



Universidad
Católica del
Uruguay

Trabajo obligatorio

Segunda Entrega

Trabajo Obligatorio

Ingeniería en Informática

Bases de Datos

Integrantes del grupo:

- Federico Becoña
- Rodrigo López
- Julio Suaya

Docentes:

- Esteban Barrios
-

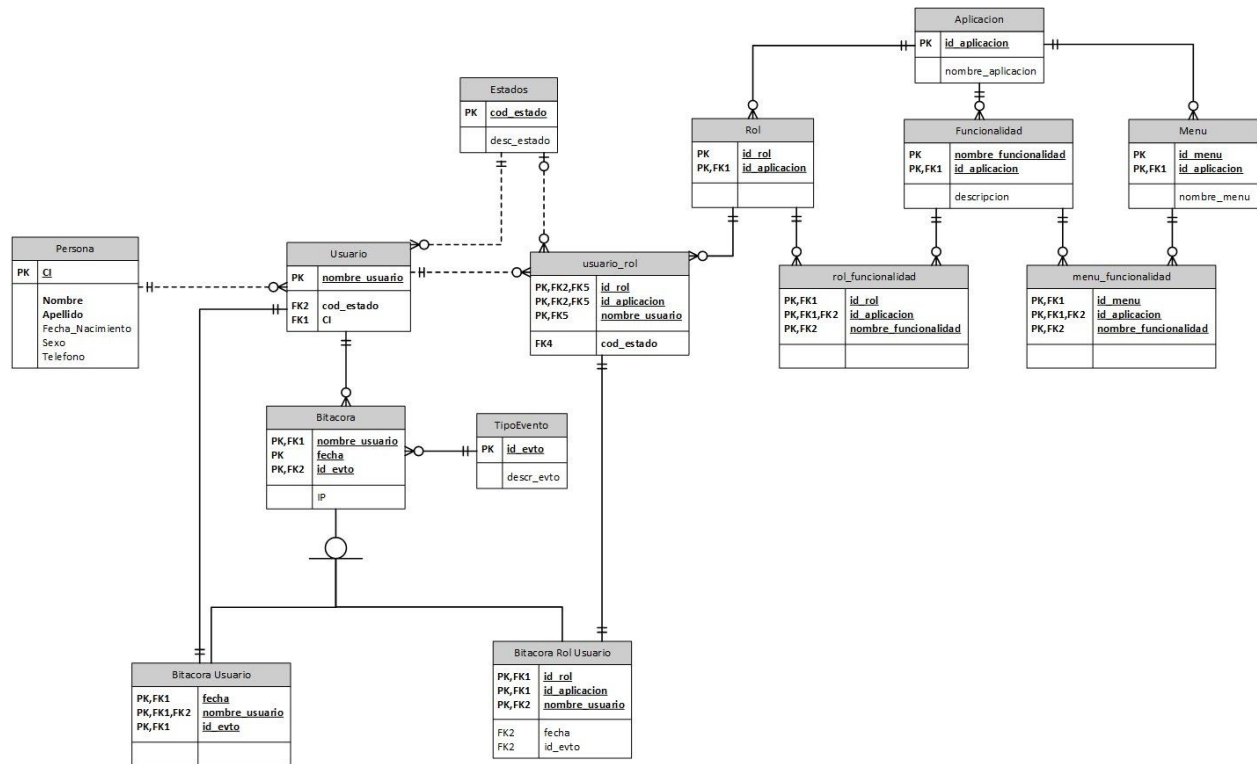
| | |
|--|-----------|
| Requerimientos | 3 |
| Modelo Entidad Relación | 5 |
| Diseño lógico | 6 |
| Especificacion | 8 |
| Estructura | 8 |
| Representación | 8 |
| Estructura de directorio Sistema | 27 |
| Estructura de directorio Aplicación Genérica | 28 |
| Estructura de la implementación Sistema Seguridad | 29 |
| Estructura de la implementación Aplicación Genérica | 37 |
| Casos de uso | 39 |

Requerimientos

- Aplicación.
 - Está compuesta por funcionalidades.
 - Tiene uno o más usuarios.
- Funcionalidad.
 - Forma parte de una aplicación.
 - Está agrupada en menús.
- Menú.
 - Forma parte de una aplicación.
 - Tienen una o más funcionalidades.
- Rol.
 - Define un conjunto de funcionalidades dentro de una aplicación a las que pueden acceder los usuarios.
- Usuario.
 - Puede acceder a una o más aplicaciones.
 - Dentro de las aplicaciones tienen uno o más roles.
 - Cuenta con un nombre y una clave que es guardada en forma de hash en el sistema para mayor seguridad.
- Persona.
 - Puede tener uno o más usuarios.
 - Contiene datos filiatorios.
- Administrador del sistema.
 - Gestión de personas.
 - Puede crear, modificar o eliminar personas del sistema.
 - Gestión de usuarios.

- Puede crear, modificar o eliminar usuarios del sistema.
- Puede vincular o desvincular un usuario con un determinado rol.
- Para la gestión de usuarios se utiliza oposición de intereses.
- Gestión de aplicaciones.
 - Puede crear o eliminar aplicaciones del sistema.
 - Dada una aplicación puede crear, modificar o eliminar menús.
 - Dada una aplicación puede crear, modificar o eliminar funcionalidades.
 - Se pueden agrupar funcionalidades en menús.
 - Puede crear, modificar o eliminar roles para una determinada aplicación.
- Auditoria.
 - Todas las operaciones efectuadas por los administradores del sistema deben ser guardadas.

Modelo Entidad Relación



Diseño lógico

```
CREATE TABLE public."aplicacion" (  
    id_aplicacion character varying(15) NOT NULL,  
    nombre_aplicacion character varying(25)  
);
```

```
CREATE TABLE public."estados" (  
    cod_estado character varying(10) NOT NULL,  
    desc_estado character varying(50)  
);
```

```
CREATE TABLE public."funcionalidad" (  
    nombre_funcionalidad character varying(15) NOT NULL,  
    id_aplicacion character varying(15) NOT NULL,  
    descripcion character varying(30)  
);
```

```
CREATE TABLE public."menu" (  
    id_menu character varying(15) NOT NULL,  
    id_aplicacion character varying(15) NOT NULL,  
    nombre_menu character varying(15)  
);
```

```
CREATE TABLE public."menu_funcionalidad" (  
    id_menu character varying(15) NOT NULL,  
    id_aplicacion character varying(15) NOT NULL,  
    nombre_funcionalidad character varying(15) NOT NULL  
);
```

```
CREATE TABLE public."persona" (  
    "CI" character varying(14) NOT NULL,  
    nombre character varying(15) NOT NULL,  
    apellido character varying(15) NOT NULL,  
    fecha_nacimiento date,  
    sexo character varying(10),  
    telefono character varying(15)  
);
```

```
CREATE TABLE public."rol" (  
    id_rol character varying(15) NOT NULL,  
    id_aplicacion character varying(15) NOT NULL  
);
```

```
CREATE TABLE public."rol_funcionalidad" (  
    id_rol character varying(15) NOT NULL,  
    id_aplicacion character varying(15) NOT NULL,  
    nombre_funcionalidad character varying(15) NOT NULL  
);
```

```
CREATE TABLE public."tipo_evento" (  
    id_evento character varying(10) NOT NULL,  
    desc_evento character varying(50)  
);
```

```
CREATE TABLE public."usuario" (  
    nombre_usuario character varying(15) NOT NULL,  
    contrasena character varying(15) NOT NULL,  
    ci character varying(10) NOT NULL,  
    cod_estado character varying(10) NOT NULL  
);
```

```
CREATE TABLE public."usuario_rol" (  
    id_rol character varying(15) NOT NULL,  
    id_aplicacion character varying(15) NOT NULL,  
    "CI" character varying(15) NOT NULL,  
    nombre_usuario character varying(15) NOT NULL  
);
```

```
CREATE TABLE public."usuario_bitacora"(  
    usuario_modificador character varying NOT NULL,  
    id_evento character varying NOT NULL,  
    fecha timestamp without time zone NOT NULL,  
    nombre_usuario character varying NOT NULL,  
);
```

```
CREATE TABLE public.usuario_rol_bitacora  
(  
    id_evento character varying NOT NULL,  
    fecha timestamp without time zone NOT NULL,  
    usuario_modificador character varying NOT NULL,  
    id_rol character varying COLLATE NOT NULL,  
    id_aplicacion character varying NOT NULL,  
    nombre_usuario character varying NOT NULL,  
);
```

Especificacion

La aplicación será construida utilizando la modalidad cliente-servidor

El motor de base de datos elegido es Postgres SQL en su version 0.0 el cual está instalado en una máquina virtual con sistema operativo Ubuntu 0.0.

