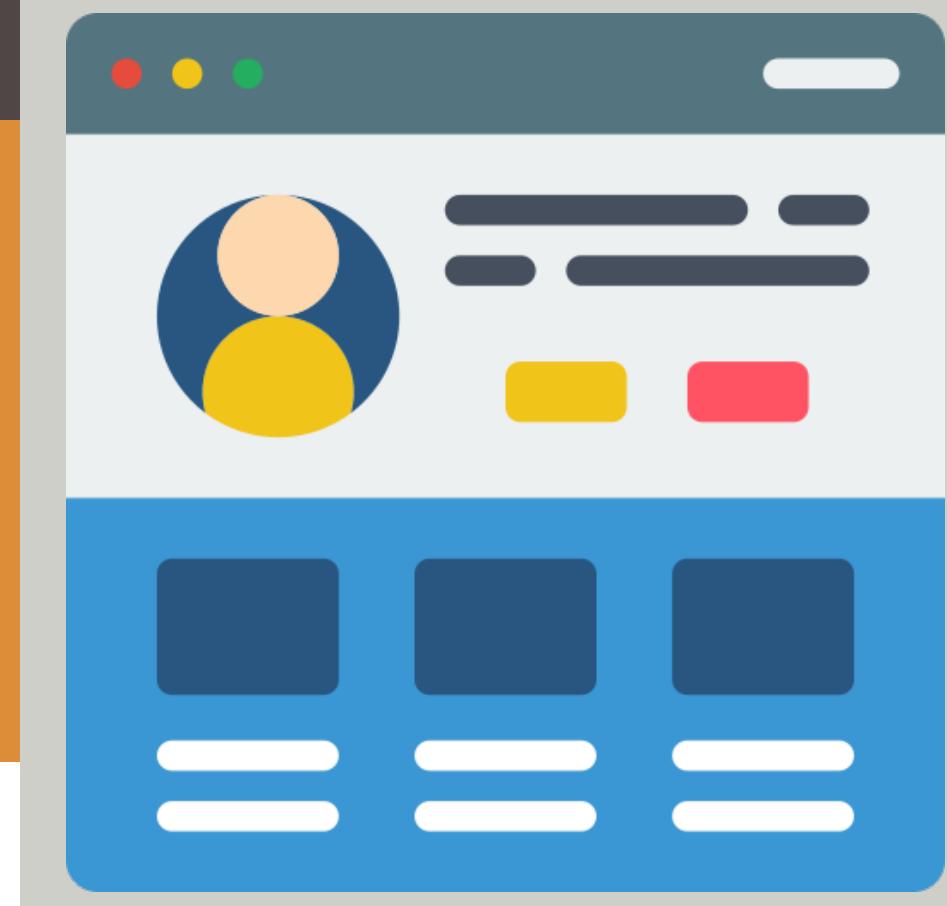




EQUIPO



INTEGRANTES:

- Camacho López Saúl
- Medina Maximiano Giselle
- Rodríguez Gallardo Raúl
- Rojas Esteban Arely Paulina

Proyecto INTEGRADOR

Universidad Politécnica de Pachuca
Ingeniería en Software
Programación Visual
Tercer Parcial

Roles

- Saúl Camacho López
- Giselle Medina Maximiano
- Raúl Rodríguez Gallardo
- Arely Paulina Rojas Esteban

Usuarios

Usuario	Clave	Cargo
ArelyPau	1234	Administrador
Ruliz07	Tacosdecanasta	Administrador
gmedinax	swimming	Administrador
Sul432	LORD432	Administrador
Alan	1	Empleado
Jona	4	Empleado
Diego	1234	Empleado

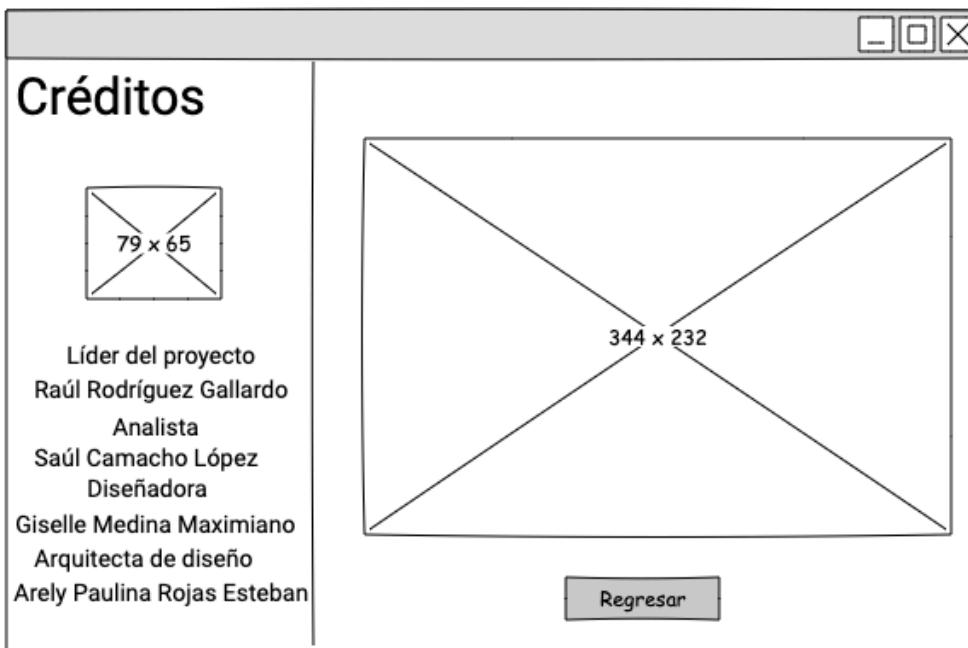
Nota: En el código de la conexión se puso como clave del root "1234".

Diseño del sistema

Sketches

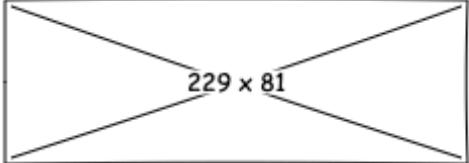


A wireframe sketch of a window titled "Iniciar Sesión". The window has standard OS X-style controls at the top right. On the left side, there is a large rectangular area labeled "Label (imagen)". On the right side, the title "Iniciar Sesión" is at the top, followed by the label "USUARIO" and a text input field. Below that is the label "CONTRASEÑA" with a text input field containing asterisks. There is also a checkbox labeled "Recuérdame" and a "Ingresar" button at the bottom.



