscripción del proyecto, así como en qué departamentos y municipios. Lo anterior, para el debido seguimiento por parte de esta Dirección.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
10/03/2025 11:25:19	12/03/2025 14:34:26	11/03/2025 15:34:26	12/03/2025 14:34:26

Acción Realizada 154351

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos de la convocatoria con **ID 11713**, filtrando aquellos cuyo estado sea **"DILIGENCIANDO"** o **"RADICADO"**.
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solicitud 154464

Asunto		Tiempo total de actividad	
Reporte convocatoria 963		00:11:21	
Descripción			
Cordial saludo. De manera atenta solicitamos del apoyo por parte de la OTSI en informarnos el número de propuestas registradas en el SIGP y el esatdo de las mismas, frente a la Convocatoria No 963 de 2025 _Orquídeas: Mujeres en Inteligencia Artificial, Ciencia y Tecnologías Cúanticas.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
11/03/2025 15:17:40	13/03/2025 09:48:50	12/03/2025 10:48:50	13/03/2025 09:48:50

Acción Realizada 154464

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos de la convocatoria con **ID 11714**, filtrando aquellos cuyo estado sea "**DILIGENCIANDO**" o "**RADICADO**".
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solicitud 154607

Asunto	Tiempo total de actividad
Convocatoria 953	00:21:12
Descripción	

Teniendo en cuenta que el cierre se acerca, y para poder reforzar la difusión de la convocatoria, quisiéramos solicitar su apoyo con la emisión de un reporte de la plataforma del SIGP, que nos dé información sobre cuántas organizaciones (entidades) se han pre-registrado hasta la fecha de la convocatoria 953 para el fortalecimiento de capacidades científicas en formas organizativas de comunidades negras en Caquetá, Chocó, Magdalena, Córdoba y Bogotá D.C. E igualmente cuántas han terminado el proceso de inscripción del proyecto, así como en qué departamentos y municipios. Lo anterior, para el debido seguimiento por parte de esta Dirección.

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
13/03/2025 14:24:31	14/03/2025 10:34:21	14/03/2025 10:34:20	

Acción Realizada 154607

Para atender la solicitud de generación de reportes, se llevó a cabo el siguiente procedimiento:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos de la convocatoria con **ID 11713**, filtrando aquellos cuyo estado sea **"DILIGENCIANDO"** o **"RADICADO"**.
- Se verificó que la consulta retornara la información requerida y optimizara el rendimiento en la base de datos SIGP.

2. Envío del Script a la DBA de Minciencias

- El script fue remitido a **Erika González**, DBA de Minciencias, para su ejecución en la base de datos SIGP.
- Se realizó el seguimiento correspondiente para garantizar la correcta ejecución y obtención de los datos.

3. Recepción del Documento y Notificación

- Se recibió el archivo **Excel** con los datos extraídos según la consulta realizada.
- Se revisó la información para verificar su integridad y consistencia con la solicitud inicial.

4. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional.

Solución de Errores

Se solucionaron errores reportados por los usuarios en la plataforma:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
153408	Inconvenientes para descargar anexos	Solucionado	-	[Ver detalle]
153615	Error Acceso Convocatoria 953	Solucionado	-	[Ver detalle]
153971	Error Descarga Certificado Evaluador	Solucionado	-	[Ver detalle]
154490	Error Actores Crear Solicitud Unidad	Pendiente Usuario	Aclaraciones	[Ver detalle]

Solicitud 153408

Asunto	Tiempo total de actividad
Reporte SIGP	00:06:52
Descripción	
Solicitud acceso a anexos de la convocatoria 951	

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
20/02/2025 11:31:52	27/02/2025 10:38:30	26/02/2025 11:38:30	27/02/2025 10:38:30

Acción Realizada 153408

Tras el reporte de un inconveniente al momento de descargar los documentos adjuntos, se realizaron las siguientes acciones:

1. Análisis del Problema

- Se revisó el caso presentado y se identificó que el error estaba relacionado con **falta de privilegios** en el sistema.
- Se verificaron los permisos asignados al usuario afectado y se confirmó que no contaba con los accesos adecuados para la descarga de archivos.

