

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

SEGUNDO 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 Auditorías	2
Creación pantalla de Detalles Auditorías	41
Creación pantalla de Reporte Control Auditoría.....	68
Creación pantalla de Reporte Auditoría por Campo	101
Auditoría de inicio y cierre de sesión	127
2. IMPLEMENTACIÓN AMBIENTE DE PRUEBAS SIGP, (No. Obligación Contrato: 2.2.1, 2.2.5, 2,2,6)	130
3. 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)	134
Introducción	135
Resumen de Actividades	135
Detalle de Casos Atendidos.....	135
Parametrización de Convocatorias	135
Parametrización de Rúbricas de Evaluación	166
Generación de Reportes.....	168
Solución de Errores	218
Actualización de Datos	230

Requerimientos.....	235
Conclusiones.....	245

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. 154455

Descripción y alcance del Requerimiento

Este requerimiento tiene como objetivo el desarrollo e integración de cuatro nuevas funcionalidades dentro del módulo de administración del Sistema Integrado de Gestión de Proyectos (SIGP). Estas funcionalidades permitirán gestionar y consultar los registros de auditoría del sistema a través de las siguientes secciones:

1. **Auditorías:** Visualización de cada cambio realizado sobre las tablas del sistema, identificando el usuario, la tabla afectada, el terminal, la fecha/hora del evento y el tipo de operación realizada.
2. **Detalle de Auditorías:** Consulta detallada de cada auditoría registrada, permitiendo visualizar los campos modificados con su valor antes y después de la operación.
3. **Reporte Control Auditoría:** Generación de reportes basados en parámetros de búsqueda como usuario, tabla y rango de fechas, que permiten visualizar de forma consolidada la cantidad de registros creados, modificados, consultados, descargados y borrados.
4. **Reporte Auditoría por Campo:** Generación y descarga de reportes personalizados que muestran, por tabla y por campo, los valores modificados según los filtros aplicados por el usuario.
5. **Registro de Accesos:** Registro y consulta de los eventos de acceso al sistema, permitiendo identificar los usuarios que iniciaron o finalizaron sesión, junto con la información relacionada.

El alcance del desarrollo incluye el diseño y desarrollo de las interfaces gráficas, la implementación de funcionalidades para la visualización, filtrado y exportación de registros, 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 Auditorías
- Creación de nueva pantalla de Detalle de Auditorías
- Creación de nueva pantalla de Reporte Control Auditoría
- Creación de nueva pantalla de Reporte Auditoría por Campo
- Implementación de lógica para registrar eventos de inicio y cierre de sesión dentro de la aplicación

Aplicativo de Gestión

Creación pantalla de Auditorías

Función:

Esta pantalla permite la consulta de auditorías registradas en el sistema. Se puede visualizar cada cambio realizado sobre las tablas de la base de datos, identificando el usuario que lo ejecutó, la tabla afectada, la terminal desde la que se realizó la operación, la fecha/hora y el tipo de operación efectuada (creación, modificación, eliminación, etc.).

Principales funcionalidades:

- **Listado de Auditorías:** Se visualizan todas las auditorías registradas en una tabla con opciones de filtrado por usuario, tabla y fecha.
- **Visualización Segura:** Esta pantalla es únicamente de consulta, no permite la creación, edición ni eliminación de registros.

Creación nuevo menú de Auditorías

```
-- funciones (menus)
-- función: Gestión Auditorías (menú)
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', 'GEST6300', 'Gestión Auditorías', 'M', null, null, null, null,
'GEST6000', 'fa fa-arrows-alt', 'HERCULES', to_date('01-04-2024 00:00:00', 'dd-mm-yyyy
hh24:mi:ss'), 'HERCULES', to_date('01-04-2024 00:00:00', 'dd-mm-yyyy hh24:mi:ss'));

-- función: Auditorías
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', 'GEST6310', 'Auditorías', 'F', null, null,
'/app/administracion/auditorias/auditorias.xhtml', null, 'GEST6100', 'fa fa-bell-o',
'HERCULES', to_date('01-04-2024 00:00:00', 'dd-mm-yyyy hh24:mi:ss'), 'HERCULES',
to_date('01-04-2024 00:00:00', 'dd-mm-yyyy hh24:mi:ss'));
-- derechos funciones (permisos)
-- derechos función: Gestión Auditorías (menú)
insert into sb_derechos_funciones (DERECHO_FUNCION_ID, APLICACION, FUNCION, GRUPO,
ESTADO, USUARIO_CREADOR, FECHA_CREACION, USUARIO_MODIFICADOR, FECHA_MODIFICACION)
values (null, 'SIGP_WEB', 'GEST6300', 'SGPS ADMINISTRADOR', 'H', 'HERCULES',
to_date('01-04-2024 00:00:00', 'dd-mm-yyyy hh24:mi:ss'), 'HERCULES', to_date('01-04-2024
00:00:00', 'dd-mm-yyyy hh24:mi:ss'));

-- derechos función: Auditorías
insert into sb_derechos_funciones (DERECHO_FUNCION_ID, APLICACION, FUNCION, GRUPO,
ESTADO, USUARIO_CREADOR, FECHA_CREACION, USUARIO_MODIFICADOR, FECHA_MODIFICACION)
values (null, 'SIGP_WEB', 'GEST6310', 'SGPS ADMINISTRADOR', 'H', 'HERCULES',
to_date('01-04-2024 00:00:00', 'dd-mm-yyyy hh24:mi:ss'), 'HERCULES', to_date('01-04-2024
00:00:00', 'dd-mm-yyyy hh24:mi:ss'));
```

*Crear nueva entidad de la tabla **SB_AUDITORIAS** 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.security;

import com.sb.colciencias.gestion.business.domain.Tabla;
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;

/**
 *
 * @author lsuarez
 */
```

```

*/
@Entity
@Table(name = "SB_AUDITORIAS", catalog = "", schema = "HERCULES")
@NamedQueries({
    @NamedQuery(name = "Auditoria.findAll", query = "SELECT a FROM Auditoria a")
    ,@NamedQuery(name = "Auditoria.findByAuditoriaId", query = "SELECT a FROM Auditoria
a WHERE a.auditoriaId = :auditoriaId")
    ,@NamedQuery(name = "Auditoria.findByFechaHora", query = "SELECT a FROM Auditoria a
WHERE a.fechaHora = :fechaHora")
    ,@NamedQuery(name = "Auditoria.findByTerminal", query = "SELECT a FROM Auditoria a
WHERE a.terminal = :terminal")
    ,@NamedQuery(name = "Auditoria.findByOperacion", query = "SELECT a FROM Auditoria a
WHERE a.operacion = :operacion")
    ,@NamedQuery(name = "Auditoria.findByTabla", query = "SELECT a FROM Auditoria a
WHERE a.tabla = :tabla")
    ,@NamedQuery(name = "Auditoria.findByVerificacion", query = "SELECT a FROM Auditoria
a WHERE a.verificacion = :verificacion")
    ,@NamedQuery(name = "Auditoria.findByUsuario", query = "SELECT a FROM Auditoria a
WHERE a.usuario = :usuario")
    ,@NamedQuery(name = "Auditoria.findByAplicacion", query = "SELECT a FROM Auditoria a
WHERE a.aplicacion = :aplicacion"))
public class Auditoria implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"SB_AUDITORIA_ID_SEQ")
    @SequenceGenerator(name = "SB_AUDITORIA_ID_SEQ", sequenceName =
"SB_AUDITORIA_ID_SEQ", allocationSize = 1)
    @Basic(optional = false)
    @NotNull
    @Column(name = "AUDITORIA_ID")
    private Long auditoriaId;
    @Column(name = "FECHA_HORA")
    @Temporal(TemporalType.TIMESTAMP)
    private Date fechaHora;
    @Basic(optional = false)
    @Column(name = "TERMINAL")
    private String terminal;
    @Basic(optional = false)
    @Column(name = "OPERACION")
    private String operacion;
    @Basic(optional = false)
    @Column(name = "VERIFICACION")
    private String verificacion;
    @Column(name = "USUARIO")
    private String usuario;

    //Joins
    @JoinColumn(name = "APLICACION", referencedColumnName = "APLICACION")
    @ManyToOne
    private Aplicacion aplicacion;
    @JoinColumn(name = "TABLA", referencedColumnName = "TABLA")
    @ManyToOne
    private Tabla tabla;

    @Transient
    private Usuario usuarioJoin;

    //Auditoria
    @Transient
    private AppSessionVO appSessionVO;

```

```
public Auditoria() {
}

public Auditoria(Long auditoriaId) {
    this.auditoriaId = auditoriaId;
}

public Auditoria(Long auditoriaId, Date fechaHora, String terminal, String
operacion) {
    this.auditoriaId = auditoriaId;
    this.fechaHora = fechaHora;
    this.terminal = terminal;
    this.operacion = operacion;
}

public Long getAuditoriaId() {
    return auditoriaId;
}

public void setAuditoriaId(Long auditoriaId) {
    this.auditoriaId = auditoriaId;
}

public Date getFechaHora() {
    return fechaHora;
}

public void setFechaHora(Date fechaHora) {
    this.fechaHora = fechaHora;
}

public String getTerminal() {
    return terminal;
}

public void setTerminal(String terminal) {
    this.terminal = terminal;
}

public String getOperacion() {
    return operacion;
}

public void setOperacion(String operacion) {
    this.operacion = operacion;
}

public String getVerificacion() {
    return verificacion;
}

public void setVerificacion(String verificacion) {
    this.verificacion = verificacion;
}

public String getUsuario() {
    return usuario;
}

public void setUsuario(String usuario) {
    this.usuario = usuario;
}
```

```
public Usuario getUsuarioJoin() {
    return usuarioJoin;
}

public void setUsuarioJoin(Usuario usuarioJoin) {
    this.usuarioJoin = usuarioJoin;
}

public Aplicacion getAplicacion() {
    return aplicacion;
}

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

public Tabla getTabla() {
    return tabla;
}

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

@Override
public int hashCode() {
    int hash = 0;
    hash += (auditoriaId != null ? auditoriaId.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 Auditoria)) {
        return false;
    }
    Auditoria other = (Auditoria) object;
    return !((this.auditoriaId == null && other.auditoriaId != null) ||
(this.auditoriaId != null && !this.auditoriaId.equals(other.auditoriaId)));
}

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

public AppSessionVO getAppSessionVO() {
    return appSessionVO;
}

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

}}
```

Crear nuevo DTO representativo de la entidad

```
/*
 * 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.transfer;

import com.sb.colciencias.gestion.business.domain.Tabla;
import com.sb.colciencias.gestion.business.domain.security.*;
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 lsuarez
 */
public class AuditoriaDTO implements Serializable {

    private static final long serialVersionUID = 1L;

    @Bind(name = "auditoriaId")
    private Long auditoriaId;
    @Bind(name = "fechaHora")
    private Date fechaHora;
    @Bind(name = "terminal")
    private String terminal;
    @Bind(name = "operacion")
    private String operacion;
    // @Bind(name = "tabla")
    // private String tabla;
    @Bind(name = "verificacion")
    private String verificacion;
    @Bind(name = "usuario")
    private String usuario;

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

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

    public AuditoriaDTO() {
    }

    public AuditoriaDTO(Long auditoriaId) {
        this.auditoriaId = auditoriaId;
    }

    public AuditoriaDTO(Long auditoriaId, Date fechaHora, String terminal, String
operacion) {
        this.auditoriaId = auditoriaId;
    }
}
```

```
        this.fechaHora = fechaHora;
        this.terminal = terminal;
        this.operacion = operacion;
    }

    public Long getAuditoriaId() {
        return auditoriaId;
    }

    public void setAuditoriaId(Long auditoriaId) {
        this.auditoriaId = auditoriaId;
    }

    public Date getFechaHora() {
        return fechaHora;
    }

    public void setFechaHora(Date fechaHora) {
        this.fechaHora = fechaHora;
    }

    public String getTerminal() {
        return terminal;
    }

    public void setTerminal(String terminal) {
        this.terminal = terminal;
    }

    public String getOperacion() {
        return operacion;
    }

    public void setOperacion(String operacion) {
        this.operacion = operacion;
    }

    public String getVerificacion() {
        return verificacion;
    }

    public void setVerificacion(String verificacion) {
        this.verificacion = verificacion;
    }

    public String getUsuario() {
        return usuario;
    }

    public void setUsuario(String usuario) {
        this.usuario = usuario;
    }

    public UsuarioDTO getUsuarioJoin() {
        return usuarioJoin;
    }

    public void setUsuarioJoin(UsuarioDTO usuarioJoin) {
        this.usuarioJoin = usuarioJoin;
    }

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

```
}

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

public TablaDTO getTabla() {
    return tabla;
}

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

@Override
public int hashCode() {
    int hash = 0;
    hash += (auditoriaId != null ? auditoriaId.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 AuditoriaDTO)) {
        return false;
    }
    AuditoriaDTO other = (AuditoriaDTO) object;
    return !((this.auditoriaId == null && other.auditoriaId != null) ||
        (this.auditoriaId != null && !this.auditoriaId.equals(other.auditoriaId)));
}

@Override
public String toString() {
    return "com.sb.colciencias.gestion.business.transfer.AuditoriaDTO[ auditoriaId="
+ auditoriaId + " ]";
}

public AppSessionVO getAppSessionVO() {
    return appSessionVO;
}

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

La clase **AuditoriaDTO** es un Data Transfer Object (DTO) que se utiliza para transportar datos relacionados con la entidad **Auditoria** 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 (Usuario, Aplicacion, Tabla):** 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 = "auditoriaId")  
private Long auditoriaId;
```

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:**
 - **auditoriaId:** Identificador único de la auditoría.
 - **fechaHora:** Fecha y hora en que se realizó la operación.
 - **terminal:** Terminal o equipo desde el cual se ejecutó la acción.
 - **operacion:** Tipo de operación realizada (crear, actualizar, eliminar, etc.).
 - **verificacion:** Información adicional sobre la validación o control de la operación.
 - **usuario:** Nombre del usuario que ejecutó la operación.
- **Relaciones con otras entidades (Joins):**

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

- Relación con el objeto UsuarioDTO correspondiente al usuario que ejecutó la auditoría.

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

- aplicacion: Referencia a la aplicación dentro del sistema que fue afectada por la operación.

```
@Bind(name = "tabla", returnType = TablaDTO.class, type = Tabla.class)
private TablaDTO tabla;
```

- tabla: Entidad de tabla específica en la base de datos sobre la cual se realizó la operación.

- **Objeto de sesión:**

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

- appSessionVO: Permite capturar información de la sesión actual del usuario, útil para trazabilidad y control.

Constructores

```
public AuditoriaDTO() { }
```

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

```
public AuditoriaDTO(Long auditoriaId) {
    this.auditoriaId = auditoriaId;
}
```

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

```
public AuditoriaDTO(Long auditoriaId, Date fechaHora, String terminal, String operacion) {
    this.auditoriaId = auditoriaId;
    this.fechaHora = fechaHora;
    this.terminal = terminal;
    this.operacion = operacion;
}
```

- **Constructor con ID, fechaHora, terminal y operacion:** 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 getUsuario() {  
    return usuario;  
}  
  
public void setUsuario(String usuario) {  
    this.usuario = usuario;  
}
```

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

Estos métodos permiten:

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

```
@Override  
public boolean equals(Object object) {  
    if (!(object instanceof AuditoriaDTO)) {  
        return false;  
    }  
    AuditoriaDTO other = (AuditoriaDTO) object;  
    return (this.auditoriaId != null && this.auditoriaId.equals(other.auditoriaId));  
}
```

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

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

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

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

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

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

Conclusión

Este DTO:

- Permite transportar información de Auditoria 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 **"Auditorías"** 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 auditorías del sistema.

```
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST6310')}">
  <li class="#{selectedPage eq 'auditoria' ? 'active':''}"><p:link
outcome="/app/administracion/auditorias/auditorias" >
  #{msg['sb.menu.gestion.administracion.auditorias.auditoria']}</p:link></li>
</ui:fragment>
```

Control de Visibilidad con Seguridad

```
<ui:fragment rendered="#{SeguridadSessController.mapaMenu.containsKey('GEST6310')}">
```

- 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('GEST6310')** verifica si el identificador 'GEST6310' 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 'auditoria' ? 'start active':''}">
```

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

Enlace de Navegación

```
<p:link outcome="/app/administracion/auditorias/auditorias" 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/auditorias/auditorias**).
- 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.auditorias.auditoria']}
```

- `<i class="fa fa-building"></i>` inserta un ícono de edificio usando **FontAwesome** para representar visualmente la funcionalidad.
- `#{msg['sb.menu.gestion.administracion.auditorias.auditoria']}` 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 auditorías. Su visibilidad depende de los permisos del usuario (**GEST6310**). 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 AuditoriaViewController

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

Declaraciones de Clase

```
@Named(value = "auditoriaViewController")
```

```
@ViewScoped  
public class AuditoriaViewController extends GenericSessionController<AuditoriaDTO>  
implements Serializable {
```

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

Inyección de Auditorías

```
@EJB  
private AdministracionServiceFacade administracionServiceFacade;  
@Inject  
private ConfigAppController configAppController;
```

- **administracionServiceFacade**: Proporciona acceso a la capa de negocio (**EJB**).
- **configAppController**: Controlador de configuración de la aplicación.

Variables de Estado

```
private LazyDataModel<AuditoriaDTO> model;  
private Date fechaInicial;  
private Date fechaFinal;  
private boolean filtroActivo;  
private AuditoriaDTO auditoriaFilterBusq;  
private List<UsuarioDTO> listaUsuarios;  
private List<TablaDTO> listaTablas;
```

- **model**: Implementación de **LazyDataModel<AuditoriaDTO>** utilizada para la carga paginada de auditorías en componentes PrimeFaces.
- **fechaInicial**: Fecha de inicio utilizada como filtro en las consultas de auditorías.
- **fechaFinal**: Fecha de fin utilizada como filtro en las consultas de auditorías.
- **filtroActivo**: Bandera booleana que indica si los filtros de búsqueda están siendo aplicados.
- **auditoriaFilterBusq**: Objeto DTO que contiene los criterios de búsqueda para filtrar auditorías.

- **listaUsuarios:** Lista de objetos UsuarioDTO utilizada para poblar el filtro por usuario.
- **listaTablas:** Lista de objetos TablaDTO utilizada para poblar el filtro por tabla.

Método cargarListas()

```
public void cargarListas() {  
    this.visibleNuevo = Boolean.FALSE;  
    this.visibleEditar = Boolean.FALSE;  
    this.visibleEliminar = Boolean.FALSE;  
    this.visibleRecargar = Boolean.FALSE;  
    LocalDate ahoraLocalDate = LocalDate.now();  
    LocalDate fechaInicialLocalDate = ahoraLocalDate.minus(1, ChronoUnit.MONTHS);  
    if (fechaFinal == null) {  
        fechaFinal = Date.from(ahoraLocalDate.atTime(23,  
59).atZone(ZoneId.systemDefault()).toInstant());  
    }  
    if (fechaInicial == null) {  
        fechaInicial =  
Date.from(fechaInicialLocalDate.atStartOfDay(ZoneId.systemDefault()).toInstant());  
    }  
  
    cargarListaTablas();  
    this.model = new LazyDataModel<AuditoriaDTO>() {
```

- **@PostConstruct:** Se ejecuta después de instanciar el **bean**.
- **visibleNuevo, visibleEditar, visibleEliminar, visibleRecargar:** **Flags** booleanas para controlar la visibilidad de botones u opciones en la interfaz.
- **fechaInicial:** Fecha por defecto establecida al primer día del mes anterior, usada como filtro de búsqueda.
- **fechaFinal:** Fecha por defecto establecida al final del día actual, usada como filtro de búsqueda.
- Llama a **cargarListaTablas()** para inicializar las listas.
- Crea un **LazyDataModel** para la carga de datos con paginación.

Implementación de LazyDataModel

```
@Override  
public List<AuditoriaDTO> 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 {
    //Filtro por aplicacion
    filters.put("aplicacion", configApplController.getAplicacion().getAplicacion());
    //Filtro por rango de fecha inicial
    if (fechaInicial != null) {
        filters.put("fechaInicial", fechaInicial);
    }
    //Filtro por rango de fecha final
    if (fechaFinal != null) {
        filters.put("fechaFinal", fechaFinal);
    }
    //Filtro Búsqueda Avanzada
    if (auditoriaFilterBusq != null) {
        filters.put("auditoriaFilterBusq", auditoriaFilterBusq);
    }
    int count = administracionServiceFacade.auditoriasListCount(filters);
    if (count > 0) {
        list = administracionServiceFacade.auditoriasList(first, pageSize, sortField,
        sortOrder.equals(SortOrder.ASCENDING) ? "A" : "D", filters);
    }
    model.setRowCount(count);
} catch (Exception ex) {
    LOG.error("Error al cargar el listado de auditorias ", ex);
}
return list;
}
```

Explicación

- Obtiene los filtros aplicados desde la tabla, especialmente el **globalFilter** si está presente y no está vacío.
- Agrega filtros adicionales manuales como:
 - **aplicacion:** Se filtra por la aplicación actual desde **configApplController**.
 - **fechaInicial y fechaFinal:** Se usan para limitar el rango de fechas de auditoría.
 - **auditoriaFilterBusq:** Filtro avanzado con más criterios de búsqueda.
- Consulta la cantidad total de registros que cumplen los filtros con **auditoriasListCount(filters)**.
- Si existen registros, obtiene la lista paginada de auditorías usando **auditoriasList(...)**, aplicando ordenamiento si está definido.

- Asigna el total de registros al modelo con **model.setRowCount(count)** para que PrimeFaces gestione correctamente la paginación.
- Captura excepciones y registra errores en el log si ocurre algún problema durante la carga.

Método getRowData()

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

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

Método cargarListaTablas()

```
public void cargarListaTablas() {
    try {
        listaTablas = new ArrayList<>();
        Map<String, Object> filters = new HashMap<>();
        filters.put("aplicacion", configApplController.getAplicacion().getAplicacion());
        int count = administracionServiceFacade.tablasListCount(filters);
        if(count > 0){
            listaTablas = administracionServiceFacade.tablasList(-1, -1, "tabla", "A",
filters);
        }
    } catch (Exception ex) {
        LOG.error("Error al cargar el listado de tablas ", ex);
    }
}
```

- Carga la lista de tablas (listaTablas) según la aplicación actual.
- Usa filtros para la consulta.
- Si hay resultados, obtiene la lista ordenada por nombre
- Maneja posibles errores con log

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

Controlador de Conversación AuditoriaConvController.java

El **AuditoriaConvController** es un controlador en Java EE que gestiona las operaciones sobre las auditorías 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<AuditoriaDTO>**, lo que le proporciona funcionalidad base para la gestión de datos en una conversación de JSF.

Inyección de Auditorías

```
@Inject
private AuditoriaViewController auditoriaViewController;
@Inject
private ColcienciasSessController colcienciasSessController;
@EJB
private ConfiguracionServiceFacade configuracionServiceFacade;
```

- **AuditoriaViewController**: Controlador de vista que gestiona el estado y la interacción con la UI.
- **ColcienciasSessController**: Proporciona configuraciones generales de la aplicación.
- **ConfiguracionServiceFacade**: Componente EJB que maneja la lógica de negocio relacionada con la administración de auditorías.

Variables Importantes

```
private final String programa = "GEST6310";

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

- **programa**: Código que identifica el módulo dentro del sistema (**GEST6310**).
- **LOG**: Logger de **Apache Log4j** para registrar eventos y errores.

Funcionalidades Implementadas

Mostrar Lista de Auditorías (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 = auditoriaViewController.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);
auditoriaViewController.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.

Limpiar Filtros (limpiarFiltros)

Resetea los filtros aplicados en la vista de auditorías.

```
public void limpiarFiltros() {
    auditoriaViewController.setAuditoriaFilterBusq(new AuditoriaDTO());
    auditoriaViewController.setFiltroActivo(false);
    System.out.println("limpiando");
}
```

Flujo:

1. Se crea un nuevo DTO vacío.
2. Se desactiva el indicador de filtros activos.
3. Se imprime mensaje de limpieza en consola.

Obtener Usuarios SIGP (getListaUsuariosSigp)

Busca usuarios según el texto ingresado y actualiza la lista en el controlador.

```
public List<UsuarioDTO> getListaUsuariosSigp(String query) {  
    List<UsuarioDTO> usuarios = new ArrayList<>();  
    try {  
        usuarios = configuracionServiceFacade.obtenerUsuariosSigp(query.replaceAll("^\\s*",  
        ""));  
        auditoriaViewController.setListaUsuarios(usuarios);  
    } catch (Exception ex) {  
        LOG.error("Error al obtener el listado de usuarios ", ex);  
    }  
    return usuarios;  
}
```

Flujo:

1. Limpia espacios al inicio del texto.
2. Consulta los usuarios que coincidan con el texto.
3. Actualiza la lista en el controlador.
4. Maneja errores en caso de excepción.

Ir a Detalles Auditoría (irDetallesAuditoria)

```
Guarda la auditoría seleccionada y redirige a la vista de detalles.  
public void irDetallesAuditoria(AuditoriaDTO auditoriaDTO) {  
    Map<String, Object> mapReferencias = new HashMap<>();  
    mapReferencias.put("auditoriaSelected", auditoriaDTO);  
    colcienciasSessController.setUtilObject(mapReferencias);  
  
    this.redirectPage("app/administracion/auditorias/detallesAuditoria.xhtml");  
}
```

Flujo:

1. Crea un mapa con la auditoría seleccionada.
2. Guarda la auditoría en sesión.
3. Redirige a la pantalla de detalles.

Consideraciones Técnicas

- **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**.

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

```
//  
//GESTIÓN AUDITORIAS (SISTEMA)  
//  
/**  
 * Consulta el listado de Auditorias 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  
 * Auditorias  
 * @return lista de Auditorias encontrados en el rango enviado y basado en  
 * los filtros enviados  
 * @throws Exception  
 */  
List<AuditoriaDTO> auditoriasList(int first, int pageSize, String sortField, String  
sortOrder, Map<String, Object> filters) throws Exception;  
/**  
 * Cuenta la cantidad de Auditorias actualmente registrados en la base de  
 * datos  
 *  
 * @param filters  
 * @return  
 * @throws Exception  
 */  
int auditoriasListCount(Map<String, Object> filters) throws Exception;  
/**  
 * Guarda el registro de Auditoria en la base de datos  
 *  
 * @param AuditoriaDTO  
 * @throws Exception  
 */  
void guardarAuditoria(AuditoriaDTO AuditoriaDTO) throws Exception;  
/**  
 * Edita el registro de Auditoria en la base de datos  
 *  
 * @param AuditoriaDTO  
 * @throws Exception  
 */  
void editarAuditoria(AuditoriaDTO AuditoriaDTO) throws Exception;  
/**  
 * Elimina el registro de Auditoria en la base de datos  
 *  
 */
```

```
* @param AuditoriaDTO
* @throws Exception
*/
void eliminarAuditoria(AuditoriaDTO AuditoriaDTO) throws Exception;
/**
* Buscar el registro de un Auditoria y lo retorna basado en el id (llave
* primaria)
*
* @param AuditoriaId
* @return el Auditoria que coincide con el id enviado
* @throws Exception
*/
AuditoriaDTO queryAuditoria(Long AuditoriaId) throws Exception;
```

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

```
//----- Gestión Auditorias (Sistema) -----//
@Override
public List<AuditoriaDTO> auditoriasList(int first, int pageSize, String sortField, String
sortOrder, Map<String, Object> filters) throws Exception {
    ObjectConverter<AuditoriaDTO, Auditoria> converter = objectConverterFactory.checkOut();
    List<AuditoriaDTO> listAuditoriaDTOs =
converter.toTransferList(auditoriaService.auditoriasList(first, pageSize, sortField,
sortOrder, filters), AuditoriaDTO.class);
    objectConverterFactory.checkIn(converter);
    return listAuditoriaDTOs;
}
@Override
public int auditoriasListCount(Map<String, Object> filters) throws Exception {
    //Como hay busqueda avanzada, toca hacer el converter con la variable
    AuditoriaDTO auditoriaFilterDTO = (AuditoriaDTO) filters.get("auditoriaFilterBusq");
    ObjectConverter<AuditoriaDTO, Auditoria> converter = objectConverterFactory.checkOut();
    Auditoria auditoriaFilterBusq = converter.toEntity(auditoriaFilterDTO, Auditoria.class);
    filters.put("auditoriaFilterBusq", auditoriaFilterBusq);
    return auditoriaService.auditoriasListCount(filters);
}
@Override
public void guardarAuditoria(AuditoriaDTO auditoriaDTO) throws Exception {
    ObjectConverter<AuditoriaDTO, Auditoria> converter = objectConverterFactory.checkOut();
    auditoriaService.guardarAuditoria(converter.toEntity(auditoriaDTO, Auditoria.class));
    objectConverterFactory.checkIn(converter);
}
@Override
public void editarAuditoria(AuditoriaDTO auditoriaDTO) throws Exception {
    ObjectConverter<AuditoriaDTO, Auditoria> converter = objectConverterFactory.checkOut();
    auditoriaService.editarAuditoria(converter.toEntity(auditoriaDTO, Auditoria.class));
    objectConverterFactory.checkIn(converter);
}
@Override
public void eliminarAuditoria(AuditoriaDTO auditoriaDTO) throws Exception {
    ObjectConverter<AuditoriaDTO, Auditoria> converter = objectConverterFactory.checkOut();
    auditoriaService.eliminarAuditoria(converter.toEntity(auditoriaDTO, Auditoria.class));
    objectConverterFactory.checkIn(converter);
}
```

```
@Override
public AuditoriaDTO queryAuditoria(Long auditoriaId) throws Exception {
    ObjectConverter<AuditoriaDTO, Auditoria> converter = objectConverterFactory.checkOut();
    AuditoriaDTO AuditoriaDTO =
    converter.toTransfer(auditoriaService.queryAuditoria(auditoriaId), AuditoriaDTO.class);
    objectConverterFactory.checkIn(converter);
    return AuditoriaDTO;
}
```

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

```
@Local
public interface AuditoriaService {

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

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

    void guardarAuditoria(Auditoria auditoria) throws SystemException;

    void editarAuditoria(Auditoria auditoria) throws SystemException;

    void eliminarAuditoria(Auditoria auditoria) throws SystemException;

    Auditoria queryAuditoria(Long auditoriaId) throws SystemException;

    List<Object[]> reporteControlAuditoriasList(Map<String, Object> filters) throws
    SystemException;

    String callSqlAuditoria(String p_aplicacion, String p_tabla, String p_campo_llave,
    String p_valor_llave, String p_fecha_inicial, String p_fecha_final) throws SystemException;

}
```

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

```
@Stateless
public class AuditoriaServiceBean extends StoreManagerImpl implements AuditoriaService {

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

    @Override
    public int auditoriasListCount(Map<String, Object> filters) throws SystemException {
```

```

        return
        this.getStoreManager(AdministracionStoreManager.class).auditoriasListCount(filters);
    }

    @Override
    public void guardarAuditoria(Auditoria auditoria) throws SystemException {
        this.crear(auditoria, Auditoria.class);
    }

    @Override
    public void editarAuditoria(Auditoria auditoria) throws SystemException {
        this.modificar(auditoria, Auditoria.class);
    }

    @Override
    public void eliminarAuditoria(Auditoria auditoria) throws SystemException {
        this.eliminar(auditoria, Auditoria.class);
    }

    @Override
    public Auditoria queryAuditoria(Long auditoriaId) throws SystemException {
        return this.buscar(auditoriaId, Auditoria.class);
    }

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

    @Override
    public String callSqlAuditoria(String p_aplicacion, String p_tabla, String
p_campo_llave, String p_valor_llave, String p_fecha_inicial, String p_fecha_final) throws
SystemException {
        return
        this.getStoreManager(AdministracionStoreManager.class).callSqlAuditoria(p_aplicacion,
p_tabla, p_campo_llave, p_valor_llave, p_fecha_inicial, p_fecha_final);
    }
}

```

Internacionalización en español e inglés

#ES	
sb.menu.gestion.administracion.auditorias.auditoria.header	= Auditorías
sb.menu.gestion.administracion.auditorias.auditoria.breadcrumb	= Auditorías
sb.menu.gestion.administracion.auditorias.auditoria.búsquedaAvanzada	= Búsqueda
sb.menu.gestion.administracion.auditorias.auditoria.auditoriaId.header	= Auditoría
sb.menu.gestion.administracion.auditorias.auditoria.usuario.header	= Usuario
sb.menu.gestion.administracion.auditorias.auditoria.usuario.placeholder	= Digite y
seleccione el usuario	

sb.menu.gestion.administracion.auditorias.auditoria.tabla.header	= Tabla
sb.menu.gestion.administracion.auditorias.auditoria.terminal.header	= Terminal
sb.menu.gestion.administracion.auditorias.auditoria.fechaHora.header	= Fecha/Hora
sb.menu.gestion.administracion.auditorias.auditoria.operacion.header	= Operación
sb.menu.gestion.administracion.auditorias.auditoria.verificacion.header	=
Verificación	
sb.menu.gestion.administracion.auditorias.auditoria.aplicacion.header	= Aplicación
sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn	= Detalle
sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn.tooltip	= Ver
Detalle	
sb.menu.gestion.administracion.auditorias.auditoria.fechaInicial.label	= Fecha
Inicial	
sb.menu.gestion.administracion.auditorias.auditoria.fechaInicial.required	= Debe
ingresar la Fecha Inicial	
sb.menu.gestion.administracion.auditorias.auditoria.fechaFinal.label	= Fecha
Final	
sb.menu.gestion.administracion.auditorias.auditoria.fechaFinal.required	= Debe
ingresar la Fecha Final	
#EN	
sb.menu.gestion.administracion.auditorias.auditoria.header	= Audits
sb.menu.gestion.administracion.auditorias.auditoria.breadcrumb	= Audits
sb.menu.gestion.administracion.auditorias.auditoria.búsquedaAvanzada	= Advanced
Search	
sb.menu.gestion.administracion.auditorias.auditoria.auditoriaId.header	= ID Audit
sb.menu.gestion.administracion.auditorias.auditoria.usuario.header	= User
sb.menu.gestion.administracion.auditorias.auditoria.usuario.placeholder	= Enter and
select the user	
sb.menu.gestion.administracion.auditorias.auditoria.tabla.header	= Table
sb.menu.gestion.administracion.auditorias.auditoria.terminal.header	= Terminal
sb.menu.gestion.administracion.auditorias.auditoria.fechaHora.header	= Date/Time
sb.menu.gestion.administracion.auditorias.auditoria.operacion.header	= Operation
sb.menu.gestion.administracion.auditorias.auditoria.verificacion.header	=
Verification	
sb.menu.gestion.administracion.auditorias.auditoria.aplicacion.header	=
Application	
sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn	= Detail
sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn.tooltip	= See Detail
sb.menu.gestion.administracion.auditorias.auditoria.fechaInicial.label	= Initial
Date	
sb.menu.gestion.administracion.auditorias.auditoria.fechaInicial.required	= You must
enter the Start Date	
sb.menu.gestion.administracion.auditorias.auditoria.fechaFinal.label	= Final Date
sb.menu.gestion.administracion.auditorias.auditoria.fechaFinal.required	= You must
enter the End Date	

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

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

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

```

<p:dataTable id="#{auditoriaViewController.modulo}Table"
    var ="auditoria"
    value="#{auditoriaViewController.modelo}"
    rowKey="#{auditoria.auditoriaId}"
    selection="#{auditoriaConvController.selected}"
    selectionMode="single"
    paginator="true"
    rows="10"
    lazy="true"
    reflow="true"
    sortBy="#{auditoria.auditoriaId}"
    sortOrder="DESCENDING"
    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#{auditoriaViewController.modulo}"
>
    <f:facet name="header">
        <div style="float:right;padding-top: 5px;">
            <ui:fragment rendered="false">
                <h:commandLink>
                    
                    <p:dataExporter type="xlsxstream"
target="#{auditoriaViewController.modulo}Table" fileName="auditorias" />
                </h:commandLink>
                <h:commandLink>
                    
                    <p:dataExporter type="pdf"
target="#{auditoriaViewController.modulo}Table" fileName="auditorias"/>
                </h:commandLink>
                <h:commandLink>
                    
                    <p:dataExporter type="csv"
target="#{auditoriaViewController.modulo}Table" fileName="auditorias"/>
                </h:commandLink>
                <h:commandLink>
                    
                    <p:dataExporter type="xml"
target="#{auditoriaViewController.modulo}Table" fileName="auditorias"/>
                </h:commandLink>
            </ui:fragment>
            <p:commandButton id="toggler" type="button" value="Columnas"
style="float:right" icon="pi pi-align-justify" />
            <p:columnToggler datasource="#{auditoriaViewController.modulo}Table"
trigger="toggler" />
        </div>
        <ui:decorate template="../../template/_tableHeader.xhtml">
            <ui:define name="comandos">
                <ui:param name="sessionController"
value="#{auditoriaViewController}" />
                <ui:param name="conversationController"
value="#{auditoriaConvController}" />
                <ui:param name="btnBusqueda" value="N" />
            </ui:define>
        </ui:decorate>
    </f:facet>
</p:dataTable>

```

```

        <ui:param name="buscarVisible" value="S" />
    </ui:define>
</ui:decorate>
</f:facet>
<p:ajax event="rowSelect"
    listener="#{auditoriaConvController.onRowSelect}"
    update="#{auditoriaViewController.modulo}Table"
    process="@this"/>
<p:ajax event="rowUnselect"
    listener="#{auditoriaConvController.onRowUnselect}"
    update="#{auditoriaViewController.modulo}Table"
    process="@this"/>
<p:column style="text-align: center; padding: 5px;"
    width="5%"
    exportable="false"
    toggleable="false"
    >
        <p:rowToggler/>
    </p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.usuario.header']}"
    filterBy="#{auditoria.usuario}"
    sortBy="#{auditoria.usuario}"
    filterMatchMode="contains"
    filterStyle="display:none;"
    style="text-align: center"
    width="15%"
    >
        <h:outputText value="#{auditoria.usuario}"/>
    </p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.tabla.header']}"
    filterBy="#{auditoria.tabla.tabla}"
    sortBy="#{auditoria.tabla.tabla}"
    filterMatchMode="contains"
    filterStyle="display:none;"
    style="text-align: center"
    >
        <h:outputText value="#{auditoria.tabla.tabla}"
title="#{auditoria.tabla.tabla}"/>
    </p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.terminal.header']}"
    filterBy="#{auditoria.terminal}"
    sortBy="#{auditoria.terminal}"
    filterMatchMode="contains"
    filterStyle="display:none;"
    style="text-align: center"
    width="15%"
    >
        <h:outputText value="#{auditoria.terminal}"/>
    </p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.fechaHora.header']}"
    filterBy="#{auditoria.fechaHora}"
    sortBy="#{auditoria.fechaHora}"
    filterMatchMode="contains"
    filterStyle="display:none;"
    style="text-align: center"
    width="15%"
    >

```

```

                <h:outputText value="#{auditoria.fechaHora}">
                    <f:convertDateTime pattern="dd/MM/yyyy HH:mm" timeZone="EST"/>
                </h:outputText>
            </p:column>
            <p:column>
                headerText="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.operacion.header']}"
                filterBy="#{auditoria.operacion}"
                sortBy="#{auditoria.operacion}"
                filterMatchMode="contains"
                filterStyle="display:none;"
                style="text-align: center"
                width="15%"
            >
                <h:outputText
                    value="#{msg['listaTipoOperacion.'.concat(auditoria.operacion)]}" />
            </p:column>
            <p:column style="text-align: center"
                width="10%"
                exportable="false"
                toggleable="false"
            >
                <p:commandLink style="color: #fff;"
                    process="@this"

                    title="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn.tooltip']}"
                    styleClass="btn btn-xs btn-primary sbTooltip"
                    immediate="true"

                    actionListener="#{auditoriaConvController.irDetallesAuditoria(auditoria)}"
                >
                    <i class="fa fa-building-o">
                        #{msg['sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn']} </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.administracion.auditorias.auditoria.auditoriaId.header']}" />
                                <h:outputText value="#{auditoria.auditoriaId}" />
                                <h:outputText
                                    value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.aplicacion.header']}" />
                                <h:outputText value="#{auditoria.aplicacion.aplicacion}" />
                                <h:outputText
                                    value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.verificacion.header']}" />
                                <h:outputText value="#{auditoria.verificacion}" />
                            </p:panelGrid>
                        </div>
                    </div>
                    <p:spacer/>
                </div>
            </p:rowExpansion>
        </p:dataTable>
    </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="#{auditoriaViewController.modulo}Table"
    var ="auditoria"
    value="#{auditoriaViewController.model}"
    rowKey="#{auditoria.auditoriaId}"
    selection="#{auditoriaConvController.selected}"
    selectionMode="single"
    paginator="true"
    rows="10"
    lazy="true"
    reflow="true"
    sortBy="#{auditoria.auditoriaId}"
    sortOrder="DESCENDING"
    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#{auditoriaViewController.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<Auditoria>) obtenida del controlador.