A wireframe sketch of a window titled "Créditos". The window has standard OS X-style controls at the top right. On the left side, there is a small square icon labeled "79 x 65" and a list of credits: "Líder del proyecto Raúl Rodríguez Gallardo", "Analista Saúl Camacho López", "Diseñadora Giselle Medina Maximiano", "Arquitecta de diseño Arely Paulina Rojas Esteban". On the right side, there is a large rectangular area with diagonal lines and dimensions "344 x 232". At the bottom right is a "Regresar" button.

Formulario de Registro de Usuario

	 229 x 81
Usuario	ID Usuario: <input type="text"/> Cargo: <input type="text"/> ▾
Usuario	Nickname: <input type="text"/>
Canciones	Contraseña: <input type="text"/>
Cantantes	Nombre: <input type="text"/>
Álbums	<input type="file"/> Imagen
	Guardar

Formulario de Consulta y Actualización de Usuarios

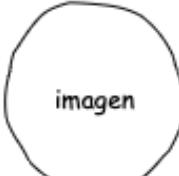
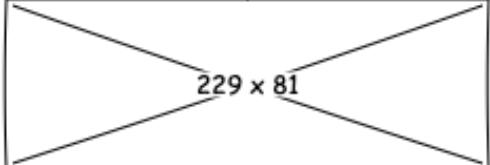
	 229 x 81																				
Usuario	<input type="text"/> Buscar Guardar Actualizar Borrar																				
Usuario	<input type="file"/> Imagen																				
Canciones																					
Cantantes																					
Álbums																					
	<table border="1"><thead><tr><th>idUsuario</th><th>Nickname</th><th>Nombre</th><th>Cargo</th></tr></thead><tbody><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></tbody></table>	idUsuario	Nickname	Nombre	Cargo																
idUsuario	Nickname	Nombre	Cargo																		

Diagram of a user interface window:

The window has a title bar with standard minimize, maximize, and close buttons. The left sidebar contains a circular profile picture labeled "Imagen" and four rectangular buttons labeled "Usuario", "Canciones", "Cantantes", and "Álbums". The main panel features a large empty rectangular area at the top with a size indicator of "229 x 81". Below this is a horizontal row of four buttons: "Buscar", "Guardar", "Actualizar", and "Borrar". A table with five columns ("idCanción", "Nombre", "Álbum", "Duración", "Género") and four rows follows.

idCanción	Nombre	Álbum	Duración	Género

Diagram of a user interface window for adding or editing a song:

The window has a title bar with standard minimize, maximize, and close buttons. The left sidebar contains a circular profile picture labeled "Imagen" and four rectangular buttons labeled "Usuario", "Canciones", "Cantantes", and "Álbums". The main panel features a large empty rectangular area with a size indicator of "229 x 81". Below this are several input fields: "ID Canción:" with an empty input box, "Nombre:" with an empty input box, "Álbum:" with an empty input box and a dropdown arrow, "Duración:" with an empty input box, and "Género:" with an empty input box and a dropdown arrow. A "Guardar" button is located at the bottom right.

Diagrama de interfaz de usuario para la visualización de datos:

Panel izquierdo (Menú):

- Imagen
- Usuario
- Usuario
- Canciones
- Cantantes
- Álbums

Panel central:

Imagen de tamaño 229 x 81.

Barras de herramientas:

- Buscar
- Guardar
- Actualizar
- Borrar

Tabla de datos:

idCantante	Nombre artístico	Nombre real	Teléfono	Dirección

Diagrama de interfaz de usuario para la inserción de datos:

Panel izquierdo (Menú):

- Imagen
- Usuario
- Usuario
- Canciones
- Cantantes
- Álbums

Panel central:

Imagen de tamaño 229 x 81.

Campos de texto:

- ID Cantante: _____
- Teléfono: _____
- Nombre artístico: _____
- Nombre real: _____
- Dirección: _____

Boton:

- Guardar

Window title bar with close, minimize, and maximize buttons.

Left sidebar:

- Imagen
- Usuario
- Canciones
- Cantantes
- Álbums

Main content area:

Placeholder image area labeled "229 x 81".

Search and action buttons: Buscar, Guardar, Actualizar, Borrar.

idAlbum	Nombre	Género	Duración	Fecha	Productor	Cantante

Window title bar with close, minimize, and maximize buttons.

Left sidebar:

- Imagen
- Usuario
- Canciones
- Cantantes
- Álbums

Main content area:

Placeholder image area labeled "229 x 81".

Form fields for creating a new album:

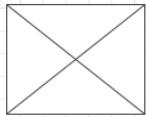
ID Álbum:	<input type="text"/>	Fecha de grabación:	<input type="text"/> ▾
Nombre:	<input type="text"/>		
Cantante:	<input type="text"/> ▾		
Productor:	<input type="text"/>		
Duración:	<input type="text"/>	Género:	<input type="text"/> ▾

Save button: Guardar

Wireframes

A wireframe of a login page. On the left, there is a large square placeholder with a large 'X' drawn through it, indicating where an image or logo would be. On the right, the page title 'Iniciar Sesión' is at the top. Below it are two input fields: one for 'CORREO ELECTRÓNICO' and another for 'CONTRASEÑA', both with placeholder text 'I'. There is also a checked checkbox labeled 'Recuérdame'. At the bottom is a rounded rectangular button labeled 'Ingresar'.

A wireframe of a credits page. On the left, the word 'Créditos' is at the top, above a small square placeholder with a large 'X'. Below it is a list of team members: 'Líder del proyecto Raúl Rodríguez Gallardo', 'Analista Saúl Camacho López', 'Diseñadora Giselle Medina Maximiano', and 'Arquitecta de diseño Arely Paulina Rojas Estaban'. On the right, there is a large square placeholder with a large 'X' drawn through it. At the bottom is a rounded rectangular button labeled 'Regresar'.



Usuario

idUsuario	Nickname	Nombre	Cargo



Usuario

ID Usuario: Cargo:

Nickname:

Contraseña:

Nombre:

Wireframe of a user management application:

- Left Sidebar:** Contains a user icon (square with an X) and four rounded rectangular buttons labeled "Usuarios", "Canciones", "Cantantes", and "Álbums".
- Right Main Area:** Features a search bar with placeholder text "I", and four buttons: "Buscar", "Guardar", "Actualizar", and "Borrar". Below the search area is a table with columns: "idCancion", "Nombre", "Álbum", "Duración", and "Género". The table has 6 rows.