2. Aplicación de la Solución

- Se procedió a **asignar los permisos necesarios** al usuario para que pudiera acceder y descargar los documentos adjuntos sin restricciones.
- Se validó la configuración en el sistema para asegurar que los permisos fueran correctamente aplicados.

3. Verificación y Comunicación al Usuario

- Se realizó una prueba posterior para garantizar que la descarga de archivos funcionara correctamente.
- Se notificó al usuario sobre la resolución del inconveniente y se le solicitó que intentara nuevamente la descarga para confirmar el acceso.

Solicitud 153615

Asunto	Tiempo total de actividad
SOLICITUD TI Fwd: Solicitud No. Solicitud No. 20250003521R	00:10:51
Descripción	
<p>Ing. Leonardo buenas tardes:</p> <p>Solicito apoyo para validar cuál es el proceder de esta solicitud ya que conozco el procedimiento a realizar según la situación presentada al ciudadano.</p> <p>Cordialmente,</p> <p>Nidia Janneth León Cabeza Contratista Oficina de Tecnologías y Sistemas de Información njleon@minciencias.gov.co Teléfono: 601 6258480 Ext. 3504 Av. Calle 26 No. 57 - 83 Torre 8, pisos 2 al 6 Bogotá, Colombia www.minciencias.gov.co</p> <p>----- Forwarded message -----</p> <p>De: César Fabián Gómez Vega <cfgomez@minciencias.gov.co> Date: mar, 18 feb 2025 a las 17:37 Subject: Solicitud No. Solicitud No. 20250003521R To: Nidia Janneth León Cabeza <njleon@minciencias.gov.co> Cc: Luis Carlos Ovalle Acosta <lcovalle@minciencias.gov.co>, Martha Lucía Quintero García <mlquintero@minciencias.gov.co></p>	

Buenas tardes, Nidia.

Comparto la petición No. Solicitud No. 20250003521R, como fecha Máxima de respuesta está establecida 2025-02-24 23:59:59

Gracias.

36fc0089-5067-425f-b134-e255476b59eb.jpg

César Fabián Gómez Vega
Jefe de Oficina Asesora
Oficina Asesora de Planeación e Innovación Institucional
cfgomez@minciencias.gov.co
Teléfono: 601 6258480 Ext. 3300
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
24/02/2025 17:55:27	25/02/2025 16:00:00	24/02/2025 18:20:53	25/02/2025 16:00:00

Acción Realizada 153615

Tras el reporte de un posible error en el acceso a la convocatoria **"953-2024 CONVOCATORIA PARA COMUNIDADES NEGRAS DE BOGOTÁ, CAQUETÁ, CÓRDOBA, MAGDALENA Y CHOCÓ"**, se realizó la siguiente verificación y respuesta:

1. Análisis del Reporte

- Se identificó que la imagen proporcionada correspondía al **03 de febrero de 2025**, fecha en la cual la convocatoria aún se encontraba en modo de **pruebas**.
- Durante esta fase, el acceso estaba restringido y solo se permitía mediante **invitación especial** a funcionarios internos.

2. Verificación de la Publicación

- Se revisó la trazabilidad de la solicitud **153242 en la mesa de servicios**, confirmando que el usuario funcional dio su visto bueno para la publicación el **18 de febrero de 2025**.
- A partir de esa fecha, la convocatoria se trasladó al **ambiente de producción** y quedó accesible para el público.

3. Corrección y Comunicación al Usuario

- Se informó al usuario que el problema reportado correspondía a una fecha en la que la convocatoria aún estaba en pruebas.
- Se proporcionó el **enlace de acceso oficial en producción**:



<https://plataformasigp.minciencias.gov.co:7003/FormularioProyectos/faces/index.xhtml>

4. Nota Aclaratoria sobre Invitaciones Especiales

- Se explicó que cuando una convocatoria aún no ha sido completamente **parametrizada, validada y aprobada por el usuario funcional**, se configura bajo un esquema de **invitación especial**.
- En este estado, el acceso se concede únicamente a **funcionarios internos** para realizar pruebas.
- Se reiteró que el **link y las credenciales** de acceso en modo de pruebas **NO deben ser compartidos con el público**, ya que son exclusivos para validaciones internas.