Postgres fue elegido porque entre otras cosas es de código abierto y se puede utilizar de forma gratuita y fue la herramienta que utilizamos durante el curso

Estructura

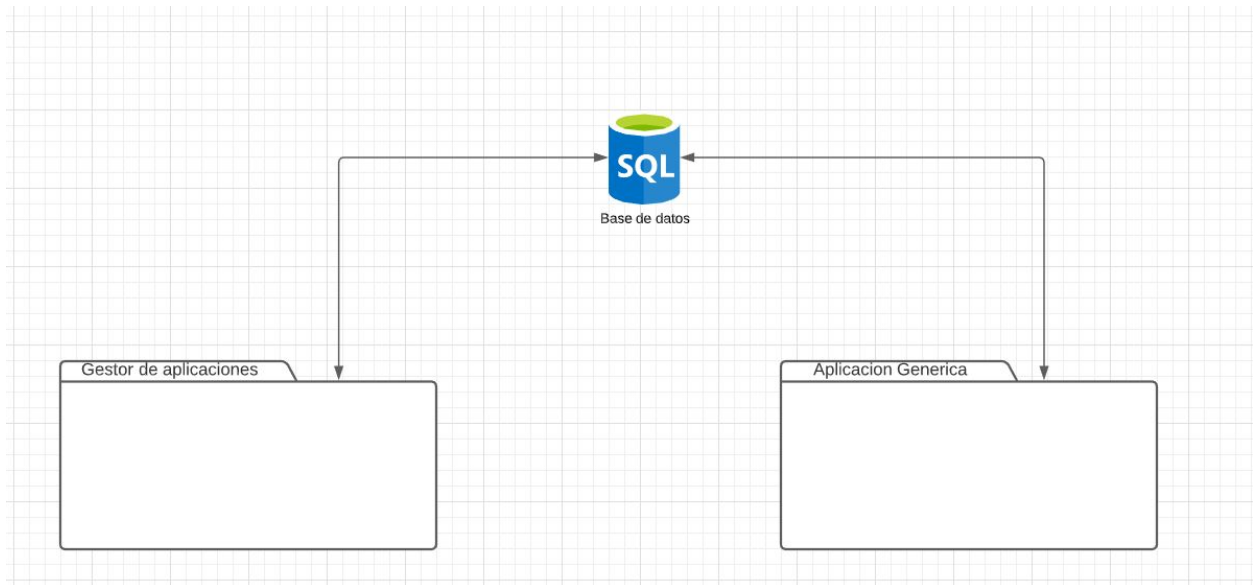
El proyecto se compone de 2 programas,

El primero es para la gestión del sistema de seguridad, este programa permite administrar las tablas necesarias para cumplir con su función, como por ejemplo crear aplicaciones, usuarios, roles, funcionalidades, menús, asignar roles a los usuarios, funcionalidades a los menús,

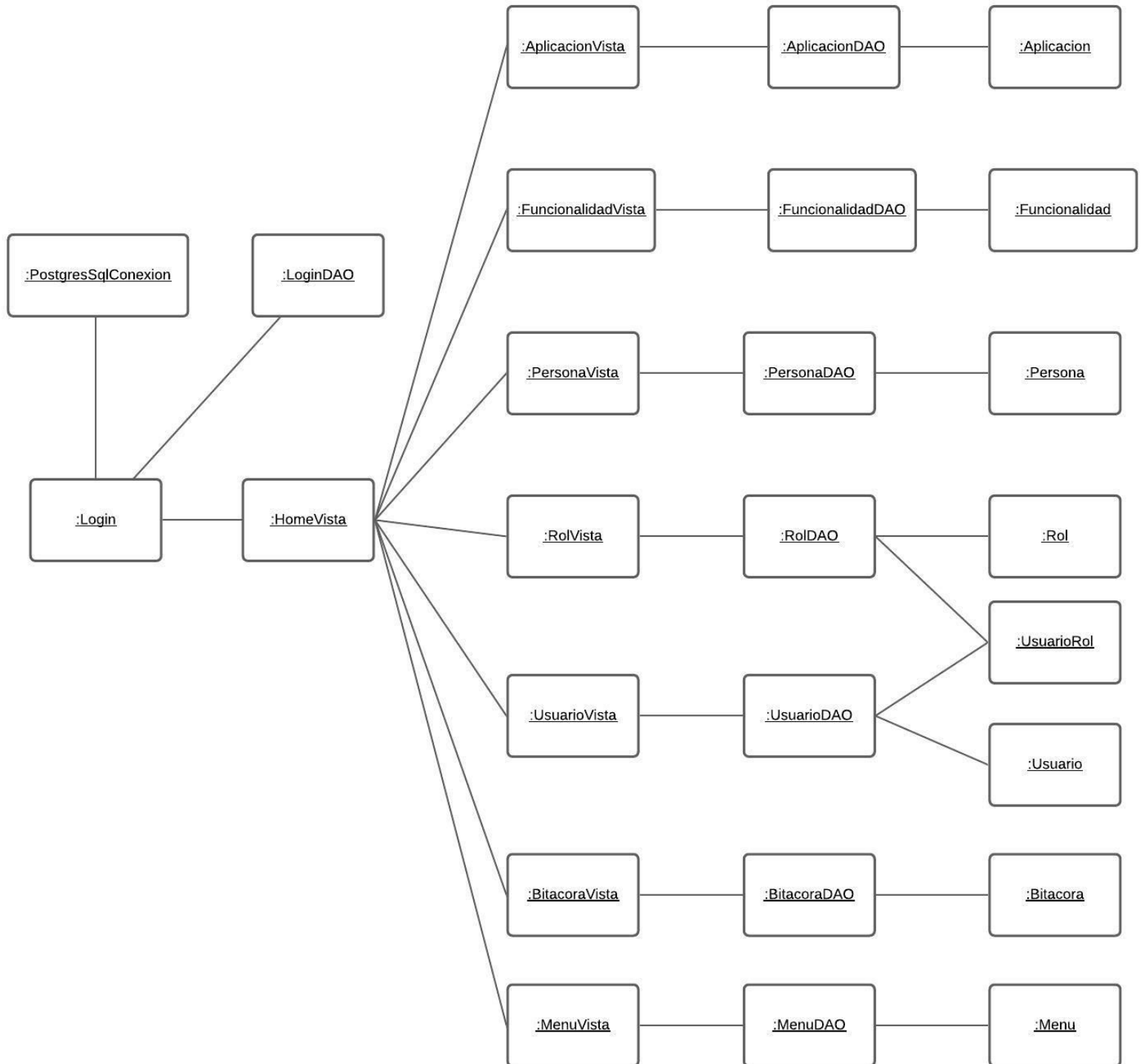
El segundo programa es una aplicación básica, mediante la cual el usuario se loguea a la base de datos, utilizando un ofuscamiento, y averigua a qué menús y funcionalidades tiene acceso.

La idea detrás de esto, es que las aplicaciones cliente, utilicen estas funcionalidades como deseen.

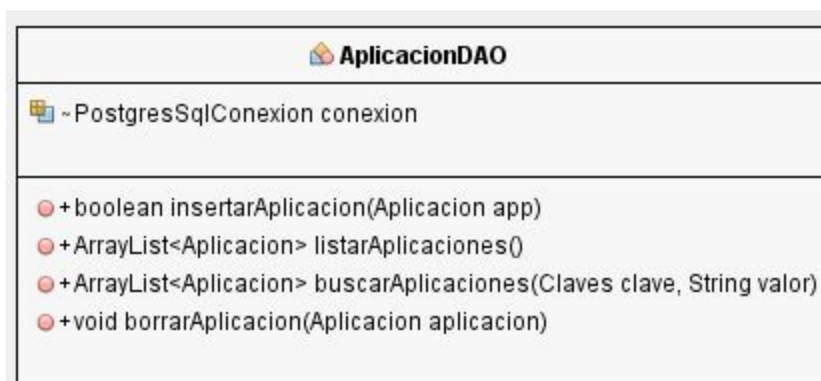
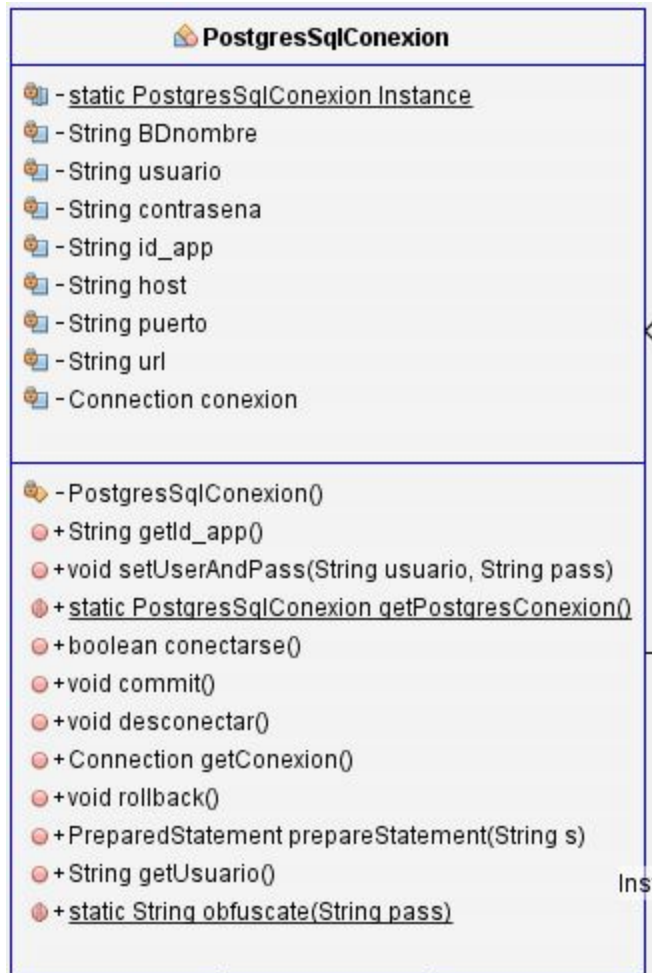
Representación