Detailed wireframe of a song creation form:

- Left Sidebar:** Contains a user icon (square with an X) and four rounded rectangular buttons labeled "Usuarios", "Canciones", "Cantantes", and "Álbums".
- Right Main Area:** Contains the following fields:
 - ID Canción:
 - Nombre:
 - Álbum:
 - Duración: Género:
- Bottom Center:** A large rounded rectangular button labeled "Guardar".

A screenshot of a mobile application interface. On the left, there is a sidebar with a user icon and five rounded rectangular buttons labeled "Usuarios", "Canciones", "Cantantes", and "Álbums". The main area features a large rectangular box with a diagonal cross. Below it is a search bar with an input field and three buttons: "Buscar", "Guardar", and "Actualizar". A "Borrar" button is located to the right of the search bar. Underneath the search bar is a table with columns labeled "idCantante", "Nombre artístico", "Nombre real", "Teléfono", and "Dirección". There are several rows of data in the table, with the last row being partially visible.

A screenshot of a mobile application interface. On the left, there is a sidebar with a user icon and five rounded rectangular buttons labeled "Usuarios", "Canciones", "Cantantes", and "Álbums". The main area features a large rectangular box with a diagonal cross. Below it is a form with five input fields. The first two fields are labeled "ID Cantante:" and "Teléfono:", each with an input field. The next three fields are labeled "Nombre artístico:", "Nombre real:", and "Dirección:", each with an input field. At the bottom right of the form is a rounded rectangular button labeled "Guardar".

Usuario

Usuarios

Canciones

Cantantes

Álbums

idAlbum	Nombre	Género	Duración	Fecha	Productor	Cantante

Usuario

Usuarios

Canciones

Cantantes

Álbums

ID Álbum:

Nombre:

Cantante:

Productor:

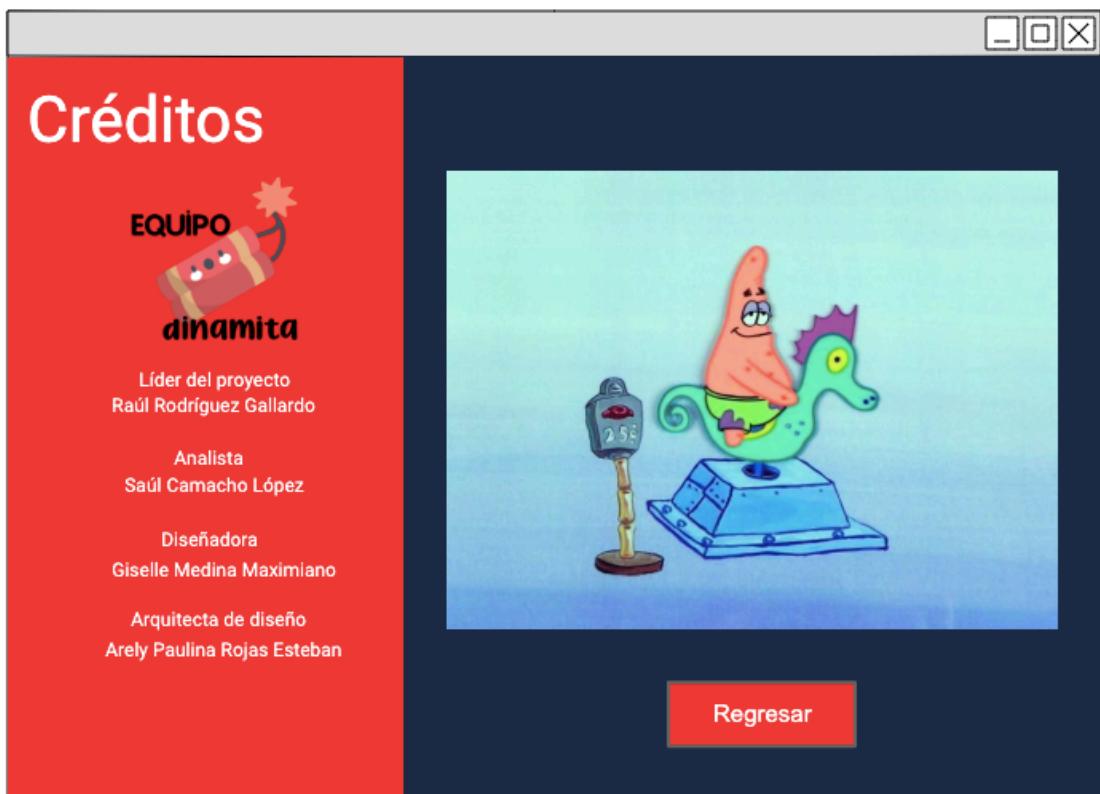
Duración:

Fecha de grabación:

Género:

Guardar

Mock-Ups



USUARIOS

Usuario

Usuarios
Canciones
Cantantes
Álbums

<i>idUsuario</i>	Nickname	Nombre	Cargo

REGISTRO USUARIO

Usuario

Usuarios
Canciones
Cantantes
Álbums

ID Usuario: Cargo:

Nickname:

Contraseña:

Nombre: Imagen

Guardar

CANCIONES

idCanción	Nombre	Álbum	Duración	Género

REGISTRO CANCIÓN

ID Canción:

Nombre:

Álbum:

Duración: Género:

Guardar

This screenshot shows a list of artists. The interface has a red sidebar on the left containing user-related links: 'Usuario', 'Usuarios', 'Canciones', 'Cantantes', and 'Álbums'. The main area is titled 'ARTISTAS' and contains a search bar, a magnifying glass icon, a floppy disk icon for saving, a circular arrow icon for refreshing, and a trash can icon for deleting. Below these icons is a table with columns: 'idCantante', 'Nombre artístico', 'Nombre real', 'Teléfono', and 'Dirección'. There are five empty rows in the table.

idCantante	Nombre artístico	Nombre real	Teléfono	Dirección

This screenshot shows a form for registering a new artist. The left sidebar is identical to the first screenshot, with links for 'Usuario', 'Usuarios', 'Canciones', 'Cantantes', and 'Álbums'. The main area is titled 'REGISTRO ARTISTA' and contains fields for inputting information: 'ID Cantante' (with a corresponding input field), 'Teléfono' (with a corresponding input field), 'Nombre Artístico' (with a corresponding input field), 'Nombre Real' (with a corresponding input field), and 'Dirección' (with a corresponding input field). A large red 'Guardar' (Save) button is located at the bottom right of the form area.