Solicitud 153971

Asunto	Tiempo total de actividad
SOLICITUD TI	00:25:39
Descripción	
<p>Buen día</p> <p>Me permito solicitar su colaboración en el aplicativo SIGP ya que no me permite generar el certificado de par evaluador código 112034 cédula 1113659059 Natali López Mejía de la convocatoria 956 aunque ya se haya finalizado la evaluación.</p> <p>[image: image.png]</p> <p>Cordialmente,</p> <p>--</p> <p>*Mar Jenny Romero Acosta*</p> <p>Secretario Ejecutivo</p> <p>Dirección De Gestión De Recursos Para La (CTel)</p> <p>*mjromero@minciencias.gov.co <mjromero@minciencias.gov.co>*</p> <p>Teléfono: 601 6258480 Ext. 4101</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>	

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
03/03/2025 11:31:40	06/03/2025 10:21:25	05/03/2025 11:21:25	06/03/2025 10:21:25

Acción Realizada 153971

Se reportó un inconveniente al generar el certificado del evaluador **Natali López Mejía**. Tras el análisis del caso, se identificó que el problema se originó porque la evaluación del proyecto no había sido cerrada correctamente.

Aunque la validación inicial detectó que la evaluación seguía abierta, el sistema no detuvo el flujo del proceso y permitió que el usuario avanzara a la **etapa de decisión**, lo que generó una inconsistencia. Se realizaron las siguientes acciones:

1. Análisis y Diagnóstico

- Se revisó el comportamiento del sistema al momento de generar el certificado.
- Se identificó que la validación de cierre de evaluación estaba funcionando parcialmente: detectaba que la evaluación no estaba cerrada, pero no detenía el flujo adecuadamente.
- Se confirmó que esta falla permitía que el usuario accediera a la ventana de diálogo para la etapa de decisión sin haber completado el cierre de la evaluación.

2. Ajuste en la Lógica de Validación

- Se modificó la validación en el sistema para que, cuando la evaluación no esté cerrada, **se detenga completamente el flujo** y no se permita avanzar hasta que el usuario realice el cierre correspondiente.

- Se implementaron pruebas para asegurar que el sistema reaccione correctamente ante este escenario.

3. Verificación y Comunicación al Usuario

- Se realizaron pruebas para validar que el problema había sido solucionado y que la nueva validación impedía inconsistencias en el flujo.
- Se notificó a los usuarios sobre la corrección del error, asegurando que ahora el sistema exige el cierre de evaluación antes de permitir avanzar a la etapa de decisión.

Solicitud 154490

Asunto	Tiempo total de actividad
SOLICITUD TI	00:06:58
Descripción <p>Buen día ing Leonardo:</p> <p>solicito apoyo para validar el inconveniente presentado en la aplicación de Reconocimiento de Actores para dar la respuesta al ciudadano con la PQRDS No. 220250005974R.</p> <p>Adjunto comunicación.</p> <p>Cordialmente, 0250005974R</p> <p> *Nidia Janneth León Cabeza*</p> <p>Contratista</p> <p>Oficina de Tecnologías y Sistemas de Información</p> <p>*njleon@minciencias.gov.co <njleon@minciencias.gov.co>*</p>	

Teléfono: 601 6258480 Ext. 3504
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
11/03/2025 22:11:12	13/03/2025 12:12:04		

Acción Realizada 154490

Se recibió una solicitud de validación sobre un inconveniente en la aplicación de **Reconocimiento de Actores**. Sin embargo, no se cuenta con suficiente información para revisar el caso ni para replicar el error reportado. Se realizaron las siguientes acciones:

1. Análisis Preliminar

- Se revisaron los reportes previos en el sistema para identificar si había registros similares al error reportado.
- Se verificó si existían logs o registros de errores en la aplicación relacionados con la solicitud.
- No se encontró información suficiente para determinar la causa del problema.