Propiedad	Descripción
rowKey	Identificador único de cada fila basado en auditoriald.
selection	Almacena la fila seleccionada en auditoriaConvController.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="#{auditoria.auditoriald}"	Define el ordenamiento inicial por auditoriald.
sortOrder="ASCENDING"	Orden ascendente por defecto.
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="#{auditoriaConvController.onRowSelect}"
  update="#{auditoriaViewController.modulo}Table"
  process="@this"/>

<p:ajax event="rowUnselect"
  listener="#{auditoriaConvController.onRowUnselect}"
  update="#{auditoriaViewController.modulo}Table"
```

```
process="@this"/>
```

- **rowSelect:** Llama a `onRowSelect()` en `auditoriaConvController` cuando el usuario selecciona una fila.
- **rowUnselect:** Llama a `onRowUnselect()` cuando el usuario deselecciona 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 Auditoria.

Ejemplo:

```
<p:column
  headerText="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.usuario.header']}"
  filterBy="#{auditoria.usuario}"
  sortBy="#{auditoria.usuario}"
  filterMatchMode="contains"
  filterStyle="display:none;"
  style="text-align: center"
  width="15%"
>
  <h:outputText value="#{auditoria.usuario}"/>
</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="15%"	Define el ancho de la columna.

Se repite este patrón para otras columnas como **tabla.tabla**, **terminal**, **fechaHora**, **operación**, **usuarioAdministrador.usuario**, **vigente**, etc.

Acciones en las Filas

```
<p:commandLink style="color: #fff;"
               process="@this"

               title="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn.tooltip']}"
               styleClass="btn btn-xs btn-primary sbTooltip"
               immediate="true"
               actionListener="#{auditoriaConvController.irDetallesAuditoria(auditoria)}"
               >
               <i class="fa fa-building-o">
               #{msg['sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn']} </i>
               </p:commandLink>
```

- **Botón para ver detalles de una auditoría seleccionada. (fa-building-o):**
 - Ejecuta **irDetallesAuditoria(auditoria)**.
 - Redirige a la vista de detalles de auditoría.

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.administracion.auditorias.auditoria.auditoriaId.header']}" />
          <h:outputText value="#{auditoria.auditoriaId}" />
          <h:outputText
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.aplicacion.header']}" />
          <h:outputText value="#{auditoria.aplicacion.aplicacion}" />
          <h:outputText
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.verificacion.header']}" />
          <h:outputText value="#{auditoria.verificacion}" />
        </p:panelGrid>
      </div>
    </div>
    <p:spacer/>
  </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 FECHAS -->
<h:panelGroup layout="block" id="paramFiltro" class="row">
  <div class="well form-horizontal" style="padding: 20px">
    <div class="row">
      <div class="col-lg-4">
        <div class="form-group">
          <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.fechaInicial.label']}"
styleClass="col-lg-6 control-label"
for="fechaInicial"/>
          <div class="col-lg-6">
            <p:calendar id="fechaInicial"
locale="es"
value="#{auditoriaViewController.fechaInicial}"
required="true"

requiredMessage="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.fechaInicial.req
uired']}"

navigator="true"
mask="true"
pattern="dd/MM/yyyy"
maxdate="0"/>
          </div>
        </div>
      </div>
      <div class="col-lg-4">
        <div class="form-group">
          <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.fechaFinal.label']}"
styleClass="col-lg-5 control-label"
for="fechaFinal"/>
          <div class="col-lg-6">
            <p:calendar id="fechaFinal"
locale="es"
value="#{auditoriaViewController.fechaFinal}"
required="true"

requiredMessage="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.fechaFinal.requi
red']}"

navigator="true"
mask="true"
pattern="dd/MM/yyyy"
maxdate="0"/>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

```

        <p:commandLink id="btnFiltrar"
                      process="paramFiltro"

update="#{auditoriaViewController.modulo}TabList:#{auditoriaViewController.modulo}Table"

                      styleClass="btn btn btn-primary pull-left"
                      >
        <i class="fa fa-search"></i>
        #{msg['sb.component.boton.filtrar']}
        </p:commandLink>
    </div>
</div>
</div>
</div>
</h:panelGroup>
<!-- FILTRO AVANZADO -->
<h:panelGroup layout="block" id="formaFiltro" 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.formatoEvaluacion.búsquedaAvanzada']}</button>
        </ui:fragment>
        <div id="demo" class="collapse #{auditoriaViewController.filtroActivo ? 'in' : ''}
well form-horizontal" style="padding: 20px">
            <div class="row">
                <div class="col-lg-12">
                    <div class="form-group">
                        <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.usuario.header']}"
                        styleClass="col-lg-3 control-label"
                        for="usuarioBusq-Aud"/>
                        <div class="col-lg-8">
                            <p:autoComplete id="usuarioBusq-Aud"

value="#{auditoriaViewController.auditoriaFilterBusq.usuarioJoin}"
                                converter="usuarioConverter"

completeMethod="#{auditoriaConvController.getListUsuariosSigp}"
                                var="usuario"
                                minQueryLength="4"
                                itemLabel="#{usuario.usuarioDescripcion}"
                                itemValue="#{usuario}"
                                forceSelection="true"
                                inputStyleClass="form-control"
                                style="width: 100%"

placeholder="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.usuario.placeholder'
]]}"

                                />
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-lg-12">
        <div class="form-group">
            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.tabla.header']}"
            styleClass="col-lg-3 control-label"
            for="tablaBusq"/>

```

```

        <div class="col-lg-8">
            <p:selectOneMenu id="tablaBusq"
value="#{auditoriaViewController.auditoriaFilterBusq.tabla}"
                                converter="tablaConverter"
                                styleClass="sbTooltip"
                                style="width:100%;"
                                autoWidth="false"
                                panelStyle="width:180px"
                                effect="fade"
                                filter="true"
                                filterMatchMode="contains"
                                >
                <f:selectItem
itemLabel="#{msg['sb.selectOneMenu.registro']}" />
                <f:selectItems
value="#{auditoriaViewController.listaTablas}"
                                var="tabla"
                                itemLabel="#{tabla.tabla}"
                                itemDescription="#{tabla.descripcionTabla}"
                                itemValue="#{tabla}"
                                />
            </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.auditorias.auditoria.terminal.header']}"
                                styleClass="col-lg-6 control-label"
                                for="terminalBusq"/>
            <div class="col-lg-6">
                <p:inputText id="terminalBusq"
value="#{auditoriaViewController.auditoriaFilterBusq.terminal}"
                                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.auditorias.auditoria.operacion.header']}"
                                styleClass="col-lg-4 control-label"
                                for="operacionBusq"/>
                <div class="col-lg-6">
                    <h:selectOneMenu id="operacionBusq"
value="#{auditoriaViewController.auditoriaFilterBusq.operacion}"
                                styleClass="form-control">
                        <f:selectItems
value="#{configApplController.listaTipoOperacion}" />
                        </h:selectOneMenu>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
        <div class="row" jsf:rendered="false">
            <div class="col-lg-6">
                <div class="form-group">
                    <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.fechaHora.header']}"
                    styleClass="col-lg-6 control-label"
                    for="fechaHoraBusq"/>
                    <div class="col-lg-6">
                        <p:calendar id="fechaHoraBusq"
                            locale="es"

value="#{auditoriaViewController.auditoriaFilterBusq.fechaHora}"
                            navigator="true"
                            mask="true"
                            pattern="dd/MM/yyyy"/>
                    </div>
                </div>
            </div>
        </div>
        <div class="hr-line-dashed"></div>
        <div class="row">
            <div class="col-lg-6">
                <div class="form-group">
                    <div class="col-lg-8 col-lg-offset-6">
                        <p:commandLink id="buscar"
                            process="formaFiltro"

actionListener="#{auditoriaConvController.lazyLoad}"
                            update="#{auditoriaViewController.modulo}Table"

                            styleClass="btn btn btn-primary pull-left"
                            >
                            <i class="fa fa-search"></i>
                            #{msg['sb.component.boton.filtrar']}
                        </p:commandLink>
                    </div>
                </div>
            </div>
            <div class="col-lg-6">
                <div class="form-group">
                    <div class="col-lg-8 col-lg-offset-4">
                        <p:commandLink id="limpiar"

update="formaFiltro,#{auditoriaViewController.modulo}Table"
                            process="@this"

actionListener="#{auditoriaConvController.limpiarFiltros}"
                            styleClass="btn btn btn-primary pull-right"

onclick="PF('items#{auditoriaViewController.modulo}').clearFilters()"
                            >
                            <i class="fa fa-times"></i>
                            #{msg['sb.component.boton.cancelar']}
                        </p:commandLink>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

```

```
</h:panelGroup>  
<p:spacer/>
```

Este código define una sección de filtro avanzado para la búsqueda de auditorías 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 Despliegado 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.auditorias.auditoria.búsquedaAvanzada']**.

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

- La clase **collapse** permite ocultar/mostrar el panel de filtros.
- Se usa EL **#{auditoriaViewController.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

- Usuario Administrador: Campo autocompletable (<p:autoComplete>) con filtro por nombre de usuario.
- Tabla Afectada: Menú desplegable (<p:selectOneMenu>) con filtro por nombre de tabla.
- Terminal: Campo de texto (<p:inputText>) para ingresar el nombre de la terminal.
- Operación Realizada: Menú desplegable (<h:selectOneMenu>) para seleccionar la operación.
- Fecha y Hora: Campo de calendario (<p:calendar>) para seleccionar una fecha específica.

- Fecha Inicial: Campo de calendario (<p:calendar>) para seleccionar la fecha de inicio del filtro.
- o Fecha Final: Campo de calendario (<p:calendar>) para seleccionar la fecha de finalización.

4. Botones de Acción

- **Botón de Buscar (<p:commandLink id="buscar">)**
 - Ejecuta el método **lazyLoad** en controlador **auditoriaConvController**.
 - Actualiza la tabla de resultados definida por **#{auditoriaViewController.modulo}Table**.
 - Limpia los filtros visuales con **PF('items#{auditoriaViewController.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 Búsqueda Avanzada

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.

Funcionamiento Filtro Fechas

1. El usuario ingresa las fechas requeridas

2. 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.

Visualización de la pantalla

Auditorías

Fecha Inicial * 10/03/2025 Fecha Final * 10/04/2025 [Filtrar](#)

Auditorías

Búsqueda Avanzada

Buscar Todos: [Columnas](#)

	Usuario ↕	Tabla ↕	Terminal ↕	Fecha/Hora ↕	Operación ↕	
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	08/04/2025 15:00	Actualizar	Detalle
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	08/04/2025 03:00	Actualizar	Detalle
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	08/04/2025 15:00	Actualizar	Detalle
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	07/04/2025 15:00	Actualizar	Detalle
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	07/04/2025 03:00	Actualizar	Detalle
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	08/04/2025 15:00	Actualizar	Detalle
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	08/04/2025 03:00	Actualizar	Detalle
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	05/04/2025 15:00	Actualizar	Detalle
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	05/04/2025 03:00	Actualizar	Detalle
●	SIGP_WEB	ADM_PARAMETROS_SISTEMA	UNKNOWN	04/04/2025 15:00	Actualizar	Detalle

Total de registros: 173 1 2 3 4 5 6 7 8 9 10 10

Creación pantalla de Detalles Auditorías

Función:

Visualiza los detalles de una auditoría específica, mostrando los campos modificados en la tabla de la base de datos.

Principales funcionalidades:

- **Visualización de Campos:** Muestra los campos que fueron afectados en la operación.

Creación nuevo menú de detalles auditorías

```
-- 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', 'GEST6311', 'Detalles Auditoría', 'F', null, null,
'/app/administracion/auditorias/detallesAuditoria.xhtml', null, 'GEST6100', 'fa fa-edit',
'HERCULES', to_date('01-04-2024', 'dd-mm-yyyy'), 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', 'GEST6311', 'SGPS_ADMINISTRADOR', 'H', 'HERCULES', to_date('01-04-
2024', 'dd-mm-yyyy'), null, null);
```

*Mapear nueva entidad (**DetalleAuditoria.java**) de la tabla **SB_DETALLES_AUDITORIAS** 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.security;

import com.sb.colciencias.gestion.business.valueObjects.AppSessionVO;
import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
```

```
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.Lob;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Transient;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;

/**
 *
 * @author lsuarez
 */
@Entity
@Table(name = "SB_DETALLES_AUDITORIAS", catalog = "", schema = "HERCULES")
@XmlRootElement
@NamedQueries({
    @NamedQuery(name = "DetalleAuditoria.findAll", query = "SELECT d FROM DetalleAuditoria d")
    , @NamedQuery(name = "DetalleAuditoria.findById", query = "SELECT d FROM DetalleAuditoria d WHERE d.detalleAuditoriaId = :detalleAuditoriaId")
    , @NamedQuery(name = "DetalleAuditoria.findByCampo", query = "SELECT d FROM DetalleAuditoria d WHERE d.campo = :campo")
    , @NamedQuery(name = "DetalleAuditoria.findByValorAntes", query = "SELECT d FROM DetalleAuditoria d WHERE d.valorAntes = :valorAntes")
    , @NamedQuery(name = "DetalleAuditoria.findByValorDespues", query = "SELECT d FROM DetalleAuditoria d WHERE d.valorDespues = :valorDespues")
    , @NamedQuery(name = "DetalleAuditoria.findByLlavePrimaria", query = "SELECT d FROM DetalleAuditoria d WHERE d.llavePrimaria = :llavePrimaria")
    , @NamedQuery(name = "DetalleAuditoria.findByVerificacion", query = "SELECT d FROM DetalleAuditoria d WHERE d.verificacion = :verificacion")
    , @NamedQuery(name = "DetalleAuditoria.findByIdAuditoriaId", query = "SELECT d FROM DetalleAuditoria d WHERE d.auditoriaId.auditoriaId = :auditoriaId"))
public class DetalleAuditoria implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "SB_USUARIO_GRUPO_ID_SEQ")
    @SequenceGenerator(name = "SB_USUARIO_GRUPO_ID_SEQ", sequenceName = "SB_USUARIO_GRUPO_ID_SEQ", allocationSize = 1)
    @Basic(optional = false)
    @NotNull
    @Column(name = "DETALLE_AUDITORIA_ID")
    private Long detalleAuditoriaId;
    @Basic(optional = false)
    @NotNull
    @Size(min = 1, max = 30)
    @Column(name = "CAMPO")
    private String campo;
    @Size(max = 4000)
    @Column(name = "VALOR_ANTES")
    private String valorAntes;
    @Size(max = 4000)
    @Column(name = "VALOR_DESPUES")
    private String valorDespues;
    @Lob
```

```
@Column(name = "DOCUMENTO")
private byte[] documento;
@Lob
@Column(name = "TEXTO")
private String texto;
@Column(name = "LLAVE_PRIMARIA")
private Short llavePrimaria;
@Size(max = 4000)
@Column(name = "VERIFICACION")
private String verificacion;

//Joins
@JoinColumn(name = "AUDITORIA_ID", referencedColumnName = "AUDITORIA_ID")
@ManyToOne(optional = false, fetch = FetchType.LAZY)
private Auditoria auditoriaId;

//Auditoria
@Transient
private AppSessionVO appSessionVO;

public DetalleAuditoria() {
}

public DetalleAuditoria(Long detalleAuditoriaId) {
    this.detalleAuditoriaId = detalleAuditoriaId;
}

public DetalleAuditoria(Long detalleAuditoriaId, String campo) {
    this.detalleAuditoriaId = detalleAuditoriaId;
    this.campo = campo;
}

public Long getDetalleAuditoriaId() {
    return detalleAuditoriaId;
}

public void setDetalleAuditoriaId(Long detalleAuditoriaId) {
    this.detalleAuditoriaId = detalleAuditoriaId;
}

public String getCampo() {
    return campo;
}

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

public String getValorAntes() {
    return valorAntes;
}

public void setValorAntes(String valorAntes) {
    this.valorAntes = valorAntes;
}

public String getValorDespues() {
    return valorDespues;
}

public void setValorDespues(String valorDespues) {
```

```
        this.valorDespues = valorDespues;
    }

    public byte[] getDocumento() {
        return documento;
    }

    public void setDocumento(byte[] documento) {
        this.documento = documento;
    }

    public String getTexto() {
        return texto;
    }

    public void setTexto(String texto) {
        this.texto = texto;
    }

    public Short getLlavePrimaria() {
        return llavePrimaria;
    }

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

    public String getVerificacion() {
        return verificacion;
    }

    public void setVerificacion(String verificacion) {
        this.verificacion = verificacion;
    }

    public Auditoria getAuditoriaId() {
        return auditoriaId;
    }

    public void setAuditoriaId(Auditoria auditoriaId) {
        this.auditoriaId = auditoriaId;
    }

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (detalleAuditoriaId != null ? detalleAuditoriaId.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 DetalleAuditoria)) {
            return false;
        }
        DetalleAuditoria other = (DetalleAuditoria) object;
        if ((this.detalleAuditoriaId == null && other.detalleAuditoriaId != null) ||
            (this.detalleAuditoriaId != null &&
            !this.detalleAuditoriaId.equals(other.detalleAuditoriaId))) {
            return false;
        }
    }
}
```

```

    }
    return true;
}

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

public AppSessionVO getAppSessionVO() {
    return appSessionVO;
}

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

```

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

El **DetalleAuditoriaDTO** (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 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.transfer;

import com.sb.colciencias.gestion.business.domain.security.*;
import com.sb.colciencias.gestion.business.util.annotation.Bind;
import com.sb.colciencias.gestion.business.valueObjects.AppSessionVO;
import java.io.Serializable;
import javax.persistence.Lob;

/**
 *
 * @author lsuarez
 */
public class DetalleAuditoriaDTO implements Serializable {

    private static final long serialVersionUID = 1L;

    @Bind(name = "detalleAuditoriaId")
    private Long detalleAuditoriaId;
    @Bind(name = "campo")
    private String campo;
    @Bind(name = "valorAntes")
    private String valorAntes;
    @Bind(name = "valorDespues")

```

```
private String valorDespues;
@Lob
@Bind(name = "documento")
private byte[] documento;
@Lob
@Bind(name = "texto")
private String texto;
@Bind(name = "llavePrimaria")
private Short llavePrimaria;
@Bind(name = "verificacion")
private String verificacion;

//Joins
@Bind(name = "auditoriaId", returnType = AuditoriaDTO.class, type = Auditoria.class)
private AuditoriaDTO auditoriaId;

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

public DetalleAuditoriaDTO() {
}

public DetalleAuditoriaDTO(Long detalleAuditoriaId) {
    this.detalleAuditoriaId = detalleAuditoriaId;
}

public DetalleAuditoriaDTO(Long detalleAuditoriaId, String campo) {
    this.detalleAuditoriaId = detalleAuditoriaId;
    this.campo = campo;
}

public Long getDetalleAuditoriaId() {
    return detalleAuditoriaId;
}

public void setDetalleAuditoriaId(Long detalleAuditoriaId) {
    this.detalleAuditoriaId = detalleAuditoriaId;
}

public String getCampo() {
    return campo;
}

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

public String getValorAntes() {
    return valorAntes;
}

public void setValorAntes(String valorAntes) {
    this.valorAntes = valorAntes;
}

public String getValorDespues() {
    return valorDespues;
}

public void setValorDespues(String valorDespues) {
```

```
        this.valorDespues = valorDespues;
    }

    public byte[] getDocumento() {
        return documento;
    }

    public void setDocumento(byte[] documento) {
        this.documento = documento;
    }

    public String getTexto() {
        return texto;
    }

    public void setTexto(String texto) {
        this.texto = texto;
    }

    public Short getLlavePrimaria() {
        return llavePrimaria;
    }

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

    public String getVerificacion() {
        return verificacion;
    }

    public void setVerificacion(String verificacion) {
        this.verificacion = verificacion;
    }

    public AuditoriaDTO getAuditoriaId() {
        return auditoriaId;
    }

    public void setAuditoriaId(AuditoriaDTO auditoriaId) {
        this.auditoriaId = auditoriaId;
    }

    public AppSessionVO getAppSessionVO() {
        return appSessionVO;
    }

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

    @Override
    public int hashCode() {
        int hash = 0;
        hash += (detalleAuditoriaId != null ? detalleAuditoriaId.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 DetalleAuditoriaDTO)) {
            return false;
        }
        DetalleAuditoriaDTO other = (DetalleAuditoriaDTO) object;
        if ((this.detalleAuditoriaId == null && other.detalleAuditoriaId != null) ||
            (this.detalleAuditoriaId != null &&
             !this.detalleAuditoriaId.equals(other.detalleAuditoriaId))) {
            return false;
        }
        return true;
    }

    @Override
    public String toString() {
        return "com.sb.colciencias.gestion.business.domain.security.DetalleAuditoria["
            + detalleAuditoriaId + " ]";
    }
}
```

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(name = "detalleAuditoriaId")
private Long detalleAuditoriaId;
@Bind(name = "campo")
private String campo;
@Bind(name = "valorAntes")
private String valorAntes;
@Bind(name = "valorDespues")
private String valorDespues;
@Lob
@Bind(name = "documento")
```

```
private byte[] documento;  
@Lob  
@Bind(name = "texto")  
private String texto;  
@Bind(name = "llavePrimaria")  
private Short llavePrimaria;  
@Bind(name = "verificacion")  
private String verificacion;
```

- **Identificadores y Descripción:**

- **detalleAuditorialId:** Identificador único del detalle de auditoría.
- **campo:** Nombre del campo de la tabla que fue modificado.
- **valorAntes:** Valor que tenía el campo antes de la operación.
- **valorDespues:** Valor del campo después de la operación.
- **documento:** Campo tipo LOB que almacena información en formato binario relacionada con el cambio.
- **texto:** Campo tipo LOB que almacena texto adicional relacionado con el cambio.
- **llavePrimaria:** Indica si el campo corresponde a una llave primaria (1 para sí, 0 para no).
- **verificacion:** Campo adicional para control o validación del cambio realizado.

```
@Bind(name = "auditoriaId", returnType = AuditoriaDTO.class, type = Auditoria.class)  
private AuditoriaDTO auditoriaId;  
@Bind(name = "appSessionVO")  
private AppSessionVO appSessionVO;
```

- **Relaciones con Otras Entidades:**

- **auditorialId:** Relación con **AuditoriaDTO**, lo que indica que cada **DetalleAuditoriaDTO** está asociada con una auditoría.
- **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.

- **DetalleAuditorialId:**

```
public Long getDetalleAuditoriaId() {  
    return detalleAuditoriaId;  
}  
public void setDetalleAuditoriaId(Long detalleAuditoriaId) {
```

```
this.detalleAuditoriaId = detalleAuditoriaId;  
}
```

- **Métodos Override:**

```
@Override  
public int hashCode() {  
    int hash = 0;  
    hash += (detalleAuditoriaId != null ? detalleAuditoriaId.hashCode() : 0);  
    return hash;  
}  
  
@Override  
public boolean equals(Object object) {  
    if (this == object) {  
        return true;  
    }  
    if (object == null || getClass() != object.getClass()) {  
        return false;  
    }  
    final DetalleAuditoriaDTO other = (DetalleAuditoriaDTO) object;  
    return this.detalleAuditoriaId != null &&  
    this.detalleAuditoriaId.equals(other.detalleAuditoriaId);  
}  
  
@Override  
public String toString() {  
    return "com.sb.colciencias.gestion.business.domain.security.DetalleAuditoria[  
    detalleAuditoriaId=" + detalleAuditoriaId + " ]";  
}}
```

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

El **DetalleAuditoriaDTO** 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.

Controlador de Vista DetalleAuditoriaViewController.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 = "detalleAuditoriaViewController")
@ViewScoped
public class DetalleAuditoriaViewController extends
GenericSessionController<DetalleAuditoriaDTO> implements Serializable {
```

- **@Named(value = "detalleAuditoriaViewController"):**
 - Define este controlador como un **bean de JSF**, permitiendo su acceso en las páginas **XHTML** con `#{detalleAuditoriaViewController}`.
- **@ViewScoped:**
 - Mantiene el estado del controlador **mientras el usuario permanezca en la misma vista**.
- **extends GenericSessionController<DetalleAuditoriaDTO>:**
 - Hereda funcionalidades de **GenericSessionController**, lo que sugiere que maneja sesiones de usuario y datos de tipo **DetalleAuditoriaDTO**.
- **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 ColcienciasSessController colcienciasSessController;
@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 ColcienciasSessController colcienciasSessController;**

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

Atributos de la Clase

```
private LazyDataModel<DetalleAuditoriaDTO> model;  
private AuditoriaDTO auditoriaSelected;  
private DetalleAuditoriaDTO detalleAuditoriaFilterBusq;  
private boolean filtroActivo;  
private static final org.apache.log4j.Logger LOG =  
org.apache.log4j.Logger.getLogger(DetalleAuditoriaViewController.class);
```

- **LazyDataModel<DetalleAuditoriaDTO> model;**
 - Representa un modelo de datos perezoso, utilizado para cargar los detalles de auditoría de manera eficiente en la vista, especialmente útil en tablas con paginación.
- **AuditoriaDTO auditoriaSelected;**
 - Almacena la auditoría seleccionada, indicando a cuál auditoría pertenecen los detalles que se están visualizando.
- **DetalleAuditoriaDTO detalleAuditoriaFilterBusq;**
 - Objeto utilizado como filtro para realizar búsquedas dentro de los detalles de auditoría, facilitando la aplicación de criterios dinámicos.
- **boolean filtroActivo;**
 - Indica si se ha aplicado algún filtro en la búsqueda de los detalles, permitiendo condicionar comportamientos en la vista.
- **Logger LOG**
 - Instancia del logger del sistema, utilizada para registrar eventos, errores o trazas dentro del controlador.

Constructor de la Clase

```
public DetalleAuditoriaViewController() {  
    super(DetalleAuditoriaDTO.class);  
}
```

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

Inicialización con @PostConstruct

```
@PostConstruct  
public void cargarListas() {  
  
    this.visibleNuevo = Boolean.FALSE;  
    this.visibleEliminar = Boolean.FALSE;  
  
    Map<String, Object> mapReferencias = (Map<String, Object>)  
colcienciasSessController.getUtilObject();  
    auditoriaSelected = (AuditoriaDTO) mapReferencias.get("auditoriaSelected");  
  
    auditoriaSelected.setUsuarioJoin(administracionServiceFacade.obtenerUsuario(auditoriaSelected.  
d.getUsuario()));  
  
    this.model = new LazyDataModel<DetalleAuditoriaDTO>()
```

- **@PostConstruct:**
 - Se ejecuta **después de que se haya creado** la instancia del controlador.
 - Llama a cargarListaSecuencias() para obtener las secuencias de tablas.
- **visibleNuevo / visibleEliminar:**
 - Banderas booleanas utilizadas para controlar la visibilidad de botones u opciones en la interfaz gráfica.
 - Por defecto están en false, ya que la pantalla de detalles de auditoría es solo de consulta.
- **auditoriaSelected:**
 - Obtiene del contexto la auditoría seleccionada previamente, permitiendo cargar sus detalles.
 - Se le asocia el usuario relacionado a través de un join con el método obtenerUsuario(...) del servicio de administración.

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<DetalleAuditoriaDTO> 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("auditoriaId", auditoriaSelected.getAuditoriaId()); // Filtro por Auditoria ID  
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 los detalles de la auditoría actual.

- 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(DetalleAuditoriaDTO tablaDTO) {  
    return tablaDTO.getDetalleAuditoriaId();  
}
```

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

```
@Override  
public DetalleAuditoriaDTO getRowData(String rowKey) {  
    try {  
        return administracionServiceFacade.queryDetalleAuditoria(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.

Métodos Getters y Setters

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

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

```
public ConfigAppController getConfigAppController() { return configAppController; }  
public void setConfigAppController(ConfigAppController configAppController) {  
    this.configAppController = configAppController; }  
}
```

- Métodos para acceder al controlador de configuración.

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 **DetalleAuditoriaConvController.java***

El **DetalleAuditoriaConvController** 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 DetalleAuditoriaDTO (CRUD), además de interactuar con servicios y controladores de sesión para manejar información y seguridad.

Anotaciones y Alcance

```
@Named(value = "detalleAuditoriaConvController")
@ConversationScoped
public class DetalleAuditoriaConvController extends
GenericConversationController<DetalleAuditoriaDTO> implements Serializable {
```

- **@Named(value = "detalleAuditoriaConvController")**: 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.

Auditorías Inyectadas

```
@EJB
private AdministracionServiceFacade administracionServiceFacade;
@Inject
private DetalleAuditoriaViewController detalleAuditoriaViewController;
@Inject
private SeguridadSessController seguridadSessController;
@Inject
private ColcienciasSessController colcienciasSessController;
```

```
@Inject  
private ConfigAppController configAppController;
```

- **AdministracionServiceFacade:** EJB para operaciones administrativas sobre tablas.
- **DetalleAuditoriaViewController:** Controlador de vista para la gestión de detalles de auditorías.
- **SeguridadSessController:** Controlador de seguridad para autenticación y permisos
- **ColcienciasSessController:** Controlador de sesión del usuario.
- **ConfigAppController:** Controlador de configuración de la aplicación.

Atributos Principales

```
private final String programa = "GEST6311";  
  
private static final org.apache.log4j.Logger LOG =  
org.apache.log4j.Logger.getLogger(DetalleAuditoriaConvController.class);  
  
public DetalleAuditoriaConvController() {  
    this.selected = new DetalleAuditoriaDTO();  
}
```

- programa: Código del programa utilizado en auditorías.
- LOG: Logger de Log4J para registrar eventos y errores.
- selected: Objeto DetalleAuditoriaDTO seleccionado, inicializado en el constructor.