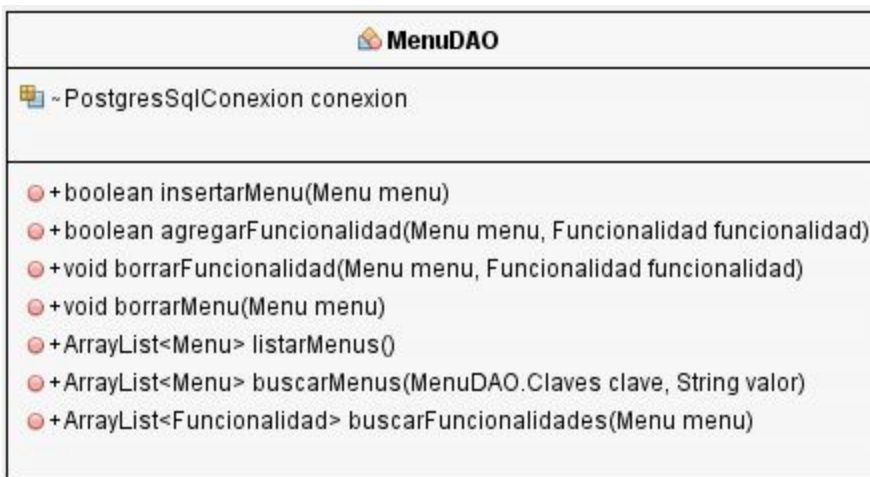
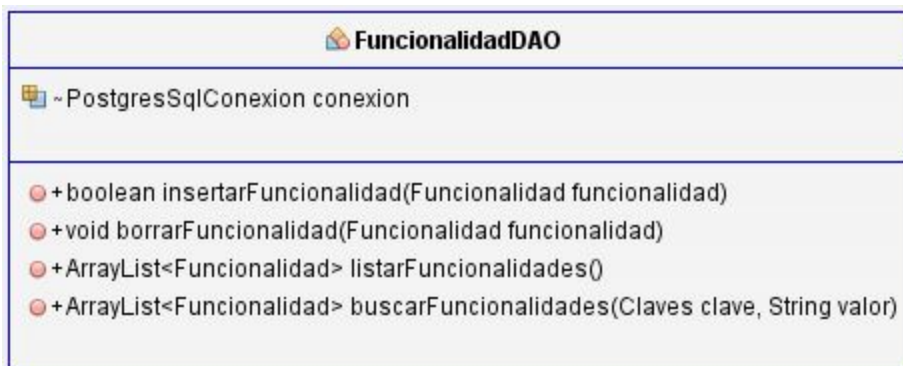
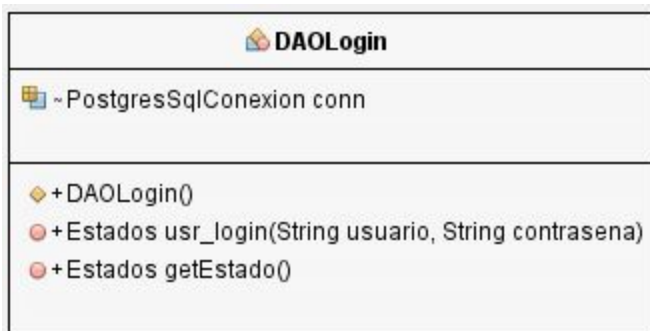
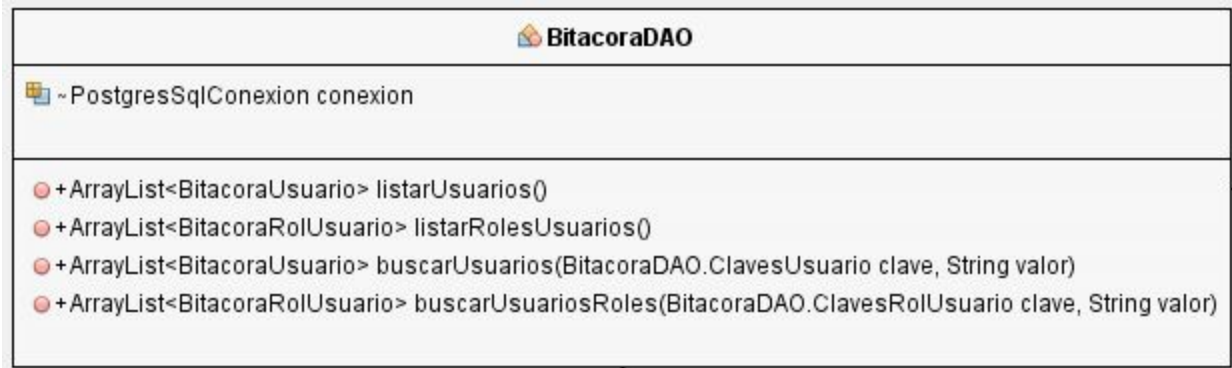


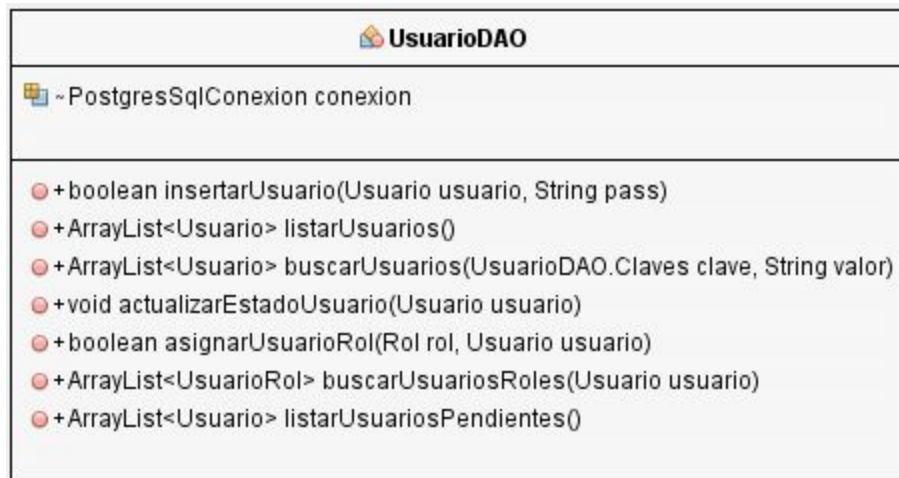
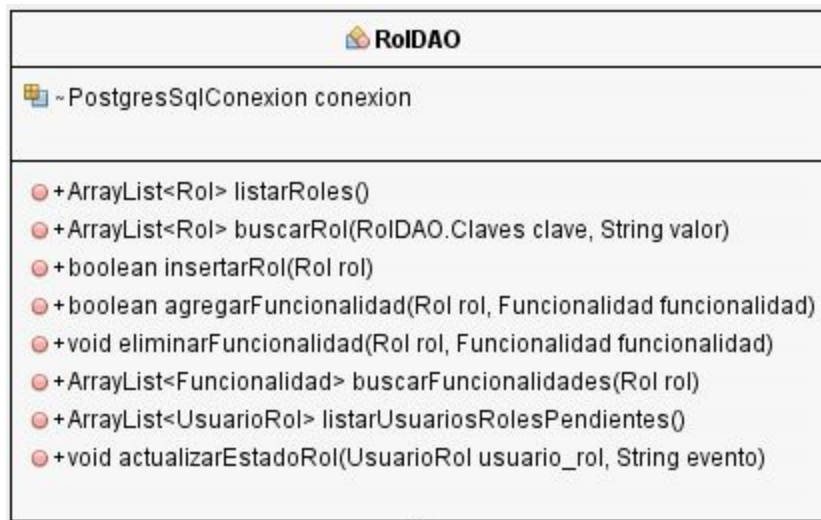
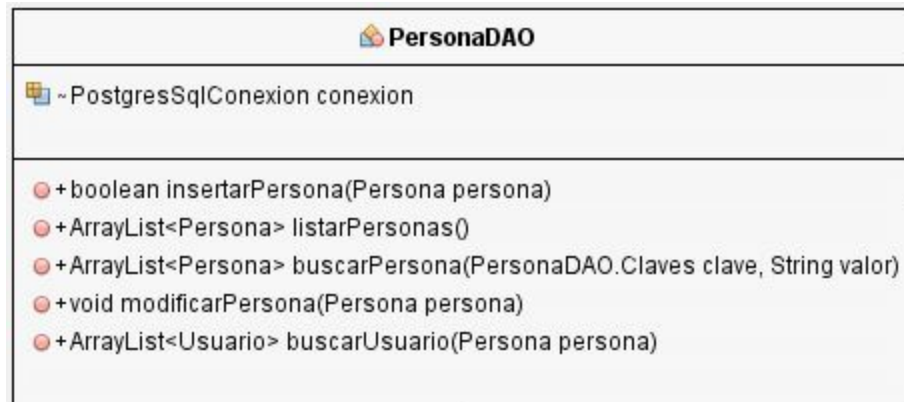
Diagramas de Colaboración Sistema Seguridad


















Diagramas de Clase Sistema Seguridad





























|  Aplicacion |
|---|
|  ~String id_aplicacion  ~String nombre_aplicacion |
|  +Aplicacion(String id_aplicacion, String nombre_aplicacion)  +String getId_aplicacion()  +String toString()  +String getNombre_aplicacion() |