ÁLBUMS

User icon

Usuario

- Usuarios
- Canciones
- Cantantes
- Álbums

idAlbum	Nombre	Género	Duración	Fecha	Productor	Cantante

REGISTRO ÁLBUM

User icon

Usuario

- Usuarios
- Canciones
- Cantantes
- Álbums

ID Álbum:

Fecha de grabación:

Nombre:

Cantante:

Productor:

Duración: Género:

Guardar

Diagrams

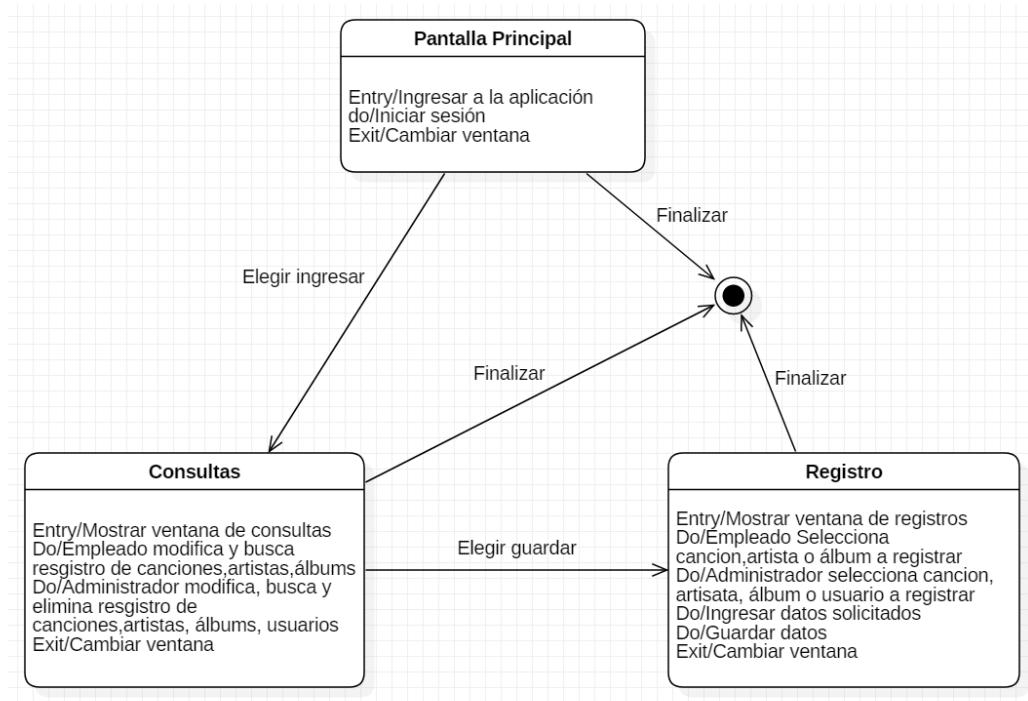
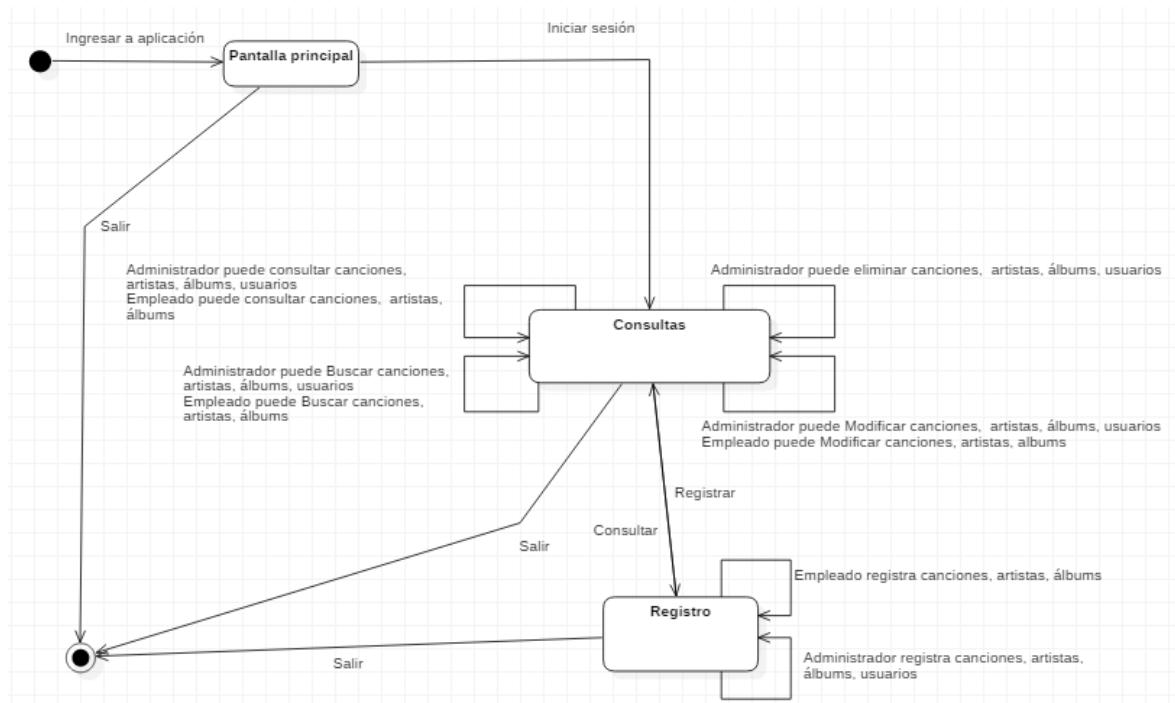
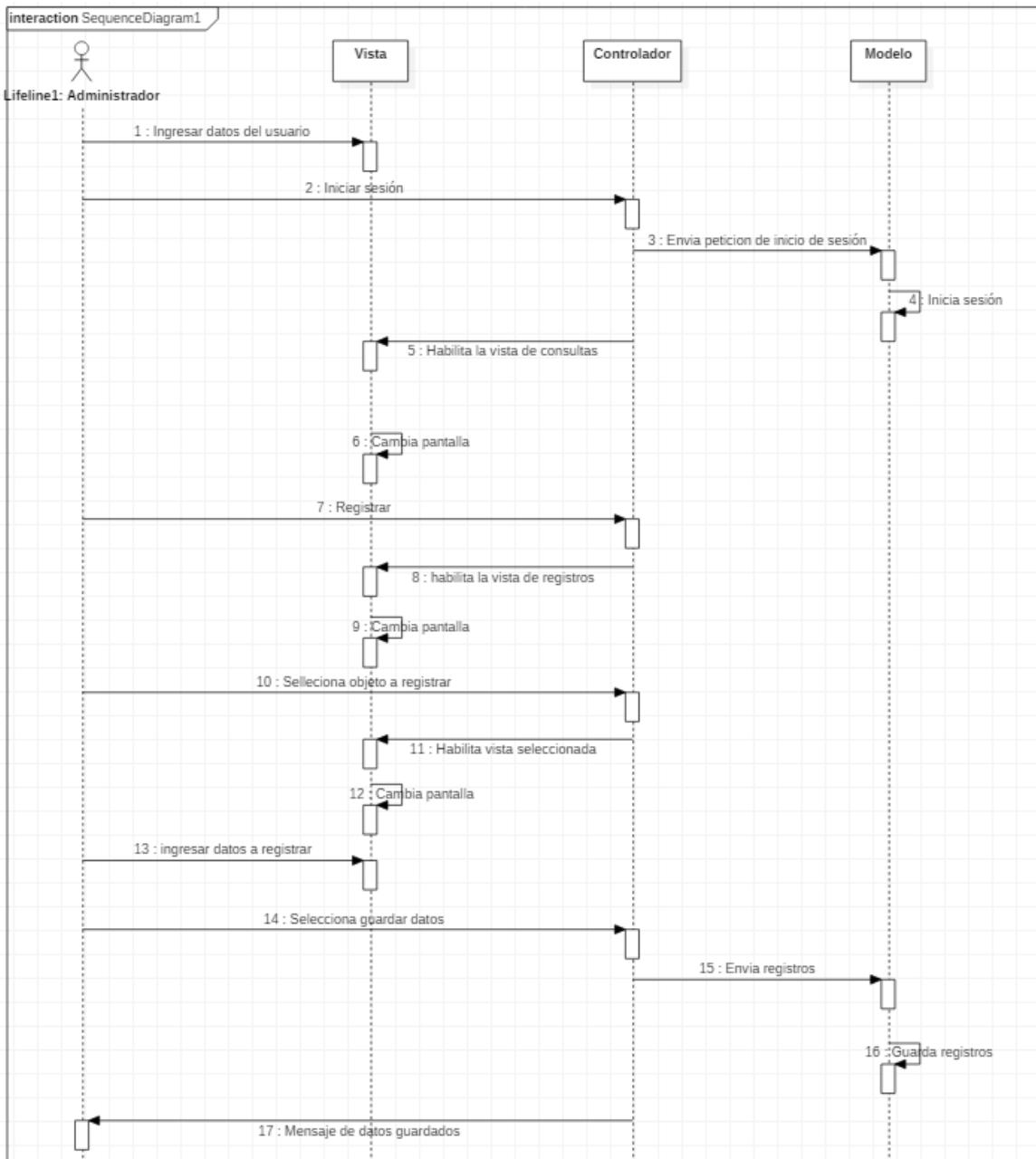
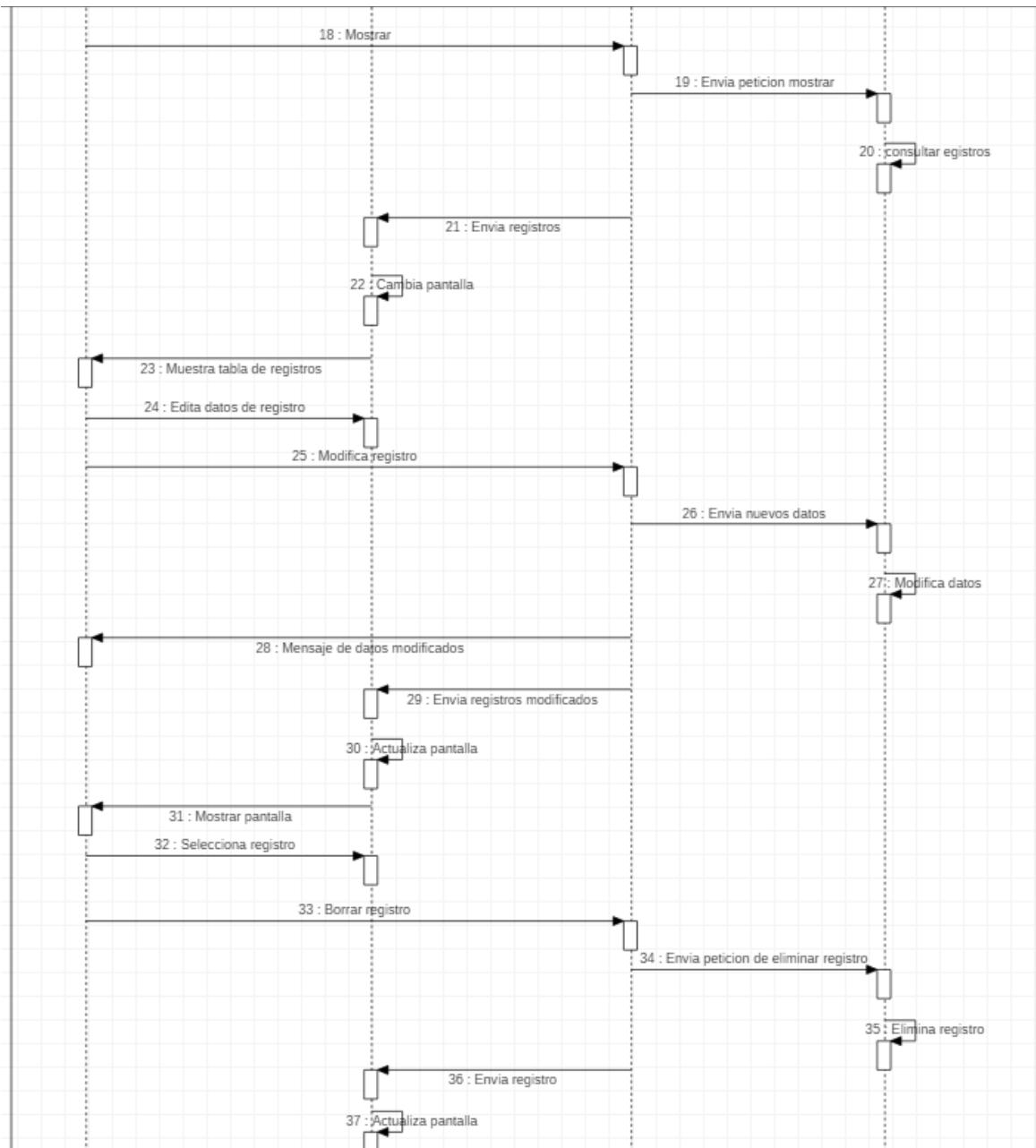
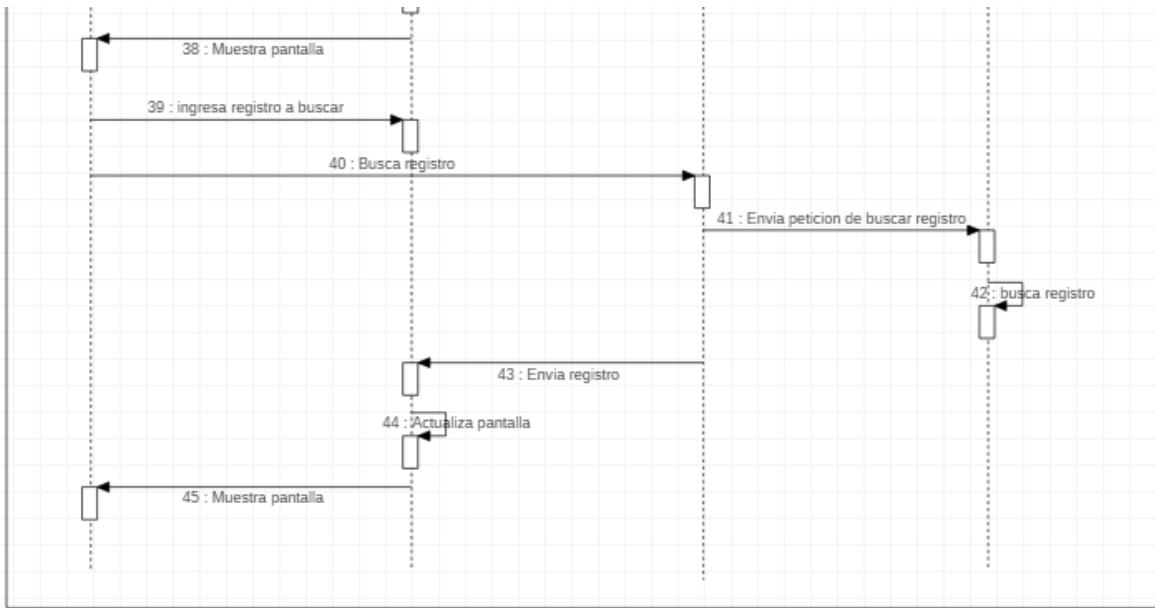