2. Solicitud de Información Adicional al Usuario

- Se contactó al usuario para solicitar detalles que permitan analizar el problema con mayor precisión.
- Se solicitó información clave como:
 - **Entidad y unidad correspondiente** que presenta el problema.
 - **Mensajes de error** recibidos en pantalla o en la consola.
 - **Capturas de pantalla** para visualizar mejor el error.
 - **Pasos para reproducir el problema**, con el fin de replicarlo en un entorno de prueba.

3. Estado Actual

- En espera de la retroalimentación del usuario con la información solicitada.
- Una vez se reciba la información, se procederá con la validación y diagnóstico del problema.

Actualización de Datos

Se realizó la siguiente actualización en la plataforma SIGP:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
154254	Cambio Nombre OTRI de Universidad de la Sabana	Solucionado	-	[Ver detalle]

Solicitud 154254

Descripción
Buenas tardes:

Con el fin de atender la solicitud trasladada por la Universidad de la Sabana en cuanto a la actualización del nombre de su OTRI (Oficina de Transferencia de Resultados de Investigación) en la plataforma SIGP, amablemente se requiere su apoyo para poder realizar dicha modificación.

En este sentido, en el SIGP la Universidad indica que el nombre con el cual se encuentra registrada su OTRI es "Unisabana Hub" y se requiere realizar su cambio por el de "Oficina de Transferencia de Conocimiento", para lo cual se adjunta la siguiente documentación:

- Certificación del cambio de nombre firmado por el representante legal.
- Certificado laboral vigente de la Jefe de la oficina de transferencia de conocimiento.
- Certificado de existencia y representación expedido por el Ministerio de Educación.
- Acta de nombramiento del representante legal, Sr. Juan Fernando Córdoba Marentes.
- Copia de la cédula de ciudadanía del representante legal.

De otra parte es indispensable resaltar que dicha OTRI sigue siendo dependiendo de la Universidad de la Sabana y por lo tanto conserva el NIT con el cual se realizó su registro y reconocimiento como actor del SNCTI.

Acción Realizada:

El usuario solicitó la actualización del nombre de la OTRI (Oficina de Transferencia de Resultados de Investigación) de la **Universidad de La Sabana** en la plataforma **SIGP**, debido a un cambio en su denominación oficial.

Acciones Tomadas:

1. Recepción y Análisis de la Solicitud

- Se revisó la solicitud para confirmar los detalles del cambio solicitado.

Inicio

Administración

Configuración

Reconocimiento

Configurar Vigencias

Solicitudes Reconocimiento

Solicitudes por Entidad

Entidades Evaluadoras

Evaluar

Consultas Reportes

Beneficios Tributarios

Proyectos/Programas

Inversión Patentes

Ayuda

Búsqueda por Entidad

Asignar Evaluadores

Entidad: UNIVERSIDAD DE LA SABANA

Q Filtrar

Cancelar

A continuación se muestran las solicitudes de reconocimiento según el Tipo de Reconocimiento al que pertenecen

Unidades I+D+i Centros Investigación Centros Desarrollo Centros Innovación Centros Ciencia **OTRI** IEBT

Parques - PCTI

Fecha Radicación	Número Solicitud	Tipo Reconocimiento	Empresa Altamente Innovadora (EAI)	Estado	Entidad	Unidad/Centro	Opciones
11/03/2025	9376	Oficina de Transferencia de Resultados de Investigación Renovación	No Aplica	RADICADA	UNIVERSIDAD DE LA SABANA	UNISABANA HUB	✓ 🔍 📄
28/09/2021	7476	Oficina de Transferencia de Resultados de Investigación	No Aplica	RECONOCIMIENTO VENCIDO	UNIVERSIDAD DE LA SABANA	UNISABANA HUB	✓ ⚙️ 🔍 📄

TOTAL DE REGISTROS: 2 << < 1 > >> 10

- Se verificó en la base de datos el nombre actual registrado para la OTRI de la Universidad de La Sabana.

```

select u.*, rowid
from cen_usuarios u
where u.usuario_id = 11620 -- solicitud Renovacion
or u.usuario_id = 9307
--and u.nombre_centro = 'Unisabana Hub'

select *
from cen_solicitudes_reco s
where s.solicitud_id = 9376 -- solicitud Renovacion
OR s.solicitud_id = 7476;