Métodos Principales

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 = detalleAuditoriaViewController.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);  
}
```

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

Prepara la interfaz para mostrar datos:

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

volverAuditorias()

```
public void volverAuditorias() {  
    Map<String, Object> mapReferencias = new HashMap<>();  
    colcienciasSessController.setUtilObject(mapReferencias);  
    this.redirectPage("app/administracion/auditorias/auditorias.xhtml");  
}
```

1. Crea un nuevo HashMap para almacenar referencias temporales.
2. Establece este mapa en el controlador de sesión (colcienciasSessController) para limpiar o reiniciar datos previos.
3. Redirige al usuario a la pantalla principal de auditorías (auditorias.xhtml).

limpiarFiltros (String cambioEstado)

```
public void limpiarFiltros() {  
    detalleAuditoriaViewController.setDetalleAuditoriaFilterBusq(new DetalleAuditoriaDTO());  
    detalleAuditoriaViewController.setFiltroActivo(false);  
    System.out.println("limpiando");  
}
```

1. Reinicia el filtro de búsqueda asignando un nuevo objeto DetalleAuditoriaDTO.
2. Desactiva el estado del filtro estableciendo filtroActivo en false.
3. Imprime en consola el mensaje "limpiando" como indicativo de depuración.

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

```
//
//GESTIÓN DETALLES AUDITORIAS (SISTEMA)
//
/**
 * Consulta el listado de Detalle Auditorias 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
 * Detalle Auditorias
 * @return lista de DetalleAuditorias encontrados en el rango enviado y
 * basado en los filtros enviados
 * @throws Exception
 */
List<DetalleAuditoriaDTO> detalleAuditoriasList(int first, int pageSize, String sortField,
String sortOrder, Map<String, Object> filters) throws Exception;
/**
 * Cuenta la cantidad de Detalle Auditorias actualmente registrados en la
 * base de datos
 *
 * @param filters
 * @return
 * @throws Exception
 */
int detalleAuditoriasListCount(Map<String, Object> filters) throws Exception;
/**
 * Guarda el registro de Detalle Auditoria en la base de datos
 *
 * @param detalleAuditoriaDTO
 * @throws Exception
 */
void guardarDetalleAuditoria(DetalleAuditoriaDTO detalleAuditoriaDTO) throws Exception;
/**
 * Edita el registro de Detalle Auditoria en la base de datos
 *
 * @param detalleAuditoriaDTO
 * @throws Exception
 */
void editarDetalleAuditoria(DetalleAuditoriaDTO detalleAuditoriaDTO) throws Exception;
/**
 * Elimina el registro de Detalle Auditoria en la base de datos
 *
 * @param detalleAuditoriaDTO
 * @throws Exception
 */
void eliminarDetalleAuditoria(DetalleAuditoriaDTO detalleAuditoriaDTO) throws Exception;
/**
 * Buscar el registro de un Detalle Auditoria y lo retorna basado en el id
 * (llave primaria)
 *
 * @param detalleAuditoriaId
 * @return el Detalle Auditoria que coincide con el id enviado
 * @throws Exception
 */
```

```
*/  
DetalleAuditoriaDTO queryDetalleAuditoria(Long detalleAuditoriaId) throws Exception;
```

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

```
//----- Gestión Detalles Auditorias (Sistema) -----  
---//  
@Override  
public List<DetalleAuditoriaDTO> detalleAuditoriasList(int first, int pageSize, String  
sortField, String sortOrder, Map<String, Object> filters) throws Exception {  
    ObjectConverter<DetalleAuditoriaDTO, DetalleAuditoria> converter =  
    objectConverterFactory.checkOut();  
    List<DetalleAuditoriaDTO> listDetalleAuditoriaDTOs =  
    converter.toTransferList(detalleAuditoriaService.detalleAuditoriasList(first, pageSize,  
sortField, sortOrder, filters), DetalleAuditoriaDTO.class);  
    objectConverterFactory.checkIn(converter);  
    return listDetalleAuditoriaDTOs;  
}  
@Override  
public int detalleAuditoriasListCount(Map<String, Object> filters) throws Exception {  
    //Como hay busqueda avanzada, toca hacer el converter con la variable  
    DetalleAuditoriaDTO detalleAuditoriaFilterDTO = (DetalleAuditoriaDTO)  
filters.get("detalleAuditoriaFilterBusq");  
    ObjectConverter<DetalleAuditoriaDTO, DetalleAuditoria> converter =  
    objectConverterFactory.checkOut();  
    DetalleAuditoria detalleAuditoriaFilterBusq =  
    converter.toEntity(detalleAuditoriaFilterDTO, DetalleAuditoria.class);  
    filters.put("detalleAuditoriaFilterBusq", detalleAuditoriaFilterBusq);  
    return detalleAuditoriaService.detalleAuditoriasListCount(filters);  
}  
@Override  
public void guardarDetalleAuditoria(DetalleAuditoriaDTO detalleAuditoriaDTO) throws  
Exception {  
    ObjectConverter<DetalleAuditoriaDTO, DetalleAuditoria> converter =  
    objectConverterFactory.checkOut();  
    detalleAuditoriaService.guardarDetalleAuditoria(converter.toEntity(detalleAuditoriaDTO,  
DetalleAuditoria.class));  
    objectConverterFactory.checkIn(converter);  
}  
@Override  
public void editarDetalleAuditoria(DetalleAuditoriaDTO detalleAuditoriaDTO) throws Exception  
{  
    ObjectConverter<DetalleAuditoriaDTO, DetalleAuditoria> converter =  
    objectConverterFactory.checkOut();  
    detalleAuditoriaService.editarDetalleAuditoria(converter.toEntity(detalleAuditoriaDTO,  
DetalleAuditoria.class));  
    objectConverterFactory.checkIn(converter);  
}  
@Override  
public void eliminarDetalleAuditoria(DetalleAuditoriaDTO detalleAuditoriaDTO) throws  
Exception {  
    ObjectConverter<DetalleAuditoriaDTO, DetalleAuditoria> converter =  
    objectConverterFactory.checkOut();  
    detalleAuditoriaService.eliminarDetalleAuditoria(converter.toEntity(detalleAuditoriaDTO,  
DetalleAuditoria.class));  
    objectConverterFactory.checkIn(converter);  
}
```

```
}
@Override
public DetalleAuditoriaDTO queryDetalleAuditoria(Long detalleAuditoriaId) throws Exception {
    ObjectConverter<DetalleAuditoriaDTO, DetalleAuditoria> converter =
    objectConverterFactory.checkOut();
    DetalleAuditoriaDTO DetalleAuditoriaDTO =
    converter.toTransfer(detalleAuditoriaService.queryDetalleAuditoria(detalleAuditoriaId),
    DetalleAuditoriaDTO.class);
    objectConverterFactory.checkIn(converter);
    return DetalleAuditoriaDTO;
}
```

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

```
@Local
public interface DetalleAuditoriaService {

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

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

    void guardarDetalleAuditoria(DetalleAuditoria detalleAuditoria) throws SystemException;

    void editarDetalleAuditoria(DetalleAuditoria detalleAuditoria) throws SystemException;

    void eliminarDetalleAuditoria(DetalleAuditoria detalleAuditoria) throws SystemException;

    DetalleAuditoria queryDetalleAuditoria(Long detalleAuditoriaId) throws SystemException;
}
```

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

```
@Stateless
public class DetalleAuditoriaServiceBean extends StoreManagerImpl implements
DetalleAuditoriaService {

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

    @Override
```

```

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

        @Override
        public void guardarDetalleAuditoria(DetalleAuditoria detalleAuditoria) throws
        SystemException {
            this.crear(detalleAuditoria, DetalleAuditoria.class);
        }

        @Override
        public void editarDetalleAuditoria(DetalleAuditoria detalleAuditoria) throws
        SystemException {
            this.modificar(detalleAuditoria, DetalleAuditoria.class);
        }

        @Override
        public void eliminarDetalleAuditoria(DetalleAuditoria detalleAuditoria) throws
        SystemException {
            this.eliminar(detalleAuditoria, DetalleAuditoria.class);
        }

        @Override
        public DetalleAuditoria queryDetalleAuditoria(Long detalleAuditoriaId) throws
        SystemException {
            return this.buscar(detalleAuditoriaId, DetalleAuditoria.class);
        }
    }

```

Internacionalización en español e inglés

```

#ES
sb.menu.gestion.administracion.auditorias.detalleAuditoria.header           =
Detalles Auditorías
sb.menu.gestion.administracion.auditorias.detalleAuditoria.breadcrumb       =
Detalles Auditorías
sb.menu.gestion.administracion.auditorias.detalleAuditoria.búsquedaAvanzada =
Búsqueda Avanzada
sb.menu.gestion.administracion.auditorias.detalleAuditoria.campo.header     =
Campo
sb.menu.gestion.administracion.auditorias.detalleAuditoria.valorAntes.header =
Valor Antes
sb.menu.gestion.administracion.auditorias.detalleAuditoria.valorDespues.header =
Valor Después
sb.menu.gestion.administracion.auditorias.detalleAuditoria.texto.header     =
Texto LOB
sb.menu.gestion.administracion.auditorias.detalleAuditoria.verificacion.header =
Verificación
sb.menu.gestion.administracion.auditorias.detalleAuditoria.auditoria.header =
Información de la Auditoría
sb.menu.gestion.administracion.auditorias.detalleAuditoria.volverAuditoria.btn =
Volver a Auditorías

#EN

```

```
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.header
= Reporte Control de Auditorías
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.breadcrumb
= Control de Auditorías
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.usuario.label
= Usuario
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.usuario.placeholder
= Digite y seleccione el usuario
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.tabla.label
= Tabla
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.descripcion.label
= Descripción
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaInicial.label
= Fecha Inicial
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaInicial.required
= Debe ingresar la Fecha Inicial
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaFinal.label
= Fecha Final
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaFinal.required
= Debe ingresar la Fecha Final
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.registros.header
= Número de Registros
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.creados.label
= Creados
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.modificados.label
= Modificados
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.consultados.label
= Consultados
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.descargados.label
= Descargados
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.borrados.label
= Borrados
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.reporte.btn
= Generar Reporte
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.detalle.header
= Detalle de Auditoría
```

Pestaña de Detalles de Auditoría de visualización de la información en la pantalla de DetalleAuditorias

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

<!-- TABLA -->
<div class="row">
    <div class="col-lg-12" style="overflow-x: scroll;">
        <p:dataTable id="#{detalleAuditoriaViewController.modulo}Table"
            var ="detalleAuditoria"
            value="#{detalleAuditoriaViewController.modelo}"
            rowKey="#{detalleAuditoria.detalleAuditoriaId}"
            selection="#{detalleAuditoriaConvController.selected}"
            selectionMode="single"
            paginator="true"
            rows="10"
            lazy="true"
            reflow="true"
```

```

        sortBy="#{detalleAuditoria.campo}"
        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#{detalleAuditoriaViewController.modulo}">
    <f:facet name="header">
        <ui:decorate template="../../template/_tableHeader.xhtml">
            <ui:define name="comandos">
                <ui:param name="sessionController"
value="#{detalleAuditoriaViewController}" />
                <ui:param name="conversationController"
value="#{detalleAuditoriaConvController}" />
                <ui:param name="btnBusqueda" value="N" />
                <ui:param name="buscarVisible" value="S" />
            </ui:define>
        </ui:decorate>
    </f:facet>
    <p:ajax event="rowSelect"
        listener="#{detalleAuditoriaConvController.onRowSelect}"
        update="#{detalleAuditoriaViewController.modulo}Table"
        process="@this"/>
    <p:ajax event="rowUnselect"
        listener="#{detalleAuditoriaConvController.onRowUnselect}"
        update="#{detalleAuditoriaViewController.modulo}Table"
        process="@this"/>
    <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.detalleAuditoria.campo.header']}"
        >
        filterBy="#{detalleAuditoria.campo}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        style="text-align: center"
        >
        <h:outputText value="#{detalleAuditoria.campo}"/>
    </p:column>
    <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.detalleAuditoria.valorAntes.heade
r']}"
        >
        filterBy="#{detalleAuditoria.valorAntes}"
        sortBy="#{detalleAuditoria.valorAntes}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        style="text-align: center"
        >
        <h:outputText value="#{detalleAuditoria.valorAntes}"/>
    </p:column>
    <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.detalleAuditoria.valorDespues.h
eader']}"
        >
        filterBy="#{detalleAuditoria.valorDespues}"
        sortBy="#{detalleAuditoria.valorDespues}"
        filterMatchMode="contains"
        filterStyle="display:none;"
        style="text-align: center"
        >
        <h:outputText value="#{detalleAuditoria.valorDespues}"/>
    </p:column>

```

```

        </p:column>
        <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.detalleAuditoria.texto.header']}
}"
            >
            <h:outputText title="#{detalleAuditoria.texto}"
value="#{detalleAuditoria.texto.length() > 100 ?
detalleAuditoria.texto.substring(0,100).concat('...') : detalleAuditoria.texto}"/>
            </p:column>
        </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.auditorias.detalleAuditoria.header']}">

```

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

2. Tabla de Datos

```

<p:dataTable id="#{detalleAuditoriaViewController.modulo}Table"
var ="detalleAuditoria"
value="#{detalleAuditoriaViewController.modelo}"
rowKey="#{detalleAuditoria.detalleAuditoriaId}"
selection="#{detalleAuditoriaConvController.selected}"
selectionMode="single"
paginator="true"
rows="10"
lazy="true"
reflow="true"
sortBy="#{detalleAuditoria.campo}"
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#{detalleAuditoriaViewController.modulo}">
```

- p:dataTable muestra los datos de detalleAuditoriaViewController.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.

3. Eventos de Selección de Fila

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

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

4. Columnas de los Detalles de Auditoría

Ejemplo de una columna con ordenamiento y filtrado:

```
<p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.detalleAuditoria.campo.header']}"
filterBy="#{detalleAuditoria.campo}"
filterMatchMode="contains"
filterStyle="display:none;"
style="text-align: center"
>
<h:outputText value="#{detalleAuditoria.campo}"/>
</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.descripcionDetalleAuditoria.

Visualización de la pantalla

Detalles Auditorías

Volver

Información de la Auditoría

Auditoría ID: 52778
 Usuario: SIGP_WEB - Sistema de Gestión de Proyectos - SIGP
 Tabla: ADM_PARAMETROS_SISTEMA
 Terminal: UNKNOWN
 Fecha/Hora: 09/04/2025 15:00
 Operación: Actualizar

Detalles Auditorías Guardar/Editar

Búsqueda Avanzada

Campo	Valor Antes	Valor Después	Texto LOB
CODIGO_PARAMETRO	FCA_GRUSCI	FCA_GRUSCI	
FECHA	09/04/2025 03:00:00	09/04/2025 15:00:00	

Total de registros: 2

Creación pantalla de Reporte Control Auditoría

Función:

Permite realizar consultas detalladas sobre los campos modificados dentro de las tablas del sistema, facilitando el análisis específico por columna.

Principales funcionalidades:

- **Filtros de Búsqueda:** Permite seleccionar parámetros como usuario, tabla, fechas y tipo de operación (creación, modificación, consulta, descarga, eliminación).
- **Consulta por Campo:** Muestra, de acuerdo con los filtros aplicados, los campos específicos que fueron modificados o utilizados en operaciones sobre las tablas.
- **Visualización Detallada:** Presenta el campo, el número de veces que fue modificado, consultado, descargado, creado o eliminado, agrupado por usuario y tabla.

Creación nuevo menú de Reporte Control Auditoría

```
-- 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', 'GEST6320', 'Reporte Control de Auditoría', 'F', null, null,
'/app/administracion/auditorias/reporteControlAuditoria.xhtml', null, 'GEST6100', 'fa fa-
print', 'HERCULES', to_date('01-04-2024', 'dd-mm-yyyy'), 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', 'GEST6320', 'SGPS_ADMINISTRADOR', 'H', 'HERCULES', to_date('01-04-
2024', 'dd-mm-yyyy'), null, null);
```

Controlador de Vista ReporteControlAuditoriaViewController.java

El **ReporteControlAuditoriaViewController** es un controlador de vista dentro del módulo de administración que gestiona la visualización y manipulación de campos del objeto (**ReporteControlAuditoriaVO**).

```

/*
 * 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;

/*
 * 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.AuditoriaDTO;
import com.sb.colciencias.gestion.business.transfer.DetalleAuditoriaDTO;
import com.sb.colciencias.gestion.business.transfer.TablaDTO;
import com.sb.colciencias.gestion.business.transfer.UsuarioDTO;
import com.sb.colciencias.gestion.business.util.SystemException;
import com.sb.colciencias.gestion.business.valueObjects.ReporteControlAuditoriaVO;
import com.sb.colciencias.gestion.web.controller.application.ConfigApplController;
import com.sb.standard.web.controller.session.GenericSessionController;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.ejb.EJB;
import javax.inject.Inject;
import javax.inject.Named;
import javax.enterprise.context.SessionScoped;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;

/**
 *
 * @author lsuarez
 */
@Named(value = "reporteControlAuditoriaViewController")
@SessionScoped
public class ReporteControlAuditoriaViewController extends
GenericSessionController<ReporteControlAuditoriaVO> implements Serializable {

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

    private ReporteControlAuditoriaVO controlAuditoriaVO;
    private List<UsuarioDTO> listaUsuarios;
    private List<TablaDTO> listaTablas;

```

```
private List<Object[]> listReporteControlAuditoria;
private List<AuditoriaDTO> listAuditorias;
private List<DetalleAuditoriaDTO> listDetallesAuditorias;

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

public ReporteControlAuditoriaViewController() {
    super(ReporteControlAuditoriaVO.class);
    controlAuditoriaVO = new ReporteControlAuditoriaVO();
}

@Inject
public void cargarListas() {
    try {
        controlAuditoriaVO.setCreadosBool(Boolean.TRUE);
        controlAuditoriaVO.setModificadosBool(Boolean.TRUE);
        controlAuditoriaVO.setConsultadosBool(Boolean.TRUE);
        controlAuditoriaVO.setDescargadosBool(Boolean.TRUE);
        controlAuditoriaVO.setBorradosBool(Boolean.TRUE);

        listaTablas = new ArrayList<>();
        Map<String, Object> filters = new HashMap<>();
        filters.put("aplicacion", configApplController.getAplicacion().getAplicacion());
        int count = administracionServiceFacade.tablasListCount(filters);
        if (count > 0) {
            listaTablas = administracionServiceFacade.tablasList(-1, -1, "tabla", "A",
filters);
        }
    } catch (Exception ex) {
        LOG.error("Error al cargar el listas ", ex);
    }
}

public void cargarReporte() {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));
    try {
        Map<String, Object> filters = new HashMap<>();
        //Filtro por aplicacion
        filters.put("aplicacion", configApplController.getAplicacion().getAplicacion());
        //Filtro Busqueda Avanzada
        if (controlAuditoriaVO != null) {
            controlAuditoriaVO.setAplicacion(configApplController.getAplicacion());
            filters.put("controlAuditoriaVOFilter", controlAuditoriaVO);
        }

        listReporteControlAuditoria =
administracionServiceFacade.reporteControlAuditoriasList(filters);

    } catch (SystemException ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        LOG.error("Error al cargar el reporte ", ex);
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        LOG.error("Error al cargar el reporte ", ex);
    } finally {
    }
}
```

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

public ConfigApplController getConfigApplController() {
    return configApplController;
}

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

public ReporteControlAuditoriaVO getControlAuditoriaVO() {
    return controlAuditoriaVO;
}

public void setControlAuditoriaVO(ReporteControlAuditoriaVO controlAuditoriaVO) {
    this.controlAuditoriaVO = controlAuditoriaVO;
}

public List<UsuarioDTO> getListaUsuarios() {
    return listaUsuarios;
}

public void setListaUsuarios(List<UsuarioDTO> listaUsuarios) {
    this.listaUsuarios = listaUsuarios;
}

public List<TablaDTO> getListaTablas() {
    return listaTablas;
}

public void setListaTablas(List<TablaDTO> listaTablas) {
    this.listaTablas = listaTablas;
}

public List<Object[]> getListReporteControlAuditoria() {
    return listReporteControlAuditoria;
}

public void setListReporteControlAuditoria(List<Object[]> listReporteControlAuditoria) {
    this.listReporteControlAuditoria = listReporteControlAuditoria;
}

public List<AuditoriaDTO> getListAuditorias() {
    return listAuditorias;
}

public void setListAuditorias(List<AuditoriaDTO> listAuditorias) {
    this.listAuditorias = listAuditorias;
}

public List<DetalleAuditoriaDTO> getListDetallesAuditorias() {
    return listDetallesAuditorias;
}

public void setListDetallesAuditorias(List<DetalleAuditoriaDTO> listDetallesAuditorias)
{
    this.listDetallesAuditorias = listDetallesAuditorias;
}
```

El **ReporteControlAuditoriaViewController** proporciona una implementación robusta para la generación y visualización del reporte de control de auditoría. Permite aplicar filtros por usuario, tabla, fecha y tipo de acción (creados, modificados, consultados, descargados, borrados) para consultar de manera detallada los eventos registrados en el sistema.

Auditorías e Inyección de Dependencias

- **@EJB AdministracionServiceFacade** Servicio de negocio que proporciona las operaciones necesarias para recuperar los datos de auditoría desde la capa de persistencia.
- **@Inject ConfigAppController** Controlador que maneja la configuración de la aplicación, especialmente la aplicación actual desde la cual se están generando los reportes.

Atributos Principales

- **ReporteControlAuditoriaVO controlAuditoriaVO** Objeto de valor que encapsula los filtros seleccionados por el usuario (usuario, tabla, fechas, tipos de acciones, etc.).
- **List<UsuarioDTO> listaUsuarios** Lista de usuarios disponibles en el sistema, usada para el autocompletado o selección en el filtro.
- **List<TablaDTO> listaTablas** Lista de tablas disponibles para seleccionar en el filtro.
- **List<Object[]> listReporteControlAuditoria** Resultado de la consulta generada por los filtros del reporte. Cada arreglo representa una fila del reporte con los valores: Usuario, Tabla, Creados, Modificados, Consultados, Descargados y Borrados.
- **List<AuditoriaDTO> listAuditorias** Lista general de auditorías encontradas (no utilizada directamente en el xhtml, pero útil para lógica interna si se quiere expandir).
- **List<DetalleAuditoriaDTO> listDetallesAuditorias** Detalles más específicos de las auditorías encontradas (usado para vistas más detalladas si se requiere).

Ciclo de Vida

- **@Inject cargarListas()** Método de inicialización que:
 1. Inicializa los checkboxes de los filtros (Creados, Modificados, Consultados, etc.) como activos por defecto.

2. Carga la lista de tablas disponibles para la aplicación actual utilizando el servicio de negocio `AdministracionServiceFacade`.
 3. Generación del Reporte
- **cargarReporte()** Construye el mapa de filtros a partir de `controlAuditoriaVO`, incluyendo el filtro de aplicación.
 1. Llama a `reporteControlAuditoriasList(filters)` en el facade para obtener los datos a mostrar.
 2. Maneja errores mediante mensajes `FacesMessage` y registros con `org.apache.log4j.Logger`.

Mensajes y Manejo de Errores

Se utiliza `FacesContext` para mostrar mensajes al usuario y `log4j` para registrar errores en la carga de datos, garantizando trazabilidad y fácil depuración.

Getters y Setters

Incluye los métodos estándar para acceder y modificar los atributos del controlador, como:

- **`getControlAuditoriaVO()` / `setControlAuditoriaVO()`**
- **`getListaUsuarios()` / `setListaUsuarios()`**
- **`getListaTablas()` / `setListaTablas()`**
- **`getListReporteControlAuditoria()` / `setListReporteControlAuditoria()`**
- **`getListAuditorias()` / `setListAuditorias()`**
- **`getListDetallesAuditorias()` / `setListDetallesAuditorias()`**

*Controlador de Conversación **ReporteControlAuditoriaConvController.java***

El ReporteControlAuditoriaConvController es un controlador de conversación en una aplicación web basada en Java EE con PrimeFaces. Maneja la gestión de entidades, proporcionando las diferentes funcionalidades de la pantalla y los registros en la interfaz de usuario.

Paquete y Dependencias

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

Dependencias Principales:

```
@Inject
private ReporteControlAuditoriaViewController reporteControlAuditoriaViewController;
@Inject
private ConfigApplController configApplController;
@Inject
private ColcienciasSessController colcienciasSessController;
@EJB
private ConfiguracionServiceFacade configuracionServiceFacade;
@EJB
private AdministracionServiceFacade administracionServiceFacade;
```

- ReporteControlAuditoriaViewController: Controlador de vista para la gestión de ReporteControlAuditoriaVO.
- ConfigApplController: Controlador de configuración de la aplicación.
- ColcienciasSessController: Controlador de sesión para objetos de uso general.
- ConfiguracionServiceFacade: Fachada de servicios que gestiona la configuración general del sistema, incluyendo entidades como parámetros del sistema, tipos de valores, etc.
- AdministracionServiceFacade: Fachada de servicios encargada de la administración de entidades clave como tablas, campos, usuarios y la consulta de auditorías para la generación del reporte.

Atributos

```
private final String programa = "GEST6320";

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

public ReporteControlAuditoriaConvController() {
    this.selected = new ReporteControlAuditoriaVO();
}
```

```
}
```

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

Métodos Principales

cargarAuditorias(Object[] objects, String operacion)

```
public void cargarAuditorias(Object[] objects, String operacion) {
    System.out.println("entra a cargarAuditorias");
    try {
        String usuario = (String) objects[0];
        String tabla = (String) objects[1];
        ReporteControlAuditoriaVO controlAuditoriaVOFilter =
reporteControlAuditoriaViewController.getControlAuditoriaVO();
        System.out.println("aplicacion: " +
configAppController.getAplicacion().getAplicacion());
        System.out.println("usuario: " + usuario);
        System.out.println("tabla: " + tabla);
        System.out.println("operacion: " + operacion);
        System.out.println("fechaInicial: " + controlAuditoriaVOFilter.getFechaInicial());
        System.out.println("fechaFinal: " + controlAuditoriaVOFilter.getFechaFinal());
        //Mapa de filtros
        Map<String, Object> filters = new HashMap<>();
        filters.put("aplicacion", configAppController.getAplicacion().getAplicacion());
        filters.put("usuario", usuario);
        filters.put("tabla", tabla);
        filters.put("operacion", operacion);
        if (controlAuditoriaVOFilter.getFechaInicial() != null) {
            filters.put("fechaInicial", controlAuditoriaVOFilter.getFechaInicial());
        }
        if (controlAuditoriaVOFilter.getFechaFinal() != null) {
            Calendar calendarFinal = Calendar.getInstance();
            calendarFinal.setTime(controlAuditoriaVOFilter.getFechaFinal());
            // Establece la hora del calendario a 23:59
            calendarFinal.set(Calendar.HOUR_OF_DAY, 23);
            calendarFinal.set(Calendar.MINUTE, 59);
            calendarFinal.set(Calendar.SECOND, 59);
            filters.put("fechaFinal", calendarFinal.getTime());
        }
        List<AuditoriaDTO> listaAuditorias = administracionServiceFacade.auditoriasList(-1,
-1, "fechaHora", "A", filters);
        reporteControlAuditoriaViewController.setListAuditorias(listaAuditorias);
    } catch (Exception ex) {
        LOG.error("Error al cargar la auditoria ", ex);
    }
}
```

Carga y lista las auditorías asociadas a un usuario, tabla y tipo de operación dentro de un rango de fechas determinado.

- **Parámetros:**
 - **objects[0]:** Usuario que realizó la operación.
 - **objects[1]:** Nombre de la tabla auditada.
 - **operacion:** Tipo de operación auditada (CREAR, MODIFICAR, CONSULTAR, etc.).
- **Lógica de ejecución:**
 1. Obtiene los filtros desde el objeto **ReporteControlAuditoriaVO** y parámetros del método.
 2. Construye un mapa de filtros incluyendo: - Aplicación actual. - Usuario y tabla seleccionados. - Tipo de operación. - Fechas inicial y final, donde la final se ajusta a las 23:59:59.
 3. Ejecuta la consulta paginada con **administracionServiceFacade.auditoriasList(...)** para recuperar la lista de auditorías.
 4. Asigna el resultado al controlador **ReporteControlAuditoriaViewController** mediante **setListAuditorias**.
- **Manejo de errores:**
 - Registra cualquier excepción con **LOG.error(...)** para facilitar trazabilidad.

cargarAuditorias(Object[] objects, String operacion)

```
public void cargarDetallesAuditoria(AuditoriaDTO auditoriaDTO) {
    try {
        System.out.println("auditoriaDTO: " + auditoriaDTO);
        Map<String, Object> filters = new HashMap<>();
        filters.put("auditoriaId", auditoriaDTO.getId()); //filtro por Auditoria ID
        List<DetalleAuditoriaDTO> listaDetalles =
        administracionServiceFacade.detalleAuditoriasList(-1, -1, "campo", "A", filters);
        System.out.println("listaDetalles SIZE: " + listaDetalles.size());
        reporteControlAuditoriaViewController.setListDetallesAuditorias(listaDetalles);
    } catch (Exception ex) {
        LOG.error("Error al cargar los detalles de la auditoria ", ex);
    }
}
```

- Consulta los detalles de una auditoría específica usando su ID.

- Construye filtros y llama a `detalleAuditoriasList()` de `AdministracionServiceFacade`.
- Asigna la lista de detalles al controlador `ReporteControlAuditoriaViewController`.
- Maneja errores con log en caso de excepción

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 define método para obtener el resultado del reporte en Facade EJB
(**AdministracionServiceFacade.java**) para funcionalidad de pantalla*

```
// _____  
//GESTIÓN AUDITORIAS (SISTEMA)  
// _____  
/**  
 * Consulta el listado de Auditorias registrados en la base de datos  
 *  
 * @param filters contiene los filtros al momento de solicitar el listado de  
 * Auditorias  
 * @return lista de Auditorias encontrados en el rango enviado y basado en  
 * los filtros enviados  
 * @throws Exception  
 */  
List<Object[]> reporteControlAuditoriasList(Map<String, Object> filters) throws  
Exception
```

*Se crea en **AuditoriaService.java** EJB método para obtener el resultado del Reporte*

```
List<Object[]> reporteControlAuditoriasList(Map<String, Object> filters) throws  
SystemException;
```

*Se crea en **AuditoriaServiceBean.java** método para obtener el resultado del Reporte*

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

- Internacionalización en español e inglés

```
# ES  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.header  
= Reporte Control de Auditorías  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.breadcrumb  
= Control de Auditorías  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.usuario.label  
= Usuario  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.usuario.placeholder  
= Digite y seleccione el usuario  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.tabla.label  
= Tabla  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.descripcion.label  
= Descripción  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaInicial.label  
= Fecha Inicial  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaInicial.required  
= Debe ingresar la Fecha Inicial  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaFinal.label  
= Fecha Final  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaFinal.required  
= Debe ingresar la Fecha Final  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.registros.header  
= Número de Registros  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.creados.label  
= Creados  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.modificados.label  
= Modificados  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.consultados.label  
= Consultados  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.descargados.label  
= Descargados  
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.borrados.label  
= Borrados
```

```
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.reporte.btn
= Generar Reporte
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.detalle.header
= Detalle de Auditoría

# EN
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.header
= Audit Control Report
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.breadcrumb
= Audit Control
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.usuario.label
= User
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.usuario.placeholder
= Enter and select the user
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.tabla.label
= Table
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.descripcion.label
= Description
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaInicial.label
= Initial Date
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaInicial.required
= You must enter the Starting Date
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaFinal.label
= End Date
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaFinal.required
= You must enter the End Date
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.registros.header
= Number of Records
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.creados.label
= Created
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.modificados.label
= Modified
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.consultados.label
= Consulted
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.descargados.label
= Consulted
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.borrados.label
= Deleted
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.reporte.btn
= Generate Report
sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.detalle.header
= Audit Detail
```

Pestaña de Filtros para generar reporte de Control de Auditoría

La pestaña **"Filtros"** en la interfaz de usuario permite configurar los criterios de búsqueda para generar el reporte de auditoría. Esta sección ofrece componentes de entrada como fechas, usuarios, tablas y operaciones (crear, modificar, consultar, descargar, borrar), permitiendo realizar una búsqueda avanzada. Además, está integrada con el modelo **ReporteControlAuditoriaVO**, asegurando que los filtros aplicados se reflejen directamente en los resultados del reporte.

```
<div class="well form-horizontal" style="padding: 20px">
  <div class="row">
    <div class="col-lg-12">
```

```

        <div class="form-group">
            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.usuario.labe
l']}"
                styleClass="col-lg-3 control-label"
                for="usuario-RepCont"/>
            <div class="col-lg-7">
                <p:autoComplete id="usuario-RepCont"

value="#{reporteControlAuditoriaViewController.controlAuditoriaVO.usuario}"
                converter="usuarioConverter"

completeMethod="#{reporteControlAuditoriaConvController.getListaUsuariosSigp}"
                var="usuario"
                minQueryLength="4"
                itemLabel="#{usuario.usuarioDescripcion}"
                itemValue="#{usuario}"
                forceSelection="true"
                inputStyleClass="form-control"
                style="width: 100%"

placeholder="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.usuari
o.placeholder']}"
            />
        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-lg-12">
        <div class="form-group">
            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.tabla.label'
]}"
                styleClass="col-lg-3 control-label"
                for="tabla"/>
            <div class="col-lg-7">
                <p:selectOneMenu id="tabla"

value="#{reporteControlAuditoriaViewController.controlAuditoriaVO.tabla}"
                converter="tablaConverter"
                styleClass="sbTooltip"
                style="width:100%;"
                autoWidth="false"
                panelStyle="width:180px"
                effect="fade"
                filter="true"
                filterMatchMode="contains"
                >
                    <f:selectItem itemLabel="#{msg['sb.selectOneMenu.registro']}" />
                    <f:selectItems
value="#{reporteControlAuditoriaViewController.listaTablas}"
                        var="tabla"
                        itemLabel="#{tabla.tabla}"
                        itemDescription="#{tabla.descripcionTabla}"
                        itemValue="#{tabla}"
                    />
                </p:selectOneMenu>
            </div>
        </div>
    </div>
</div>

```

```

</div>
<div class="row">
  <div class="col-lg-6">
    <div class="form-group">
      <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaInicial
.label']}"
                                styleClass="col-lg-6 control-label"
                                for="fechaInicial"/>
      <div class="col-lg-5">
        <p:calendar id="fechaInicial"
                    locale="es"

value="#{reporteControlAuditoriaViewController.controlAuditoriaVO.fechaInicial}"
                    required="true"

requiredMessage="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fe
chaInicial.required']}"

                    navigator="true"
                    mask="true"
                    pattern="dd/MM/yyyy"
                    maxdate="0"/>
      </div>
    </div>
  </div>
  <div class="col-lg-6">
    <div class="form-group">
      <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fechaFinal.l
abel']}"
                                styleClass="col-lg-3 control-label"
                                for="fechaFinal"/>
      <div class="col-lg-5">
        <p:calendar id="fechaFinal"
                    locale="es"

value="#{reporteControlAuditoriaViewController.controlAuditoriaVO.fechaFinal}"
                    required="true"

requiredMessage="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.fe
chaFinal.required']}"

                    navigator="true"
                    mask="true"
                    pattern="dd/MM/yyyy"
                    maxdate="0"/>
      </div>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-lg-10 col-lg-offset-1">
    <div class="panel panel-info">
      <div class="panel-heading" style="font-size: 14px; text-align: left;">
        <i class="fa fa-check modal-icon">

#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.registros.header']}
        </i>
      </div>
      <div class="panel-body">
        <div class="row">
          <!-- creados -->

```

```

        <div class="col-lg-4">
            <div class="form-group">
                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.creados.labe
1']}"
                                styleClass="col-lg-6 control-label"
                                for="creados"/>
                <div class="col-lg-4">
                    <h:selectBooleanCheckbox id="creados"

value="#{reporteControlAuditoriaViewController.controlAuditoriaVO.creadosBool}"
                                styleClass="js-switch"
                                >
                    </h:selectBooleanCheckbox>
                </div>
            </div>
        </div>
        <!-- modificados -->
        <div class="col-lg-4">
            <div class="form-group">
                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.modificados.
label']}"
                                styleClass="col-lg-6 control-label"
                                for="modificados"/>
                <div class="col-lg-4">
                    <h:selectBooleanCheckbox id="modificados"

value="#{reporteControlAuditoriaViewController.controlAuditoriaVO.modificadosBool}"
                                styleClass="js-switch"
                                >
                    </h:selectBooleanCheckbox>
                </div>
            </div>
        </div>
        <!-- consultados -->
        <div class="col-lg-4">
            <div class="form-group">
                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.consultados.
label']}"
                                styleClass="col-lg-6 control-label"
                                for="consultados"/>
                <div class="col-lg-4">
                    <h:selectBooleanCheckbox id="consultados"

value="#{reporteControlAuditoriaViewController.controlAuditoriaVO.consultadosBool}"
                                styleClass="js-switch"
                                >
                    </h:selectBooleanCheckbox>
                </div>
            </div>
        </div>
        <!-- descargados -->
        <div class="col-lg-4">
            <div class="form-group">
                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.descargados.
label']}"
                                styleClass="col-lg-6 control-label"
                                for="descargados"/>

```

```

        <div class="col-lg-4">
            <h:selectBooleanCheckbox id="descargados"
value="#{reporteControlAuditoriaViewController.controlAuditoriaVO.descargadosBool}"
                                styleClass="js-switch"
                                >
            </h:selectBooleanCheckbox>
        </div>
    </div>
</div>
<!-- borrados -->
<div class="col-lg-4">
    <div class="form-group">
        <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.borrados.la
el']}"
                                styleClass="col-lg-6 control-label"
                                for="borrados"/>
        <div class="col-lg-4">
            <h:selectBooleanCheckbox id="borrados"
value="#{reporteControlAuditoriaViewController.controlAuditoriaVO.borradosBool}"
                                styleClass="js-switch"
                                >
            </h:selectBooleanCheckbox>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="row">
    <div class="col-lg-6" style="text-align: right;">
        <div class="form-group">
            <p:commandLink id="btnReporte"
process="panelReporteControl"
styleClass="btn btn btn-primary"
update="@form,
#{reporteControlAuditoriaViewController.modulo}Table"
actionListener="#{reporteControlAuditoriaViewController.cargarReporte}"
                                >
                <i class="fa fa-file-archive-o"></i>
            </p:commandLink>
        </div>
    </div>
    <div class="col-lg-6">
        <div class="form-group">
            <p:commandLink id="limpiar"
update="#{reporteControlAuditoriaViewController.modulo}Form"
process="@this"
actionListener="#{reporteControlAuditoriaConvController.limpiarFiltros}"
styleClass="btn btn btn-primary"
onclick="PF('items#{reporteControlAuditoriaViewController.modulo}').clearFilters()"
        </p:commandLink>
        </div>
    </div>
</div>

```

```
>  
    <i class="fa fa-trash-o"></i>  
    #{msg['sb.component.boton.limpiar']}  
</p:commandLink>  
</div>  
</div>  
</div>
```

Componentes Principales

1. **Campo de Usuario (p:autoComplete):** Permite seleccionar un usuario del sistema para filtrar el reporte. La búsqueda es asistida y se activa a partir de 4 caracteres.

- **ID:** *usuario-RepCont*
- **Valor:** *controlAuditoriaVO.usuario*
- **Método:** *getListaUsuariosSigp()*
- **Selección forzada:** *true*
- **Conversor:** *usuarioConverter*
- **Estilo:** *campo autocompletable de ancho completo*

2. **Selección de Tabla (p:selectOneMenu):** Filtra el reporte según la tabla seleccionada. Soporta filtrado interno y muestra descripción adicional.

- **ID:** *tabla*
- **Valor:** *controlAuditoriaVO.tabla*
- **Lista:** *listaTablas*
- **Filtro:** *habilitado con contains*
- **Conversor:** *tablaConverter*
- **Estilo:** *menú desplegable ancho completo con efecto "fade"*

3. Rango de Fechas (p:calendar): Permite seleccionar las fechas inicial y final del período del reporte.

- **ID inicial:** fechaInicial
- **ID final:** fechaFinal
- **Valor:** controlAuditoriaVO.fechaInicial / fechaFinal
- **Formato:** dd/MM/yyyy
- **Validaciones:** campos requeridos, con fecha máxima de hoy

4. Filtros por Tipo de Operación (h:selectBooleanCheckbox): Se incluyen varios checkboxes para filtrar según el tipo de acción realizada

- **Operaciones:** creados, modificados, consultados, descargados, borrados
- **Valor:** controlAuditoriaVO.<operación>Bool
- **Estilo:** js-switch (botones tipo switch personalizados)

5. Botones de Acción

- **Generar Reporte (p:commandLink):** Ejecuta el método cargarReporte y actualiza la tabla de resultados.
 - **ID:** btnReporte
 - **Actualiza:** @form y #{modulo}Table
 - **Estilo:** btn-primary con icono fa-file-archive-o
- **Limpiar Filtros (p:commandLink):** Limpia todos los campos y filtros activos, reseteando la vista.
 - **ID:** limpiar
 - **Método:** limpiarFiltros()
 - **Estilo:** btn-primary con icono fa-trash-o

- **Extra:** limpia filtros visuales del dataTable con `PF('items#{modulo}').clearFilters()`

Pestaña de Tabla de resultados del reporte generado

Esta tabla muestra el reporte de control de auditorías generadas desde el sistema. Se despliega cuando el usuario selecciona la opción "Generar reporte", presentando un resumen detallado de las acciones realizadas sobre los registros de cada módulo.