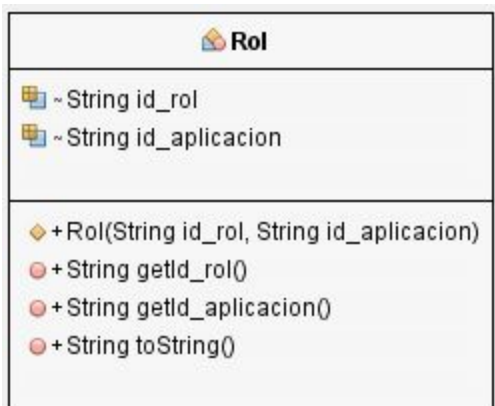
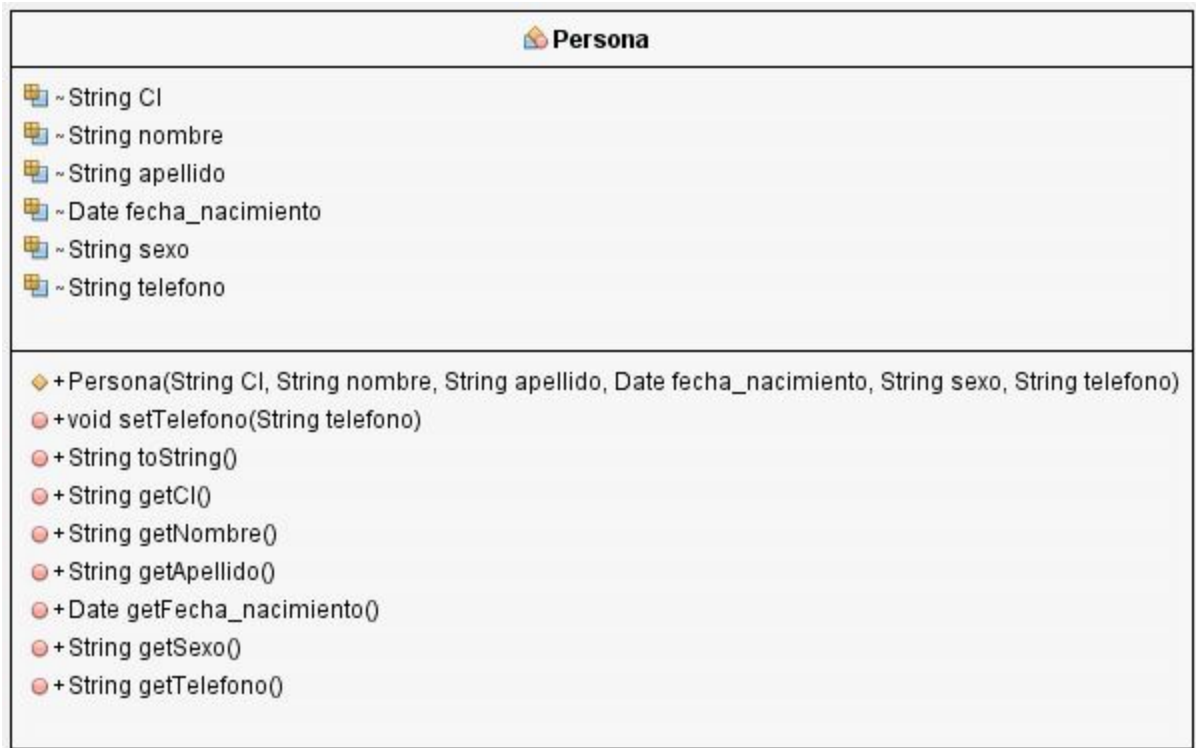
| <<abstract>>  Bitacora |
|--|
|  ~String id_evto  ~String usuario_modificador  ~Timestamp fecha |
|  +Bitacora(String id_evto, String usuario_modificador, Timestamp fecha)  +String getId_evto()  +String getUsuario_modificador()  +Timestamp getFecha() |

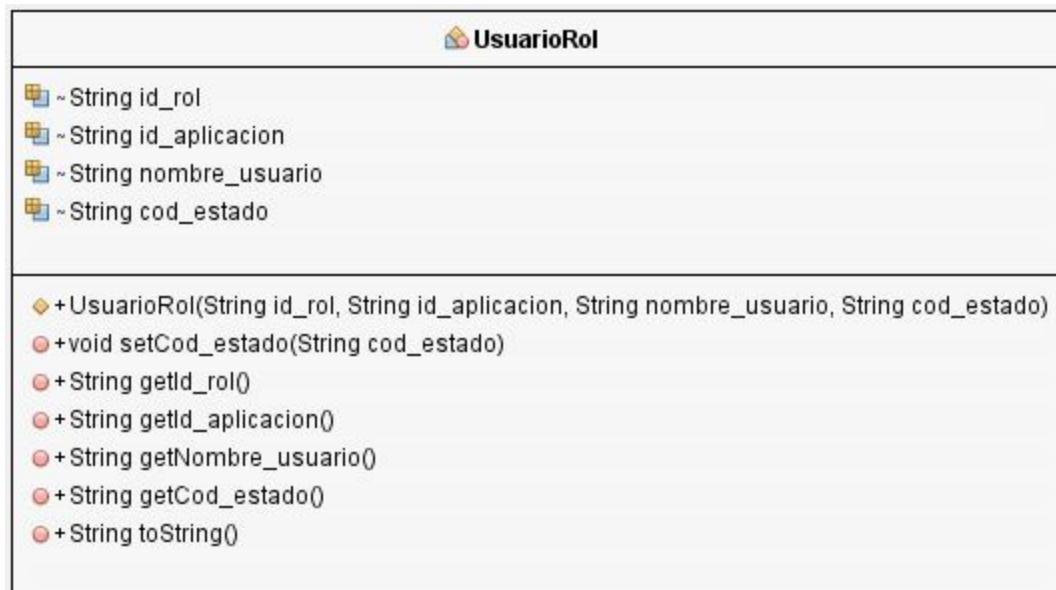
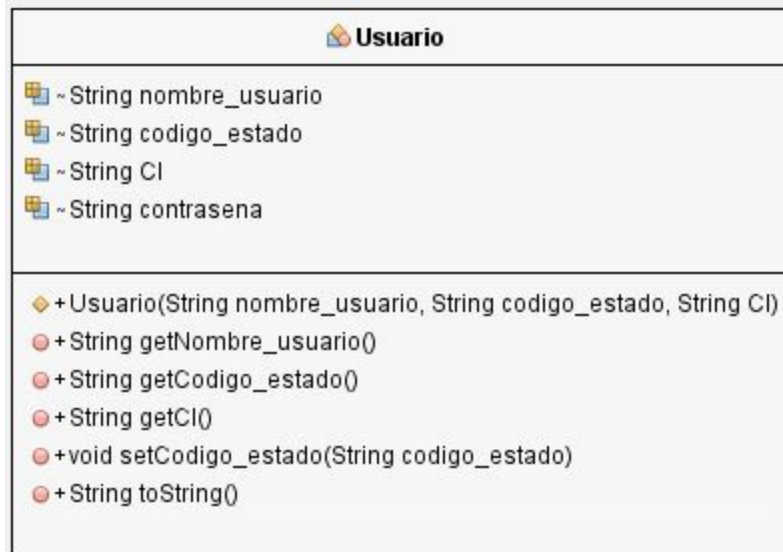
|  BitacoraRolUsuario |
|--|
|  ~String nombre_usuario  ~String id_rol  ~String id_aplicacion |
|  +BitacoraRolUsuario(String id_evto, String usuario_modificador, Timestamp fecha, String nombre_usuario, String id_rol, String id_aplicacion)  +String getId_rol()  +String getId_aplicacion()  +String getNombre_usuario() |

|  BitacoraUsuario |
|---|
|  ~String nombre_usuario |
|  +BitacoraUsuario(String id_evto, String usuario_modificador, Timestamp fecha, String nombre_usuario)  +String getNombre_usuario() |

|  Funcionalidad |
|---|
| <div>  ~String descripcion </div> <div>  ~String id_aplicacion </div> <div>  ~String nombre_funcionalidad </div> |
| <div>  +Funcionalidad(String nombre_funcionalidad, String id_aplicacion, String descripcion) </div> <div>  +String toString() </div> <div>  +String getDescripcion() </div> <div>  +String getId_aplicacion() </div> <div>  +String getNombre_funcionalidad() </div> |

|  Menu |
|---|
| <div>  ~String id_menu </div> <div>  ~String nombre </div> <div>  ~String id_aplicacion </div> |
| <div>  +Menu(String id_menu, String nombre, String id_aplicacion) </div> <div>  +String getId_menu() </div> <div>  +String getNombre() </div> <div>  +String toString() </div> <div>  +String getId_aplicacion() </div> |
















AplicacionVista

-  - AplicacionDAO app_dao
-  - javax.swing.JLabel errorMsg
-  - javax.swing.ButtonGroup buttonGroup1
-  - javax.swing.JButton insertButton
-  - javax.swing.JTextField insertIdField
-  - javax.swing.JTextField insertNameField
-  - javax.swing.JButton jButton8
-  - javax.swing.JButton jButton9
-  - javax.swing.JLabel jLabel1
-  - javax.swing.JLabel jLabel2
-  - javax.swing.JPanel jPanel1
-  - javax.swing.JPanel jPanel2
-  - javax.swing.JPanel jPanel3
-  - javax.swing.JScrollPane jScrollPane2
-  - javax.swing.JTabbedPane jTabbedPane1
-  - javax.swing.JButton searchButton
-  - javax.swing.JComboBox<String> searchColumn
-  - javax.swing.JTextField searchField
-  - javax.swing.JTable searchResultsTable









+ AplicacionVista()