```

Resultado de la Consulta

Todas las Filas Recuperadas: 2 en 0,009 segundos

	USUARIO_ID	USUARIO	CLAVE	TIPO_USUARIO	UNIDAD_ID	CENTRO_ID	ENTIDAD_ID	NOMBRE_CENTRO	EMAIL	TIPO_RECONOCIMIENTO	F...	U...	ACTIVO	ROWID
1	9307	SRA74762021	f1b012531...	D	(null)	(null)	11116	UNISABANA HUB	otriunisa@unisabana.edu.co	46358	...	(...)	N	AAODERAAAGADLaaAAQ
2	11620	SRA74762021	ae86d2349...	D	(null)	(null)	11116	UNISABANA HUB	otri@unisabana.edu.co	46358	...	(...)	S	AAODERAAAGADYJNAAD

2. Ejecución del Cambio en la Base de Datos

- Se realizó la actualización del nombre en la base de datos, modificando:
 - Nombre anterior:** "UNISABANA HUB"
 - Nombre actualizado:** "Oficina de Transferencia de Conocimiento"

```

UPDATE cen_usuarios u
SET u.nombre_centro = 'Oficina de Transferencia de Conocimiento'
WHERE u.usuario_id = 11620;

```

- Se verificó que la modificación se reflejara correctamente en la base de datos

```
-- Script de actualización nombre OTRI "Unisabana Hub" a "Oficina de Transferencia de Conocimiento",
update cen_usuarios u
set u.nombre_centro = 'Oficina de Transferencia de Conocimiento'
where u.usuario_id = 11620;
```

Ida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 2 en 0,003 segundos

USUARIO_ID	USUARIO	CLAVE	TIPO_USUARIO	UNIDAD_ID	CENTRO_ID	ENTIDAD_ID	NOMBRE_CENTRO	EMAIL	TIPO_RECONOCIMIENTO	F...	U...	ACTIVO
9307	SR274762021	f1b012531...	D	(null)	(null)	11116	UNISABANA HUB	otriunisa@unisabana.edu.co	46358	...	...	N
11620	SR274762021	ee56d23d5...	D	(null)	(null)	11116	Oficina de Transferencia de Conocimiento	otri@unisabana.edu.co	46358	...	...	S

- Se verificó que la modificación se reflejara correctamente en la plataforma SIGP.

Inicio

Administración

Configuración

Reconocimiento

Configurar Vigencias

Solicitudes Reconocimiento

Solicitudes por Entidad

Entidades Evaluadoras

Evaluar

Consultas Reportes

Beneficios Tributarios

Proyectos/Programas

Inversión Patentes

Ayuda

Entidad: UNIVERSIDAD DE LA SABANA

Filtrar Cancelar

A continuación se muestran las solicitudes de reconocimiento según el Tipo de Reconocimiento al que pertenecen

Unidades I+D+i Centros Investigación Centros Desarrollo Centros Innovación Centros Ciencia **OTRI** IEBT

Parques - PCTI

Fecha Radicación	Número Solicitud	Tipo Reconocimiento	Empresa Altamente Innovadora (EAI)	Estado	Entidad	Unidad/Centro	Opciones
11/03/2025	9376	Oficina de Transferencia de Resultados de Investigación <i>Renovación</i>	No Aplica	RADICADA	UNIVERSIDAD DE LA SABANA	Oficina de Transferencia de Conocimiento	✓ 🔍 📄
28/09/2021	7476	Oficina de Transferencia de Resultados de Investigación	No Aplica	RECONOCIMIENTO VENCIDO	UNIVERSIDAD DE LA SABANA	UNISABANA HUB	✓ ⚙️ 🔍 📄

TOTAL DE REGISTROS: 2 << 1 >> 10

3. Notificación al Usuario

- Se envió un mensaje de confirmación al usuario notificando que el cambio fue realizado exitosamente.
- Se adjuntaron **soportes** con capturas de pantalla del sistema, mostrando el estado **ANTES y DESPUÉS** de la actualización.
- Se solicitó al usuario una **validación final** para confirmar que la actualización se reflejara correctamente en la plataforma.