La información se organiza por usuario, tabla afectada y los distintos tipos de operación: creaciones, modificaciones, consultas, descargas y eliminaciones. Cada cifra en las columnas de acciones es interactiva; al hacer clic, se despliega una ventana emergente con el detalle de las auditorías correspondientes.

```
<!-- TABLA -->
<div class="row">
  <div class="col-lg-12" style="overflow-x: scroll;">
    <p:dataTable id="#{reporteControlAuditoriaViewController.modulo}Table"
      var ="reporteControl"
      value="#{reporteControlAuditoriaViewController.listReporteControlAudito
ria}"
      rowKey="#{reporteControl[0]}"
      paginator="true"
      rows="10"
      lazy="true"
      reflow="true"
      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#{reporteControlAuditoriaViewController.modulo}"
    >
      <f:facet name="header">
        <div style="float:right;padding-top: 5px;">
          <ui:fragment rendered="false">
            <h:commandLink>
              
              <p:dataExporter type="xlsxstream"
target="#{reporteControlAuditoriaViewController.modulo}Table" fileName="auditorias" />
            </h:commandLink>
          </h:commandLink>
        </div>
      </f:facet>
    </p>
  </div>
</div>
```

```


<p:dataExporter type="pdf"
target="#{reporteControlAuditoriaViewController.modulo}Table" fileName="auditorias"/>
</h:commandLink>
<h:commandLink>

<p:dataExporter type="csv"
target="#{reporteControlAuditoriaViewController.modulo}Table" fileName="auditorias"/>
</h:commandLink>
<h:commandLink>

<p:dataExporter type="xml"
target="#{reporteControlAuditoriaViewController.modulo}Table" fileName="auditorias"/>
</h:commandLink>
</ui:fragment>
</div>
</f:facet>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.usuario
.label']}"
style="text-align: center"
width="15%"
>
<h:outputText value="#{reporteControl[0]}/>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.tabla.1
abel']}"
style="text-align: center"
>
<h:outputText value="#{reporteControl[1]}" title="#{reporteControl[1]}/>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.creados
.label']}"
style="text-align: center"
width="12%"
>
<p:commandLink
actionListener="#{reporteControlAuditoriaConvController.cargarAuditorias(reporteControl,
'C')}"
update=":auditoriaForm"
oncomplete="$('#auditoriaDialog').modal();"
>
<h:outputText value="#{reporteControl[3]}"
title="#{msg['sb.component.boton.detalles']}" style="color: #0000ff; "/>
</p:commandLink>
</p:column>
<p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.modific
ados.label']}"
style="text-align: center"
width="12%"
>
<p:commandLink
actionListener="#{reporteControlAuditoriaConvController.cargarAuditorias(reporteControl,
'A')}"
update=":auditoriaForm"

```

```

                                oncomplete="$('#auditoriaDialog').modal();"
                                immediate="true"
                                >
                                <h:outputText value="#{reporteControl[4]}"
title="#{msg['sb.component.boton.detalles']}" style="color: #0000ff; "/>
                                </p:commandLink>
                                </p:column>
                                <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.consult
ados.label']}"
                                style="text-align: center"
                                width="12%"
                                >
                                <p:commandLink
actionListener="#{reporteControlAuditoriaConvController.cargarAuditorias(reporteControl,
'E')}"
                                update=":auditoriaForm"
                                oncomplete="$('#auditoriaDialog').modal();"
                                immediate="true"
                                >
                                <h:outputText value="#{reporteControl[5]}"
title="#{msg['sb.component.boton.detalles']}" style="color: #0000ff; "/>
                                </p:commandLink>
                                </p:column>
                                <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.descarg
ados.label']}"
                                style="text-align: center"
                                width="12%"
                                >
                                <p:commandLink
actionListener="#{reporteControlAuditoriaConvController.cargarAuditorias(reporteControl,
'D')}"
                                update=":auditoriaForm"
                                oncomplete="$('#auditoriaDialog').modal();"
                                immediate="true"
                                >
                                <h:outputText value="#{reporteControl[6]}"
title="#{msg['sb.component.boton.detalles']}" style="color: #0000ff; "/>
                                </p:commandLink>
                                </p:column>
                                <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.reporteControlAuditoria.borrado
s.label']}"
                                style="text-align: center"
                                width="12%"
                                >
                                <p:commandLink
actionListener="#{reporteControlAuditoriaConvController.cargarAuditorias(reporteControl,
'B')}"
                                update=":auditoriaForm"
                                oncomplete="$('#auditoriaDialog').modal();"
                                immediate="true"
                                >
                                <h:outputText value="#{reporteControl[7]}"
title="#{msg['sb.component.boton.detalles']}" style="color: #0000ff; "/>
                                </p:commandLink>
                                </p:column>
                                </p:dataTable>
                                </div>
                                </div>

```

Estructura General

La tabla se encuentra contenida en un <div> con clases de Bootstrap para garantizar la responsividad:

```
<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 principal que define la tabla es:

```
<p:dataTable> (PrimeFaces)
```

Propiedades Principales de <p:dataTable>

```
<p:dataTable id="#{auditoriaViewController.modulo}Table"
    var="auditoria"
    value="#{auditoriaViewController.modelo}"
    rowKey="#{auditoria.auditoriaId}"
    selection="#{auditoriaConvController.selected}"
    selectionMode="single"
    paginator="true"
    rows="10"
    lazy="true"
    reflow="true"
    sortBy="#{auditoria.auditoriaId}"
    sortOrder="DESCENDING"
    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#{auditoriaViewController.modulo}">
```

- Explicación atributos:

Propiedad	Descripción
id	ID único de la tabla, generado dinámicamente según el módulo.
var	Variable usada para iterar sobre los elementos.
value	Fuente de datos (lista de objetos Auditoria).
rowKey	Clave única para identificar cada fila (auditoriaId).
selection	Objeto seleccionado.
selectionMode="single"	Permite seleccionar solo una fila.
paginator="true"	Activa la paginación.
rows="10"	Muestra 10 filas por página.
lazy="true"	Usa LazyDataModel para carga diferida.

reflow="true"	Optimiza la visualización en dispositivos móviles.
sortBy / sortOrder	Orden inicial de los datos (por ID, descendente).
paginatorTemplate	Personaliza la paginación.
rowsPerPageTemplate	Selector de filas por página.
paginatorPosition	Ubica la paginación abajo.
emptyMessage	Mensaje si no hay registros.
widgetVar	Nombre para manipular vía JavaScript (PF('itemsXXX')).

Eventos de Selección de Filas

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

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

- Explicación eventos:

Evento	Acción que realiza
rowSelect	Llama a onRowSelect() cuando se selecciona una fila.
rowUnselect	Llama a onRowUnselect() cuando se deselecta una fila.
update	Refresca la tabla después del evento.
process="@this"	Solo procesa el componente que dispara el evento.

Columnas de la Tabla

Ejemplo:

```
<p:column
  headerText="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.usuario.header']}"
  filterBy="#{auditoria.usuario}"
  sortBy="#{auditoria.usuario}"
  filterMatchMode="contains"
  filterStyle="display:none;"
  style="text-align: center"
  width="15%">
  <h:outputText value="#{auditoria.usuario}" />
</p:column>
```

- Explicación propiedades:

Propiedad	Descripción
-----------	-------------

headerText	Título visible, extraído de mensajes internacionales.
filterBy	Campo para filtrar datos.
sortBy	Habilita ordenamiento por ese campo.
filterMatchMode	Tipo de coincidencia (en este caso, contiene).
filterStyle	Oculto el filtro visual.
width / style	Personaliza diseño y alineación.

Este patrón se repite para otras columnas como:

- tabla.tabla
- terminal
- fechaHora
- operación
- usuarioAdministrador.usuario
- vigente

Acciones en las Filas

```
<p:commandLink style="color: #fff;"
    process="@this"

    title="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn.tooltip']}"
    styleClass="btn btn-xs btn-primary sbTooltip"
    immediate="true"
    actionListener="#{auditoriaConvController.irDetallesAuditoria(auditoria)}">
    <i class="fa fa-building-o">
#{msg['sb.menu.gestion.administracion.auditorias.auditoria.detalle.btn']} </i>
</p:commandLink>
```

- Ejecuta irDetallesAuditoria(auditoria) para redirigir a la vista de detalles.

Dialogo de Auditorías y Detalle de Auditorías

Este componente corresponde al diálogo modal que se despliega al dar clic en los números de una de las columnas de acciones en la tabla de auditoría principal, permitiendo al usuario ver los registros históricos detallados de cambios realizados sobre datos de una tabla específica del sistema.

```
<!-- Dialogo de Auditoria -->
<div class="modal modal-scroll fade bs-modal-md" id="auditoriaDialog" tabindex="-1"
    role="dialog" aria-hidden="true">
```

```

<div class="modal-dialog modal-lg">
  <h:form id="auditoriaForm">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal"><span aria-
hidden="true">&times;</span><span class="sr-
only">#{msg['sb.component.boton.cerrar']}</span></button>
      </div>
      <div class="modal-body form-horizontal">
        <div class="panel panel-info">
          <div class="panel-heading">
            <h3 class="panel-title" style="text-transform: uppercase; text-
align: center">
              <i class="fa fa-check"/>
              #{msg['sb.menu.gestion.administracion.auditorias.auditoria.header']}
            </h3>
          </div>
          <div class="panel-body">
            <p:dataList
value="#{reporteControlAuditoriaViewController.listAuditorias}" var="auditoria"
id="listaAuditorias"
rows="1"
itemStyleClass="text-align:center"
paginator="true"
paginatorPosition="bottom"
currentPageReportTemplate="#{msg['sb.tabla.totalRegi-
stros']}&#160;{totalRecords}"
paginatorTemplate="{CurrentPageReport}
{FirstPageLink} {PreviousPageLink} {PageLinks} {NextPageLink} {LastPageLink}
{RowsPerPageDropdown}"
emptyMessage="#{msg['sb.datatable.registro.empty']}"
type="definition"
lazy="true"
rowIndexVar="rowIndex"
>
              <h:outputFormat
value="#{msg['sb.datatable.registro.numero']}"
style="font-size: 16px">
                <f:param value="#{rowIndex+1}" />
                <f:param
value="#{reporteControlAuditoriaViewController.listAuditorias.size()}" />
              </h:outputFormat>
              <p:spacer/>
              <div class="row">
                <div class="col-lg-8 col-lg-offset-2">
                  <p:panelGrid
columns="2" columnClasses="label,value" layout="grid" styleClass="ui-panelgrid-blank">
                    <h:outputText
value="#{msg['sb.menu.gestion.id']}" />
                    <h:outputText value="#{auditoria.auditoriaId}"
/>
                    <h:outputText
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.usuario.header']}" />
                    <h:outputText value="#{auditoria.usuario}" />
                    <h:outputText
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.tabla.header']}" />
                    <h:outputText value="#{auditoria.tabla.tabla}"
/>
                    <h:outputText
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.fechaHora.header']}" />
                    <h:outputText value="#{auditoria.fechaHora}"

```

```

HH:mm" timeZone="EST"/>
</h:outputText>
<h:outputText
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.operacion.header']}" />
<h:outputText
value="#{msg['listaTipoOperacion.'.concat(auditoria.operacion)]}" />
<h:outputText
value="#{msg['sb.menu.gestion.administracion.auditorias.auditoria.terminal.header']}" />
<h:outputText value="#{auditoria.terminal}" />
</p:panelGrid>
<p:separator/>
<div class="row">
    <div class="col-lg-12" style="text-align:
center;">
        <p:commandButton
value="#{msg['sb.component.boton.detalles']}"
        actionListener="#{reporteCo
ntrolAuditoriaConvController.cargarDetallesAuditoria(auditoria)}"
        update="auditoriaForm"
oncomplete="PF('detallesDia
log').show();"
        />
    </div>
</div>
</div>
</div>
</p:dataList>
</div>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-white"
onclick="$('#auditoriaDialog').modal('hide')">#{msg['sb.component.boton.cerrar']}</button>
</div>
</div>
<p:dialog widgetVar="detallesDialog"
header="#{msg['sb.menu.gestion.administracion.auditorias.reporteContro
lAuditoria.detalle.header']}"
width="900"
modal="true"
showEffect="fade"
hideEffect="fade"
resizable="false">
    <p:outputPanel id="panelDetail" style="text-align:center;">
        <p:dataTable var="detalle"
value="#{reporteControlAuditoriaViewController.listDetalles
Auditorias}"
        style="width:100%; margin:auto;">
            <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.detalleAuditoria.campo.header']
}">
                <h:outputText value="#{detalle.campo}" title="#{detalle.campo}"
/>
            </p:column>
            <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.detalleAuditoria.valorAntes.heade
r']}">
                <h:outputText value="#{detalle.valorAntes}"
title="#{detalle.valorAntes}" />
            </p:column>

```

```

        <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.detalleAuditoria.valorDespues.h
eader']}">
            <h:outputText
value="#{detalle.valorDespues}" title="#{detalle.valorDespues}" />
        </p:column>
        <p:column
headerText="#{msg['sb.menu.gestion.administracion.auditorias.detalleAuditoria.texto.header'
]}">
            <h:outputText id="textoId"
value="#{detalle.texto.length() > 250
?
detalle.texto.substring(0,250).concat('...')}
: detalle.texto}"
title="#{detalle.texto}" />
        </p:column>
    </p:dataTable>
</p:outputPanel>
</p:dialog>
</h:form>
</div>
</div>

```

Estructura del Diálogo

```
<div class="modal modal-scroll fade bs-modal-md" id="auditoriaDialog" ...>
```

- Se define un modal Bootstrap con scroll habilitado, diseño "fade" y tamaño medio.
- id="auditoriaDialog" permite que sea llamado desde otras partes del sistema, por ejemplo, con \$('#auditoriaDialog').modal('show').

h:form – Formulario Principal

```
<h:form id="auditoriaForm">
```

- Contenedor de todos los componentes JSF (PrimeFaces).
- Será actualizado dinámicamente cada vez que se selecciona una auditoría desde la vista principal.

p:dataList – Lista Paginada de Auditorías

```
<p:dataList value="#{reporteControlAuditoriaViewController.listAuditorias}" var="auditoria" ...>
```

- Se itera sobre la lista listAuditorias cargada desde el controlador.
- El contenido está paginado de uno en uno (rows="1"), mostrando una auditoría a la vez.
- El diseño busca resaltar la información detallada de cada registro.
- Campos que se muestran:
 - auditoriald: ID de la auditoría.
 - usuario: Usuario que ejecutó la operación.
 - tabla: Nombre de la tabla afectada.
 - fechaHora: Fecha y hora del evento (formateada dd/MM/yyyy HH:mm).
 - operacion: Tipo de operación (INSERT, UPDATE, DELETE, etc).
 - terminal: IP o equipo desde donde se ejecutó la acción.

Botón "Detalles"

```
<p:commandButton value="#{msg['sb.component.boton.detalles']}"  
actionListener="#{reporteControlAuditoriaConvController.cargarDetallesAuditoria(auditoria)}"  
update="auditoriaForm"  
oncomplete="PF('detallesDialog').show();" />
```

Llama a la función que carga los detalles del registro específico seleccionado.

- Al completarse, abre el diálogo secundario detallesDialog para mostrar los cambios campo por campo.

p:dialog - Modal Secundario de Detalles

```
<p:dialog widgetVar="detallesDialog" ...>
```

Este modal muestra los detalles específicos del cambio ejecutado:

- campo: Nombre del campo modificado.

- valorAntes: Valor anterior antes de la modificación.
- valorDespues: Valor nuevo después de la modificación.
- texto: Texto descriptivo o extendido del cambio. Si es muy largo, se muestra truncado a 250 caracteres con

```
<h:outputText id="textold"
  value="{detalle.texto.length() > 250
    ? detalle.texto.substring(0,250).concat('...')
    : detalle.texto}"
  title="{detalle.texto}" />
```

- El atributo title muestra el contenido completo como tooltip.

Consideraciones Técnicas:

- La vista aprovecha componentes de PrimeFaces (p:) y JSF (h:) para combinar la potencia de back y front.
- Usa lazy="true" para cargar datos de manera eficiente, mejorando el rendimiento.
- El modal se cierra con el botón de Bootstrap o mediante lógica JS (\$('#auditoriaDialog').modal('hide')).

Visualización de la pantalla

Reporte Control de Auditorías

Usuario

Tabla

Fecha Inicial * Fecha Final *

✓ Número de Registros

Creados ☒ Modificados ☒ Consultados ☒

Descargados ☒ Borrados ☒

Usuario	Tabla	Creados	Modificados	Consultados	Descargados	Borrados
DCANGREJO	ACCESO AL SISTEMA			1		
ENT3712024	ACCESO AL SISTEMA			10		
EVA_AMMARTINEZAR	PRY_EVALUACIONES		118			
EVA_AMMARTINEZAR	PRY_EVALUACIONES_CALIFICACION		287			
EVA_ASANCHEZG	PRY_EVALUACIONES		1			
EVA_JFCARDONA	CEN_EVALUACIONES		44			
EVA_JFCARDONA	PRY_EVALUACIONES		187			
EVA_JJMARTINEZ	PRY_EVALUACIONES		118			
EVA_JJMARTINEZ	PRY_EVALUACIONES_CALIFICACION		282			
EVA_LACORDOBA	PRY_EVALUACIONES		157			
Total de registros: 111						

1 2 3 4 5 6 7 8 9 10 10

AUDITORÍAS

Registro 1 de 118

ID	48011
Usuario	EVA_AMMARTINEZAR
Tabla	PRY_EVALUACIONES
Fecha/Hora	02/07/2024 08:51
Operación	Actualizar
Terminal	0:0:0:0:0:0:1

Ver Detalles

Total de registros: 118

1 2 3 4 5 6 7 8 9 10

Cerrar

Detalle de Auditoría

Campo	Valor Antes	Valor Después	Texto LOB
EVALUACION_PROYECTO_ID	130696	130696	
PUNTAJE	0	15	

Creación pantalla de Reporte Auditoría por Campo

Función:

Permite generar un reporte con el detalle de los valores modificados por tabla y campo dentro del sistema.

Principales funcionalidades:

- **Parámetros de generación:** El usuario selecciona los criterios (usuario, tabla, fechas, tipo de operación) para construir el reporte.
- **Descarga directa:** No se visualiza información en pantalla, el resultado se descarga al dar clic en el botón.

Creación nuevo menú de configuración de Campos por DetalleAuditorias

```
-- 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', 'GEST6330', 'Reporte Auditoría por Campo', 'F', null, null,
'/app/administracion/auditorias/reporteAuditoriaCampo.xhtml', null, 'GEST6100', 'fa fa-
film', 'HERCULES', to_date('01-04-2024', 'dd-mm-yyyy'), 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', 'GEST6330', 'SGPS_ADMINISTRADOR', 'H', 'HERCULES', to_date('01-04-
2024', 'dd-mm-yyyy'), null, null);
```

Controlador de Vista ReporteAuditoriaCampoViewController.java

El **ReporteAuditoriaCampoViewController** es un controlador de vista dentro del módulo de administración que gestiona la visualización y manipulación de campos del objeto (**ReporteControlAuditoriaVO**).

```
/*
 * 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.AuditoriaDTO;
import com.sb.colciencias.gestion.business.transfer.CampoDTO;
import com.sb.colciencias.gestion.business.transfer.DetalleAuditoriaDTO;
import com.sb.colciencias.gestion.business.transfer.TablaDTO;
```

```
import com.sb.colciencias.gestion.business.transfer.UsuarioDTO;
import com.sb.colciencias.gestion.business.util.SystemException;
import com.sb.colciencias.gestion.business.valueObjects.ReporteControlAuditoriaVO;
import com.sb.colciencias.gestion.web.controller.application.ConfigApplController;
import com.sb.standard.web.controller.session.GenericSessionController;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.ejb.EJB;
import javax.inject.Inject;
import javax.inject.Named;
import javax.enterprise.context.SessionScoped;
import javax.faces.application.FacesMessage;
import javax.faces.context.FacesContext;

/**
 *
 * @author lsuarez
 */
@Named(value = "reporteAuditoriaCampoViewController")
@SessionScoped
public class ReporteAuditoriaCampoViewController extends
GenericSessionController<ReporteControlAuditoriaVO> implements Serializable {

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

    private ReporteControlAuditoriaVO controlAuditoriaVO;
    private List<UsuarioDTO> listaUsuarios;
    private List<TablaDTO> listaTablas;
    private List<CampoDTO> listaCampos;
    private List<Object> listaValores;
    private List<Object[]> listReporteAuditoriaCampo;
    private List<AuditoriaDTO> listAuditorias;
    private List<DetalleAuditoriaDTO> listDetallesAuditorias;

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

    public ReporteAuditoriaCampoViewController() {
        super(ReporteControlAuditoriaVO.class);
        controlAuditoriaVO = new ReporteControlAuditoriaVO();
    }

    @Inject
    public void cargarListas() {
        try {
            controlAuditoriaVO.setCreadosBool(Boolean.TRUE);
            controlAuditoriaVO.setModificadosBool(Boolean.TRUE);
            controlAuditoriaVO.setConsultadosBool(Boolean.TRUE);
            controlAuditoriaVO.setDescargadosBool(Boolean.TRUE);
            controlAuditoriaVO.setBorradosBool(Boolean.TRUE);

            listaTablas = new ArrayList<>();
            Map<String, Object> filters = new HashMap<>();
            filters.put("aplicacion", configApplController.getAplicacion().getAplicacion());
            int count = administracionServiceFacade.tablasListCount(filters);
```

```

        if (count > 0) {
            listaTablas = administracionServiceFacade.tablasList(-1, -1, "tabla", "A",
filters);
        }
    } catch (Exception ex) {
        LOG.error("Error al cargar el listas ", ex);
    }
}

public void cargarReporte() {
    FacesMessage message = new FacesMessage(FacesMessage.SEVERITY_INFO,
        resolveExpression("#{msg['sb.mensajes.titulo.exito']}"),
        resolveExpression("#{msg['sb.mensajes.transaccion.satisfactorio']}"));
    try {
        Map<String, Object> filters = new HashMap<>();
        //Filtro por aplicacion
        filters.put("aplicacion", configApplController.getAplicacion().getAplicacion());
        //Filtro Busqueda Avanzada
        if (controlAuditoriaVO != null) {
            controlAuditoriaVO.setAplicacion(configApplController.getAplicacion());
            filters.put("controlAuditoriaVOFilter", controlAuditoriaVO);
        }

        listReporteAuditoriaCampo =
administracionServiceFacade.reporteControlAuditoriasList(filters);
    } catch (SystemException ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        LOG.error("Error al cargar el reporte ", ex);
    } catch (Exception ex) {
        message = new FacesMessage(FacesMessage.SEVERITY_ERROR, "ERROR",
resolveExpression(ex.getMessage()));
        LOG.error("Error al cargar el reporte ", ex);
    } finally {
        FacesContext.getCurrentInstance().addMessage(null, message);
    }
}

public ConfigApplController getConfigApplController() {
    return configApplController;
}

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

public ReporteControlAuditoriaVO getControlAuditoriaVO() {
    return controlAuditoriaVO;
}

public void setControlAuditoriaVO(ReporteControlAuditoriaVO controlAuditoriaVO) {
    this.controlAuditoriaVO = controlAuditoriaVO;
}

public List<UsuarioDTO> getListaUsuarios() {
    return listaUsuarios;
}

public void setListaUsuarios(List<UsuarioDTO> listaUsuarios) {
    this.listaUsuarios = listaUsuarios;
}

```

```
}

public List<TablaDTO> getListaTablas() {
    return listaTablas;
}

public void setListaTablas(List<TablaDTO> listaTablas) {
    this.listaTablas = listaTablas;
}

public List<CampoDTO> getListaCampos() {
    return listaCampos;
}

public void setListaCampos(List<CampoDTO> listaCampos) {
    this.listaCampos = listaCampos;
}

public List<Object> getListaValores() {
    return listaValores;
}

public void setListaValores(List<Object> listaValores) {
    this.listaValores = listaValores;
}

public List<Object[]> getListReporteAuditoriaCampo() {
    return listReporteAuditoriaCampo;
}

public void setListReporteAuditoriaCampo(List<Object[]> listReporteAuditoriaCampo) {
    this.listReporteAuditoriaCampo = listReporteAuditoriaCampo;
}

public List<AuditoriaDTO> getListAuditorias() {
    return listAuditorias;
}

public void setListAuditorias(List<AuditoriaDTO> listAuditorias) {
    this.listAuditorias = listAuditorias;
}

public List<DetalleAuditoriaDTO> getListDetallesAuditorias() {
    return listDetallesAuditorias;
}

public void setListDetallesAuditorias(List<DetalleAuditoriaDTO> listDetallesAuditorias)
{
    this.listDetallesAuditorias = listDetallesAuditorias;
}
```

El **ReporteAuditoriaCampoViewController** proporciona una implementación robusta para la generación y visualización del reporte de auditoría por campo. Permite aplicar filtros por usuario, tabla, campo llave, valor llave, rango de fechas y tipo de acción (creados, modificados, consultados, descargados, borrados) para consultar de manera detallada los eventos registrados en el sistema.

Auditorías e Inyección de Dependencias

- **@EJB AdministracionServiceFacade** Servicio de negocio que proporciona las operaciones necesarias para recuperar los datos de auditoría desde la capa de persistencia.
- **@Inject ConfigApplController** Controlador que maneja la configuración de la aplicación, especialmente la aplicación actual desde la cual se están generando los reportes.

Atributos Principales

- **ReporteControlAuditoriaVO controlAuditoriaVO** Objeto de valor que encapsula los filtros seleccionados por el usuario (usuario, tabla, fechas, tipos de acciones, etc.).
- **List<UsuarioDTO> listaUsuarios** Lista de usuarios disponibles en el sistema, usada para selección en los filtros o para visualizaciones más detalladas.
- **List<TablaDTO> listaTablas** Lista de tablas disponibles para la aplicación actual.
- **List<CampoDTO> listaCampos** Lista de campos asociados a la tabla seleccionada. Permite segmentar el análisis por campos específicos.
- **List<Object> listaValores** Lista de valores específicos de los campos seleccionados. Usado para filtrar con más detalle.
- **List<Object[]> listReporteAuditoriaCampo** Resultado del reporte generado a partir de los filtros. Cada arreglo representa una fila del reporte, que incluye información por campo auditado.
- **List<AuditoriaDTO> listAuditorias** Lista general de auditorías encontradas (puede usarse para vistas generales o procesamiento interno).
- **List<DetalleAuditoriaDTO> listDetallesAuditorias** Detalles más específicos de las auditorías, útil para vistas más detalladas o informes internos.

Ciclo de Vida

- **@Inject cargarListas()** Método de inicialización que:
 - Inicializa los checkboxes de filtros (Creados, Modificados, Consultados, etc.) como activos por defecto.
 - Carga la lista de tablas disponibles de la aplicación actual, mediante el servicio `AdministracionServiceFacade`.
- **cargarReporte()** Método principal para generar el reporte:
 - **Construye el mapa de filtros incluyendo el filtro por aplicación y los datos de controlAuditoriaVO.**
 - **Invoca `reporteControlAuditoriasList(filters)` en el facade para recuperar los resultados del reporte.**
 - **Captura y muestra mensajes de éxito o error mediante `FacesContext`, y registra errores con `org.apache.log4j.Logger`.**

Mensajes y Manejo de Errores

- Se usa `FacesContext` para mostrar mensajes amigables al usuario.
- Los errores se registran con `log4j`, permitiendo trazabilidad y depuración del proceso de generación de reportes.

Getters y Setters

Incluye los métodos estándar para acceder y modificar los atributos del controlador:

- **`getControlAuditoriaVO()` / `setControlAuditoriaVO()`**
- **`getListaUsuarios()` / `setListaUsuarios()`**
- **`getListaTablas()` / `setListaTablas()`**
- **`getListaCampos()` / `setListaCampos()`**

- **`getListaValores()` / `setListaValores()`**

- `getListReporteAuditoriaCampo() / setListReporteAuditoriaCampo()`
- `getListAuditorias() / setListAuditorias()`
- `getListDetallesAuditorias() / setListDetallesAuditorias()`

*Controlador de Conversación **ReporteAuditoriaCampoConvController.java***

El **ReporteControlAuditoriaConvController** es un controlador de conversación en una aplicación web basada en Java EE con PrimeFaces. Maneja la gestión de entidades, proporcionando las diferentes funcionalidades de la pantalla.

Paquete y Dependencias

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

Dependencias Principales:

```
@Inject
private ReporteAuditoriaCampoViewController reporteControlAuditoriaViewController;
@Inject
private ConfigApplController configApplController;
@Inject
private ColcienciasSessController colcienciasSessController;
@EJB
private ConfiguracionServiceFacade configuracionServiceFacade;
@EJB
private AdministracionServiceFacade administracionServiceFacade;
```

- **ReporteAuditoriaCampoViewController:** Controlador de vista para la gestión de **ReporteControlAuditoriaVO**.
- **ConfigApplController:** Controlador de configuración de la aplicación.
- **ColcienciasSessController:** Controlador de sesión para objetos de uso general.
- **ConfiguracionServiceFacade:** Fachada de servicios que gestiona la configuración general del sistema, incluyendo entidades como parámetros del sistema, tipos de valores, etc.
- **AdministracionServiceFacade:** Fachada de servicios encargada de la administración de entidades clave como tablas, campos, usuarios y la consulta de auditorías para la generación del reporte.

Atributos

```
private final String programa = " GEST6330";

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

public ReporteAuditoriaCampoConvController() {
    this.selected = new ReporteControlAuditoriaVO();
}
```

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

Métodos Principales

cargarCampos(TablaDTO tablaDTO)

```
public void cargarCampos(TablaDTO tablaDTO) {
    try {
        List<CampoDTO> listaCampos = new ArrayList<>();
        Map<String, Object> filters = new HashMap<>();
        filters.put("aplicacion", configApplController.getAplicacion().getAplicacion());
        //filtro por Aplicacion
        filters.put("tablaId", tablaDTO.getTablaId()); //filtro por Tabla ID
        listaCampos = administracionServiceFacade.campoList(-1, -1, "campo", "A", filters);
        reporteAuditoriaCampoViewController.setListaCampos(listaCampos);
        reporteAuditoriaCampoViewController.getControlAuditoriaVO().setCampoLlave(null);
        reporteAuditoriaCampoViewController.getControlAuditoriaVO().setCampoValor(null);
    } catch (Exception ex) {
        LOG.error("Error al cargar los campos por tabla ", ex);
    }
}
```

Método que permite cargar dinámicamente los campos asociados a una tabla específica seleccionada por el usuario, dentro del contexto de la aplicación actual.

Parámetros:

- **tablaDTO:** Objeto que representa la tabla seleccionada. Se utiliza para identificar qué campos deben ser recuperados.

Lógica de ejecución:

1. Inicializa una nueva lista vacía de campos (listaCampos).

2. Construye un mapa de filtros que incluye:
 - aplicacion: Se obtiene desde ConfigApiController, indicando la aplicación actual.
 - tablaId: Identificador único de la tabla seleccionada.
3. Ejecuta la consulta de campos con administracionServiceFacade.campoList(...) usando los filtros construidos.
4. Asigna el resultado al controlador principal usando setListaCampos(listaCampos).
5. Limpia los valores de los campos campoLlave y campoValor del ReporteControlAuditoriaVO para evitar valores inconsistentes en la interfaz.
6. En caso de error, registra el mensaje usando log4j.

Manejo de errores:

- Registra cualquier excepción con LOG.error(...) para facilitar trazabilidad.

completeValoresCampos(String queryStr)

```
public List<Object> completeValoresCampos(String queryStr) {
    List<Object> listaValores = new ArrayList<>();
    try {
        String tabla =
reporteAuditoriaCampoViewController.getControlAuditoriaVO().getTabla().getTabla();
        String campo =
reporteAuditoriaCampoViewController.getControlAuditoriaVO().getCampoLlave().getCampo();
        System.out.println("tabla: " + tabla);
        System.out.println("campo: " + campo);
        listaValores = administracionServiceFacade.obtenerValoresCampo(tabla, campo,
queryStr.replaceAll("^\\s*", ""));
        System.out.println("listaValores SIZE: " + listaValores.size());
        reporteAuditoriaCampoViewController.setListaValores(listaValores);
    } catch (Exception ex) {
        LOG.error("Error al cargar los valores de campos por tabla ", ex);
    }
    return listaValores;
}
```

Método utilizado para cargar dinámicamente los posibles valores de un campo llave dentro de una tabla seleccionada, permitiendo funcionalidades de autocompletado basadas en una cadena de búsqueda ingresada por el usuario.

Parámetros

- queryStr: Cadena de texto ingresada por el usuario para buscar coincidencias en los valores del campo.

Lógica de Ejecución

1. Recupera el nombre de la tabla y el campo seleccionados desde el ReporteControlAuditoriaVO, que representan el contexto actual del formulario.
2. Elimina los espacios en blanco al inicio del queryStr utilizando replaceAll("^\\s*", "") para evitar errores en la consulta.
3. Ejecuta el método administracionServiceFacade.obtenerValoresCampo(...) para recuperar la lista de valores que coinciden con el criterio de búsqueda:
 - tabla: Nombre de la tabla.
 - campo: Nombre del campo llave.
4. queryStr: Cadena de búsqueda limpia.
5. Asigna la lista de resultados al controlador mediante setListaValores(listaValores) para que puedan ser mostrados en el componente de autocompletado.
6. En caso de producirse una excepción, se registra el error usando log4j.
7. Retorna la lista de valores encontrados.

Manejo de errores:

- Registra cualquier excepción con LOG.error(...) para facilitar trazabilidad.

generarReporte()

```
public void generarReporte() {
    System.out.println("generarReporte");

    PrimeFaces.current().executeScript("PF('blockContent').show();");
    FacesContext context = FacesContext.getCurrentInstance();
    HttpServletResponse response = (HttpServletResponse)
context.getExternalContext().getResponse();
    String nombreReporte = "ReporteAuditoria";
    try {
        ReporteControlAuditoriaVO controlAuditoriaVO =
reporteAuditoriaCampoViewController.getControlAuditoriaVO();
        Date fechaActual = new Date();
        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);
        //Filtro por Aplicacion
        hashMap.put("APLICACION", configApplController.getAplicacion().getAplicacion());
        //Filtro por Tabla
        if (controlAuditoriaVO.getTabla() != null) {
            hashMap.put("TABLA", controlAuditoriaVO.getTabla().getTabla());
        }
        //Filtro por Campo Llave
        if (controlAuditoriaVO.getCampoLlave() != null) {
            hashMap.put("CAMPO_LLAVE", controlAuditoriaVO.getCampoLlave().getCampo());
        }
        //Filtro por Campo Valor
        if (controlAuditoriaVO.getCampoValor() != null) {
            hashMap.put("CAMPO_VALOR", controlAuditoriaVO.getCampoValor().toString());
        }
        //Filtro por rango de fecha inicial
        String formatFechaInicial = null;
        if (controlAuditoriaVO.getFechaInicial() != null) {
            DateFormat df = new SimpleDateFormat("dd/MM/yyyy");
            Date p_fecha_inicial = controlAuditoriaVO.getFechaInicial();
            formatFechaInicial = df.format(p_fecha_inicial);
            hashMap.put("FECHA_INICIAL", formatFechaInicial);
        }
        //Filtro por rango de fecha final
        String formatFechaFinal = null;
        if (controlAuditoriaVO.getFechaFinal() != null) {
            DateFormat df2 = new SimpleDateFormat("dd/MM/yyyy");
            Date p_fecha_final = controlAuditoriaVO.getFechaFinal();
        }
    }
}
```

```

        Calendar calendar = Calendar.getInstance();
        calendar.setTime(p_fecha_final);
        // Establece la hora del calendario a 23:59
        calendar.set(Calendar.HOUR_OF_DAY, 23);
        calendar.set(Calendar.MINUTE, 59);
        calendar.set(Calendar.SECOND, 59);
        formatFechaFinal = df2.format(calendar.getTime());
        hashMap.put("FECHA_FINAL", formatFechaFinal);
    }
    //Registros creados
    hashMap.put("p_creados", controlAuditoriaV0.getCreadosBool() ? "S" : "N");
    //Registros modificados
    hashMap.put("p_modificados", controlAuditoriaV0.getModificadosBool() ? "S" : "N");
    //Registros consultados
    hashMap.put("p_consultados", controlAuditoriaV0.getConsultadosBool() ? "S" : "N");
    //Registros descargados
    hashMap.put("p_descargados", controlAuditoriaV0.getDescargadosBool() ? "S" : "N");
    //Registros borrados
    hashMap.put("p_borrados", controlAuditoriaV0.getBorradosBool() ? "S" : "N");
    //Parametro de sentencia segun filtro seleccionado
    String parametros_where = this.whereConsulta(hashMap);
    hashMap.put("parametros_where", parametros_where + "");
    System.out.println("parametros_where: " + parametros_where);
    //Parametro usuario conectado
    hashMap.put("USUARIO_CONECTADO",
seguridadSessController.getSelected().getDescripcionUsuario() + "");
    //
    byte reporte[] = reporteUtil.generarReporte(hashMap, nombreReporte, "EXCEL",
context);
    response.setContentType("application/xls");
    response.setHeader("Content-Disposition", "attachment; filename=\"" + nombreReporte
+ fechaActual.getTime() + ".xls\"");
    response.setContentLength(reporte.length);
    try (ServletOutputStream outputStream = response.getOutputStream()) {
        outputStream.write(reporte, 0, reporte.length);
        outputStream.flush();
    }
    context.responseComplete();