-  -// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents void initComponents()
-  -void searchColumnActionPerformed(java.awt.event.ActionEvent evt)
-  -void searchButtonActionPerformed(java.awt.event.ActionEvent evt)
-  -void insertButtonActionPerformed(java.awt.event.ActionEvent evt)
-  -void insertNameFieldActionPerformed(java.awt.event.ActionEvent evt)
-  -void jButton8ActionPerformed(java.awt.event.ActionEvent evt)
-  -void jButton9ActionPerformed(java.awt.event.ActionEvent evt)
-  + static void main(String args)










































BitacoraVista

-  ~BitacoraDAO bit_dao
-  - javax.swing.ButtonGroup buttonGroup1
-  - javax.swing.JButton jButton8
-  - javax.swing.JButton jButton9
-  - javax.swing.JPanel jPanel1
-  - javax.swing.JPanel jPanel4
-  - javax.swing.JScrollPane jScrollPane2
-  - javax.swing.JScrollPane jScrollPane3
-  - javax.swing.JTabbedPane jTabbedPane1
-  - javax.swing.JButton searchButtonRolUsr
-  - javax.swing.JButton searchButtonUsr
-  - javax.swing.JComboBox<String> searchColumn
-  - javax.swing.JComboBox<String> searchColumn1
-  - javax.swing.JTextField searchField
-  - javax.swing.JTextField searchField1
-  - javax.swing.JTable searchResultsTable
-  - javax.swing.JTable searchResultsTable1








 + BitacoraVista()

-  -// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents void initComponents()
-  -void searchColumnActionPerformed(java.awt.event.ActionEvent evt)
-  -void searchButtonUsrActionPerformed(java.awt.event.ActionEvent evt)
-  -void jButton9ActionPerformed(java.awt.event.ActionEvent evt)
-  -void jButton8ActionPerformed(java.awt.event.ActionEvent evt)
-  -void searchColumn1ActionPerformed(java.awt.event.ActionEvent evt)
-  -void searchButtonRolUsrActionPerformed(java.awt.event.ActionEvent evt)
-  + static void main(String args)

FuncionalidadVista

-  - FuncionalidadDAO fun_dao
-  - RolDAO rol_dao
-  - AplicacionDAO app_dao
-  - javax.swing.JButton BotonRolFuncionalidad
-  - javax.swing.JComboBox<Funcionalidad> CBoxAsignarFuncionalidad
-  - javax.swing.JComboBox<Rol> CBoxAsignarRol
-  - javax.swing.JComboBox<Aplicacion> CBoxIFApp
-  - javax.swing.JComboBox<Rol> CBoxRol
-  - javax.swing.JComboBox<Rol> CBoxRolFuncionalidad
-  - javax.swing.JComboBox<Usuario> CBoxUsuario
-  - javax.swing.JLabel ErrorMsg
-  - javax.swing.JLabel MsgAsignarRol
-  - javax.swing.JTable TablaRolFuncionalidad
-  - javax.swing.JButton insertButton
-  - javax.swing.JTextField insertDescripcionField
-  - javax.swing.JTextField insertNameField
-  - javax.swing.JButton jButton1
-  - javax.swing.JButton jButton2
-  - javax.swing.JButton jButton8
-  - javax.swing.JButton jButton9
-  - javax.swing.JLabel jLabel1
-  - javax.swing.JLabel jLabel2
-  - javax.swing.JLabel jLabel3
-  - javax.swing.JLabel jLabel4
-  - javax.swing.JLabel jLabel5
-  - javax.swing.JLabel jLabel6
-  - javax.swing.JLabel jLabel7
-  - javax.swing.JLabel jLabel8
-  - javax.swing.JPanel jPanel1
-  - javax.swing.JPanel jPanel2
-  - javax.swing.JPanel jPanel3
-  - javax.swing.JPanel jPanel4
-  - javax.swing.JPanel jPanel5
-  - javax.swing.JPanel jPanel6
-  - javax.swing.JScrollPane jScrollPane1
-  - javax.swing.JScrollPane jScrollPane2
-  - javax.swing.JTabbedPane jTabbedPane1
-  - javax.swing.JButton searchButton
-  - javax.swing.JComboBox<String> searchColumn
-  - javax.swing.JTextField searchField
-  - javax.swing.JTable searchResultsTable

+ FuncionalidadVista()

-  - // <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents void initComponents()
-  - void searchColumnActionPerformed(java.awt.event.ActionEvent evt)
-  - void searchButtonActionPerformed(java.awt.event.ActionEvent evt)
-  - void insertButtonActionPerformed(java.awt.event.ActionEvent evt)
-  - void jButton8ActionPerformed(java.awt.event.ActionEvent evt)
-  - void jButton9ActionPerformed(java.awt.event.ActionEvent evt)
-  - void BotonRolFuncionalidadActionPerformed(java.awt.event.ActionEvent evt)

Home

- javax.swing.JButton jButton1
- javax.swing.JButton jButton2
- javax.swing.JButton jButton3
- javax.swing.JButton jButton4
- javax.swing.JButton jButton5
- javax.swing.JButton jButton6
- javax.swing.JButton jButton7
- javax.swing.JButton jButton8
- javax.swing.JPanel jPanel1

+ Home()

```
// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents void initComponents()
-void jButton2ActionPerformed(java.awt.event.ActionEvent evt)
-void jButton5ActionPerformed(java.awt.event.ActionEvent evt)
-void jButton7ActionPerformed(java.awt.event.ActionEvent evt)
-void jButton2MouseClicked(java.awt.event.MouseEvent evt)
-void jButton1MouseClicked(java.awt.event.MouseEvent evt)
-void jButton3MouseClicked(java.awt.event.MouseEvent evt)
-void jButton4MouseClicked(java.awt.event.MouseEvent evt)
-void jButton6MouseClicked(java.awt.event.MouseEvent evt)
-void jButton5MouseClicked(java.awt.event.MouseEvent evt)
-void jButton7MouseClicked(java.awt.event.MouseEvent evt)
-void jButton8ActionPerformed(java.awt.event.ActionEvent evt)
+ static void main(String args)
```





































Login


- DAOLogin log_dao
- javax.swing.JButton jButton1
- javax.swing.JLabel jLabel1
- javax.swing.JLabel jLabel2
- javax.swing.JLabel jLabel3
- javax.swing.JPanel jPanel1
- javax.swing.JPanel jPanel2
- javax.swing.JScrollPane jScrollPane1
- javax.swing.JLabel msg_login
- javax.swing.JPasswordField pass_input
- javax.swing.JTextPane user_input












+ Login()

```
// <editor-fold defaultstate="collapsed" desc="Generated Code"> //GEN-BEGIN: initComponents void initComponents()
-void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
-void jButton1MouseClicked(java.awt.event.MouseEvent evt)
+ static void main(String args)
```






















MenuVista


-  - MenuDAO menu_dao
-  - FuncionalidadDAO func_dao
-  - AplicacionDAO app_dao
-  - javax.swing.JButton BotonAgregarFunc
-  - javax.swing.JComboBox<Aplicacion> CBoxAppls
-  - javax.swing.JComboBox<Funcionalidad> CBoxFuncionalidad
-  - javax.swing.JComboBox<Menu> CBoxMenu
-  - javax.swing.JComboBox<Menu> CBoxMenuFunc
-  - javax.swing.JComboBox<Aplicacion> ComboAppAgregarFunc
-  - javax.swing.JLabel ErrorMsg
-  - javax.swing.JLabel MsgAgregarFunc
-  - javax.swing.JTable ResMenuVista
-  - javax.swing.JButton botonBuscarMenuFunc
-  - javax.swing.ButtonGroup buttonGroup1
-  - javax.swing.JButton insertButton
-  - javax.swing.JTextField insertIdMenuField
-  - javax.swing.JTextField insertNameField
-  - javax.swing.JButton jButton10
-  - javax.swing.JButton jButton9
-  - javax.swing.JLabel jLabel1
-  - javax.swing.JLabel jLabel2
-  - javax.swing.JLabel jLabel3
-  - javax.swing.JLabel jLabel4
-  - javax.swing.JLabel jLabel5
-  - javax.swing.JLabel jLabel6
-  - javax.swing.JLabel jLabel7
-  - javax.swing.JPanel jPanel1
-  - javax.swing.JPanel jPanel2
-  - javax.swing.JPanel jPanel3
-  - javax.swing.JPanel jPanel4
-  - javax.swing.JPanel jPanel5
-  - javax.swing.JScrollPane jScrollPane1
-  - javax.swing.JScrollPane jScrollPane2
-  - javax.swing.JTabbedPane jTabbedPane1
-  - javax.swing.JButton searchButton
-  - javax.swing.JComboBox<String> searchColumn
-  - javax.swing.JTextField searchField
-  - javax.swing.JTable searchResultsTable








 + MenuVista()

-  -// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents void initComponents()
-  -void searchColumnActionPerformed(java.awt.event.ActionEvent evt)
-  -void searchButtonActionPerformed(java.awt.event.ActionEvent evt)
-  -void insertButtonActionPerformed(java.awt.event.ActionEvent evt)
-  -void jButton9ActionPerformed(java.awt.event.ActionEvent evt)
-  -void jButton10ActionPerformed(java.awt.event.ActionEvent evt)
-  -void BotonAgregarFuncActionPerformed(java.awt.event.ActionEvent evt)
-  -void botonBuscarMenuFuncActionPerformed(java.awt.event.ActionEvent evt)
-  -void CBoxMenuActionPerformed(java.awt.event.ActionEvent evt)
-  -void ComboAppAgregarFuncActionPerformed(java.awt.event.ActionEvent evt)
-  + static void main(String args)