Diagrama de secuencia

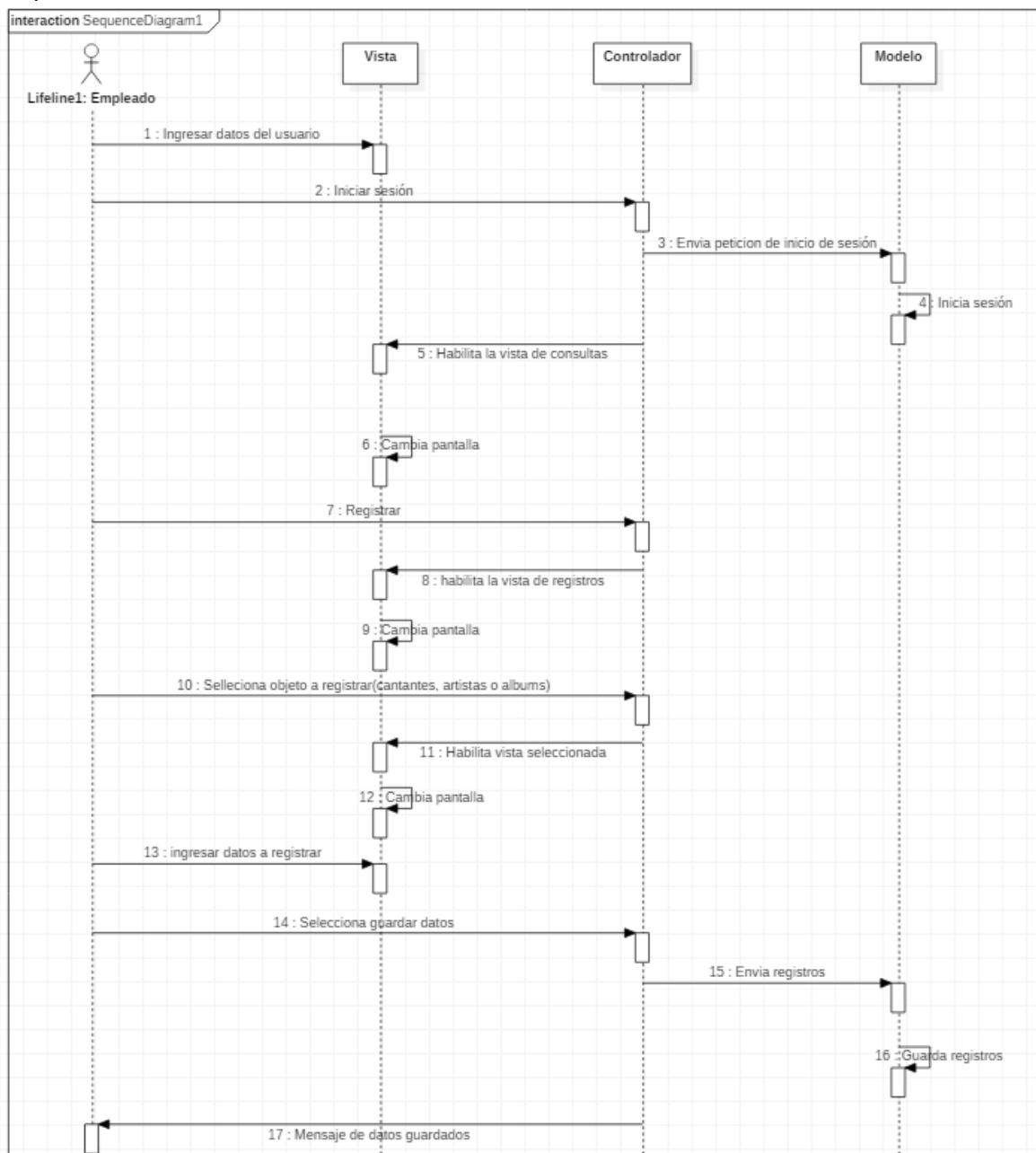
Administrador