Requerimientos

Se recibieron solicitudes de implementación de nuevas funcionalidades en el sistema:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
153264	Anonimizar reportes de evaluación	Pendiente Proveedor	Documentación	[Ver detalle]
154299	Desarrollo Nuevo Informe Evaluación Tercer Evaluador	Pendiente Proveedor	Documentación	[Ver detalle]

Solicitud 153264

Asunto	Tiempo total de actividad
Requerimiento de Anonimización de reportes del tercer evaluador	00:27:11
Descripción	
<p>Se solicita a la Oficina de Tecnologías de la Información y Sistemas (OTSI) la generación de un nuevo informe para los proponentes cuando se asigna un tercer evaluador en las respectivas convocatorias que cursan en el ministerio. Actualmente, el informe solo presenta la menor dispersión entre las evaluaciones, por lo que es necesario que el nuevo informe incluya explícitamente las tres evaluaciones completas realizadas por los evaluadores asignados, sin realizar promedios ni omitir ninguna de ellas.</p> <p>El informe debe ser de carácter cualitativo, presentando de manera estructurada los comentarios y observaciones de cada evaluador organizados según los criterios y subcriterios establecidos en los términos de referencia de la convocatoria. Finalmente, el informe debe contener una anotación que remita expresamente a los términos de referencia de la convocatoria respectiva, para que los proponentes puedan revisar las condiciones aplicables en los casos en que haya un segundo evaluador. Con este nuevo informe se busca garantizar mayor transparencia y claridad en la retroalimentación a los proponentes, asegurando que puedan comprender la evaluación de su propuesta de manera integral.</p>	

esta solicitud debe ser tramitada a Software Builder cuando nos generan este nuevo reporte

Cordialmente,

Oficina de Tecnologías y Sistemas de Información
Ministerio de Ciencia, Tecnología e Innovación

--

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
18/02/2025 16:03:28	17/03/2025 16:44:57		

Acción Realizada 153264

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.

Estado Actual:

- ☒ Estimación de esfuerzos completada.
- ☒ Desarrollo implementado.
- ☒ Despliegue en pruebas realizado.
- ☐ Pendiente de aprobación para producción.
- ☐ Despliegue final en producción.

Solicitud 154299

Asunto	Tiempo total de actividad
SOLICITUD TI Requerimiento Desarrollo nuevo Informe Evaluación	00:02:27
Descripción	
<p>Buen día:</p> <p>Para formalizar el Requerimiento Desarrollo nuevo Informe Evaluación.</p> <p>Cordialmente,</p> <p>*Nidia Janneth León Cabeza*</p> <p>Contratista</p> <p>Oficina de Tecnologías y Sistemas de Información</p> <p>*njleon@minciencias.gov.co <njleon@minciencias.gov.co>*</p> <p>Teléfono: 601 6258480 Ext. 3504</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>----- Forwarded message -----</p> <p>De: Daniel Rodrigo Tarquino Mosquera <drtarquino@minciencias.gov.co></p> <p>Date: jue, 6 mar 2025 a las 11:41</p> <p>Subject: Formatos de Requerimientos</p> <p>To: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co></p> <p>Cc: Nidia Janneth León Cabeza <njleon@minciencias.gov.co></p> <p>Buenos días,</p>	

Remito los formatos solicitados .

Gracias por todo su apoyo.

Daniel Rodrigo Tarquino Mosquera

Asesor

VICEMINISTERIO DE TALENTO Y APROPIACIÓN SOCIAL DEL CONOCIMIENTO

drtarquino@minciencias.gov.co <drtarquino@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>>

--

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

08/03/2025 17:08:24	10/03/2025 18:02:57		
------------------------	------------------------	--	--

Acción Realizada 154299

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.

Conclusiones

Durante este período se brindó soporte técnico en la plataforma SIGP, permitiendo la parametrización de convocatorias, 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.

Estado Actual:

- ✓ Estimación de esfuerzos completada.
- ✓ Desarrollo implementado.
- ✓ Despliegue en pruebas realizado.
- ✓ Pendiente de aprobación para producción.
- ✓ Despliegue final en producción.

Firma:

César A. Riaño D.

Ing. César Augusto Riaño Delgado
Gerente Proyecto