PersonaVista

-  - PersonaDAO per_dao
-  - javax.swing.JLabel Cellphone
-  - javax.swing.JLabel ErrorMsg
-  - javax.swing.ButtonGroup buttonGroup1
-  - javax.swing.JButton insertButton
-  - javax.swing.JTextField insertCellphoneField
-  - com.github.lgooddatepicker.components.DatePicker insertDateField
-  - javax.swing.JTextField insertIdField
-  - javax.swing.JTextField insertNameField
-  - javax.swing.JTextField insertSexField
-  - javax.swing.JTextField insertSurnameField
-  - javax.swing.JButton jButton8
-  - javax.swing.JButton jButton9
-  - javax.swing.JLabel jLabel2
-  - javax.swing.JLabel jLabel3
-  - javax.swing.JLabel jLabel4
-  - javax.swing.JLabel jLabel5
-  - javax.swing.JLabel jLabel6
-  - javax.swing.JPanel jPanel1
-  - javax.swing.JPanel jPanel2
-  - javax.swing.JPanel jPanel3
-  - javax.swing.JScrollPane jScrollPane2
-  - javax.swing.JTabbedPane jTabbedPane1
-  - javax.swing.JButton searchButton
-  - javax.swing.JComboBox<String> searchColumn
-  - javax.swing.JTextField searchField
-  - javax.swing.JTable searchResultsTable

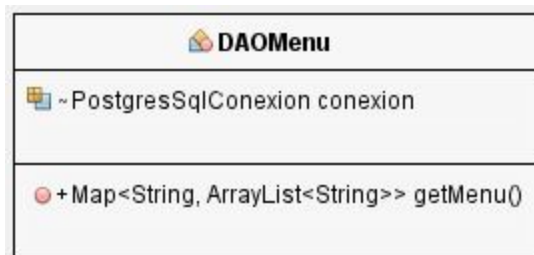
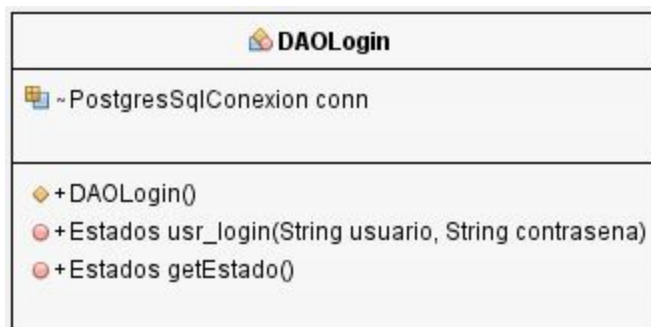
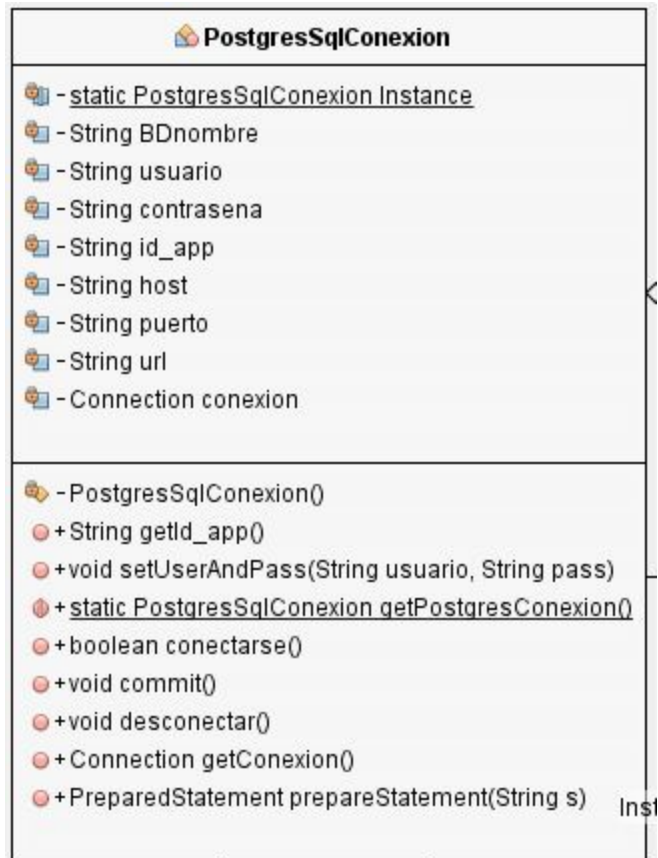
 + PersonaVista()










-  -// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents void initComponents()
-  - void searchColumnActionPerformed(java.awt.event.ActionEvent evt)
-  - void searchButtonActionPerformed(java.awt.event.ActionEvent evt)
-  - void insertButtonActionPerformed(java.awt.event.ActionEvent evt)
-  - void jButton8ActionPerformed(java.awt.event.ActionEvent evt)
-  - void jButton9ActionPerformed(java.awt.event.ActionEvent evt)
-  + static void main(String args)










- RoIDAO rol_dao
- UsuarioDAO usr_dao
- AplicacionDAO app_dao
- javax.swing.JButton BotonUsuarioRol
- javax.swing.JButton BtnAsignar
- javax.swing.JButton BtnRevocar
- javax.swing.JComboBox<Aplicacion> CBoxIFApp
- javax.swing.JComboBox<Rol> CBoxRol
- javax.swing.JComboBox<UsuarioRol> CBoxRolRevocar
- javax.swing.JComboBox<Usuario> CBoxUsuario
- javax.swing.JComboBox<Usuario> CBoxUsuarioRevocar
- javax.swing.JComboBox<Usuario> CBoxUsuarioRol
- javax.swing.JLabel ErrorMsg
- javax.swing.JLabel MsgAsignarRol
- javax.swing.JLabel MsgAsignarRol1
- javax.swing.JTable TablaRolUsuario
- javax.swing.ButtonGroup buttonGroup1
- javax.swing.JButton insertButton
- javax.swing.JTextField insertIdRolField
- javax.swing.JButton jButton10
- javax.swing.JButton jButton9
- javax.swing.JLabel jLabel1
- javax.swing.JLabel jLabel2
- javax.swing.JLabel jLabel3
- javax.swing.JLabel jLabel4
- javax.swing.JLabel jLabel5
- javax.swing.JLabel jLabel6
- javax.swing.JLabel jLabel7
- javax.swing.JPanel jPanel1
- javax.swing.JPanel jPanel2
- javax.swing.JPanel jPanel3
- javax.swing.JPanel jPanel4
- javax.swing.JPanel jPanel5
- javax.swing.JPanel jPanel6
- javax.swing.JPanel jPanel7
- javax.swing.JScrollPane jScrollPane1
- javax.swing.JScrollPane jScrollPane2
- javax.swing.JScrollPane jScrollPane3
- javax.swing.JTabbedPane jTabbedPane1
- javax.swing.JTable rolUsuarioPorAprobar
- javax.swing.JButton searchButton
- javax.swing.JComboBox<String> searchColumn
- javax.swing.JTextField searchField
- javax.swing.JTable searchResultsTable








- + RoIvista()
- // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents void initComponents()
- void searchColumnActionPerformed(java.awt.event.ActionEvent evt)
- void searchButtonActionPerformed(java.awt.event.ActionEvent evt)
- void insertButtonActionPerformed(java.awt.event.ActionEvent evt)
- void jButton9ActionPerformed(java.awt.event.ActionEvent evt)