    PrimeFaces.current().executeScript("PF('blockContent').hide();");
} catch (Exception ex) {
    PrimeFaces.current().executeScript("toast.error(' " +
this.resolveExpression(ex.getMessage()) + "')");
    //LOG.error("Error al generar el reporte de usuarios ", ex);
}
}
}

```

Genera y descarga un archivo Excel (.xls) con los registros de auditoría por campo, aplicando los filtros definidos en la pantalla de búsqueda. Incluye filtros como: aplicación actual, tabla seleccionada, campo llave, campo valor, rango de fechas (ajustando la fecha final a las 23:59:59), y los tipos de operación auditados (crear, modificar, consultar, descargar o eliminar).

Lógica de ejecución

1. Bloquea la interfaz para mostrar al usuario que se está generando el reporte (PF('blockContent').show()).
2. Inicializa contexto y response: Se obtiene el FacesContext y el HttpServletResponse para preparar la descarga del archivo Excel.
3. Crea el mapa de parámetros (hashMap) que se enviará al generador de reportes:
 - Se recuperan los valores de filtros definidos en el ReporteControlAuditoriaVO, como:
 - Aplicación.
 - Tabla seleccionada.
 - Campo llave.
 - Campo valor.
 - Fechas inicial y final (la fecha final se ajusta a las 23:59:59).
 - Tipos de operación auditada (crear, modificar, consultar, descargar, borrar).
4. Agrega los logos al reporte:
 - Consulta los parámetros LOGIN_LOGO y LOGIN_LOGO2 desde base de datos.
 - Extrae la imagen en base64 y la asigna al mapa de parámetros para que se muestren en el encabezado del reporte.
5. Construye una cláusula WHERE dinámica usando el método whereConsulta(...), con base en los filtros aplicados, y la añade al mapa.
6. Agrega el nombre del usuario conectado como metadato adicional en el reporte.
7. Genera el reporte llamando a ReporteUtil.generarReporte(...), pasándole los parámetros, el nombre del reporte (ReporteAuditoria), el formato (EXCEL) y el contexto JSF.
8. Configura la respuesta HTTP para que el navegador inicie automáticamente la descarga del archivo .xls.

9. Escribe el archivo al ServletOutputStream y finaliza la respuesta con context.responseComplete().
10. Desbloquea la interfaz una vez finalizada la generación del reporte.

Manejo de errores:

- Registra cualquier excepción con LOG.error(...) para facilitar trazabilidad.

Este controlador facilita la gestión de campos en la aplicación web, asegurando operaciones seguras y auditadas dentro de la plataforma.

*Se define método para obtener la consulta del reporte en Facade EJB
(AdministracionServiceFacade.java) para funcionalidad de pantalla*

```
/**
 * Llama la funcion que me devuelve la consulta para mostrar el reporte de
 * Auditorias por Campo
 *
 * @param p_aplicacion
 * @param p_tabla
 * @param p_campo_llave
 * @param p_valor_llave
 * @param p_fecha_inicial
 * @param p_fecha_final
 * @return Query devuelto por la funcion
 * @throws Exception
 */
String callSqlAuditoria(String p_aplicacion, String p_tabla, String p_campo_llave, String
p_valor_llave, String p_fecha_inicial, String p_fecha_final) throws Exception; Exception
```

Se crea en AuditoriaService.java EJB método para obtener la consulta del Reporte

```
String callSqlAuditoria(String p_aplicacion, String p_tabla, String p_campo_llave, String
p_valor_llave, String p_fecha_inicial, String p_fecha_final) throws SystemException;
```

Se crea en AuditoriaServiceBean.java método para obtener el resultado del Reporte

```
@Override
public String callSqlAuditoria(String p_aplicacion, String p_tabla, String p_campo_llave,
String p_valor_llave, String p_fecha_inicial, String p_fecha_final) throws SystemException {
```

```
return
this.getStoreManager(AdministracionStoreManager.class).callSqlAuditoria(p_aplicacion,
p_tabla, p_campo_llave, p_valor_llave, p_fecha_inicial, p_fecha_final);
}
```

- Internacionalización en español e inglés

```
# ES
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.header
= Reporte Auditoría por Campo
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.breadcrumb
= Control de Auditorías
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.usuario.label
= Usuario
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.usuario.placeholder
= Digite y seleccione el Usuario
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.tabla.label
= Tabla
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.tabla.required
= Debe seleccionar una Tabla
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.descripcion.label
= Descripción
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoLlave.label
= Campo Llave
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoValor.label
= Valor Llave
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoValor.required
= Debe ingresar y seleccionar el Valor Llave
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoValor.placeholder
= Ingrese y seleccione el Valor Llave
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaInicial.label
= Fecha Inicial
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaInicial.required
= Debe ingresar la Fecha Inicial
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaFinal.label
= Fecha Final
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaFinal.required
= Debe ingresar la Fecha Final
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.registros.header
= Número de Registros
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.creados.label
= Creados
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.modificados.label
= Modificados
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.consultados.label
= Consultados
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.descargados.label
= Descargados
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.borrados.label
= Borrados
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.reporte.btn
= Generar Reporte
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.detalle.header
= Detalle de Auditoría
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.formatoReporte.label
= Formato Reporte
```

```
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.formatoReporte.required
= Debe seleccionar el formato del reporte

# EN
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.header
= Audit Control Report
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.breadcrumb
= Audit Control
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.usuario.label
= User
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.usuario.placeholder
= Enter and select the user
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.tabla.label
= Table
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.tabla.required
= You must select a Table
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.descripcion.label
= Description
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoLlave.label
= Key Field
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoValor.label
= Key Value
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoValor.required
= You must enter and select the Key Value
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoValor.placeholder
= Enter and select the Key Value
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaInicial.label
= Initial Date
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaInicial.required
= You must enter the Starting Date
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaFinal.label
= End Date
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaFinal.required
= You must enter the End Date
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.registros.header
= Number of Records
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.creados.label
= Created
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.modificados.label
= Modified
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.consultados.label
= Consulted
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.descargados.label
= Consulted
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.borrados.label
= Deleted
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.reporte.btn
= Generate Report
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.detalle.header
= Audit Detail
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.formatoReporte.label
= Report Format
sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.formatoReporte.required
= You must select the report format
```

Pestaña de Filtros para generar reporte de Auditoría por Campo

La pestaña **"Filtros"** en la interfaz de usuario permite configurar los criterios de búsqueda para generar el reporte de auditoría por campo. Esta sección ofrece componentes de entrada como fechas, tablas, campo llave, valor llave y operaciones (crear, modificar, consultar, descargar, borrar), permitiendo realizar una búsqueda avanzada. Además, está integrada con el modelo **ReporteControlAuditoriaVO**, asegurando que los filtros aplicados se reflejen directamente en los resultados del reporte.

```
<div class="well form-horizontal" style="padding: 20px">
  <div class="row">
    <div class="col-lg-12">
      <div class="form-group">
        <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.tabla.label']}"
"
          styleClass="col-lg-3 control-label"
          style="text-align: right"
          for="tabla"/>
        <div class="col-lg-7">
          <p:selectOneMenu id="tabla"

value="#{reporteAuditoriaCampoViewController.controlAuditoriaVO.tabla}"
          converter="tablaConverter"
          required="true"

requiredMessage="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.tabla.required']}"
          styleClass="sbTooltip"
          style="width:100%;"
          autoWidth="false"
          panelStyle="width:180px"
          effect="fade"
          filter="true"
          filterMatchMode="contains"
          >
          <p:ajax event="change" update="campollave" process="@this"

listener="#{reporteAuditoriaCampoConvController.cargarCampos(reporteAuditoriaCampoViewContro
ller.controlAuditoriaVO.tabla)}"
          />
          <f:selectItem itemLabel="#{msg['sb.selectOneMenu.registro']}" />
          <f:selectItems
value="#{reporteAuditoriaCampoViewController.listaTablas}"
          var="tabla"
          itemLabel="#{tabla.tabla}"
          itemDescription="#{tabla.descripcionTabla}"
          itemValue="#{tabla}"
          />
          </p:selectOneMenu>
        </div>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-lg-12">
      <div class="form-group">
```

```

        <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoLlave.labe
el']}"
        styleClass="col-lg-3 control-label"
        style="text-align: right"
        for="campoLlave"/>
        <div class="col-lg-7">
            <p:selectOneMenu id="campoLlave"
value="#{reporteAuditoriaCampoViewController.controlAuditoriaVO.campoLlave}"
            converter="campoConverter"
            required="false"
            styleClass="sbTooltip"
            style="width:100%;"
            autoWidth="false"
            panelStyle="width:180px"
            effect="fade"
            filter="true"
            filterMatchMode="contains"
            >
            <p:ajax event="change" update="campoValor" process="@this"
listener="#{reporteAuditoriaCampoConvController.limpiarCampoValor()}"
            >
            </p:ajax>
            <f:selectItem itemLabel="#{msg['sb.selectOneMenu.registro']}" />
            <f:selectItems
value="#{reporteAuditoriaCampoViewController.listaCampos}"
            var="campo"
            itemLabel="#{campo.campo}"
            itemDescription="#{campo.descripcionCampo}"
            itemValue="#{campo}"
            />
            </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.auditorias.reporteAuditoriaCampo.campoValor.labe
el']}"
            styleClass="col-lg-3 control-label"
            style="text-align: right"
            for="campoValor"/>
            <div class="col-lg-7">
                <p:autoComplete id="campoValor"
value="#{reporteAuditoriaCampoViewController.controlAuditoriaVO.campoValor}"
                converter="genericObjectConverter"
                required="false"

requiredMessage="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.camp
oValor.required']}"

completeMethod="#{reporteAuditoriaCampoConvController.completeValoresCampos}"
                var="valor"
                itemLabel="#{valor.toString()}"
                itemValue="#{valor.toString()}"

```

```

forceSelection="true"
inputStyleClass="form-control"
panelStyleClass="custom-panel"
style="width: 100%;"
queryDelay="50"
maxResults="10"

disabled="#{reporteAuditoriaCampoViewController.controlAuditoriaV0.campoLlave == null}"

placeholder="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.campoValor.placeholder']}"
    >
    </p:autoComplete>
    </div>
    </div>
    </div>
    </div>
    <div class="row">
        <div class="col-lg-6">
            <div class="form-group">
                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaInicial.label']}"
                styleClass="col-lg-6 control-label"
                style="text-align: right"
                for="fechaInicial"/>
            <div class="col-lg-5">
                <p:calendar id="fechaInicial"
                    locale="es"

value="#{reporteAuditoriaCampoViewController.controlAuditoriaV0.fechaInicial}"
                    required="true"

requiredMessage="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaInicial.required']}"
                    navigator="true"
                    mask="true"
                    pattern="dd/MM/yyyy"
                    maxdate="0"/>
            </div>
        </div>
        <div class="col-lg-6">
            <div class="form-group">
                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaFinal.label']}"
                styleClass="col-lg-3 control-label"
                style="text-align: right"
                for="fechaFinal"/>
            <div class="col-lg-5">
                <p:calendar id="fechaFinal"
                    locale="es"

value="#{reporteAuditoriaCampoViewController.controlAuditoriaV0.fechaFinal}"
                    required="true"

requiredMessage="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.fechaFinal.required']}"
                    navigator="true"
                    mask="true"

```

```

pattern="dd/MM/yyyy"
maxdate="0"/>

</div>
</div>
</div>
<p:spacer/>
<div class="row">
  <div class="col-lg-10 col-lg-offset-1">
    <div class="panel panel-info">
      <div class="panel-heading" style="font-size: 14px; text-align: left;">
        <i class="fa fa-check modal-icon">

#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.registros.header']]
      </i>
    </div>
    <div class="panel-body">
      <div class="row">
        <!-- creados -->
        <div class="col-lg-4">
          <div class="form-group">
            <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.creados.label'
]]}"
            styleClass="col-lg-6 control-label"
            for="creados"/>
            <div class="col-lg-4">
              <h:selectBooleanCheckbox id="creados"

value="#{reporteAuditoriaCampoViewController.controlAuditoriaV0.creadosBool}"
              styleClass="js-switch"
              >
            </h:selectBooleanCheckbox>
          </div>
        </div>
      </div>
      <!-- modificados -->
      <div class="col-lg-4">
        <div class="form-group">
          <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.modificados.la
bel']}"
          styleClass="col-lg-6 control-label"
          for="modificados"/>
          <div class="col-lg-4">
            <h:selectBooleanCheckbox id="modificados"

value="#{reporteAuditoriaCampoViewController.controlAuditoriaV0.modificadosBool}"
            styleClass="js-switch"
            >
          </h:selectBooleanCheckbox>
        </div>
      </div>
      <!-- consultados -->
      <div class="col-lg-4">
        <div class="form-group">
          <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.consultados.la
bel']}"
          styleClass="col-lg-6 control-label"

```

```

                                for="consultados"/>
                                <div class="col-lg-4">
                                    <h:selectBooleanCheckbox id="consultados"
value="#{reporteAuditoriaCampoViewController.controlAuditoriaV0.consultadosBool}"
                                styleClass="js-switch"
                                >
                                    </h:selectBooleanCheckbox>
                                </div>
                            </div>
                        </div>
                        <!-- descargados -->
                        <div class="col-lg-4">
                            <div class="form-group">
                                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.descargados.la
bel']}"
                                styleClass="col-lg-6 control-label"
                                for="descargados"/>
                                <div class="col-lg-4">
                                    <h:selectBooleanCheckbox id="descargados"
value="#{reporteAuditoriaCampoViewController.controlAuditoriaV0.descargadosBool}"
                                styleClass="js-switch"
                                >
                                    </h:selectBooleanCheckbox>
                                </div>
                            </div>
                        </div>
                        <!-- borrados -->
                        <div class="col-lg-4">
                            <div class="form-group">
                                <p:outputLabel
value="#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.borrados.label
']}"
                                styleClass="col-lg-6 control-label"
                                for="borrados"/>
                                <div class="col-lg-4">
                                    <h:selectBooleanCheckbox id="borrados"
value="#{reporteAuditoriaCampoViewController.controlAuditoriaV0.borradosBool}"
                                styleClass="js-switch"
                                >
                                    </h:selectBooleanCheckbox>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
            <div class="row">
                <div class="col-lg-6" style="text-align: right;">
                    <div class="form-group">
                        <p:commandLink id="reporte"
                            process="panelReporteControl, fechaInicial, fechaFinal"
actionListener="#{reporteAuditoriaCampoConvController.generarReporte()}"
                            styleClass="btn btn btn-primary sbTooltip"

```

```

                                title="{msg['sb.component.boton.generar.reporte']}"
                                update="{reporteAuditoriaCampoViewController.modulo}Form,
messages"
                                ajax="false"
                                >
                                <i class="fa fa-file-excel-o"></i>
#{msg['sb.menu.gestion.administracion.auditorias.reporteAuditoriaCampo.reporte.btn']}
                                </p:commandLink>
                                </div>
</div>
<div class="col-lg-6">
    <div class="form-group">
        <p:commandLink id="limpiar"
                                update="{reporteAuditoriaCampoViewController.modulo}Form"
                                process="@this"

actionListener="{reporteAuditoriaCampoConvController.limpiarFiltros}"
                                styleClass="btn btn btn-primary"
                                >
                                <i class="fa fa-trash-o"></i>
                                #{msg['sb.component.boton.limpiar']}
                                </p:commandLink>
        </div>
    </div>
</div>
</div>

```

Componentes Principales

1. Selección de Tabla (p:selectOneMenu)

Permite seleccionar la tabla del sistema sobre la que se desea generar el reporte. Al cambiar la tabla, se actualiza dinámicamente el campo llave.

- **ID:** tabla
- **Valor:** controlAuditoriaVO.tabla
- **Lista:** listaTablas
- **Convertor:** tablaConverter
- **Filtro:** habilitado (filter="true", filterMatchMode="contains")
- **Eventos:**
 - **change:** ejecuta cargarCampos(tabla) y actualiza campoLlave

- **Estilo:** menú desplegable de ancho completo (style="width:100%;"), con efecto fade
- **Extras:**
 - Descripciones adicionales con itemDescription="#{tabla.descripcionTabla}"
 - autoWidth="false", panelStyle="width:180px"
 - Campo requerido con mensaje personalizado

2. Selección de Campo Llave (p:selectOneMenu)

Se habilita después de seleccionar la tabla. Permite elegir un campo específico de esa tabla para aplicar el filtro de valor.

- **ID:** campoLlave
- **Valor:** controlAuditoriaVO.campoLlave
- **Lista:** listaCampos
- **Convertor:** campoConverter
- **Filtro:** habilitado (filter="true", filterMatchMode="contains")
- **Eventos:**
 - change: ejecuta limpiarCampoValor() y actualiza campoValor
- **Estilo:** menú desplegable de ancho completo con efecto fade
- **Extras:**
 - Descripciones con itemDescription="#{campo.descripcionCampo}"
 - autoWidth="false", panelStyle="width:180px"

3. Campo de Valor (p:autoComplete)

Campo autocompletable que permite filtrar por el valor del campo seleccionado. Se activa al seleccionar un campo llave.

- **ID:** campoValor
- **Valor:** controlAuditoriaVO.campoValor
- **Método de búsqueda:** completeValoresCampos
- **Convertor:** genericObjectConverter
- **Selección forzada:** true
- **Deshabilitado cuando:** campoLlave == null
- **Estilo:** campo autocompletable de ancho completo con panelStyleClass="custom-panel"
- **Extras:**
 - queryDelay="50", maxResults="10"
 - placeholder definido con mensaje i18n
 - Usa itemLabel y itemValue como valor.toString()

4. Rango de Fechas (p:calendar)

Permite seleccionar la fecha inicial y final para el reporte. Campos obligatorios con validación hasta la fecha actual.

- **ID inicial:** fechaInicial
- **ID final:** fechaFinal
- **Valor:** controlAuditoriaVO.fechaInicial / fechaFinal
- **Formato:** dd/MM/yyyy
- **Validaciones:**

- Ambos campos requeridos
- Fecha máxima: hoy (maxdate="0")

- **Extras:**

- navigator="true", mask="true"
- Localización en español (locale="es")

5. Filtros por Tipo de Operación (h:selectBooleanCheckbox)

Filtros por tipo de acción realizada: creación, modificación, consulta, descarga y borrado. Cada opción es un switch visual.

- **Operaciones:**

- **creados:** controlAuditoriaVO.creadosBool
- **modificados:** controlAuditoriaVO.modificadosBool
- **consultados:** controlAuditoriaVO.consultadosBool
- **descargados:** controlAuditoriaVO.descargadosBool
- **borrados:** controlAuditoriaVO.borradosBool

- **Estilo:** js-switch (botones tipo switch personalizados)

- **Ubicación:** agrupados dentro de un panel panel-info

6. Botón Generar Reporte (p:commandLink)

Ejecuta la generación del reporte con los filtros seleccionados.

- **ID:** reporte
- **Acción:** generarReporte()
- **Process:** panelReporteControl, fechaInicial, fechaFinal

- **Estilo:** btn btn-primary con icono de reporte (fa-file-archive-o)

Visualización de la pantalla

Reporte Auditoría por Campo

Tabla *

Seleccione...

Campo Llave

Seleccione...

Valor Llave

Ingrese y seleccione el Valor Llave

Fecha Inicial *

Fecha Final *

✓ Número de Registros

Creados

Modificados

Consultados

Descargados

Borrados

Generar Reporte

Limpiar

Visualización del reporte

Reporte de Auditoría del Sistema SIGP Ministerio de Ciencia, Tecnología e Innovación Minciencias										
APLICACION:		SIGP WEB								
TABLA:		SIGP_CONVOCATORIAS								
CAMPO:										
VALOR:										
FECHA INICIAL:		02/04/2015								
FECHA FINAL:		11/04/2025								
OPERACIONES:		* CREAR * DESCARGAR * ACTUALIZAR * BORRAR * ENTRAR/CONSULTAR								
AUDITORIA ID	DETALLE AUDITORIA ID	FECHA	OPERACION	USUARIO	TERMINAL	CAMPO	VALOR ANTES	VALOR DESPUES	CÓDIGO VALOR ANTES	CÓDIGO VAL
19292	175861	03/08/2020 14:22:51	ACTUALIZAR	JHUERFANO	186.29.183.169	CONVOCATORIA_ID	1020	1020	1020	1020
19292	175853	03/08/2020 14:22:51	ACTUALIZAR	JHUERFANO	186.29.183.169	AVAL_OBLIGATORIO		N		N
19292	175854	03/08/2020 14:22:51	ACTUALIZAR	JHUERFANO	186.29.183.169	CHEQUEA_APORTE_SENA	N	S	N	S
19292	175855	03/08/2020 14:22:51	ACTUALIZAR	JHUERFANO	186.29.183.169	CONVOCATORIA_REGIONAL	N	S	N	S

Auditoría de inicio y cierre de sesión

Función:

Esta funcionalidad permite registrar los eventos de inicio y cierre de sesión dentro del sistema de auditoría, asociándolos a una tabla virtual denominada ACCESO_AL_SISTEMA. Dado que estos eventos no implican una modificación directa sobre una tabla específica, se utiliza esta referencia simbólica con el fin de mantener la trazabilidad uniforme de todas las acciones del usuario. Cada registro almacena el usuario que realizó la operación, la terminal desde la cual accedió, la fecha y hora del evento, y el tipo de operación efectuada (entrada o salida).

Se definen métodos de registro de auditoría en la clase SeguridadWSImpl

La clase **SeguridadWSImpl** actúa como el controlador principal encargado de gestionar el proceso de autenticación de usuarios dentro del sistema. Es la implementación de la interfaz Seguridad y se encarga de manejar las distintas rutas de inicio de sesión, tanto para usuarios internos (mediante autenticación LDAP) como para usuarios externos (autenticación basada en base de datos).

Además de validar credenciales y establecer las sesiones de usuario, esta clase también registra eventos de auditoría al momento del inicio de sesión, asigna permisos y parámetros personalizados, y realiza validaciones adicionales como el cambio obligatorio de contraseña.

Métodos clave

registrarAuditoria(String accion, String usuario, Date fecha)

```
public void registrarAuditoria(String accion, String usuario, Date fecha) {
    try {
        //Obtener terminal
        HttpServletRequest request = (HttpServletRequest)
FacesContext.getCurrentInstance().getExternalContext().getRequest();
        String ipAddress = request.getHeader("X-FORWARDED-FOR");
        if (ipAddress == null) {
            ipAddress = request.getRemoteAddr();
        }

        //Auditoria Inicio de Sesión
        AuditoriaDTO auditoriaDTO = new AuditoriaDTO();
        auditoriaDTO.setFechaHora(fecha);
        auditoriaDTO.setTerminal(ipAddress);
        auditoriaDTO.setOperacion(accion);
        auditoriaDTO.setUsuario(usuario);
        auditoriaDTO.setAplicacion(configApplController.getAplicacion());
    }
}
```

```
auditoriaDTO.setTabla(appSessController.getTablaAccesoAlSistema());  
administracionServiceFacade.guardarAuditoria(auditoriaDTO);  
} catch (Exception ex) {  
    Logger.getLogger(SeguridadConvController.class.getName()).log(Level.SEVERE,  
null, ex);  
}  
}
```

Registra una auditoría específicamente relacionada con el inicio (entrada) o cierre (salida) de sesión de un usuario, incluyendo la IP del terminal desde el cual se realiza.

Parámetros:

- **acción:** Indica el tipo de operación:
 - "E" para Entrada (inicio de sesión).
 - "S" para Salida (cierre de sesión).
- **usuario:** Nombre de usuario que inicia o cierra sesión.
- **fecha:** Fecha y hora en que ocurrió la acción.

Pasos clave:

1. Obtener IP del cliente:
 - Primero intenta extraer la IP desde la cabecera X-FORWARDED-FOR (por si hay proxy).
 - Si no está disponible, utiliza la IP remota directamente desde request.
2. Preparar el objeto de auditoría (AuditoriaDTO), asigna:
 - Fecha y hora (fechaHora)
 - IP del terminal (terminal)
 - Tipo de operación (E o S)
 - Usuario
 - Aplicación (desde configAppController)
 - Tabla asociada al acceso (desde appSessController)

3. Guardar auditoría:

- Envía el objeto al servicio `administracionServiceFacade.guardarAuditoria(...)` para su persistencia.

4. Manejo de errores:

- Cualquier excepción es capturada y registrada como error severo en el log.

Notas adicionales:

- Este método es parte del flujo de seguridad, útil para trazar los accesos al sistema.
- Usado comúnmente en controladores relacionados con login/logout.

*Se define método de guardar Auditoría en Facade EJB
(**AdministracionServiceFacade.java**) para registrar Auditoría de Sesión*

```
/**
 * Guarda el registro de Auditoria en la base de datos
 *
 * @param AuditoriaDTO
 * @throws Exception
 */
void guardarAuditoria(AuditoriaDTO AuditoriaDTO) throws Exception;
```

*Se implementa método de guardar Auditoría EJB
(**AdministracionFacadeServiceBean.java**) para registrar Auditoría de Sesión*

```
public void guardarAuditoria(AuditoriaDTO auditoriaDTO) throws Exception {
    ObjectConverter<AuditoriaDTO, Auditoria> converter = objectConverterFactory.checkOut();
    auditoriaService.guardarAuditoria(converter.toEntity(auditoriaDTO, Auditoria.class));
    objectConverterFactory.checkIn(converter);
}
```

*Se crea método de guardar auditoría en la interfaz **AuditoriaService.java** para registrar Auditoría de inicio de Sesión.*

```
void guardarAuditoria(Auditoria auditoria) throws SystemException;
```

*Se crea método de guardar auditoría en **AuditoriaServiceBean.java** para registrar Auditoría de inicio de Sesión.*

```
@Override  
public void guardarAuditoria(Auditoria auditoria) throws SystemException {  
    this.crear(auditoria, Auditoria.class);  
}
```

- Guarda en base de datos el objeto de auditoría recibido.
- Se usa para registrar eventos como inicio o cierre de sesión.
- Llama internamente al método crear(...) para persistir.
- Puede lanzar SystemException si falla el guardado.

2. IMPLEMENTACIÓN AMBIENTE DE PRUEBAS SIGP, (No. Obligación Contrato: 2.2.1, 2.2.5, 2,2,6)

Solicitud de requerimiento No. 156699

Implementar un entorno de pruebas funcional para el sistema SIGP utilizando Oracle WebLogic Server 12cR2 y Oracle WebTier sobre Oracle Linux 7. Este entorno permitirá pruebas de integración, despliegue y validación sin impactar la operación en producción.

Detalles Técnicos de Implementación

1. Preparación del Sistema Operativo

Se utilizó una máquina con Oracle Linux 7.6 o superior. Se realizaron las siguientes configuraciones previas como usuario root:

Instalación de paquetes requeridos:

```
yum install -y oracle-database-server-12cR2-preinstall.x86_64
yum install -y binutils gcc gcc-c++ glibc glibc-devel ksh libaio libaio-devel \
              libstdc++ libstdc++-devel sysstat zip unzip \
              curl-devel expat-devel gettext-devel openssl-devel zlib-devel
Ajuste de límites de recursos del sistema:
Archivo /etc/security/limits.conf:
```

```
* soft nfile 2048
* hard nfile 16384
* soft nproc 2047
* hard nproc 16384
Archivo /etc/pam.d/login:
```

```
session required /lib/security/pam_limits.so
Archivo /etc/profile:
```

```
if [ "$USER" = "oracle" ]; then
    ulimit -u 16384 -n 65536
fi
Parámetros del kernel en /etc/sysctl.conf:
```

```
fs.file-max = 6815744
kernel.sem = 250 32000 100 128
kernel.shmmni = 4096
kernel.shmall = 1073741824
kernel.shmmax = 4398046511104
net.ipv4.ip_local_port_range = 9000 65500
Aplicado con: sysctl -p
```

2. Instalación del JDK 8

```
useradd -m -g oinstall -d /home/oracle -s /bin/bash oracle
cd /u01/instaladores
tar -xzf jdk-8u211-linux-x64.tar.gz
chown -R oracle:oinstall jdk1.8.0_212
```

3. Instalación de WebLogic Server 12cR2

```
su - oracle
cd /u01/instaladores
export JAVA_HOME=/u01/instaladores/jdk1.8.0_212
export PATH=$JAVA_HOME/bin:$PATH
java -jar fmw_12.2.1.4.0_wls.jar
Se seleccionó instalación tipo "Skip Auto Updates"
Se definió el directorio /u01/oracle/product/fmw/P_WLS/12.2.1.4
```

4. Creación del Dominio WebLogic

- Usuario: weblogic
- Clave: wLMinCie24
- Dominio: base_domain
- Puerto del AdminServer: 7001
- Se creó el archivo boot.properties para autenticación automática:

```
cd $DOMAIN_HOME/servers/AdminServer/security
vi boot.properties
username=weblogic
password=wLMinCie24
```

5. Instalación y Configuración del WebTier

```
cd /u01/instaladores
./ohs_121200_linux64.bin
```

- Se instaló en /u01/oracle/product/fmw/P_OHS/12.1.2
- Se configuró instancia ohs1 con Node Manager (nodemanager/MinCiencias24)

6. Configuración de DADs (mod_plsql)

Archivo:

/u01/oracle/product/fmw/P_OHS/12.1.2/user_projects/domains/base_domain/config/fmwconfig/components/OHS/instances/ohs1/mod_plsql/dads.conf

Ejemplo:

```
<Location /seg>
  SetHandler pls_handler
  Allow from All
  PlsqlDatabaseUsername sigp_web
  PlsqlDatabasePassword Sigp2025
  PlsqlDatabaseConnectString inga.colciencias.local:1530:sigpc.colciencias.local
  PlsqlNLSLanguage SPANISH_SPAIN.WE8MSWIN1252
  PlsqlAuthenticationMode Basic
  PlsqlDocumentTablename SIGP_WEB.ADM_DOCUMENTOS
  PlsqlDocumentPath docs
</Location>
```

7. Creación de Servicios Linux

Servicio para WebTier: /etc/init.d/httpora

```
chmod 750 /etc/init.d/httpora
chkconfig --add httpora
chkconfig httpora on
Servicio para WebLogic: /etc/systemd/system/wls_adminserver.service
```

```
[Unit]
Description=WebLogic Adminserver Service

[Service]
Type=simple
User=oracle
ExecStart=/u01/oracle/product/fmw/P_WLS/12.2.1.4/user_projects/domains/base_domain/bin/start
WebLogic.sh
ExecStop=/u01/oracle/product/fmw/P_WLS/12.2.1.4/user_projects/domains/base_domain/bin/stopwe
bLogic.sh