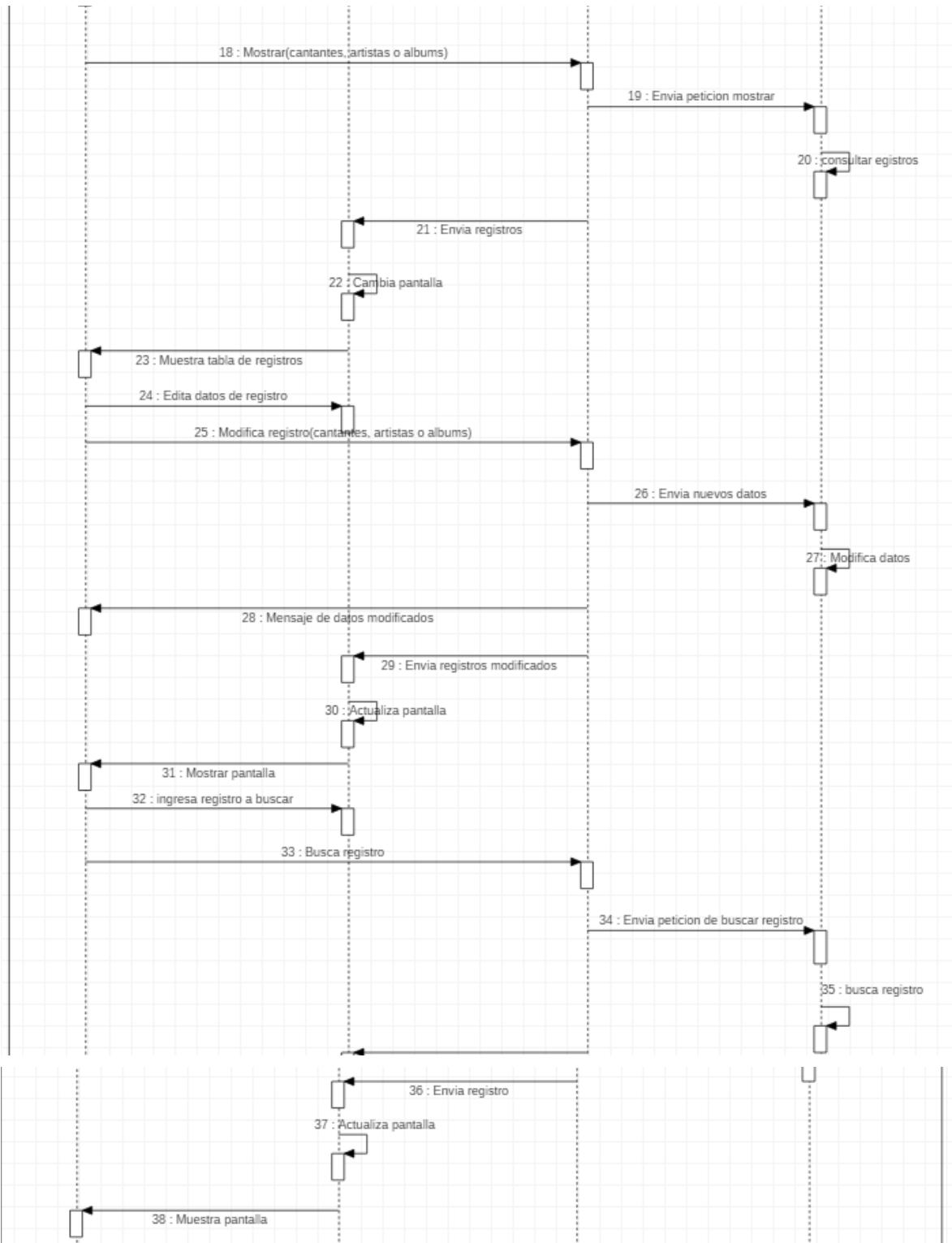






Empleado





Caso de uso

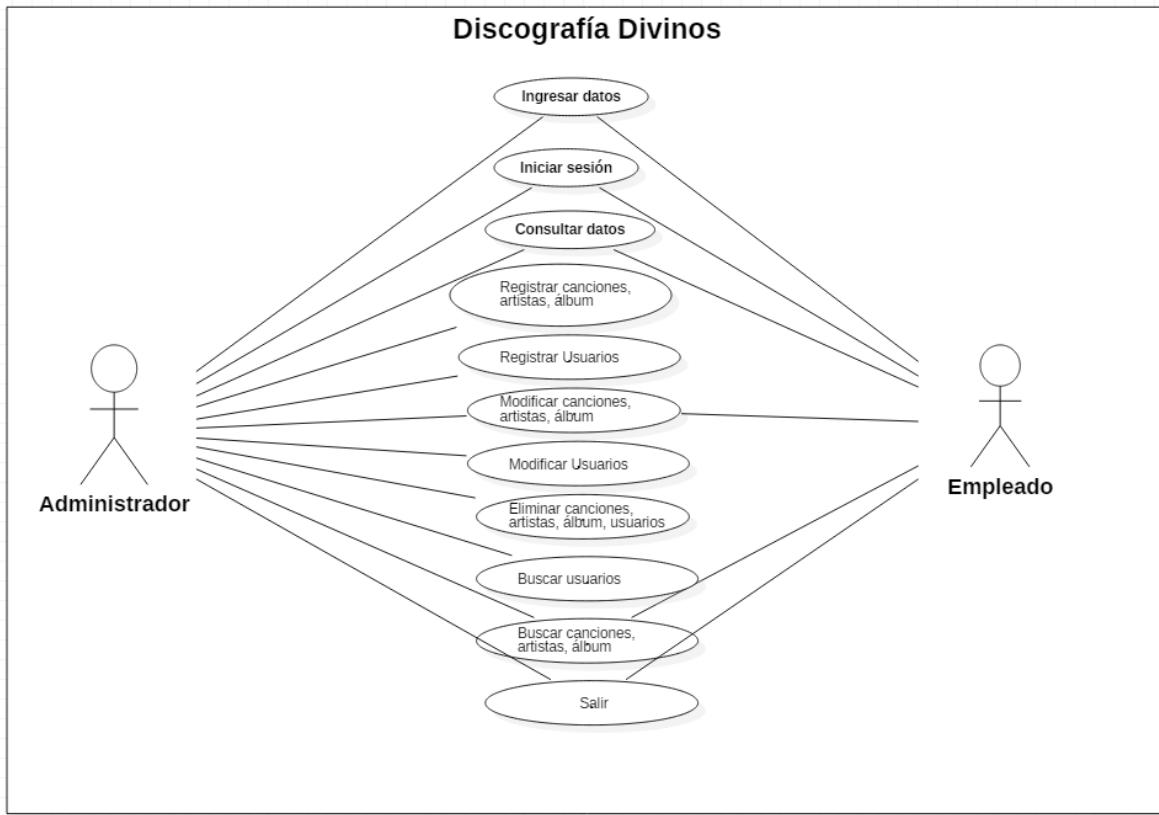