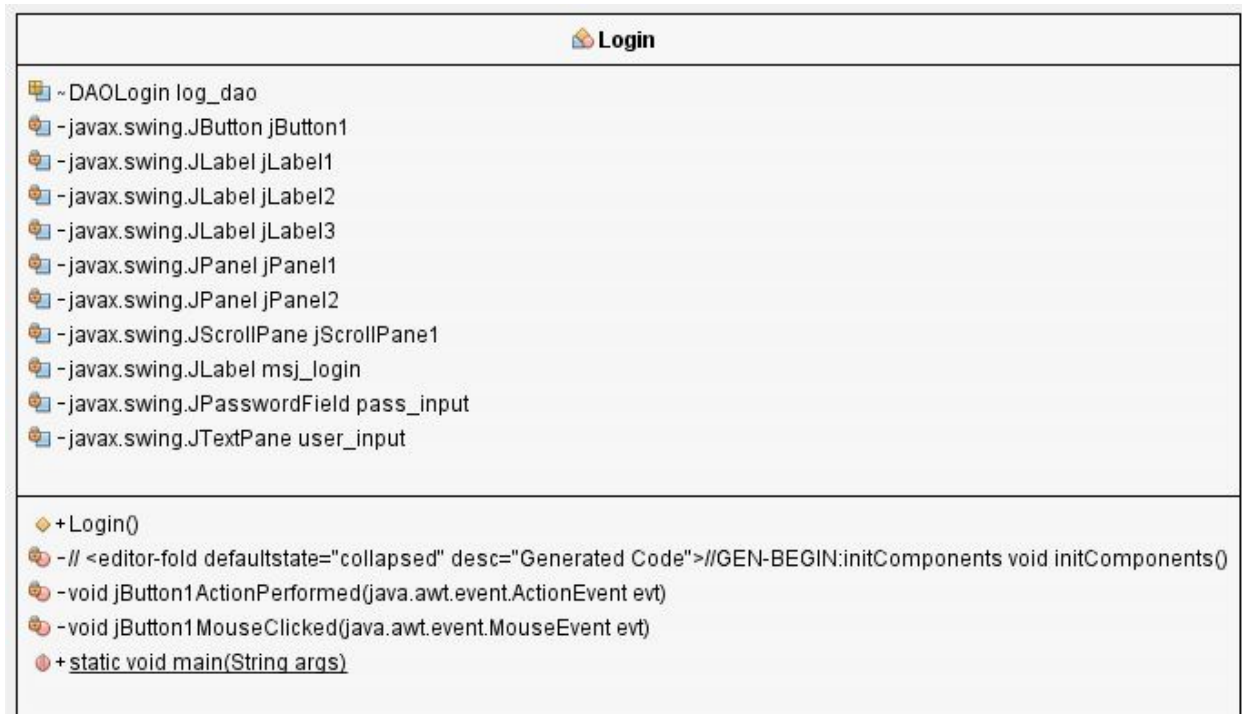
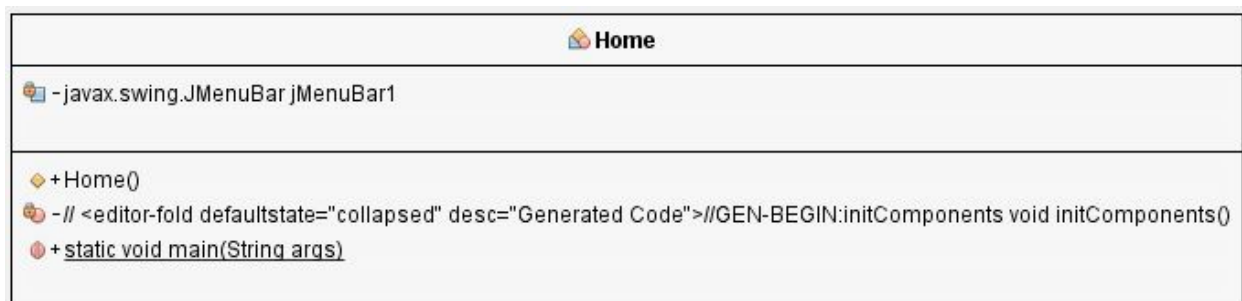
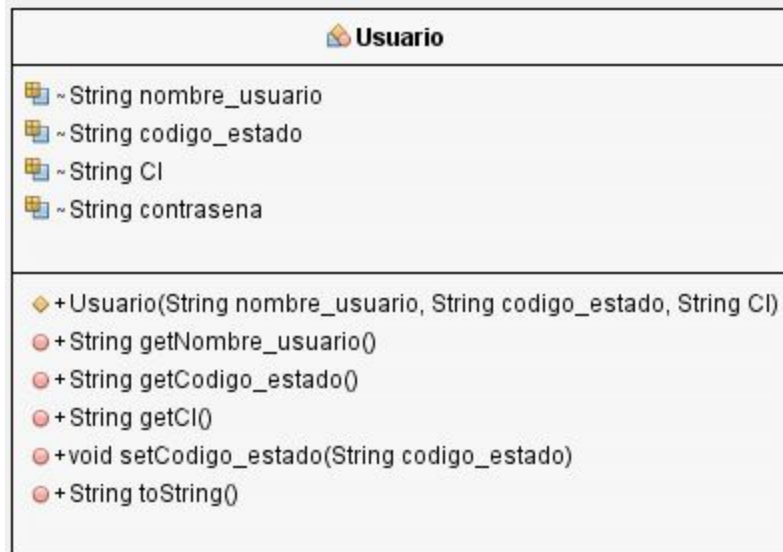
Diagramas de Clase Aplicación Genérica



|  Funcionalidad |
|---|
| <div>  ~String descripcion  ~String id_aplicacion  ~String nombre_funcionalidad </div> |
| <div>  +Funcionalidad(String nombre_funcionalidad, String id_aplicacion, String descripcion)  +String toString()  +String getDescripcion()  +String getId_aplicacion()  +String getNombre_funcionalidad() </div> |

|  Menu |
|---|
| <div>  ~String id_menu  ~String nombre  ~String id_aplicacion </div> |
| <div>  +Menu(String id_menu, String nombre, String id_aplicacion)  +String getId_menu()  +String getNombre()  +String toString()  +String getId_aplicacion() </div> |

|  Rol |
|--|
| <div>  ~String id_rol  ~String id_aplicacion </div> |
| <div>  +Rol(String id_rol, String id_aplicacion)  +String getId_rol()  +String getId_aplicacion()  +String toString() </div> |



Estructura de directorio Sistema

Conexion

- PostgresSqlConexion

DAO

- AplicacionDAO
- BitacoraDAO
- DAOLogin
- FuncionalidadDAO
- MenuDAO
- PersonaDAO
- RolDAO
- UsuarioDAO

Modelo

- Aplicacion
- Bitacora
- BitacoraRolUsuario
- BitacoraUsuario
- Funcionalidad
- Menu
- Persona
- Rol
- Usuario
- UsuarioRol

Vista

- AplicacionVista
- BitacoraVista
- FuncionalidadVista
- Home
- Login
- MenuVista
- PersonaVista
- RolVista
- UsuarioVista

Estructura de directorio Aplicación Genérica

Conexion

PostgresSqlConexion

DAO

DAOLogin

DAOMenu

Modelo

Funcionalidad

Menu

Rol

Usuario

Vista

Home

Login

Estructura de la implementación Sistema Seguridad

Clase: PostgresSqlConexion.java

Atributos

- Instance
Es la instancia del objeto.
- BDnombre
Es el nombre de la base de datos.
- Usuario
Es el nombre de acceso a la base de datos del usuario.
- contraseña
Es la contraseña de usuario de acceso a la base de datos.
- Id_app
Es el identificador de la aplicación del sistema
- Host
Es la dirección IPv4 de la conexión a la base de datos.
- puerto
Es el puerto de la conexión a la base de datos
- url
Dirección conexión a la base de datos
- Conexion
Conexión a la base de datos.

Metodos

- PostgresSqlConexion()
Constructor con singleton para el atributo "conexion".
- setUserAndPass(String usuario, String pass)
Establece el usuario y la contraseña de "conexion".
- conectarse()
Permite conectarse a la base de datos con "conexion".
- commit()
Hace un commit con "conexion" a la base de datos.
- desconectar()
Realiza la desconexión de la base de datos para "conexion"..
- rollback()
Hace rollback para "conexión".
- prepareStatement(String s)
Devuelve un preparedStatement para "conexión" según la entrada s.
- obfuscate(String pass)
Ofusca la cadena que recibe por parámetro.
- getProgresConexion()
- getId_app()
- getUsuario()
- getConexion()

Clase: AplicacionDAO

Enum: Claves

Atributos

- Conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- insertarAplicacion(Aplicacion app)
En la base de datos inserta la aplicación y el rol de administrador de dicha aplicación dándole además ese permiso al superusuario postgres.
- listarAplicaciones()
Lista toda la aplicaciones en las que el usuario actual es administrador.
- buscarAplicaciones(Clave clave, String valor)
Busca aplicaciones en la base de datos que tengan como administrador al usuario actual y además tengan el parámetro “valor” en el campo correspondiente a la “clave” que se recibe.

Clase: BitacoraDAO

Enum: ClavesUsuario

Enum: ClavesRolUsuario

Atributos

- conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- listarUsuarios()
Devuelve los registros de la bitácora de usuarios.
- listarRolesUsuarios()
Devuelve los registros de la bitácora de roles usuarios.
- buscarUsuarios(BitacoraDAO.ClavesUsuario clave, String valor)
Devuelve los registros de la bitácora de usuarios que tengan el parámetro valor en el campo correspondiente a la clave que se recibe.
- buscarUsuariosRoles(BitacoraDAO.ClavesRolUsuario clave, String valor)
Devuelve los registros de la bitácora de roles usuarios que tengan el parámetro “valor” en el campo correspondiente a la “clave” que se recibe.

Clase: DAOLogin

Enum: Estados

Atributos

- Conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- usr_login(String usuario,String contrasena)
Devuelve el estado del usuario que quiere loguearse al sistema para verificar si puede ingresar o no.
- getEstado()
Devuelve el estado del usuario que quiere loguearse al sistema.

Clase: FuncionalidadDAO

Enum: Claves

Atributos

- Conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- insertarFuncionalidad(Funcionalidad funcionalidad)
Inserta la funcionalidad en la base de datos.
- listarFuncionalidades()
Devuelve las funcionalidades de la base de datos que pertenecen a aplicaciones de las cuales el usuario actual es administrador.
- buscarFuncionalidades(Claves clave, String valor)
Devuelve las funcionalidades de la base de datos que pertenecen a aplicaciones de las cuales el usuario actual es administrador y que cumplan con tener el parámetro "valor" en el campo correspondiente a la "clave" que se recibe..

Clase : MenuDAO

Enum: Claves

Atributos

- Conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- insertarMenu(Menu menu)
Inserta el menu en la base de datos.
- agregarFuncionalidad(Menu menu, Funcionalidad funcionalidad)

Realiza una inserción en la tabla menu_funcionalidad de la base de datos para vincular el menu y la funcionalidad que se reciben por parámetro.

- listarMenus()
Devuelve los menús pertenecientes a las aplicaciones de las cuales el usuario actual es administrador.
- buscarMenus(MenuDAO.Claves clave, String valor)
Devuelve los menús pertenecientes a las aplicaciones de las cuales el usuario actual es administrador y además cumplen con tener el parámetro valor en el campo correspondiente a la clave que se recibe.
- buscarFuncionalidades(Menu menu)
Devuelve las funcionalidades pertenecientes a un menú según los datos registrados en la base de datos.