[Install]
WantedBy=multi-user.target
```

8. Configuración del Firewall

firewall-cmd --permanent --add-port=7001/tcp

firewall-cmd --reload

9. Configuración del Datasource SIGP

Desde la consola WebLogic (<http://<host>:7001/console>):

- Nombre: SigpDS
- JNDI: jdbc/SigpDS
- URL de conexión: jdbc:oracle:thin:@inga.colciencias.local:1530/sigpc.colciencias.local
- Usuario/clave: sigp_web / Sigp2025
- Se probó y activó correctamente.

10. Despliegue de Aplicaciones

Acciones realizadas desde la consola:

1. Subida de archivo WAR: SIGP.war
2. Target: AdminServer
3. Aplicación desplegada y activa
4. Verificación vía navegador:
<http://<host>:7001/SIGP>

Resultado

El ambiente de pruebas del SIGP fue instalado y verificado satisfactoriamente. Todos los servicios están activos y las aplicaciones operativas. Este entorno está disponible para pruebas funcionales y de validación antes del paso a producción.

3. 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: 15/Marzo/2025 – 15/Abril/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	4
Parametrización de Rúbricas de Evaluación	2
Generación de Reportes	24
Solución de Errores	6
Actualización de Datos	2
Requerimientos	3
Total	41

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
154949	Parametrización: 037-2024 Convocatoria 37 “Macroproyectos Nacionales”.	Solucionado	Producción	[Ver detalle]
155300	Parametrización: 966-2025 CONVOCATORIA COLOMBIA INTELIGENTE: CIENCIA Y TECNOLOGÍAS CUÁNTICAS E	Solucionado	Producción	[Ver detalle]

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
	INTELIGENCIA ARTIFICIAL PARA LOS TERRITORIOS			
155728	Parametrización: Convocatoria 961-2024 Comunidades Energeticas Ajustes Parametrizacion	Solucionado	Producción	[Ver detalle]
155923	Parametrización: Convocatoria Colombia Inteligente 2025 Observaciones TdR (solicitud inicial 155300)	Pendiente Usuario	Pruebas	[Ver detalle]

Solicitud 154949

Asunto	Tiempo total de actividad
SIGP	00:24:14
Descripción	
<p>Secretaría Técnica OCAD</p> <p>Adjuntos</p> <p>lun, 17 mar, 15:48 (hace 3 días)</p> <p>para mí, Iván, Juan, Karen, Miguel</p> <p>Buenas tardes,</p> <p>Se reenvia correo con la información suministrada por el profesional Miguel Ángel Castillo, la cual se remitió para revisión de la Dirección y de esta forma la pudieran enviar para la parametrización con el líder técnico Convocatoria 37.</p> <p>Gracias por su atención,</p> <p>Secretaría Técnica OCAD secretaria_tec_ocad@minciencias.gov.co</p>	

Teléfono: 601 6258480
Av. Calle 26 No. 57 - 83
Torre 8, pisos 2 al 6
Bogotá, Colombia
www.minciencias.gov.co

----- Forwarded message -----

De: Miguel Ángel Castillo Reina <macastillo@minciencias.gov.co>
Date: vie, 7 mar 2025 a las 11:28
Subject: Re: cuadro parametrización Convo 37
To: Daniel Rodrigo Tarquino Mosquera <drtarquino@minciencias.gov.co>
Cc: Iván Rodrigo Luna Castro <irluna@minciencias.gov.co>, Juan Pablo Cuenca Mayorga <jpcuenca@minciencias.gov.co>, Karen Viviana Navarrete Sánchez <kvnavarrete@minciencias.gov.co>

Cordial saludo,

Nos permitimos remitir archivo de "Parametrización del formulario SIGP" para su revisión. Tomando como referencia la información de los Términos de Referencia se introducen algunos cambios (señalados en azul). Así mismo considerando que la información del formulario SIGP resultará fundamental para que el líder de la Convocatoria solicite la información de pares evaluadores, se sugiere incluir las sublíneas temáticas así como información detallada de la atención a las demandas territoriales. También se añadió mayor especificidad a la localización geográfica del proyecto.

De otro lado, identificamos que en los TdR no se listan adecuadamente los Anexos No. 13 y 14 que corresponden según la página web a "Anexo 13. Enfoque diferencial e interseccional" y "Anexo 14. Niveles de madurez tecnológico". Esto para su consideración.

Captura de pantalla 2025-03-06 a la(s) 10.22.40 p.m..png

Sin otro particular,

Miguel Ángel Castillo R. - Juan Puablo Cuenca M.

El jue, 6 mar 2025 a las 16:37, Daniel Rodrigo Tarquino Mosquera
(<drtarquino@minciencias.gov.co>) escribió:

Buenas tardes,

Se remite cuadro de parametrización para su revisión y observaciones.

Recomiendo de manera especial para mañana viernes marzo 7 _ 2025, antes de las
11:00 a.m.

Quedo atento

Daniel Rodrigo Tarquino Mosquera
Asesor
Secretario Técnico del OCAD de Ctel/Supervisión SGR
drtarquino@minciencias.gov.co
Teléfono: 601 6258480
Av. Calle 26 No. 57 - 83
Torre 8, pisos 2 al 6
Bogotá, Colombia
www.minciencias.gov.co

----- Forwarded message -----

De: Miguel Ángel Castillo Reina <macastillo@minciencias.gov.co>

Date: jue, 6 mar 2025 a las 13:11

Subject: Re: Convo 37 anexo 4

To: Maritza Guzman Portilla <mguzman@minciencias.gov.co>, Daniel Rodrigo
Tarquino Mosquera <drtarquino@minciencias.gov.co>

Estimados,

De acuerdo con lo solicitado por ustedes me permito remitir versiones .PDF y .XLSX de las "Demandas Territoriales" del bienio 2023-2024 en formato de Anexo para su revisión e inclusión dentro de los documentos soporte de la Convocatoria 37. Sugiero así mismo actualizar estos documentos en la carpeta Drive donde se encuentran los archivos de la Convocatoria para mejoras en la trazabilidad.

Estaré atento a sus comentarios. Un saludo especial,

M.

El jue, 6 mar 2025 a las 9:24, Miguel Ángel Castillo Reina
(<macastillo@minciencias.gov.co>) escribió:

Estimada Maritza,

Considerando que los Términos de Referencia indican en su página 11 lo siguiente:

(...) Los proyectos presentados deberán elegir o abordar mínimo una (1) de las siguientes líneas temáticas y podrán abordar más de una sublínea temática por línea seleccionada. Participarán al menos tres (3) departamentos de mínimo dos (2) regiones diferentes del SGR. Para cada departamento participante, deberá seleccionarse como mínimo una (1) de las demandas territoriales en cualquier reto

Me permito relacionar las "Demandas Territoriales" definidas para el bienio 2023-2024 y que se encuentran publ

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
20/03/2025 10:18:16	08/04/2025 06:49:18	07/04/2025 18:22:32	08/04/2025 06:49:18

Acción Realizada 154949

1. Revisión de los términos de referencia

- Se analizan los términos de referencia de la **Convocatoria 037 Macroproyectos Nacionales** 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
 - GESTIÓN TERRITORIAL
 - Financiaciones
 - ASIGNACIÓN CTEI SGR – 2023-2024
 - Menús
 - PRY - Función General Proyectos
 - PRY - Información General Proyecto
 - PRY - Cambiar Clave
 - PRY - Proyecto
 - PRY - Roles Entidad
 - PRY - Gran Área OCDE, Área OCDE y Disciplina OCDE
 - PTY – Palabras Claves
 - PRY - Descripciones
 - PRY – Personal
 - PRY – Tipo Personal
 - PRY - Generalidades Personal
 - PRY – Presupuesto
 - PRY – Rubros
 - PRY – Presupuesto SGR

- PRY – Documentos
- PRY - Requisitos
- PRY – Validaciones
- PRY – Grupos
- PRY - Región de Ejecución (múltiples) y Departamentos Priorizados (múltiples)
- PRY - Demandas Territoriales por Departamento Priorizado
- PRY - Región a la que se solicitan recursos (Selección única)
- PRY - Línea Temática / SubLinea Temática
- PRY - Enfoque Diferencial
- 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 - Información Adicional

- Rubros
 - ADMINISTRATIVOS
 - CAPACITACIÓN Y EVENTOS
 - EQUIPOS Y SOFTWARE EQUIPOS
 - GASTOS DE VIAJE
 - INFRAESTRUCTURA
 - MATERIALES, INSUMOS Y DOCUMENTACIÓN
 - OTROS
 - PROTECCIÓN DEL CONOCIMIENTO Y DIVULGACIÓN
 - SEGUIMIENTO MATERIALES
 - SERVICIOS TECNOLÓGICOS Y PRUEBAS MATERIALES
 - TALENTO HUMANO

- Descripciones
 - 1 RESUMEN EJECUTIVO RESUMEN EJECUTIVO
 - 2 ALINEACIÓN CON LA POLÍTICA PÚBLICA ALINEACIÓN CON LA POLITICA PÚBLICA
 - 3 IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA
 - 4 ANTECEDENTES ANTECEDENTES
 - 5 JUSTIFICACIÓN JUSTIFICACIÓN

- 6 ARTICULACIÓN DEL PROYECTO EN RESPUESTA A LOS RETOS ESTRATÉGICOS Y DEMANDAS TERRITORIALES, A PARTIR DE LOS ALCANCES TEMÁTICOS
 - 7 MARCO CONCEPTUAL
 - 8 ANÁLISIS DE PARTICIPANTES
 - 9 POBLACIÓN
 - 10 OBJETIVO GENERAL
 - 11 OBJETIVOS ESPECÍFICOS
 - 12 DESCRIPCIÓN DE PERFILES, ACTIVIDADES Y PLAN DE TRABAJO DEL TALENTO HUMANO A VINCULAR
 - 13 ANÁLISIS DE LAS ALTERNATIVAS
 - 14 METODOLOGÍA PARA DESARROLLAR LA ALTERNATIVA SELECCIONADA
 - 15 CADENA DE VALOR
 - 16 SOSTENIBILIDAD DEL PROYECTO
 - 17 RESULTADOS E IMPACTOS ESPERADOS A NIVEL NACIONAL Y/O REGIONAL
 - 18 CRONOGRAMA
 - 19 ANÁLISIS DE RIESGOS
 - 20 INDICADORES DE GESTIÓN
 - 21 ASPECTOS ÉTICOS
 - 22 IDONEIDAD Y TRAYECTORIA DEL PROPONENTE Y DEMÁS PARTICIPANTES
 - 23 ANÁLISIS DE LICENCIAS Y PERMISOS
 - 24 RESUMEN DE FUENTES DE FINANCIACIÓN
 - 25 BIBLIOGRAFÍA
- Campos (Se ocultan)
- CADENA PRODUCTIVA
 - CIUDAD DE FORMACION
 - CLASIFICACIÓN
 - CODIGO CIU
 - DURACIÓN PROYECTO HORAS
 - EJECUCIÓN CRONOGRAMA
 - ESTADO EMPRESA
 - LINEA TEMATICA
 - LUGAR DE EJECUCIÓN

- NOMBRE CLÚSTER
- PERMISOS
- PROPÓSITO GENERAL
- SECTOR
- TEMÁTICAS ORIENTADORAS POR FOCO ESTRATÉGICO
- TIPO DE FINANCIACIÓN
- TIPO PERSONA
- VALOR INSCRIPCIÓN

- Financiadoras
 - SISTEMA GENERAL DE REGALIAS

- Requisitos
 - Entidades Sector productivo
 - Carta aval entidad Internacional
 - Aval del Comité de Ética de la Investigación o de Bioética
 - Soportes del Anexo 5 Requisitos de Documentos
 - Proyectos de CTel gestionados, ejecutados o en ejecución del proponente y de los aliados (Anexo 5)
 - Presupuesto (Anexo 3)
 - Documento técnico (Anexo 2)
 - Carta unificada de aval, compromiso institucional y modelo de gobernanza (Anexo 1B)
 - Adecuado desempeño entidades aliadas (Diferente al proponente)
 - Adecuado desempeño del proponente

- Roles Entidades
 - PROPONENTE
 - ALIADO

- Roles del Personal
 - EQUIPO FORMULADOR

- Líneas temáticas de la Convocatoria
 - Línea Temática 1. Proyectos de CTel para lograr un aprovechamiento sano y sostenible de la biodiversidad

Sublíneas Temáticas

- 1.1. Generación de conocimiento científico sobre las diferentes aplicaciones y usos de la biodiversidad en los diferentes sectores productivos, por ejemplo, el sector agroindustrial, salud, químico entre otros.
- 1.2. Desarrollo de modelos, herramientas e instrumentos con un enfoque de sostenibilidad ambiental que permitan la generación, transformación y uso de potenciales productos biobasados.
- 1.3. Alternativas que contribuyan a la solución de problemas ambientales empleando organismos vivos como bacterias, hongos, plantas y enzimas, en aras de promover la sostenibilidad y reducir el impacto humano en el medio ambiente (Biotecnología ambiental).
- 1.4. Economía ambiental para la comprensión, desarrollo, implementación de soluciones y evaluación de la interacción entre las actividades económicas y el medio ambiente.
- Línea Temática 2. Proyectos de CTel para la prevención, mitigación, adaptación y gestión del riesgo climático
 - Sublineas Temáticas
 - 2.1. Ciencias atmosféricas y del clima para la comprensión de la atmósfera terrestre y su interacción con los océanos, los suelos y la biósfera, en consonancia con los fenómenos meteorológicos y su incidencia sobre el planeta, la transición energética y las sociedades humanas.
 - 2.2. CTel aplicada a la hidrología, en tanto evaluación, comprensión y gestión de los procesos de precipitación, evaporación, infiltración, escorrentía, almacenamiento en suelos, aguas subterráneas y cuerpos de agua superficiales como ríos, lagos y océanos.
 - 2.3. Desarrollo de tecnologías sostenibles que reduzcan las emisiones netas de gases de efecto invernadero (emisiones brutas – emisiones capturadas).
 - 2.4. Desarrollo de tecnologías sostenibles que contribuyan a la recuperación de suelos y capacidades

productivas agropecuarias en áreas afectadas por desastres ambientales por acción antrópica o natural.

- 2.5. Desarrollo de tecnologías sostenibles que contribuyan a la resiliencia agropecuaria en territorios históricamente afectados por el cambio climático y por los eventos climáticos como el Niño y La Niña.
 - 2.6. Desarrollo de tecnologías sostenibles para la restauración y conservación de las capacidades ecosistémicas en áreas estratégicas para la protección de cuerpos de agua superficiales y subterráneos.
- Línea Temática 3. Proyectos de CTel para la conservación y gestión sostenible de áreas marinas y costeras.

Sublíneas Temáticas

- 3.1. Monitoreo avanzado de ecosistemas marinos: Desarrollo y uso de sistemas tecnológicos avanzados, como sensores submarinos, vehículos autónomos no tripulados (AUVs) tomas satelitales y plataformas de análisis de datos basadas en inteligencia artificial, para monitorear continuamente los cambios en las condiciones de los ecosistemas marinos y proporcionar información para un manejo adaptativo.
- 3.2. Innovación en estrategias de conservación: Creación de metodologías científicas y tecnologías para la protección de especies marinas en peligro, restauración de hábitats críticos, y establecimiento de áreas marinas protegidas basadas en datos obtenidos mediante herramientas como sistemas de información geográfica (SIG) y modelación de ecosistemas.
- 3.3. Captura y almacenamiento de carbono azul: Investigación en biología y ecología para maximizar el potencial de los ecosistemas marinos costeros (manglares, praderas marinas y marismas) en la captura y almacenamiento de carbono, incluyendo el desarrollo de tecnologías para medir su capacidad de secuestro de carbono de manera precisa y continua.
- 3.4. Tecnologías para la gestión de residuos marinos: Desarrollo de innovaciones tecnológicas para la

recolección, clasificación y tratamiento de residuos sólidos en ecosistemas marinos, incluyendo tecnologías avanzadas para la eliminación de microplásticos y dispositivos automatizados para limpieza costera y marina.

- Línea Temática 4. Proyectos de CTel para las ciencias humanas en concurso con la sostenibilidad ambiental

Sublineas temáticas

- 4.1. Investigación orientada a la identificación de la correlación entre la cultura, el pensamiento, el comportamiento, la historia, el lenguaje, el arte y las formas de organización social, con la degradación del aire, el agua y/o los ecosistemas terrestres y/o marino costeros.
 - 4.2. Ciencia, Tecnología e Innovación para el desarrollo, implementación y apropiación de sistemas de monitoreo social y colectivo que empoderen a las comunidades en justicia ambiental para la adecuada planeación y conservación del aire, el agua y/o los ecosistemas terrestres y/o marino costeros.
 - 4.3. Ciencia, Tecnología e Innovación para el desarrollo, implementación y apropiación de tecnologías amigables para la identificación temprana de riesgos ambientales potencialmente peligrosos y sus impactos en el ordenamiento territorial.
 - 4.4. Investigación orientada a establecer la incidencia de las emergencias climáticas y la degradación de los ecosistemas terrestres y marinos costeros sobre el desplazamiento y las migraciones forzadas.
 - 4.5. Investigación orientada a establecer la correlación entre las emergencias climáticas y la degradación de los ecosistemas terrestres y/o marino costeros, con la pobreza monetaria y las economías ilegales.
- Línea Temática 5. Proyectos de CTel para la aplicación de tecnologías 4.0 para la restauración, conservación y/o monitoreo de ecosistemas

Sublineas Temáticas

- 5.1. Sensores inteligentes para monitorear variables ambientales como calidad del agua, temperatura del suelo o niveles de contaminación en entornos terrestres o submarinos.
- 5.2. Análisis de datos complejos para predecir, por ejemplo, impactos climáticos y planificar restauraciones mediante el uso de inteligencia artificial.
- 5.3. Big data para la recolección y análisis ambiental masivo de datos para entender patrones y tendencias, como por ejemplo el empleo de datos satelitales y drones para evaluar la deforestación y pérdida de hábitats marinos.
- 5.4. Procesos de simulación virtual que muestren los impactos positivos de restaurar un ecosistema terrestre o marino costero empleando realidad aumentada y virtual.
- 5.5. Empleo de robótica avanzada para la automatización de tareas repetitivas y peligrosas, por ejemplo, en los procesos de eliminación de especies invasoras, áreas contaminadas o con aplicación agroecológica.
- 5.6. Empleo de satélites y teledetección para el monitoreo a gran escala de ecosistemas terrestres y marino-costeros, por ejemplo, para la observación de cambios en la cobertura forestal o en las corrientes oceánicas.
- 5.7. Empleo de sistemas de impresión 3D para la fabricación de estructuras artificiales para favorecer la recuperación de ecosistemas, como por ejemplo la creación de arrecifes artificiales para promover la biodiversidad marina.
- 5.8. Percepción remota y restauración de áreas de difícil acceso mediante el uso de, por ejemplo, drones que plantan semillas en áreas degradadas o vehículos submarinos que ayudan en la reforestación de pastos marinos.

- Información complementaria
 - ¿Cuántos Doctores(as) vinculará en su proyecto de CTel?
 - 2 ¿Cuántos Estudiantes de Maestría vinculará en su proyecto de CTel?
 - 3 ¿Cuántos Jóvenes Investigadores e Innovadores (recién graduado) vinculará en su proyecto de CTel?
 - 4 ¿Cuántos Jóvenes Investigadores e Innovadores (estudiante de pregrado) vinculará en su proyecto de CTel?
- 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 154949		
Número de Solicitud	Resumen de Solicitud	
154949	SEGP	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
27/03/2025 17:49:26	27/03/2025 17:49:26	00:11:48
Descripción del usuario		
Notificación manual Buenas tardes.		
Se finalizó la configuración de la convocatoria 037-2024 Convocatoria 37 "Macroproyectos Nacionales" de acuerdo al formato de parametrización de convocatoria diligenciado por ustedes. En estos momentos se encuentra la convocatoria como invitación especial para poder realizar las pruebas, es decir, el acceso es por un link diferente al público y con credenciales de invitado para poder hacer el pre-registro.		
Revisando la configuración, encontramos que ustedes solicitan registrar múltiples líneas temáticas y una sublínea temática. Actualmente el formulario no tiene esa parametrización, sino solo seleccionar una línea y una sublínea (ver sección adjunto 1). Existe otra parametrización que permite seleccionar más de una línea temática pero no permite relacionar una sublínea (ver sección adjunto 2) y esta última es la forma como se han venido manejando las convocatorias anteriores de regalías. Pueden definir la opción que mejor les parezca o en dado caso, la parametrización como ustedes quieren, sería un nuevo desarrollo a ser implementado en el sistema.		
Los datos de acceso son los siguientes:		
URL: https://plataformasigpmiencias.govco:7003/FormularioProyectos/faces/proyecto/invitacionEspecial.xhtml		
Usuario: INV7532025		
Clave: NUAUpQnM		
* Seleccionar la convocatoria "037-2024 Convocatoria 37 "Macroproyectos Nacionales".		
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		

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 154949		
Número de Solicitud	Resumen de Solicitud	
154949	SIGP	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, 58	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
04/04/2025 21:30:41	04/04/2025 21:30:41	00:10:01
Descripción del usuario		
Notificación manual		
Cordial saludo,		
<p>En atención a su solicitud, nos permitimos informar que ya se realizaron los ajustes solicitados en la parametrización de la Convocatoria 037-2024 "Macroproyectos Nacionales" dentro del sistema.</p> <p>No obstante, es importante aclarar que algunos de los ajustes recientemente requeridos no fueron incluidos originalmente en el formato de matriz de parametrización diligenciado por el equipo responsable, el cual constituye la base para realizar la configuración de la convocatoria. Específicamente:</p> <p>1. En el apartado de contenido del proyecto, se observa que en el formato diligenciado inicialmente solo se incluyeron cuatro (4) ítems, a pesar de que en los Términos de Referencia (TdR), numeral 12.1, se detallan más de veinte (20) elementos que deben ser contemplados. Dado que el sistema se configura estrictamente con base en la información suministrada en dicho formato, es responsabilidad del equipo encargado asegurarse de que esté completo y conforme a lo requerido en los TdR, ya que lo que no se relacione en el formato no podrá ser incorporado en la parametrización.</p> <p>2. En cuanto al formulario de registro del tipo de personal, los TdR indican que "Cada proyecto debe incluir seis (6) jóvenes investigadores e innovadores, distribuidos de manera que tres (3) sean estudiantes de pregrado y tres (3) sean profesionales recién graduados". Esta condición no fue reflejada en el formato de parametrización, ya que solo se registraron dos jóvenes investigadores estudiantes sin especificar las restricciones establecidas. Esta información, al no estar consignada, tampoco fue implementada inicialmente.</p> <p>Reiteramos que la configuración del sistema depende directamente de la calidad y completitud del formato entregado. Cualquier condición, restricción o validación adicional debe estar debidamente especificada en dicho formato para ser incluida oportunamente en la configuración. Por ello, es fundamental que este sea diligenciado con todo el detalle necesario desde el inicio del proceso, especialmente cuando existen disposiciones específicas en los TdR.</p> <p>Así mismo, es fundamental tener en cuenta que el proceso de parametrización de una convocatoria requiere tiempos definidos para su desarrollo, los cuales incluyen etapas de configuración, validación, pruebas internas, recepción de observaciones y ajustes posteriores. Estas actividades implican planificación, disponibilidad técnica y humana, así como el respeto por los espacios establecidos para su ejecución. En ese sentido, agradecemos que, para futuras convocatorias, se programen y coordinen con la debida antelación, evitando solicitudes que impliquen respuestas o acciones fuera de los tiempos razonables de trabajo, a fin de garantizar un proceso ordenado y eficiente.</p> <p>Quedamos atentos a cualquier observación adicional o requerimiento que se derive del presente proceso.</p> <p>—</p> <p>Cordialmente,</p> <p>Ing. Leonardo Suárez Baldovino Software Builders Ltda. ✉ lasb@softwarebuilders.co 🌐 www.softwarebuilders.co 📍 Carrera 700 # 51 - 57</p>		

8. Validación final del usuario

- Se espera la aprobación del usuario funcional para confirmar que la convocatoria está parametrizada correctamente.

7/4/25, 18:05 Correo de Minciencias (Ministerio de Ciencia Tecnología e Innovación) - Envío de formato de aceptacion de parametrizacion convocado...



Nidia Janneth León Cabeza <njleon@minciencias.gov.co>

Envío de formato de aceptacion de parametrizacion convocatoria 37

1 mensaje

Sebastián Camilo Jiménez Cortes <scjimenez@minciencias.gov.co>

7 de abril de 2025, 17:29

Para: Nidia Janneth León Cabeza <njleon@minciencias.gov.co>, Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>

Cordial saludo por medio del presente correo envío el formato de aprobación de pruebas de la parametrización de la convocatoria 37 luego de realizadas las pruebas para su respectiva firma, para continuar con el trámite la convocatoria se encuentra publicada en el siguiente link:

<https://minciencias.gov.co/convocatorias/plan-convocatorias-asctei-2023-2024/convocatoria-no-37-macroproyectos-nacionales> con fecha de apertura 31 de diciembre del 2024 al 30 de abril de 2025

--



Sebastián Camilo Jiménez Cortes

Contratista

Dirección de Ciencia

scjimenez@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](#)



Parametrización y Pruebas formulario de mecanismos VIGENTE.pdf

196K

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 037-2024 Convocatoria 37 “Macroproyectos Nacionales” en el sistema SIGP.

Solicitud 155300

Asunto	Tiempo total de actividad
Parametrización Convocatoria Colombia INteligente 2025	00:12:10
Descripción <p>Buenos días</p> <p>De manera atenta, se solicita a la Oficina de Tecnologías y Sistemas de Información atender las necesidades de parametrización que se adjuntan, con el fin de adelantar los ajustes o actividades necesarias para su correcta configuración. Lo anterior para la captura de datos, soportes y anexos requeridos a los proponentes en el marco de la convocatoria *Colombia INteligente 2025 . *</p> <p>Agradezco de antemano su atención y colaboración.</p> <p>--</p> <p>*Emiro Javier Tovar Martínez*</p> <p>Contratista</p> <p>Dirección de Desarrollo Tecnológico e Innovación</p> <p>*ejtovar@minciencias.gov.co <ejtovar@minciencias.gov.co>*</p> <p>Teléfono: 601 6258480 Ext. 6403</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
27/03/2025 12:28:04	12/05/2025 16:26:24	12/05/2025 16:21:34	12/05/2025 16:26:24

Acción Realizada 155300

1. Revisión de los términos de referencia

- Se analizan los términos de referencia de la **Convocatoria 966-2025 CONVOCATORIA COLOMBIA INTELIGENTE: CIENCIA Y TECNOLOGÍAS CUÁNTICAS E INTELIGENCIA ARTIFICIAL PARA LOS TERRITORIOS** para comprender los requisitos y lineamientos.
- Se identifican las condiciones específicas que deben ser parametrizadas en el sistema.

2. Análisis e identificación de la información

- Se definen los datos que deben ser parametrizados en la convocatoria.
- Se establece la estructura del formulario de proyectos para capturar la información requerida.

3. Diligenciamiento de la matriz de parametrización

- Se completa la **matriz de parametrización**, documentando cada campo, regla de validación y configuración necesaria en la convocatoria.
- Se parametriza lo siguiente:
 - Financiaciones
 - 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 - Generalidades Personal
 - PRY - Productos - Generación de nuevo conocimiento
 - PRY - Roles Entidad
 - PRY - Proyecto
 - PRY - Cambiar Clave
 - PRY - Requisitos
 - PRY - Presupuesto
 - PRY - Rubros
 - PRY - Información General Proyecto
 - PRY – Cronograma
 - PRY - Enfoque Diferencial
 - PRY - Descripciones
 - PRY - Productos
 - PRY – Personal
 - PRY – Documentos
 - PRY - Gran Área OCDE, Área OCDE y Disciplina OCDE
 - PRY - Línea Temática / SubLinea Temática
 - PRY - Nivel de Madurez Tecnológica
 - PRY - Información Adicional
- Rubros
- ADECUACIÓN DE LABORATORIOS Y PLANTAS PILOTO
 - APOYO A ESTANCIAS POSTDOCTORALES
 - ASESORÍA ESPECIALIZADA
 - BIBLIOGRAFÍA
 - CAPACITACIÓN
 - CERTIFICACIONES
 - COMPONENTE DE ESTRATEGIA DE FORMACION
 - EMPAQUETAMIENTO TÉCNICO-COMERCIAL E INTEGRACIÓN

- EQUIPOS
- ESTRATEGIA DE APROPIACIÓN Y USO DE LA TECNOLOGÍA
- GASTOS DE DESPLAZAMIENTO
- GASTOS DE PROPIEDAD INTELECTUAL
- MATERIALES E INSUMOS
- PROTOTIPADO Y VALIDACIÓN TÉCNICO-COMERCIAL
- PROTOTIPAJE Y VALIDACIÓN TÉCNICO-COMERCIAL
- SALIDAS DE CAMPO
- SEMILLEROS DE INVESTIGACIÓN
- SERVICIOS TECNOLÓGICOS
- VINCULACIÓN JOVEN INVESTIGADOR / INNOVADOR
- PERSONAL CIENTÍFICO Y TÉCNICO

Se configuran Montos Roles Personal por cada Rubro de tipo Personal

- Descripciones

- 1 JUSTIFICACIÓN DEL EJE Y LA LÍNEA TEMÁTICA
- 2 EQUIPO DE TRABAJO TÉCNICO - CIENTÍFICO
- 3 EQUIPO DE TRABAJO FINANCIERO - ADMINISTRATIVO
- 4 EQUIPO DE TRABAJO ENFOQUE DIFERENCIAL
- 5 GRADO DE NOVEDAD Y DIFERENCIACIÓN
- 6 APLICACIONES Y USUARIOS
- 7 BENEFICIOS Y VENTAJAS PARA LAS COMUNIDADES Y/U ORGANIZACIONES
- 8 RESUMEN DEL PROYECTO
- 9 DESCRIPCIÓN DEL PROBLEMA, DESAFÍO U OPORTUNIDAD IDENTIFICADO EN LAS COMUNIDADES Y/U ORGANIZACIONES
- 10 PROPUESTA DE SOLUCIÓN PARA LAS COMUNIDADES Y/U ORGANIZACIONES
- 11 JUSTIFICACIÓN
- 12 MARCO CONCEPTUAL
- 13 ESTADO DEL ARTE
- 14 OBJETIVOS
- 15 METODOLOGÍA PROPUESTA
- 16 RESULTADOS ESPERADOS
- 17 BENEFICIOS O PRODUCTOS ESPERADOS
- 18 IMPACTO AMBIENTAL

- 19 IMPACTOS DE LOS RESULTADOS EN LAS COMUNIDADES Y/U ORGANIZACIONES
- 20 ESTRATEGIA DE SOSTENIBILIDAD
- 21 BENEFICIOS O PRODUCTOS ESPERADOS, ENTREGABLES PARA LAS COMUNIDADES Y/O ORGANIZACIONES
- 22 ESTRATEGIA DE APROPIACIÓN Y USO DE LA TECNOLOGÍA DESARROLLADA EN LAS COMUNIDADES Y/U ORGANIZACIONES
- 23 COMPONENTE DE ESTRATEGIA DE FORMACION Y USO DE LA TECNOLOGÍA EN LAS COMUNIDADES
- 24 PROPIEDAD INTELECTUAL
- 25 RIESGOS DEL PROYECTO
- 26 BIBLIOGRAFÍA

- Campos (Se ocultan)
 - CADENA PRODUCTIVA
 - CIUDAD DE FORMACION
 - CLASIFICACIÓN
 - CODIGO CIU
 - DURACIÓN PROYECTO HORAS
 - ESTADO EMPRESA
 - LINEA TEMATICA
 - NOMBRE CLÚSTER
 - PERMISOS
 - PROGRAMA NACIONAL DE CTOL PRINCIPAL
 - PROPÓSITO GENERAL
 - SECTOR
 - TEMÁTICAS ORIENTADORAS POR FOCO ESTRATÉGICO
 - TIPO PERSONA
 - VALOR INSCRIPCION

- Financiadoras
 - MINICIENCIAS

- Requisitos
 - Soportes Enfoque Diferencial Requisitos de Documentos
 - Semilleros de investigación certificados de cada joven expedido por la IES, indicado semestre cursando

- Soportes Vinculación de Estudiantes de Maestría
- Soportes Vinculación de doctor para realizar Estancias Posdoctorales
- Soportes Vinculación de Jóvenes Investigadores e Innovadores
- Hoja de Vida Investigador Principal
- Evidencias del trámite o la existencia de permisos y licencias ambientales
- Aval del Comité de Ética de la Investigación o de Bioética de la Institución que presenta el Proyecto
- Carta de Experiencia Empresa Nacional
- Certificado de Existencia y Representación Legal o documento equivalente con vigencia no superior a treinta (30) días para Organizaciones Locales
- Certificado de Existencia y Representación Legal o documento equivalente con vigencia no superior a treinta (30) días para entidad ejecutora o empresa nacional
- Carta de aval unificada y compromiso institucional
- Roles Entidades
 - EJECUTOR
 - COLABORADORA
 - EMPRESA ALIADA
 - ORGANIZACIÓN LOCAL-REGIONAL
- Roles del Personal
 - DESARROLLADOR
 - ESTUDIANTE DE MAESTRÍA
 - INVESTIGADOR PRINCIPAL
 - JOVEN INVESTIGADOR E INNOVADOR
 - PERSONAL DE APOYO
 - PERSONAL TÉCNICO
 - POSDOCTORADO
- Líneas Temáticas
 - 1. INTELIGENCIA ARTIFICIAL
 - Sublínea
 - Aplicación de IA en Diagnóstico Médico y Medicina Personalizada: Uso de modelos de IA en el análisis de imágenes médicas para mejorar la precisión en el diagnóstico temprano de

enfermedades (como cáncer, enfermedades cardiovasculares o trastornos neurológicos), personalizar tratamientos según datos genómicos y clínicos, u optimizar la atención sanitaria a través de la monitorización en tiempo real y la gestión eficiente de recursos.

- Desarrollo de soluciones basadas en IA que fortalezcan la enseñanza de programación en IA y mejoren la experiencia de aprendizaje en entornos formales e informales: Estas herramientas deben contemplar la personalización de contenidos, la reducción de brechas digitales y la promoción de competencias en niños, jóvenes y adultos, impulsando la inclusión y el acceso a oportunidades formativas en IA en todo el territorio.
- Energías Renovables y Transición Energética: Aplicación de IA para el pronóstico, control y uso sostenible de fuentes limpias (solar, eólica, biomasa), complementada con modelos predictivos para la toma de decisiones en redes energéticas, impulsando la transición hacia sistemas más limpios y eficientes.
- Gestión de la Biodiversidad y Bioeconomía: Aplicaciones de IA para clasificar especies, monitorear ecosistemas y reforzar estrategias de conservación, al tiempo que se fomenta la innovación colaborativa con conocimientos locales para transformar recursos biológicos en bienes o servicios de alto valor agregado.
- Gestión de Riesgos y Conservación de Fauna con IA Aeroespacial: Integración de imágenes satelitales y plataformas aeroespaciales (drones, globos estratosféricos) con modelos de IA en tiempo real para la detección temprana de desastres (inundaciones, incendios, deslizamientos) y la protección de especies silvestres en riesgo. Esta combinación de tecnologías facilita la formulación de planes de contingencia ante eventos extremos y el monitoreo de hábitats, contribuyendo al control de la caza ilegal y al uso sostenible de la biodiversidad.
- Sistemas Agroalimentarios Inteligentes: Empleo de IA en agricultura de precisión, gestión de recursos hídricos y trazabilidad de productos, orientado a la soberanía alimentaria.

○ 2. TECNOLOGIAS CUÁNTICAS

Sublinea

- Computación Cuántica aplicada al Análisis de Datos Complejos: Uso de computación cuántica para potenciar la gestión y el procesamiento de grandes volúmenes de datos en agricultura, energía, bioeconomía, salud, entre otros sectores.
 - Comunicaciones Cuánticas: Estudio e implementación de canales cuánticos que faciliten la transmisión de información de forma segura, reduciendo las vulnerabilidades a ataques de interceptación y mejorando la confiabilidad
 - Criptografía Cuántica para la Seguridad de Datos: Desarrollo de protocolos cuánticos que garanticen la inviolabilidad de las comunicaciones y la protección de información sensible en entornos gubernamentales o privados.
 - Desarrollo de Algoritmos Cuánticos para la Inteligencia Artificial: Investigar la integración de la computación cuántica en modelos de aprendizaje automático, acelerando procesos de entrenamiento y toma de decisiones.
 - Optimización Cuántica en Logística y Transporte: Aplicación de algoritmos cuánticos para resolver problemas complejos de planificación de rutas, ubicación de instalaciones y eficiencia en cadenas de suministro.
 - Quantum Sensing y Metrología: Diseño de sensores cuánticos ultra precisos que permitan medir variables ambientales, geológicas o biomédicas con niveles de exactitud imposibles con la tecnología clásica.
 - Simulación Cuántica para la Investigación de Nuevos Materiales: Simular y diseñar materiales avanzados para soluciones ambientales, energéticas o médicas, aprovechando la capacidad de la física cuántica de describir fenómenos a escalas subatómicas.
- Mensajes Personalizados
 - Información Complementaria
 - ¿A qué Región hacen parte las IES, las Empresas Nacionales y Organizaciones locales – regionales que hacen parte a la alianza?

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 Observaciones sobre el formato de parametrización de la convocatoria Colombia Inteligente 2025

Mostrar la actividad para Solicitud 155300		
Número de Solicitud	Resumen de Solicitud	
155300	Parametrización Convocatoria Colombia Inteligente 2025	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
05/04/2025 21:00:59	05/04/2025 21:00:59	00:05:29
Descripción del usuario		
Notificación manual		
Estimados/as,		
Reciban un cordial saludo.		
Revisando el formato de parametrización diligenciado para la convocatoria Colombia Inteligente 2025, se han identificado algunas observaciones que impiden continuar con el proceso de configuración en el sistema. A continuación, se detallan los hallazgos:		
1. Temática del proyecto cuando se selecciona "Otra" como línea temática: El formato solicita registrar una línea temática y su respectiva sublínea. Sin embargo, cuando la línea temática es "Otra", se debe permitir diligenciar un campo de texto de hasta treinta (30) caracteres con la temática específica del proyecto. Actualmente, esta funcionalidad no existe dentro de las opciones parametrizables y se requeriría como un nuevo desarrollo.		
2. Falta de claridad en los roles de las entidades asociadas al proyecto: En la sección "Dirigida a", se mencionan tipos de entidades como Instituciones de Educación Superior, Empresa Nacional, Organizaciones Locales-Regionales y Entidades Colaboradoras, pero no se especifican los roles que estas deben cumplir (por ejemplo: ejecutora, coejecutora, aliada, etc.) ni la cantidad (mínima o máxima) de entidades a registrar por cada rol (si aplica). Esta información es necesaria para realizar una adecuada configuración.		
3. Roles del personal del proyecto no especificados: No se identifican los roles que debe asumir el personal que será vinculado al proyecto (como Investigador Principal, Joven Investigador, Doctor, entre otros), ni la cantidad de personas por rol ni si es obligatorio que estén registrados en CVLAC. Esta información es clave para parametrizar los formularios y validaciones respectivas.		
Dado lo anterior, no es posible continuar con la parametrización de la convocatoria hasta tanto no se cuente con la información completa y clara, conforme a las instrucciones definidas en el formato de parametrización de convocatorias.		
Agradecemos su atención y quedamos atentos a las correcciones o aclaraciones necesarias para avanzar con el proceso.		
...		
Cordialmente,		
Ing. Leonardo Suárez Baldovino Software Builders Ltda. lasb@softwarebuilders.co www.softwarebuilders.co Carrera 700 # 51 - 57		

- Se reciben aclaraciones
-

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 155300		
Número de Solicitud	Resumen de Solicitud	
155300	Parametrización Convocatoria Colombia Inteligente 2025	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
08/04/2025 11:53:20	08/04/2025 11:53:20	00:04:35
Descripción del usuario		
Notificación manual		
Buenos días Emiro Iovar,		
Se finalizó la configuración y parametrización de la Convocatoria Colombia Inteligente 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: INV7552025		
Clave: n3yht8rm		
URL: https://plataformasg.minciencias.gov.co:7003/FormularioProyectos/faces/proyecto/invitacionEspecial.xhtml		
* Seleccionar la convocatoria "Convocatoria Colombia Inteligente 2025".		
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		

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.

20/4/25, 4:57 p.m. Correo de Minciencias (Ministerio de Ciencia Tecnología e Innovación) - SOLICITUD TI - Ajustes Parametrización Convocatoria C...



Nidia Janneth León Cabeza <njleon@minciencias.gov.co>

SOLICITUD TI - Ajustes Parametrización Convocatoria Colombia Inteligente 2025
1 mensaje

Emiro Javier Tovar Martínez <ejtovar@minciencias.gov.co> 16 de abril de 2025, 11:52
Para: Mesa de Servicios Minciencias <mesadeservicios@minciencias.gov.co>
Cc: Nidia Janneth León Cabeza <njleon@minciencias.gov.co>, Iván Rodrigo Luna Castro <irluna@minciencias.gov.co>, Ana Lucia Caicedo Laurido <alcaicedo@minciencias.gov.co>, Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>, José Fernando García Vidal <jfgarciav@minciencias.gov.co>

Buenos días

Luego de hacer las pruebas al formulario de la Convocatoria Colombia Inteligente 2025, se identificaron unos ajustes los cuales solicitamos en el documento adjunto.

Así las cosas, quedamos atentos a los ajustes solicitados para continuar con las pruebas al formulario y garantizar que el formulario aliado con los TdR de la convocatoria.