Clase : PersonaDAO

Enum: Claves

Atributos

- Conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- insertarPersona(Persona persona)
Inserta a la persona en la base de datos.
- listarPersonas()
Devuelve las personas que hay en la base de datos.
- buscarPersona(PersonaDAO.Claves clave, String valor)
Devuelve a las personas de la base de datos que tenga el parámetro valor en el campo correspondiente a la clave que se recibe..
- buscarUsuario(Persona persona)
Devuelve todos los usuarios registrados en la base de datos para una persona.

Clase : RolDAO

Enum: Claves

Atributos

- Conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- listarRoles()
Devuelve los roles que tiene el usuario en sus aplicaciones.
- buscarRol(RolDAO.Claves clave, String valor)

Devuelve los roles que tiene el usuario en sus aplicaciones y que también tengan el parámetro valor en el campo correspondiente a la clave recibida.

- insertarRol(Rol rol)
Inserta el rol recibido en la base de datos.
- agregarFuncionalidad(Rol rol, Funcionalidad funcionalidad)
Inserta el dato que vincula la funcionalidad con el rol que se reciben en la tabla rol_funcionalidad de la base de datos.
- buscarFuncionalidades(Rol rol)
Devuelve todas las funcionalidades correspondientes al rol recibido.
- listarUsuariosRolesPendientes()
Devuelve a todos los usuarios que tengan roles pendientes de aprobación.
- actualizarEstadoRol(UsuarioRol usuario_rol, String evento)
Actualiza el estado de la asignación del rol para el usuario y registra esa modificación en la bitácora de usuario rol.

Clase : UsuarioDAO

Enum: Claves

Atributos

- Conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- insertarUsuario(Usuario usuario, String pass)
Inserta un usuario con estado de aprobación pendiente a la base de datos, además registra en ella la trama de auditoría en la tabla de bitácora usuario.
- listarUsuarios()
Devuelve los usuarios que hay en la base de datos.
- buscarUsuarios(UsuarioDAO.Claves clave, String valor)
Devuelve los usuarios que hay en la base de datos que cumplan con tener el parámetro valor en el campo correspondiente a la clave recibida .
- actualizarEstadoUsuario(Usuario usuario)
Modifica el estado de confirmación del usuario y registra ese cambio en la bitácora de usuario.
- asignarUsuarioRol(Rol rol, Usuario usuario)
Inserta la asignación del rol recibido al usuario que se recibe con estado pendiente en la base de datos, además registra en ella la trama de auditoría en la tabla de bitácora usuario rol.
- buscarUsuariosRoles(Usuario usuario)
Devuelve los roles del usuario que se recibe que se encuentren en las aplicaciones de las cuales el usuario actual es administrador.
- listarUsuariosPendientes()
Devuelve los usuarios con estado de confirmación pendiente dentro de las aplicaciones que tienen al usuario actual como un administrador de app.

Clase : Aplicacion

Atributos

- Id_aplicacion
- nombre_aplicacion

Metodos

- Aplicacion(String id_aplicacion, String nombre_aplicacion)
- getId_aplicacion()
- getNombre_aplicacion()
- toString()

Clase : Bitacora

Atributos

- Id_evto
- usuario_modificador
- fecha

Metodos

- Bitacora()
- getId_evto()
- getUsuario_modificador()
- getFecha()

Clase : BitacoraRolUsuario

Atributos

- nombre_usuario
- id_rol
- id_aplicacion

Metodos

- BitacoraRolUsuario(String id_evto, String usuario_modificador, Timestamp fecha, String nombre_usuario, String id_rol, String id_aplicacion)
- getId_rol()
- getId_aplicacion()
- getNombre_usuario()

Clase : BitacoraUsuario

Atributos

- Nombre_usuario

Metodos

- BitacoraUsuario(String id_evto, String usuario_modificador, Timestamp fecha, String nombre_usuario)
- getNombre_usuario()

Clase : Funcionalidad

Atributos

- descripcion
- Id_aplicacion
- Nombre_funcionalidad

Metodos

- Funcionalidad(String nombre_funcionalidad, String id_aplicacion, String descripcion)
- toString()
- getDescripcion()
- getId_aplicacion()
- getNombre_funcionalidad()

Clase : Menu

Atributos

- id_menu
- nombre
- Id_aplicacion

Metodos

- Menu(String id_menu, String nombre, String id_aplicacion)
- getId_menu()
- getNombre()
- toString()
- getId_aplicacion()

Clase : Persona

Atributos

- CI
- nombre
- apellido
- fecha_nacimiento
- sexo
- Telefono

Metodos

- Persona(String CI, String nombre, String apellido, Date fecha_nacimiento, String sexo, String telefono)
- setTelefono(String telefono)
- getCI()
- getNombre()
- getApellido()
- getFecha_nacimiento()

- getSexo()
- getTelefono()
- toString()

Clase : Rol

Atributos

- Id_rol
- Id_aplicacion

Metodos

- Rol(String id_rol, String id_aplicacion)
- getId_rol()
- getId_aplicacion()
- toString()

Clase : Usuario

Atributos

- nombre_usuario
- codigo_estado
- CI

Metodos

- Usuario(String nombre_usuario, String codigo_estado, String CI)
- getNombre_usuario()
- getCodigo_estado()
- getCI()
- setCodigo_estado(String codigo_estado)
- toString()
-

Clase : UsuarioRol

Atributos

- id_rol
- id_aplicacion
- Nombre_usuario
- Cod_estado

Metodos

- UsuarioRol(String id_rol, String id_aplicacion, String nombre_usuario, String cod_estado)
- setCod_estado(String cod_estado)
- getId_rol()
- getId_aplicacion()

- getNombre_usuario()
- getCod_estado()
- toString()

Estructura de la implementación Aplicación Genérica

Atributos

- Instance
Es la conexión a la base de datos.
- BDnombre
Es el nombre de la base de datos.
- Usuario
Es el nombre de acceso a la base de datos del usuario.
- contraseña
Es la contraseña de usuario de acceso a la base de datos.
- Id_app
Es el identificador de la aplicación del sistema
- Host
Es la dirección IPv4 de la conexión a la base de datos.
- puerto
Es el puerto de la conexión a la base de datos
- url
Dirección conexión a la base de datos
- Conexion
Conexión a la base de datos.

Metodos

- PostgresSqlConexion()
Constructor con singleton para el atributo "conexion".
- setUserAndPass(String usuario, String pass)
Establece el usuario y la contraseña de "conexion".
- conectarse()
Permite conectarse a la base de datos con "conexion".
- commit()
Hace un commit con "conexion" a la base de datos.
- desconectar()
Realiza la desconexión de la base de datos para "conexion"..
- rollback()
Hace rollback para "conexión".
- prepareStatement(String s)
Devuelve un preparedStatement para "conexión" según la entrada s.
- obfuscate(String pass)
Ofusca la cadena que recibe por parámetro.

- getProgresConexion()
- getId_app()
- getUsuario()
- getConexion()

Clase: DAOLogin

Enum: Estados

Atributos

- Conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- usr_login(String usuario,String contrasena)
Devuelve el estado del usuario que quiere loguearse al sistema para verificar si puede ingresar o no.
- getEstado()
Devuelve el estado del usuario que quiere loguearse al sistema.

Clase: DAOMenu

Enum: Estados

Atributos

- Conexion
Se utiliza esta conexión para interactuar con la base de datos.

Metodos

- getMenu()
Devuelve un map con los nombres de los menús y las funcionalidades correspondientes a las cuales tiene acceso el usuario actual .

Casos de uso

| | |
|---|--|
| CU-001 – Crear Usuario | |
| Descripción | Se crea un usuario nuevo |
| Actores | Administrador |
| Pre-condiciones | El administrador está logueado |
| Post-condiciones | El usuario es creado en la base de datos y queda pendiente de aprobacion |
| Curso Básico | |
| Administrador(cliente) | Sistema |
| 1 - Ingresa a el menu usuarios en la pestaña Insertar | |
| | 2 – Despliega el formulario de creacion de usuario |
| 3 - Selecciona la persona a la que se asocia el usuario, y inserta usuario y contraseña | |
| | 4 - El usuario se inserta en la base de datos |
| | 5 - Se registra en la bitacora la creacion del usuario. |
| Cursos Alternativos | |
| | 4.1 El usuario no pudo insertarse |
| | 4.1.1 Se despliega mensaje de error |
| | 4.1.2 Se vuelve al paso 3 |

| | |
|--|--|
| CU-001 – Crear Usuario | |
| Descripción | Se aprueba un usuario nuevo |
| Actores | Administrador |
| Pre-condiciones | El administrador está logueado. El usuario esta pendiente de aprobación. |
| Post-condiciones | El usuario es aprobado. |
| Curso Básico | |
| Administrador(cliente) | Sistema |
| 1 - Ingresa a el menu usuarios en la pestaña Aprobar | |
| | 2 – Despliega el combobox de usuarios pendientes de aprobacion |
| 3 - Selecciona el usuario y da aprobar | |
| | 4 - El usuario es aprobado |
| | 5 - Se registra en la bitacora la aprobación del usuario. |
| Cursos Alternativos | |
| | 3.1 No se encuentra el usuario. |
| | 3.2 fin del caso de uso. |

*El comportamiento del 4 ojos y el guardado en bitácora funciona de forma equivalente en la asignación de rol a un usuario.