Saludos,



Emiro Javier Tovar Martínez
Contratista
Dirección de Desarrollo Tecnológico e Innovación
ejtovar@minciencias.gov.co
Teléfono: 601 6258480 Ext. 6403
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](#)

2 adjuntos

 Parametrización convocatoria Colombia Inteligente 2025 - v2.xlsx
344K

 Observaciones de la plataforma V2.docx
326K

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.

Validación final del usuario

- Se espera la aprobación del usuario funcional para confirmar que la convocatoria está parametrizada correctamente.

Fwd: Solicitud de apertura CONVOCATORIA COLOMBIA INTELIGENTE: CIENCIA Y TECNOLOGÍAS CUÁNTICAS E INTELIGENCIA ARTIFICIAL PARA LOS TERRITORIOS Recibidos x

 **Emiro Javier Tovar Martínez** <ejtovar@minciencias.gov.co>
para Nidia, Harold, José, Ana, Iván, mi

vie, 25 abr, 21:36

 Traducir al español

Buenas tardes,

Luego de realizar las pruebas al formulario, amablemente solicitamos la habilitación del formulario SIGP para la convocatoria 966 de 2025: CONVOCATORIA COLOMBIA INTELIGENTE: CIENCIA Y TECNOLOGÍAS CUÁNTICAS E INTELIGENCIA ARTIFICIAL PARA LOS TERRITORIOS.

Apertura: 25 abril 2025
Cierre: 26 mayo 2025 04:00 pm

Se envía adjunto la evidencia de las pruebas realizadas a conformidad.

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 "966-2025 CONVOCATORIA COLOMBIA INTELIGENTE: CIENCIA Y TECNOLOGÍAS CUÁNTICAS E INTELIGENCIA ARTIFICIAL PARA LOS TERRITORIOS."

Solicitud 155728

Asunto	Tiempo total de actividad
SOLICITUD TI Ajustes parametrización	00:15:33
Descripción Ing. Leonardo buen día: Por favor generar los ajustes solicitados a la parametrización de la convocatoria. Una vez finalizada la gestión por favor realizar la notificación al funcionario del área técnica para la revisión correspondiente.	

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: Iván Rodrigo Luna Castro <irluna@minciencias.gov.co>

Date: mié, 2 abr 2025 a las 17:31

Subject: Ajustes parametrización

To: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>, Nidia Janneth León Cabeza <njleon@minciencias.gov.co>

Cc: Oscar Daniel Sánchez Plazas <odsanchez@minciencias.gov.co>, Solbrigitte Merchán Vaca <smerchan@minciencias.gov.co>

"Estimados miembros de la Oficina de Tecnología y de Información (OTSI),

Adjunto a este mensaje encontrará un documento que expone los errores detectados en la parametrización de la convocatoria de Comunidades Energéticas. Solicito comedidamente su apoyo para subsanar dichas inconsistencias, a fin de proceder con la publicación del documento sin dilación.

Les agradezco por su pronta atención y gestión."

--

Iván Rodrigo Luna Castro

Director Técnico

Dirección de Ciencia

irluna@minciencias.gov.co <irluna@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
03/04/2025 11:30:22	12/05/2025 16:17:48		

Acción Realizada 155728

- Se reciben los ajustes de la parametrización de la convocatoria

- Se notifica al usuario de la publicación de la convocatoria en ambiente de pruebas con los ajustes realizados

Mostrar la actividad para Solicitud 155728		
Número de Solicitud	Resumen de Solicitud	
155728	SOLICITUD TI Ajustes parametrización	
Tipo de actividad	Analista	¿Interno?
Notificación manual	Software Builder, SB	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
03/04/2025 14:52:39	03/04/2025 14:52:39	00:11:37
Descripción del usuario		
Notificación manual		
Buenas tardes Oscar Sánchez,		
<p>Los ajustes solicitados a la parametrización de la convocatoria de Comunidades Energéticas han sido realizados y ya se encuentran desplegados en el ambiente de pruebas.</p> <p>Por favor, proceder con la revisión correspondiente. Quedo atento/a a cualquier observación adicional.</p> <p>Saludos,</p> <p>--</p> <p>Cordialmente,</p> <p>Ing. Leonardo Suárez Baldovino Software Builders Ltda. ✉ leandro@softwarebuilders.co www.softwarebuilders.co 📍 Carrera 700 # 51 - 57</p>		

- Se realiza una segunda notificación para realización de pruebas de parametrización de la convocatoria

Mostrar la actividad para Solicitud 155728		
Número de Solicitud	Resumen de Solicitud	
155728	SOLICITUD TI Ajustes parametrización	
Tipo de actividad	Analista	¿Interno?
Notificación manual	León Cabeza, Nidia Janneth	No
Marca de tiempo del sistema	Fecha de la actividad	Tiempo invertido
12/05/2025 16:15:20	12/05/2025 16:15:20	00:02:28
Descripción del usuario		
Notificación manual		
Buen día		
<p>Los ajustes solicitados a la parametrización de la convocatoria Comunidades Energéticas han sido implementados y ya se encuentran desplegados en el ambiente de pruebas.</p> <p>Por favor, proceder con la revisión correspondiente y realizar las pruebas necesarias para validar el correcto funcionamiento. Quedo atento(a) a cualquier observación adicional o resultado que surja durante este proceso.</p> <p>Cordialmente,</p> <p>Nidia Janneth León Cabeza Oficina de Tecnologías y Sistemas de Información</p> <p>Descripción del sistema</p> <p>Notificación manual</p> <p>Notificar "Sánchez Plazas, Oscar Daniel" "Henao Martinez, Kevin Fernando" "Software Builder, SB" "Méndez Guevara, Harold Alexander" "León Cabeza, Nidia Janneth" "Meneses Guastar, Juan Esteban"</p> <p>Título del mensaje: Solicitud 155728 Segunda Notificación manual - Pruebas funcionales de parametrización convocatoria</p>		

- La solicitud se encuentra en estado “PENDIENTE DE USUARIO” para realización de pruebas por parte del usuario funcional

Solicitud 155923

Asunto	Tiempo total de actividad
Parametrizacion Convocatoria Colombla INTeligente 2025	00:07:58
Descripción	
Buenos dias	

De manera atenta y teniendo en cuenta las observaciones realizadas, se solicita a la Oficina de Tecnologías y Sistemas de Información atender las necesidades de parametrización que se adjuntan, con el fin de adelantar los ajustes o actividades necesarias para su correcta configuración. Lo anterior para la captura de datos, soportes y anexos requeridos a los proponentes en el marco de la convocatoria *ColombIA Inteligente 2025 . *

Agradezco de antemano su atención y colaboración.

--

Emiro Javier Tovar Martínez

Contratista

Dirección de Desarrollo Tecnológico e Innovación

ejtovar@minciencias.gov.co <ejtovar@minciencias.gov.co>

Teléfono: 601 6258480 Ext. 6403

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
07/04/2025 11:39:03	09/04/2025 11:07:51	08/04/2025 12:07:51	09/04/2025 11:07:51

Acción Realizada 155923

1. Revisión de los términos de referencia

- Se revisan los ajustes solicitados en la convocatoria

2. Ajustes de parametrización

- Se ajusta en el sistema SIGP la parametrización de la convocatoria

3. Notificación al usuario

- Se notifica al usuario de que los ajustes están para ser validados y se cierra la solicitud, debido a que se procede a llevar trazabilidad en la solicitud inicial 155300

Mostrar la actividad para Solicitud 155923			
Número de Solicitud		Resumen de Solicitud	
155923		Parametrización Convocatoria Colombia Inteligente 2025	
Tipo de actividad		Analista	¿Interno?
Notificación manual		Software Builder, SB	No
Marca de tiempo del sistema		Fecha de la actividad	Tiempo invertido
08/04/2025 12:00:05		08/04/2025 12:00:05	00:06:38
Descripción del usuario			
Notificación manual			
Buen día Emiro Tovar,			
Cordial saludo.			
En atención a la solicitud 155923, me permito informar que la parametrización requerida fue realizada teniendo en cuenta las observaciones indicadas por el usuario. Cabe aclarar que la notificación del despliegue en el ambiente de pruebas fue gestionada previamente a través de la solicitud inicial 155300, y no mediante la presente solicitud.			
Por lo anterior, esta solicitud 155923 se dará por cerrada, con el fin de mantener una única trazabilidad del proceso de parametrización en la mesa de servicios.			
Quedo atento(a) a cualquier inquietud adicional.			
Cordialmente,			
Ing. Leonardo Suárez Baldovino			
Software Builders Ltda.			
lasb@softwarebuilders.co www.softwarebuilders.co			
Carrera 700 # 51 - 57			

estado y la acción realizada:

Parametrización de Rúbricas de Evaluación

Se atendieron las siguientes solicitudes relacionadas con la configuración y ajuste de convocatorias en la plataforma SIGP:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
155015	- Rubrica de Evaluación Convocatoria 958-2024	Solucionado	Producción	[Ver detalle]

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
155970	- Rúbrica de Evaluación Convocatoria 962-2024 SENAINNOVA	Solucionado	Producción	[Ver detalle]

Solicitud 155015

Asunto			Tiempo total de actividad
rúbrica de evaluación de la Convocatoria 958			00:08:14
Descripción			
Solicito la parametrización de la rúbrica de evaluación de la Convocatoria 958			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
20/03/2025 18:35:03	28/03/2025 12:12:38	27/03/2025 12:20:04	28/03/2025 12:12:38

Acción Realizada 155015

- Se realiza la parametrización de la rúbrica de evaluación de la Convocatoria 958-2024 CONV TRANSFORMACIÓN DE LA PLANEACIÓN MINERO ENERGÉTICA PARA LA CONVERGENCIA TERRITORIAL con éxito.
- Se notifica al usuario Sánchez Plazas Oscar Daniel

Solicitud 155970

Asunto	Tiempo total de actividad

Parametrización convocatoria 962		00:20:30	
Descripción			
Solicitamos muy amablemente la parametrización del formulario de evaluación en la plataforma SIGP para la convocatoria No. 962 - Senalnova. Muchas gracias.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
07/04/2025 16:46:36	11/04/2025 11:07:11		

Acción Realizada 155970

- Se realiza la parametrización de la rúbrica de evaluación de la Convocatoria 962-2024 SENAINNOVA con éxito
- Se notifica al usuario funcional Juan Raúl Mendoza Zamudio de que la rúbrica se encuentra para pruebas y se le indica el proceso de como probarla

Generación de Reportes

Se gestionaron diversas solicitudes de generación de reportes, asegurando la disponibilidad de la información requerida:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
154723	- Reporte General Conv 962-2024 SENAINNOVA 17_03_2025	Solucionado		[Ver detalle]
154782	- Reporte de Personal de Proyectos Convocatoria 951	Solucionado	-	[Ver detalle]
154809	- Reporte de Area_linea tematica_palabras claves de Proyectos Convocatoria 958-2024	Solucionado	-	[Ver detalle]

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
154815	- Reportes Cierre Convocatoria 963 - Orquídeas	Solucionado	-	[Ver detalle]
154824	- Reporte Conv 953-2024 19_03_2025	Solucionado	-	[Ver detalle]
155007	- Reporte Conv 953-2024 21_03_2025	Solucionado	-	[Ver detalle]
155026	- Reportes Cierre Convocatoria 953-2024 Comunidades Negras	Solucionado	-	[Ver detalle]
155088	- Reporte Conv 953-2024 25_03_2025	Solucionado	-	[Ver detalle]
155204	- Reporte Contrapartidas Conv 962-2024 SENAINNOVA	Solucionado	-	[Ver detalle]
155503	- Reporte Pre-subsanación de requisitos convocatoria 963-2025 Orquídeas	Solucionado	-	[Ver detalle]
155560	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 01_04_2025	Solucionado	-	[Ver detalle]
155673	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 02_04_2025 17_00	Solucionado	-	[Ver detalle]
155734	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 03_04_2025 17_00	Solucionado	-	[Ver detalle]
155759	- Reporte de Rol Entidades por Convocatoria 962-2024 SENAINNOVA	Solucionado	-	[Ver detalle]
155773	- Reporte general de proyectos a evaluación Conv 962-2024 SENAINNOVA	Solucionado	-	[Ver detalle]
155788	- Solicitud de ficha de propuesta del proyecto 108705	Solucionado	-	[Ver detalle]
155801	- Reporte presupuesto proyectos Conv 951-2024	Solucionado	-	[Ver detalle]
155837	- Informe de SIGP de todos los mecanismos de 2024	Solucionado	-	[Ver detalle]
155882	- Reporte Conv 964-2025 Formación Maestrías Pacífico Nariñense 07_04_2025 17_00	Solucionado	-	[Ver detalle]

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
155983	- Reporte Entidades Proyectos Convocatoria 953-2024 08_04_2025	Solucionado	-	[Ver detalle]
156131	- Reporte Conv 964-2025 Formación Maestrías Pacifico Nariñense 10_04_2025 18_00	Solucionado	-	[Ver detalle]
156214	- Anonimización de Evaluadores Conv 962-2024 SENAINNOVA	Solucionado	-	[Ver detalle]
156216	- Reporte Post-subsanacion de requisitos convocatoria 963-2025 Orquídeas	Solucionado	-	[Ver detalle]
156346	- Reporte Evaluación de la Conv 958-2024	Solucionado	-	[Ver detalle]

Solicitud 154723

Asunto		Tiempo total de actividad	
Reporte SIGP		00:05:01	
Descripción			
Cordial saludo, Leonardo Suárez			
Agradecemos de su colaboración para generar un reporte general de la convocatoria 962 plataforma SIGP, incluyendo el cuadro de observación general y botón de subsanabilidad			
Miguel Bernal			
Equipo de verificación de mecanismos DGR			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

17/03/2025 08:56:40	18/03/2025 11:39:48	17/03/2025 12:39:47	18/03/2025 11:39:48
------------------------	------------------------	------------------------	------------------------

Acción Realizada 154723

Para atender la solicitud de generación del **Reporte General Convocatoria 962-2024 SENAINNOVA**, se llevaron a cabo las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL que permite extraer los proyectos registrados en la convocatoria con ID 11709, filtrando aquellos cuyo estado sea "RADICADO".
- La consulta incluyó los campos requeridos para el reporte general, contemplando además el cuadro de observación general y la información relacionada con la subsanabilidad.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente en la base de datos del aplicativo SIGP, aprovechando los permisos de acceso otorgados.
- Se generó un archivo Excel con los resultados obtenidos, estructurado según las necesidades del reporte.

3. Entrega del Reporte al Usuario Funcional

- Se notificó al usuario funcional Miguel Bernal sobre la disponibilidad del reporte.
- El archivo fue enviado por correo electrónico para su respectiva revisión y uso.

Solicitud 154782

Asunto	Tiempo total de actividad
---------------	----------------------------------

SOLICITUD TI Fwd: Listado del personal relacionado en cada una de las propuestas presentadas de la convocatoria 951	00:05:00
<p>Descripción</p> <p>Para gestión de Software Builders.</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: lun, 17 mar 2025 a las 13:04</p> <p>Subject: Fwd: Listado del personal relacionado en cada una de las propuestas presentadas de la convocatoria 951</p> <p>To: Nidia Janneth León Cabeza <njleon@minciencias.gov.co></p> <p>Buenas tardes Nidia sí por favor me puedes ayudar con este requerimiento</p> <p>Gracias</p> <p>----- Forwarded message -----</p> <p>De: Iván Rodrigo Luna Castro <irluna@minciencias.gov.co></p> <p>Date: lun, 17 mar 2025 a las 11:43</p> <p>Subject: Fwd: Listado del personal relacionado en cada una de las propuestas presentadas de la convocatoria 951</p>	

To: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>
Cc: Oscar Daniel Sánchez Plazas <odsanchez@minciencias.gov.co>

Cordial saludo

Solicito de su colaboración un Informe de SIGP en cuanto al registro y listado del personal asociado a cada una de las propuestas presentadas en la convocatoria 951. Esto para dar respuesta a PQRS.

Quedo atento a sus comentarios

----- Forwarded message -----

De: Oscar Daniel Sánchez Plazas <odsanchez@minciencias.gov.co>

Date: lun, 17 mar 2025 a las 11:28

Subject: Listado del personal relacionado en cada una de las propuestas presentadas de la convocatoria 951

To: Iván Rodrigo Luna Castro <irluna@minciencias.gov.co>

Saludo cordial,

Solicito atentamente el listado del personal asociado a cada una de las propuestas presentadas en la convocatoria 951. Esta información es requerida para dar trámite a una PQRS.

Agradezco su colaboración.

Atentamente,

--

Oscar Daniel Sánchez Plazas

Contratista

Dirección de Ciencia

odsanchez@minciencias.gov.co <odsanchez@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>>

--

Iván Rodrigo Luna Castro

Director Técnico

Dirección de Ciencia

[*irluna@minciencias.gov.co](mailto:irluna@minciencias.gov.co) <irluna@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>>

--

Harold Alexander Méndez Guevara

Jefe Oficina de Tecnologías y Sistemas de Información

<p>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</p> <p>No imprimas si no es necesario. Seamos responsables con la protección del medio ambiente.</p> <p>Declinación de Responsabilidades - Disclaimer <https://storage.googleapis.com/efor-static/CLCS/AVISO%20DE%20CONFIDENCIALIDAD%20-%20DISCLAIMER.pdf></p> <p>--</p> <p>No imprimas si no es necesario. Seamos responsables con la protección del medio ambiente.</p> <p>Declinación de Responsabilidades - Disclaimer <https://storage.googleapis.com/efor-static/CLCS/AVISO%20DE%20CONFIDENCIALIDAD%20-%20DISCLAIMER.pdf></p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
17/03/2025 16:00:39	20/03/2025 08:18:45	19/03/2025 09:18:45	20/03/2025 08:18:45

Acción Realizada 154782

Para atender la solicitud de generación del **Reporte de Listado del Personal Relacionado en las Propuestas de la Convocatoria 951**, se realizaron las siguientes acciones:

1. **Elaboración del Script de Consulta**

- Se diseñó un script SQL para obtener la información del personal vinculado a los proyectos registrados en la convocatoria con ID 11691, filtrando aquellos con estado "REVISADO".
- La consulta incluyó los datos requeridos para identificar el recurso humano asociado a cada propuesta presentada.

2. Ejecución del Script y Generación del Reporte

- Se ejecutó el script directamente en la base de datos del aplicativo SIGP.
- Se generó un archivo Excel con la información extraída, estructurada de acuerdo con los requerimientos funcionales.

3. Entrega del Reporte al Usuario Funcional

- Se notificó a la usuaria funcional Nidia León sobre la disponibilidad del reporte.
- El archivo Excel fue enviado por correo electrónico para su revisión y utilización.

Solicitud 154809

Asunto	Tiempo total de actividad
Solicitud de base de datos de evaluadores - Convocatoria 958	00:09:47
Descripción <p>Buen día ng. Leonardo:</p> <p>Se solicito el apoyo para generar un script que permita la consulta de palabras clave y áreas de conocimiento para realizar un proceso de coincidencia con la base de datos de ScienTI, con el objetivo de optimizar la identificación y categorización de evaluadores que cumplan con los criterios requeridos para su participación en el proceso. Adjunto a este correo electrónico un archivo con las palabras clave consideradas para la selección de estos evaluadores.</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: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>

Date: lun, 17 mar 2025 a las 16:41

Subject: Fwd: Solicitud de base de datos de evaluadores - Convocatoria 958

To: Nidia Janneth León Cabeza <njleon@minciencias.gov.co>, <lasb@softwarebuilders.co>

Buenas tardes, me está llamando el Viceministro Iván Luna con el fin de dar trámite URGENTE para hoy de la base de datos de los evaluadores anonimizada, de acuerdo con lo indicado en el correo de abajo. Lo que observo es que el correo de solicitud es del día 27 de febrero. ¿Qué pasaría?

----- Forwarded message -----

De: Iván Rodrigo Luna Castro <irluna@minciencias.gov.co>

Date: jue, 27 feb 2025 a las 14:05

Subject: Solicitud de base de datos de evaluadores - Convocatoria 958

To: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>

Cc: Oscar Daniel Sánchez Plazas <odsanchez@minciencias.gov.co>, Solbrigitte

Merchán Vaca <smerchan@minciencias.gov.co>

Cordial saludo

Estimado

Me dirijo a usted en seguimiento a la Convocatoria 958 y con el fin de solicitar la base de datos de los evaluadores que pueden participar en la misma. Adjunto a este correo electrónico encontrarán un archivo con las palabras clave que son consideradas para la selección de estos evaluadores.

Agradezco de antemano su colaboración y atención a esta solicitud. Quedo a su disposición en caso de que requiera información adicional.

Cordialmente,

Adjuntos:

- Archivo con las palabras clave

https://docs.google.com/spreadsheets/d/1UGxZSAYASI_puJ0UcMG0JiZVR32zTHv9/edit?usp=sharing&ouid=109308250351627756437&rtpof=true&sd=true

--

Iván Rodrigo Luna Castro

Director Técnico

Dirección de Ciencia

irluna@minciencias.gov.co <irluna@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>>

--

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
18/03/2025 09:33:20	20/03/2025 11:20:16	19/03/2025 12:20:16	20/03/2025 11:20:16

Acción Realizada 154809

En atención a la solicitud relacionada con la **identificación y categorización de evaluadores para la Convocatoria 958-2024 “Transformación de la Planeación Minero Energética para la Convergencia Territorial”**, se llevaron a cabo las siguientes actividades:

1. Elaboración del Script de Consulta

- Se generó un script SQL orientado a extraer las **palabras clave y áreas de conocimiento** asociadas a los proyectos registrados en la convocatoria con ID correspondiente.
- Esta consulta tuvo como objetivo optimizar el proceso de emparejamiento entre proyectos y evaluadores en la base de datos ScienTI.

2. Análisis y Categorización de Evaluadores

- En conjunto con la Ingeniera Erika González, se realizó el análisis de los datos obtenidos y la respectiva categorización de los evaluadores, considerando las áreas de conocimiento y las palabras clave identificadas.
- Se aplicaron criterios técnicos para asegurar la pertinencia de la asignación de evaluadores a los proyectos postulados.

3. Entrega del Reporte Final

- El reporte consolidado fue entregado al Ingeniero Harold Méndez, jefe de la Oficina de Tecnologías de la Información (OTSI), cumpliendo con los requerimientos de la solicitud.

Solicitud 154815

Asunto		Tiempo total de actividad	
Reporte convocatoria 963		00:12:05	
Descripción			
<p>Buenas tardes apreciados compañeros,</p> <p>Agradecemos que se puedan generar los reportes de la CV 963 - Orquídeas, la cual cierra el día de mañana 19 de marzo de 2025 a las 4:00 p.m., en el aplicativo SIGP, con la siguiente información: (1) reporte general, (2) reporte de presupuesto de los proponentes (monto solicitado y contrapartida), (3) información de la Doctora y de la joven investigadora, (4) categoría de enfoque diferencial, (5) CvLAC y (6) eje al que se presenta la propuesta (Total: 6). Es importante que estos insumos sean remitidos, únicamente, en formato Excel para la adecuada revisión de la información.</p> <p>Por favor copiar esta solicitud al Ingeniero Leonardo Suárez</p> <p>Agradecemos de antemano su atención y colaboración para llevar a cabo cada proceso de este 2025.</p> <p>Cordialmente,</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
18/03/2025 15:06:54	21/03/2025 11:37:09	20/03/2025 12:37:09	21/03/2025 11:37:09

Acción Realizada 154815

En respuesta a la solicitud de generación de reportes al cierre de la **Convocatoria 963-2025 "Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas"**, se ejecutaron las siguientes actividades para cumplir con los requerimientos establecidos antes del cierre del 19 de marzo de 2025 a las 4:00 p.m.:

1. Elaboración de Scripts de Consulta

- Se diseñaron múltiples scripts SQL para obtener los datos requeridos desde el aplicativo SIGP, con base en la convocatoria con ID correspondiente.
- Las consultas permitieron extraer la información estructurada para los siguientes reportes:
 - Reporte general de la convocatoria.
 - Reporte de presupuesto de los proponentes (monto solicitado y contrapartida).
 - Reporte de contrapartidas discriminadas por entidad participante.
 - Reporte de personal (incluyendo información de la Doctora y de la joven investigadora).
 - Reporte de categoría de enfoque diferencial.

2. Ejecución de Consultas y Generación de Archivos Excel

- Los scripts fueron ejecutados directamente en la base de datos SIGP, generando seis archivos Excel conforme a los lineamientos definidos para facilitar la revisión de la información por parte de los usuarios funcionales.

3. Entrega de los Reportes al Usuario Funcional

- Se notificó a las funcionarias **Carolina Munera** y **Jennifer Estrada** sobre la disponibilidad de los reportes.
- Todos los archivos fueron enviados en formato Excel, tal como fue solicitado, para su análisis y validación oportuna.

Solicitud 154824

Asunto	Tiempo total de actividad

Reporte convocatoria 953			00:06:43
Descripción			
<p>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.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
18/03/2025 16:52:54	20/03/2025 10:36:26	19/03/2025 11:36:26	20/03/2025 10:36:26

Acción Realizada 154824

En atención a la solicitud de generación del **Reporte de la Convocatoria 953-2024**, se llevaron a cabo las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para extraer los proyectos asociados a la convocatoria con ID 11713, filtrando aquellos cuyo estado fuera “DILIGENCIANDO” o “RADICADO”.
- La consulta se estructuró para garantizar la obtención de la información requerida de forma clara y precisa.

2. Ejecución del Script y Generación del Reporte

- Se ejecutó el script directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con los resultados, estructurado según las necesidades del reporte solicitado.

3. Entrega del Reporte al Usuario Funcional

- Se notificó a la usuaria funcional **Alejandra Acosta** sobre la disponibilidad del reporte.
- El archivo fue enviado por correo electrónico para su correspondiente revisión.

Solicitud 155007

Asunto		Tiempo total de actividad	
Reportes SIGP		00:04:55	
Descripción			
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
20/03/2025 17:04:25	24/03/2025 10:50:06	21/03/2025 11:50:06	24/03/2025 10:50:06

Acción Realizada 155007

En respuesta a la solicitud de generación de reporte de la **Convocatoria 953-2024**, se realizaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se elaboró un script SQL para obtener los proyectos registrados en la convocatoria con ID 11713, filtrando por los estados “DILIGENCIANDO” y “RADICADO”.
- Se verificó la precisión de los datos retornados, asegurando que correspondieran a los criterios definidos por el usuario funcional.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con los resultados obtenidos, estructurado conforme a los requerimientos del reporte.

3. Entrega del Reporte al Usuario Funcional

- Se notificó a la usuaria funcional **Alejandra Acosta** sobre la disponibilidad del reporte.
- El documento fue enviado por correo electrónico para su revisión y uso.

Solicitud 155026

Asunto	Tiempo total de actividad
reporte de la convocatoria 953	00:05:45
Descripción Buenos días amablemente solicito de su acostumbrada colaboración para la generación del reporte de la convocatoria 953 que cierra el día de hoy por la plataforma SIGP.	

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
21/03/2025 08:13:32	24/03/2025 16:00:00	21/03/2025 18:12:22	24/03/2025 16:00:00

Acción Realizada 155026

En cumplimiento de la solicitud recibida para la generación de reportes al cierre de la **Convocatoria 953-2024 - Convocatoria para Comunidades Negras de Bogotá, Caquetá, Córdoba, Magdalena y Chocó**, cuyo cierre fue el 21 de marzo de 2025, se realizaron las siguientes acciones:

1. Elaboración de Scripts de Consulta

- Se diseñaron scripts SQL para extraer la información correspondiente a la convocatoria con ID 11713.
- Las consultas permitieron generar los siguientes reportes requeridos por los usuarios funcionales:
 - Reporte general de la convocatoria.
 - Reporte de presupuesto de los proponentes (monto solicitado y contrapartida).
 - Reporte de contrapartidas discriminadas por entidad participante.
 - Reporte de personal vinculado a las propuestas.

2. Ejecución de Consultas y Generación de Archivos Excel

- Se ejecutaron los scripts directamente sobre la base de datos SIGP.
- Se generaron archivos Excel con los datos solicitados, estructurados para facilitar su análisis y revisión.

3. Entrega de los Reportes al Usuario Funcional

- Se notificó a las funcionarias **Carolina Munera** y **Jennifer Estrada** sobre la disponibilidad de los reportes.
- Los documentos fueron enviados en formato Excel conforme a los lineamientos establecidos.

Solicitud 155088

Asunto		Tiempo total de actividad	
Reportes SIGP		00:09:32	
Descripción			
quisiéramos solicitar su apoyo con la emisión de un reporte de la plataforma del SIGP, que nos dé información sobre cuántas organizaciones radicaron efectivamente su proyecto, así como en qué departamentos y municipios			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
21/03/2025 17:30:01	26/03/2025 09:30:33	25/03/2025 09:49:55	26/03/2025 09:30:33

Acción Realizada 155088

En atención a la solicitud de generación de reporte de la **Convocatoria 953-2024**, se llevaron a cabo las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener los proyectos asociados a la convocatoria con ID 11713, filtrando aquellos cuyo estado fuera “RADICADO”.

- Se aseguró que la consulta cumpliera con los criterios solicitados para brindar información precisa y relevante.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente en la base de datos SIGP.
- Se generó un archivo Excel con los datos obtenidos, conforme a las especificaciones del reporte.

3. Entrega del Reporte al Usuario Funcional

- Se notificó a la usuaria funcional **Alejandra Acosta** sobre la disponibilidad del reporte.
- El documento fue enviado para su revisión y uso.

Solicitud 155204

Asunto	Tiempo total de actividad
Reporte convocatoria 962	00:04:06
Descripción Cordial saludo, Leonardo Suárez Agradecemos de su colaboración para generar un reporte de contrapartidas proyectos por entidad por de la convocatoria 962 plataforma SIGP. Miguel Bernal Equipo de verificación de mecanismos DGR --	

<p>*Miguel Camilo Bernal Bolaños*</p> <p>Contratista</p> <p>DGR Equipo Mecanismos</p> <p>*mcbernal@minciencias.gov.co <mcbernal@minciencias.gov.co>*</p> <p>Teléfono: 601 6258480 Ext. 3203</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
26/03/2025 11:59:04	27/03/2025 15:41:41	26/03/2025 16:41:41	27/03/2025 15:41:41

Acción Realizada 155204

En atención a la solicitud de generación de reportes de contrapartidas de proyectos por entidad, correspondientes a la **Convocatoria 962-2024 SENAINNOVA** en la plataforma SIGP, se realizaron las siguientes actividades:

1. Generación de Reportes

- Se elaboraron y ejecutaron los scripts necesarios para obtener el presupuesto general y las contrapartidas de los proyectos asociados a la convocatoria 962.

- Los reportes fueron generados en formato Excel, facilitando el análisis y revisión de la información financiera.

2. Entrega de los Reportes al Usuario Funcional

- Se notificó y envió la documentación al usuario funcional **Miguel Bernal** para su correspondiente uso y validación.

Solicitud 155503

Asunto		Tiempo total de actividad	
Reporte convocatoria 963		00:08:52	
Descripción			
<p>Buenas tardes apreciados compañeros,</p> <p>Dado que el día de hoy cierra la primera verificación de requisitos y mañana 01/04/2025, da inicio la etapa de subsanación de la CV 963 -Orquídeas, agradecemos que nos puedan remitir el reporte de los estados y observaciones generadas por el equipo en el SIGP, a cada propuesta, siempre que se pueda, en el menor tiempo posible.</p> <p>Gracias por su apoyo</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
31/03/2025 15:10:18	02/04/2025 09:30:41	01/04/2025 10:24:40	02/04/2025 09:30:41

Acción Realizada 155503

En atención a la solicitud relacionada con la **Convocatoria 963-2025 Orquídeas**, se llevaron a cabo las siguientes acciones:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los requisitos de los proyectos inscritos en la convocatoria 963-2025 Orquídeas.
- La consulta fue estructurada para brindar información precisa sobre el cumplimiento de requisitos por proyecto.

2. Ejecución del Script y Notificación

- Se ejecutó el script directamente en la base de datos correspondiente.
- Se envió un correo notificando la ejecución del script y la disponibilidad de la información extraída.

Solicitud 155560

Asunto	Tiempo total de actividad
Reportes SIGP	00:06:45
Descripción Buenos días. Solicito su amable colaboración para el envío de un reporte de proyectos inscritos en la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 10:00am. Quedo atento a sus comentarios. Cordialmente, --	

Deiver Andrey Acosta Linares

Contratista

Dirección de Ciencia

daacosta@minciencias.gov.co <daacosta@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
01/04/2025 10:08:30	02/04/2025 09:57:24	01/04/2025 10:57:23	02/04/2025 09:57:24

Acción Realizada 155560

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 10:00 a.m., se ejecutaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades participantes en la convocatoria 964-2025.

- La consulta fue estructurada para reflejar la información actualizada con corte al horario solicitado.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente sobre la base de datos SIGP.
- Se generó un archivo Excel con la información obtenida, preparado para su análisis y revisión.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 155673

Asunto	Tiempo total de actividad
Reportes SIGP	00:04:45
Descripción <p>Buenas tardes, solicito su amable colaboración para el envío de un reporte de proyectos inscritos en la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 5:00pm del día de hoy.</p> <p>Muchas gracias.</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>	

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
02/04/2025 12:16:24	03/04/2025 16:00:00	02/04/2025 17:20:17	03/04/2025 16:00:00

Acción Realizada 155673

En respuesta a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 5:00 p.m. del día en curso, se llevaron a cabo las siguientes acciones:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado actualizado de los proyectos y entidades inscritos en la convocatoria 964-2025, con corte a la hora solicitada.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente en la base de datos SIGP.
- Se generó un archivo Excel con la información solicitada para facilitar su revisión y análisis.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su uso.

Solicitud 155734

Asunto	Tiempo total de actividad
Reportes SIGP	00:03:40
<p>Descripción</p> <p>Buenas tardes, solicito su amable colaboración para el envío de un reporte de proyectos inscritos en la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 5:00pm del día de hoy.</p> <p>Muchas gracias.</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>	

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
03/04/2025 12:25:13	04/04/2025 16:00:00	03/04/2025 17:52:24	04/04/2025 16:00:00

Acción Realizada 155734

En atención a la solicitud de generación de reporte para la **Convocatoria 964-2025 Formación en Maestrías Pacífico Nariñense**, con corte a las 5:00 p.m. del día en curso, se efectuaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener el estado de los proyectos y las entidades inscritas en la convocatoria 964-2025, acorde al corte horario solicitado.

2. Ejecución del Script y Generación del Reporte

- Se ejecutó el script directamente en la base de datos SIGP.
- Se generó el reporte en formato Excel, facilitando la revisión de la información.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Deiver Acosta** para su correspondiente uso.

Solicitud 155759

Asunto	Tiempo total de actividad

REPORTE (SIGP) - Conv.962		00:03:11	
Descripción			
<p>Por este medio, me permito solicitar amablemente un reporte con el nombre y NIT de todas las entidades participantes en la Conv.962-SenaInnova, ya sea como beneficiario o como aliado, junto con el ID del proyecto con el fin de realizar una validación interna.</p> <p>Muchas gracias por su valiosa colaboración.</p> <p>Cordialmente,</p> <p>--</p> <p>*Juan Raul Mendoza Zamudio*</p> <p>Profesional Especializado</p> <p>Equipo de Atención al Ciudadano</p> <p>*rmendoza@minciencias.gov.co <rmendoza@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

03/04/2025 19:05:28	07/04/2025 09:14:53	04/04/2025 10:14:53	07/04/2025 09:14:53
------------------------	------------------------	---------------------	------------------------

Acción Realizada 155759

En atención a la solicitud de generación de reporte con el nombre y NIT de todas las entidades participantes en la **Convocatoria 962-2024 SENAINNOVA para Fomentar Acceso y Adopción de Tecnologías Avanzadas en Sector Agroalimentar**, ya sea como beneficiario o aliado, junto con el ID del proyecto, se realizaron las siguientes actividades:

1. Elaboración del Script de Consulta

- Se diseñó un script SQL para obtener la información de las entidades asociadas a los proyectos de la convocatoria 962-2024, incluyendo nombre, NIT y el ID del proyecto, con el objetivo de facilitar una validación interna.

2. Ejecución del Script y Generación del Reporte

- El script fue ejecutado directamente en la base de datos SIGP.
- Se generó el reporte en formato Excel con la información requerida.

3. Entrega del Reporte al Usuario Funcional

- Se notificó y envió el reporte al usuario funcional **Juan Mendoza** para su uso.

Solicitud 155773

Asunto	Tiempo total de actividad
Reportes SIGP	00:11:50
Descripción Stres. OTSI , desde la DDTI , y en el marco del proceso de evaluación de la Conv. 962 - Senalnnova , nos permitimos solicitar amablemente la base de datos completa de la	

información registrada en SIGP de los proyectos de dicha convocatoria. Muchas gracias por su pronta gestión y quedamos atentos .			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
04/04/2025 09:07:30	10/04/2025 10:28:00	09/04/2025 11:28:00	10/04/2025 10:28:00

Acción Realizada 155773

En atención a la solicitud recibida desde la OTSI, a través de la DDTI, referente al proceso de evaluación de la **Convocatoria 962 - SenalInnova**, se realizaron las siguientes acciones:

1. **Solicitud de Precisión sobre el Reporte**
 - Se contactó al solicitante para solicitar mayor precisión y detalles específicos sobre la información requerida de la base de datos completa de los proyectos registrados en SIGP para la convocatoria 962.
2. **Seguimiento a la Solicitud**
 - Se mantuvo comunicación para clarificar el alcance del reporte solicitado y asegurar la correcta gestión de la información.

Solicitud 155788

Asunto	Tiempo total de actividad
SOLICITUD TI -- URGENTE -- SOLICITUD DE FICHA DE PROPUESTA BPIN 2024000100132	00:06:12
Descripción	

Buenos días

Cordial saludo

Solicito comedidamente *"Ficha de la propuesta registrada en SIGP"* del proyecto ID 108705 (BPIN 2024000100132) correspondiente a la convocatoria No 35

La anterior solicitud se eleva a fines de dar respuesta a INCIDENTE DE DESACATO DEL JUZGADO QUINTO LABORAL DEL CIRCUITO DE MANIZALES (Respuesta a requerimiento previo a desacato en tutela 17-001-31-05-005-2024-10178-00)

Agradezco su atencion

--

María José Uribe Sánchez

Contratista

Despacho Del Ministro - Secretaria Técnica OCAD

mjuribe@minciencias.gov.co <mjuribe@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
04/04/2025 10:49:42	07/04/2025 12:08:45	04/04/2025 13:08:45	07/04/2025 12:08:45

Acción Realizada 155788

En atención a la solicitud para obtener la *Ficha de la propuesta registrada en SIGP* correspondiente al proyecto con ID 108705 (BPIN 2024000100132) de la convocatoria No. 35, como parte de la respuesta al incidente de desacato del Juzgado Quinto Laboral del Circuito de Manizales, se llevaron a cabo las siguientes acciones:

1. Generación de la Ficha del Proyecto

- Se elaboró la ficha detallada del proyecto solicitado, extrayendo la información pertinente desde la base de datos SIGP.

2. Entrega de la Ficha al Usuario Funcional

- Se envió la ficha generada a la usuaria funcional **María José Uribe Sánchez**, quien la utilizará para la respuesta al requerimiento judicial.

Solicitud 155801

Asunto	Tiempo total de actividad
REPORTE (SIGP)	00:07:03
Descripción Solicito atentamente un informe sobre los recursos que fueron requeridos en la financiación correspondiente a las propuestas de la Convocatoria 951-2024. La pronta disposición de esta información resulta de gran importancia para el avance de las gestiones pertinentes. Agradeceríamos que nos pudieran facilitar este reporte	

a la mayor brevedad posible. Agradezco de antemano su diligencia y colaboración en este requerimiento.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
04/04/2025 13:24:55	07/04/2025 15:54:42	04/04/2025 16:54:41	07/04/2025 15:54:42

Acción Realizada 155801

En respuesta a la solicitud de un informe sobre los recursos requeridos para la financiación de las propuestas presentadas en la **Convocatoria 951-2024 - Fortalecimiento del Conocimiento Geocientífico y Tecnológico de las Fuentes**, se llevaron a cabo las siguientes acciones:

1. Generación de Scripts de Consulta

- Se desarrollaron scripts de consulta que permiten identificar los montos de recursos solicitados por los proyectos que superaron la etapa de registro.

2. Elaboración del Informe General

- A partir de la información obtenida, se consolidó un informe general en formato Excel, en el cual se evidencia el detalle de los recursos requeridos para la financiación de las propuestas.

3. Entrega del Informe

- El documento fue enviado y notificado al usuario funcional **Oscar Sánchez** para su respectiva revisión y gestión.

Solicitud 155837

Asunto	Tiempo total de actividad

Informe de SIGP de todos los mecanismos de 2024			00:08:07
Descripción			
Se requiere un informe de SIGP de todos los mecanismos de 2024 con la respectiva caracterización según: grupo étnico, género, edades. Todos los datos excepto los confidenciales.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
04/04/2025 17:29:05	09/04/2025 15:17:07	08/04/2025 16:17:07	09/04/2025 15:17:07

Acción Realizada 155837

En atención a la solicitud de generación del **Informe de SIGP de todos los mecanismos registrados durante la vigencia 2024**, se realizaron las siguientes acciones:

1. Desarrollo del Script de Consulta

- Se construyó un script que permite extraer los datos de **enfoque diferencial** tanto a nivel de los proyectos como del personal vinculado, incluyendo información sobre **grupo étnico, género y rango de edad**.

2. Consolidación del Informe

- Se generó un documento consolidado con los datos extraídos del aplicativo SIGP, permitiendo una caracterización completa de los mecanismos implementados en 2024.

3. Entrega del Informe

- El informe en formato Excel fue remitido a la usuaria funcional **Stephanie Pérez** para su análisis y uso correspondiente.

Solicitud 155882

Asunto	Tiempo total de actividad
Reportes SIGP	00:16:57
Descripción <p>Buenos días.</p> <p>Solicito su amable colaboración para envío de un reporte de proyectos inscritos en la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 5:00pm.</p> <p>Quedo atento a sus comentarios.</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>	

<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
07/04/2025 08:52:41	08/04/2025 06:49:31	07/04/2025 18:21:37	08/04/2025 06:49:31

Acción Realizada 155882

En atención a la solicitud del **reporte de proyectos inscritos en la Convocatoria 964-2025 FORMACIÓN EN MAESTRÍAS PACÍFICO NARIÑENSE**, con corte a las 5:00 p.m. del día 2025-XX-XX, se llevaron a cabo las siguientes acciones:

1. Construcción de Scripts de Consulta

- Se desarrollaron scripts que permiten extraer información detallada del **estado de los proyectos y las entidades participantes** correspondientes a la convocatoria 964-2025.

2. Generación del Documento Excel

- Se consolidó la información en un archivo en formato Excel que incluye el estado actual de los proyectos registrados hasta el corte establecido.

3. Entrega del Reporte

- El documento fue **notificado y enviado** al usuario funcional **Deiver Acosta** para su revisión y análisis correspondiente.

Solicitud 155983

Asunto	Tiempo total de actividad

SOLICITUD TI Fwd: Solicitud Apoyo: Convocatoria 953	00:07:05
<p>Descripción</p> <p>Buen día ing. Leonardo:</p> <p>Se solicita amablemente el reporte de los proyectos con el correo institucional de las entidades para generar el contacto y/o apoyo en el periodo de subsanación de requisitos por parte del área técnica.</p> <p>Cordialmente,</p> <p>*Nidia Janneth León Cabeza*</p> <p>Contratista Oficina de Tecnologías y Sistemas de Información *njleon@minciencias.gov.co <njleon@minciencias.gov.co>*</p> <p>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: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co> Date: lun, 7 abr 2025 a las 21:18 Subject: Fwd: Solicitud Apoyo: Convocatoria 953 To: Nidia Janneth León Cabeza <njleon@minciencias.gov.co>, Leonardo Suarez <lasb@softwarebuilders.co>, Erika Maritza González Chávez <emgonzalez@minciencias.gov.co>, <Juan.esteban.meneses.guastar@gmail.com>, Elbert Gregorio Miranda Pineda <egmiranda@minciencias.gov.co></p> <p>Buenas tardes por favor colaborar a la Directora Angela Melo en sus requerimientos, es urgente, a continuación los requerimientos: Convocatoria 953</p>	

1. Como Ministerio nos encontramos actualmente en la etapa de verificación de requisitos y a partir del 4 de abril se abre la fase de subsanación, por lo que requerimos una base de las organizaciones inscritas que incluya los correos electrónicos, para poder generar un comunicado general para invitar a revisar la subsanación de requisitos.

2. Adicionalmente quisiéramos solicitar un espacio con el equipo o profesionales encargados de la etapa de evaluación (parametrización, bases de datos anonimizadas y demás procesos) para resolver dudas y poder avanzar de manera eficiente.

Fecha subsanación de requisitos Del 4 de abril al 10 de abril de 2025 hasta las 5:00 p.m.

Por favor indicar fecha de reunión martes 8 de abril 4pm

----- Forwarded message -----

De: Ángela Vanessa Melo Gámez <avmelo@minciencias.gov.co>

Date: lun, 7 abr 2025 a las 20:30

Subject: Solicitud Apoyo: Convocatoria 953

To: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>

Cc: Alejandra Margarita Acosta Villamil <amacosta@minciencias.gov.co>

Estimado Dr Harold,

Reciba un caluroso saludo.

En primer lugar, desde la Dirección agradecemos todo su apoyo y el de su equipo en el proceso de inscripción de la Convocatoria 953. En el marco de la implementación del procedimiento de evaluación de esta última, solicitamos su amable apoyo con las siguientes dos solicitudes:

1. Como Ministerio nos encontramos actualmente en la etapa de verificación de requisitos y a partir del 4 de abril se abre la fase de subsanación, por lo que requerimos una base de las organizaciones inscritas que incluya los correos electrónicos, para poder generar un comunicado general para invitar a revisar la subsanación de requisitos.

2. Adicionalmente quisiéramos solicitar un espacio con el equipo o profesionales encargados de la etapa de evaluación (parametrización, bases de datos anonimizadas y demás procesos) para resolver dudas y poder avanzar de manera eficiente.

Quedamos muy atentas a su respuesta y de antemano agradecemos su gestión.

Cordial saludo.

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.minc			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
08/04/2025 06:23:55	09/04/2025 09:30:09	08/04/2025 10:14:09	09/04/2025 09:30:09

Acción Realizada 155983

En atención a la solicitud del **reporte de proyectos con el correo institucional de las entidades** de la **Convocatoria 953-2024 PARA COMUNIDADES NEGRAS DE BOGOTÁ, CAQUETÁ, CÓRDOBA, MAGDALENA Y CHOCÓ**, con el fin de facilitar el contacto y/o apoyo en el periodo de subsanación de requisitos, se ejecutaron las siguientes actividades:

1. Desarrollo de Script de Consulta

- Se creó un script que permite obtener los proyectos en **estado RADICADO** junto con la información institucional de las entidades participantes, incluyendo el **correo institucional**.

2. Generación de Documento Excel

- Se consolidó la información obtenida en un archivo Excel para facilitar su análisis y seguimiento por parte del área técnica.

3. Entrega del Reporte

- El documento fue **notificado y enviado** a la usuaria funcional **Ángela Vanessa Melo** para su gestión correspondiente.

Solicitud 156131

Asunto	Tiempo total de actividad

Reporte convocatoria 964		00:11:57	
Descripción			
<p>Buenas tardes.</p> <p>Solicito su amable colaboración para envío de un reporte de proyectos inscritos en la Convocatoria 964 Formación en Maestrías Pacífico Nariñense, con corte a las 6:00pm.</p> <p>Quedo atento a sus comentarios.</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

09/04/2025 16:43:46	11/04/2025 16:00:00	10/04/2025 18:09:32	11/04/2025 16:00:00
------------------------	------------------------	---------------------	------------------------

Acción Realizada 156131

En atención a la solicitud del **reporte de proyectos inscritos en la Convocatoria 964-2025 FORMACIÓN EN MAESTRÍAS PACÍFICO NARIÑENSE**, con corte a las **6:00 p.m.**, se realizaron las siguientes actividades:

1. Desarrollo de Script de Consulta

- Se elaboraron los scripts necesarios para obtener la información actualizada del estado de los proyectos y las entidades asociadas a la convocatoria mencionada.

2. Generación de Documento Excel

- La información extraída fue organizada y consolidada en un archivo Excel para su análisis y entrega.

3. Entrega del Reporte

- El documento fue **notificado y enviado** al usuario funcional **Deiver Acosta**, según lo solicitado.

Solicitud 156214

Asunto	Tiempo total de actividad
SOLICITUD TI Fwd: Solicitud de base anonimizada	00:03:30
Descripción Para generar trazabilidad de la solicitud del reporte anonimizado con la salvedad de entrega realizada por correo al jefe de la oficina.	

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: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>

Date: mié, 9 abr 2025 a las 19:09

Subject: Fwd: Solicitud de base anonimizada

To: Erika Maritza González Chávez <emgonzalez@minciencias.gov.co>, Leonardo Suarez <lasb@softwarebuilders.co>

Cc: Nidia Janneth León Cabeza <njleon@minciencias.gov.co>

Buenas tardes ingenieros por favor mañana generar la base de datos anonimizada de acuerdo a los criterios adjuntos y también de acuerdo a lo especificado en el procedimiento adjunto, por favor leer el procedimiento. La base de datos ANONIMIZADA por favor me la envían sólo a mí en un archivo excel encriptado con una clave no fácil de descifrar. La base de datos con los nombres por favor ERIKA conservarla en su equipo hasta tanto nos entreguen los códigos de identificación para selección de evaluadores. La clave me la dice personalmente para que yo personalmente decirle al Director Técnico de la convocatoria. De acuerdo al procedimiento los códigos que coloquemos deben ser diferentes a los ID de los evaluadores y los números no deben haber sido utilizados en otra convocatoria.

Mejor dicho leer el procedimiento adjunto para no cometer errores.

Cordial Saludo,

----- Forwarded message -----

De: Juan Raul Mendoza Zamudio <rmendoza@minciencias.gov.co>

Date: mié, 9 abr 2025 a las 15:54

Subject: Solicitud de base anonimizada

To: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co>

Cc: Ruben Darío Ortiz Morales <rdortiz@minciencias.gov.co>

Respetado Ing. Harold Alexander Méndez buenas tardes.

Teniendo en cuenta el procedimiento de Evaluación de Programas, Propuestas o Proyectos de CTel (Numeral 16), nos permitimos solicitar la base de datos anonimizada de los potenciales evaluadores, en el marco del proceso de evaluación de los proyectos y propuestas presentados a la Conv. 962 - Senalnova.

Adjuntamos los criterios definidos por las palabras claves.

Quedamos atentos.

Cordialmente,

--

Juan Raul Mendoza Zamudio

Profesional Especializado

Equipo de Atención al Ciudadano

rmendoza@minciencias.gov.co <rmendoza@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>>

--

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**

10/04/2025 18:04:30	14/04/2025 11:31:39	11/04/2025 12:31:39	14/04/2025 11:31:39
------------------------	------------------------	---------------------	------------------------

Acción Realizada 156214

En cumplimiento de la solicitud relacionada con la **Convocatoria 962-2024 SENAINNOVA**, se llevaron a cabo las siguientes actividades:

1. Generación del Reporte de Evaluadores

- Se construyó un reporte con el **listado de evaluadores** que cumplen con los criterios establecidos en cuanto a:
 - **Grandes áreas**
 - **Áreas**
 - **Disciplinas de conocimiento**
 - **Palabras clave asociadas**

2. Entrega del Reporte

- El documento resultante fue **generado y enviado** al **Jefe de la OTSI, Harold Méndez**, como parte de la gestión solicitada.

Solicitud 156216

Asunto	Tiempo total de actividad
Reportes SIGP	00:06:27
Descripción	
Buenas tardes apreciados compañeros,	

En atención al proceso de verificación de la CV 963 -Orquídeas y la identificación de posibles inconsistencias, agradecemos que nos puedan remitir el día de mañana el reporte de los estados de las propuestas (revisado, rechazado, radicado), el estado de verificación de cada requisito (sí, no, no aplica), el campo de observaciones generales de cada proyecto y el campo de observaciones de cada requisito, siempre que se pueda, en el menor tiempo posible.

Gracias por su apoyo

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
10/04/2025 19:06:16	14/04/2025 11:26:17	11/04/2025 12:26:17	14/04/2025 11:26:17

Acción Realizada 156216

En atención a la solicitud relacionada con la **Convocatoria 963 – Orquídeas: Mujeres en Inteligencia Artificial, Ciencias y Tecnologías Cuánticas 2025**, se realizó la siguiente gestión:

1. Generación del Reporte de Requisitos

Se construyó un reporte que contiene la siguiente información detallada por proyecto:

- **Estado de las propuestas:** revisado, rechazado o radicado.
- **Estado de verificación de cada requisito:** sí, no, no aplica.
- **Observaciones generales** de cada proyecto.
- **Observaciones específicas por requisito.**

2. Entrega del Reporte

- El documento fue remitido a **Jennifer Estrada**, con la información completa requerida para el seguimiento y análisis del estado de los proyectos en la convocatoria mencionada.

Solicitud 156346

Asunto		Tiempo total de actividad	
Reportes SIGP		00:13:02	
Descripción			
<p>Por medio de la presente, me dirijo a ustedes para solicitar amablemente la base de datos consolidada de las evaluaciones correspondientes al cierre de la Convocatoria 958, denominada "CONVOCATORIA TRANSFORMACIÓN DE LA PLANEACIÓN MINERO ENERGÉTICA PARA LA CONVERGENCIA TERRITORIAL Y LA TRANSICIÓN ENERGÉTICA JUSTA".</p> <p>Agradecería que la información suministrada contenga la totalidad de los campos diligenciados para cada evaluador y cada propuesta, incluyendo de manera explícita el puntaje asignado a cada criterio y subcriterio de evaluación.</p> <p>Su pronta y completa colaboración en esta solicitud es fundamental para el correcto desarrollo de las etapas subsiguientes.</p> <p>Agradezco de antemano su atención y diligencia.</p>			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
14/04/2025 12:00:19	16/04/2025 09:29:49	15/04/2025 10:29:49	16/04/2025 09:29:49

Acción Realizada 156346

En atención a la solicitud relacionada con la **Convocatoria 958-2024 – "Transformación de la Planeación Minero Energética para la Convergencia Territorial y la Transición Energética Justa"**, se realizaron las siguientes actividades:

1. Generación del Reporte de Evaluaciones

- Se construyó un **informe consolidado** que contiene los resultados de las **evaluaciones por criterio y por evaluador** correspondientes al cierre de la convocatoria.

2. Entrega del Reporte

- El documento fue **notificado y enviado** al usuario funcional **Oscar Daniel Sánchez Plazas** para su análisis y gestión correspondiente.

Solución de Errores

Se solucionaron errores reportados por los usuarios en la plataforma:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
154840	- Error Eliminar proyectos 11420 y 11421 Convocatoria Orquídeas	Solucionado	-	[Ver detalle]
155201	- Error asociar grupos y productos Solicitud 9378	Solucionado	-	[Ver detalle]
155508	- Acceder al proceso de acreditación	Solucionado	-	[Ver detalle]
155777	- Atención solicitud PQRDS Rad. No. 20250010720R – Convocatoria 963-2025	Solucionado		[Ver detalle]
156015	- Lentitud de plataforma SIGP	Solucionado		[Ver detalle]

Solicitud 154840

Asunto	Tiempo total de actividad
Proyectos 11420	00:07:29
Descripción	
Buenos días estimados, por solicitud de la ciudadana Janet Fernandez, agradezco eliminar los registros de los proyectos 11420 y 11420 en el formulario SIGP, con el fin de poder crear nuevamente otro proyecto con el	

mismo nombre y participar en la convocatoria de orquídeas que cierra el día de hoy

Agradezco su colaboración quedo atento a cualquier inquietud

--

Juan David Guzman Zambrano

Equipo de Atención al Ciudadano

jdguzman@minciencias.gov.co <jdguzman@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
19/03/2025 09:34:44	20/03/2025 09:51:33	19/03/2025 10:51:33	20/03/2025 09:51:33

Asunto: Eliminación de registros duplicados para proyectos en SIGP - Convocatoria Orquídeas

Descripción del problema:

La ciudadana Janet Fernández solicitó la eliminación de los registros correspondientes a los

proyectos con ID **11420** en el formulario SIGP. Esto con el objetivo de poder crear un nuevo proyecto con el mismo nombre y participar en la convocatoria **Orquídeas**, cuyo cierre es hoy.

Análisis y acciones realizadas:

- Se verificó la existencia de los registros duplicados en el sistema SIGP.
- Se procedió a eliminar ambos registros para evitar conflictos y permitir la creación correcta del nuevo proyecto.

Resultado:

Los registros fueron eliminados exitosamente del sistema. El usuario podrá crear el nuevo proyecto sin inconvenientes.

Notificación:

Se envió comunicación al usuario funcional **Juan Guzmán**, informando la resolución del caso.

Solicitud 155201

Asunto	Tiempo total de actividad
Verificación de información Plataforma Reconocimiento de Actores 9378	00:06:04
Descripción	
<p>Ing. Leonardo buenas tardes:</p> <p>Solicito apoyo para la revisión de la solicitud no. 9378, ya que no le permite asociar grupos ni productos, esto con el fin de dar oportuna respuesta la PQRDS no. 20250007717R.</p> <p>Adjunto información del ciudadano e imagenes de evidencia aportadas por el ciudadano.</p> <p>Cordialmente,</p>	

20250007717R 20250007717R

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
26/03/2025 11:58:46	28/03/2025 12:12:38	27/03/2025 10:10:45	28/03/2025 12:12:38

Asunto: Revisión y solución para la solicitud No. 9378 - Problemas para asociar grupos y productos

Descripción del problema:

Se solicita apoyo para la revisión de la solicitud No. 9378, ya que el usuario reporta que no es posible asociar grupos ni productos. Esta revisión es necesaria para dar respuesta oportuna a la PQRDS No. 20250007717R.

Análisis y diagnóstico:

En atención a la PQRDS No. 20250007717R, se realizó una revisión detallada en el sistema, encontrando lo siguiente:

1. La entidad **ENTIDAD NACIONAL DE COLOMBIA**, a la cual pertenece el centro Instituto de Genética, no ha registrado los grupos **COL0086813** y **COL0218153** como propios en el formulario de Entidades. (Se adjunta evidencia visual que respalda esta situación).
2. En cuanto a la imposibilidad de agregar productos al grupo **COL0204283**, se verificó que la solicitud ya ha registrado un total de 23 productos de manera exitosa. (Se adjunta imagen soporte).

Acciones realizadas:

- Se informa al usuario que para poder asociar los grupos COL0086813 y COL0218153, la entidad debe registrarlos como propios en el sistema.
- Se confirma que no existe limitación técnica para agregar productos al grupo COL0204283, ya que se han registrado productos correctamente.

Estado:

Pendiente acción por parte del usuario para registro de grupos en la entidad.

Solicitud 155508

Asunto	Tiempo total de actividad
Acceder al proceso de acreditación	00:11:59
Descripción	
Buenas tardes: Acudo a su colaboración, considerando que desde la empresa Seguros Bolívar nos indican que venían diligenciando información en el SIGP para acceder al proceso de acreditación de reconocimiento de actores como una Unidad I+D+i de Empresa y/o EAI. Dicho proceso se inició el mes Abril del año 2024 (Código de solicitud 8893).	

En este sentido indican que para los siguientes meses, en los cuales realizaron la consulta en el aplicativo les apareció el siguiente mensaje "Su solicitud de reconocimiento está en estudio por Minciencias y se encuentra en estado EN EVALUACIÓN".

No obstante, para en la última revisión que adelantaron este año les aparece el siguiente mensaje notificación "El plazo para enviar la solicitud ha expirado, es necesario anular la solicitud y crear una nueva"

Considerando que ellos manifiestan querer dar continuidad al proceso, amablemente acudo a su colaboración con el fin de poder aclararles los siguientes aspectos:

1. Desde el punto de vista del manejo de las ventanas de tiempo de la plataforma, indicar cuales son las vigencias dentro de las cuales debían presentar o registrar la información correspondiente.
2. Cómo deben proceder para dar continuar con el proceso o si se requiere dar inicio con una nueva radicación y asignación de código.

Así mismo, se adjunta como respaldo copia del correo electrónico enviado por la entidad en torno a la solicitud planteada.

Quedo atenta a su amable orientación al respecto.

Cordialmente,

Luz Marina Pulido
DDTI

Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre
31/03/2025 15:16:41	02/04/2025 13:54:40	01/04/2025 14:54:40	02/04/2025 13:54:40

Asunto: Consulta y gestión sobre solicitud de reconocimiento de actores No. 8893 - Seguros Bolívar

Descripción del problema:

El usuario funcional de Seguros Bolívar reportó que, al diligenciar información en el SIGP para la solicitud de reconocimiento de actores (solicitud No. 8893, creada en abril de 2024), el sistema inicialmente mostraba el mensaje:

"Su solicitud de reconocimiento está en estudio por Minciencias y se encuentra en estado EN EVALUACIÓN".

Actualmente, sin embargo, el sistema indica que el plazo expiró y que deben anular la solicitud y crear una nueva.

Análisis y diagnóstico:

Se revisó la solicitud No. 8893 y se confirmó que:

- Fue creada en abril de 2024, pero nunca fue radicada ante Minciencias.
- La solicitud fue anulada el 11/03/2025 debido al vencimiento del plazo de radicación establecido.
- Se verificó la auditoría del sistema para asegurar la correcta gestión de estados y tiempos.

Acciones realizadas:

- Se informó al usuario sobre los tiempos establecidos para la radicación de solicitudes.
- Se recomendó anular la solicitud actual y crear una nueva para continuar con el proceso.
- Se programó una capacitación sobre el proceso de reconocimiento de actores para el día 02/04/2025, con el fin de aclarar dudas y evitar futuros inconvenientes.

Solicitud 155777

Asunto	Tiempo total de actividad

SOLICITUD TI Fwd: IMPORTANTE_Radicado 20250011277R Asignado en AZ Digital	00:15:24
Descripción Buen día ing. Leonardo: Solicito apoyo para generar la auditoría del proyecto 114281, ya que actualmente este se encuentra en estado "RADICADO". Adjunto la información de la PQRDS. Corddialmente, *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: Carolina Múnera Camacho <cmunera@minciencias.gov.co> Date: jue, 3 abr 2025 a las 18:38 Subject: IMPORTANTE_Radicado 20250011277R Asignado en AZ Digital To: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co> Cc: Leonardo Suarez <lasb@softwarebuilders.co>, Nidia Janneth León Cabeza <njleon@minciencias.gov.co>, Yuldana del Socorro Córdoba Londoño <ydscondoba@minciencias.gov.co> Respetado Doctor Harold	

A través del AZ Digital reasigne la PQR con radicado 20250011277R, en la cual informan que presentaron dificultades en la plataforma al momento de la postulación.

Se debe emitir respuesta por el AZ el día de mañana 4 de abril dado que cuenta con vencimiento.

Recomiendo de manera especial el trámite señalado, dado que la responsabilidad de cierre recae en la Dirección de Vocaciones y Formación.

Cordialmente

Carolina Múnera Camacho

Contratista

Equipo de Seguimiento Convenios y Mecanismos

cmunera@minciencias.gov.co <cmunera@minciencias.gov.co>

Teléfono: 601 6258480 Ext. 5306

Av. Calle 26 No. 57 - 83

Torre 8, pisos 2 al 6

Bogotá, Colombia

www.minciencias.gov.co

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
04/04/2025 09:33:53	07/04/2025 15:07:32	04/04/2025 16:07:32	07/04/2025 15:07:32

Acción Realizada 155777

Atención solicitud PQRDS Rad. No. 20250010720R – Convocatoria 963-2025:

- Se revisó el caso reportado por una usuaria que presentaba dificultades al postular su proyecto en la plataforma.
- Se verificó que el proyecto identificado con el ID 114281 fue radicado exitosamente el 19 de marzo de 2025.
- Se explicó el proceso adecuado para el registro y validación de entidades, así como los pasos para diligenciar el formulario de proyectos y completar la postulación.
- Se notificó y envió respuesta al usuario solicitante.

Solicitud 156042

Asunto	Tiempo total de actividad
SIGP Gestion de plataforma	00:08:37
Descripción	

Buenas tardes,

Por el presente solicito apoyo con la gestión de la plataforma SIGP. Puesto que uno de nuestro actores, el Instituto de Genética de la Universidad Nacional de Colombia, que se encuentran postulando su solicitud de reconocimiento No. 9378, no ha podido vincular los siguientes grupos de investigación: COL0086813 del profesor William Usaquén Martínez y COL0218153 de la profesora Natalia Olaya Morales.

Quisiéramos saber la razón o la dificultad de ello.

Agradezco su gestión.

Cordialmente,

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

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
08/04/2025 15:28:12	09/04/2025 16:00:00	08/04/2025 17:20:34	09/04/2025 16:00:00

Asunto: Consulta y gestión sobre solicitud de reconocimiento de actores No. 8893 - Seguros Bolívar

Descripción del problema:

El usuario funcional de Seguros Bolívar reportó que, al diligenciar información en el SIGP para la solicitud de reconocimiento de actores (solicitud No. 8893, creada en abril de 2024), el sistema inicialmente mostraba el mensaje:

"Su solicitud de reconocimiento está en estudio por Minciencias y se encuentra en estado EN EVALUACIÓN".

Actualmente, sin embargo, el sistema indica que el plazo expiró y que deben anular la solicitud y crear una nueva.

Análisis y diagnóstico:

Se revisó la solicitud No. 8893 y se confirmó que:

- Fue creada en abril de 2024, pero nunca fue radicada ante Minciencias.
- La solicitud fue anulada el 11/03/2025 debido al vencimiento del plazo de radicación establecido.
- Se verificó la auditoría del sistema para asegurar la correcta gestión de estados y tiempos.

Acciones realizadas:

- Se informó al usuario sobre los tiempos establecidos para la radicación de solicitudes.
- Se recomendó anular la solicitud actual y crear una nueva para continuar con el proceso.
- Se programó una capacitación sobre el proceso de reconocimiento de actores para el día 02/04/2025, con el fin de aclarar dudas y evitar futuros inconvenientes.

Actualización de Datos

Se realizó la siguiente actualización en la plataforma SIGP:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
154904	- Actualización Nombre de Centro de Investigación Clínica de la Costa	Solucionado	-	[Ver detalle]
155018	- Actualización Razón Social PARQUESOFT PASTO	Solucionado	-	[Ver detalle]

Solicitud 154904

Asunto	Tiempo total de actividad
SOLICITUD TI Fwd: Solicitud de PQRS 20250007308R	00:07:47
Descripción <p>Buenas tardes ing. Leonardo:</p> <p>Solicito amablemente la actualización de la nombre del centro como lo describe la comunicación que antecede este.</p> <p>Adjunto resumen de la PQRDS y acto administrativo.</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>	

Torre 8, pisos 2 al 6
Bogotá, Colombia
www.minciencias.gov.co

----- Forwarded message -----

De: Alexandra Orjuela Ruiz <aorjuela@minciencias.gov.co>
Date: mié, 19 mar 2025 a las 10:03
Subject: Solicitud de PQRS 20250007308R
To: Nidia Janneth León Cabeza <njleon@minciencias.gov.co>

Estimada Nidia,

Recibe un cordial saludo.

Por el presente, te quiero informar que se ha solicitado el traslado de la PQRS 20250007308R a la OTSI. Esta solicitud corresponde a la solicitud de cambiar el nombre del actor de la solicitud No. 9357. Es decir, pasar de CENTRO DE INVESTIGACIÓN DE CLINICA a CENTRO DE INVESTIGACION CLINICA DE LA COSTA.

Quedo atenta a cualquier inquietud.

Cordialmente,

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

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
19/03/2025 16:47:14	21/03/2025 11:35:23	20/03/2025 12:35:23	21/03/2025 11:35:23

Acción Realizada 154909

- Se atendió la solicitud de cambio de nombre del centro de investigación registrada en la solicitud No. 9357.
- Se actualizó el nombre del centro de investigación de *CENTRO DE INVESTIGACIÓN DE CLÍNICA* a *CENTRO DE INVESTIGACIÓN CLÍNICA DE LA COSTA* en el sistema.
- Se verificó la correcta actualización y se confirmó que el cambio quedó reflejado correctamente.

Solicitud 155018

Asunto	Tiempo total de actividad
SOLICITUD TI Fwd: Respuesta: Solicitud No. 20250007210R 	00:06:06
Descripción <p>Ing. Leonardo:</p> <p>Por favor realizar la actualización de la razón social de la entidad.</p> <p>Adjunto certificado de representación y resumen de la solicitud de PQRDS 20250007210R para realizar el cambio.</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: Nidia Janneth León Cabeza <njleon@minciencias.gov.co></p> <p>Date: jue, 20 mar 2025 a las 20:27</p> <p>Subject: Respuesta: Solicitud No. 20250007210R </p> <p>To: Harold Alexander Méndez Guevara <hamendez@minciencias.gov.co></p> <p>Cc: Jairo Arias Urón <jarias@minciencias.gov.co></p>	

Ing. Harold, se genera respuesta:

Reciba un cordial saludo en nombre del Ministerio de Ciencia, Tecnología e Innovación. En respuesta a su solicitud, le informamos que se realizó la actualización de la Entidad “Parque del Software de Pasto ParqueSoft Pasto” a “Parque del Conocimiento e Innovación de Nariño ParqueSoft Nariño”. Esperamos que la información proporcionada haya resuelto su solicitud de manera satisfactoria. El mejoramiento continuo de la calidad y la eficiencia en la atención al ciudadano es una de las metas prioritarias de nuestra entidad. Por lo tanto, no dude en contactarnos a través de nuestros canales institucionales si tiene alguna duda o comentario adicional.

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
20/03/2025 20:32:11	24/03/2025 11:44:18	21/03/2025 12:44:18	24/03/2025 11:44:18

Acción Realizada 155018

- Se actualizó con éxito el nombre de la entidad de "PARQUESOFT PASTO" a "FUNDACIÓN PARQUE DEL CONOCIMIENTO Y LA INNOVACIÓN DE NARIÑO - PARQUESOFT NARIÑO" en el sistema.
- La actualización se realizó conforme al certificado de existencia y representación legal o de inscripción de documentos presentado.
- Se verificó y confirmó la correcta modificación en la base de datos.

Requerimientos

Se recibieron solicitudes de implementación de nuevas funcionalidades en el sistema:

N° Solicitud	Descripción	Estado	Observaciones	Acción Realizada
154297	- Pantalla de Configuración de Dependencias_Tablas_Campos	Pendiente de Usuario	Produccion	[Ver detalle]
154455	- Integración módulo de auditoría en SIGP	Pendiente de Usuario	Produccion	[Ver detalle]
156699	- Implementación ambiente de pruebas SIGP	Pendiente de Usuario	Pruebas	[Ver detalle]

Solicitud 154297

Asunto	Tiempo total de actividad
SOLICITUD TI Implementación de mejoras al módulo de administración - SIGP	01:08:35
<p>Descripción</p> <p>Buenas tardes, se crea la solicitud para validar el alcance del requerimiento al módulo de administración en el Sistema Integral de Gestión de Proyectos - SIGP.</p> <p>Cordialmente,</p> <p><i>*Nidia Janneth León Cabeza*</i> Contratista Oficina de Tecnologías y Sistemas de Información <i>*njleon@minciencias.gov.co <njleon@minciencias.gov.co>*</i> 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 ----- De: José Ignacio Español Perez <jiespanol@minciencias.gov.co> Date: mié, 5 mar 2025 a las 14:29 Subject: Re: Pendientes SIGP MinCiencias To: César Riaño <cesar_riano@softwarebuilders.co> Cc: Leonardo Andres Suarez Baldovino <lasb@software-b.com>, Harold Mendez <hamendez@minciencias.gov.co>, Nidia Janneth León Cabeza <njleon@minciencias.gov.co></p>	

Respetado César, cordial saludo.

De acuerdo con su solicitud, me es necesario conocer si para el plan de trabajo con el que usted elaboró su modelo para atender el contrato, cuenta con los soportes documentales que le fueron remitidos en su momento, para atender las diferentes solicitudes que corresponden a la aplicación de mejoras y actualizaciones a la plataforma SIGP. Esto con el propósito de contar con estos insumos y poder validar su vigencia y de esta manera, proceder con la generación del ticket, y que a partir de este se surtan los tiempos proyectados de acuerdo con su plan de trabajo.

Quedo atento a sus comentarios.

José Ignacio Español Pérez

Profesional Especializado

Oficina de Tecnologías y Sistemas de Información

jiespanol@minciencias.gov.co <jiespanol@minciencias.gov.co>

Teléfono: 601 6258480 Ext. 3507

Av. Calle 26 No. 57 - 83

Torre 8, pisos 2 al 6

Bogotá, Colombia

www.minciencias.gov.co

El vie, 28 feb 2025 a la(s) 4:56 p.m., César Riaño (cesar_riano@softwarebuilders.co) escribió:

> Buenas tardes José Ignacio,

>

> Para dar entrega al Producto Número 2 del contrato 237-2025, Módulo de

> Administración que comprende:

>

> - Creación pantalla de configuración de Dependencias.

> - Creación pantalla de configuración de Tablas.

> - Creación pantalla de configuración de Campos.

>
> Se requiere de la creación de la correspondiente solicitud en la mesa de
> servicio, con el fin de dar seguimiento al proceso de instalación, entrega
> de documentación y despliegue en ambiente de producción del mismo.
>
> Solicitamos nos colaboren con la creación de la solicitud, para iniciar la
> entrega correspondiente.
>
> También para recordarles los siguientes temas pendientes:
>
> - Entrega de equipo en MinCiencias para que el Ing. Leonardo pueda
> realizar sus actividades de soporte y se puedan realizar los despliegues en
> el servidor.
> - Solicitar el OtroSi acordado en la reunión de Inicio 21-Feb-2025 para
> pasar el Producto 17 del Pago 6 al Pago 4.
> - La asignación del funcionario de Minciencias para realizar la
> transferencia de conocimiento del SIGP.
>
>
> Cordialmente,
>
>
>
>
>
> Ing. César A. Riaño D.
> Gerente General
> Software Builders
> 601 4713433 | 320 3425285 <320%203425292>
> cesar_riano@softwarebuilders.co <cirj@software-b.com>
> www.softwarebuilders.co <http://softwarebuilders.co>
> Carrera 70D # 51 - 57
>
>
>
>
>

> > 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 16:31:48	07/05/2025 09:37:22		

Acción Realizada 154297

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 154455

Asunto	Tiempo total de actividad
SOLICITUD TI Integración módulo de auditoría en SIGP	00:32:47
Descripción <p>Buenas tardes:</p> <p>Modulo de Auditoria</p> <ul style="list-style-type: none"> * Revisar y resolver las Inconsistencias en el registro de Auditoría del sistema SIGP * Implementar el registro de Auditoría al momento de Ingreso y cierre de sesión de personas y proyectos * Implementar la Creación de componentes para la gestión de auditoría dentro del sistema SIGP <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>--</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
11/03/2025 14:28:26	07/05/2025 09:43:15		

Acción Realizada 154455

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 156699

Asunto		Tiempo total de actividad	
Implementación ambiente de pruebas		00:34:17	
Descripción			
Se requiere la configuración de un ambiente de pruebas para el Sistema Integral de Gestión de Proyectos (SIGP), con el fin de validar nuevas funcionalidades, realizar pruebas de integración y asegurar la estabilidad del sistema antes de su paso a producción.			
Se anexa el formato de solicitud con la especificación del requerimiento.			
Fecha/hora de apertura	Última modificación	Fecha/hora de resolución	Fecha/hora de cierre

23/04/2025 20:05:18	13/05/2025 15:04:09		
------------------------	------------------------	--	--

Acción Realizada 156699

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.

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.

Firma:

César A. Riaño D.

Ing. César Augusto Riaño Delgado
Gerente Proyecto