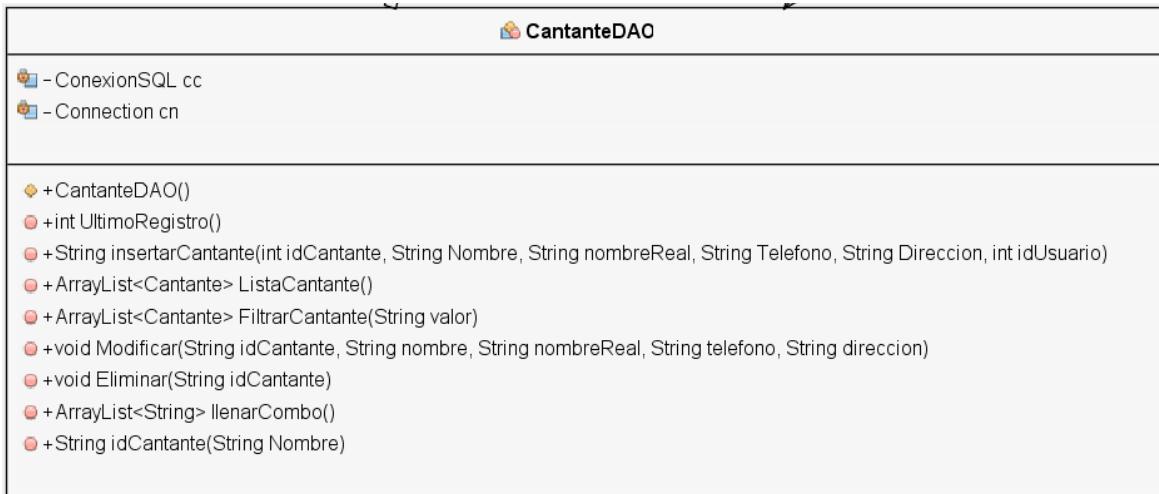
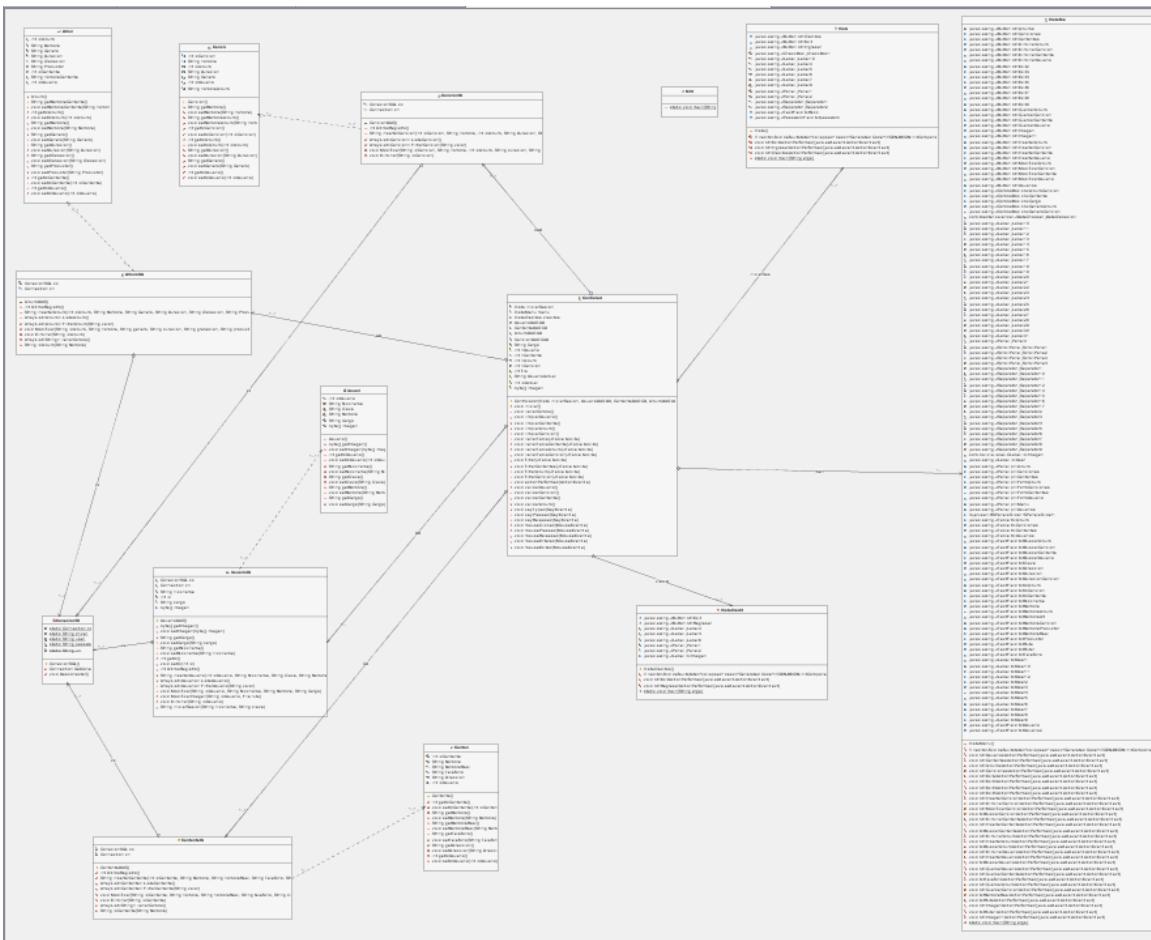


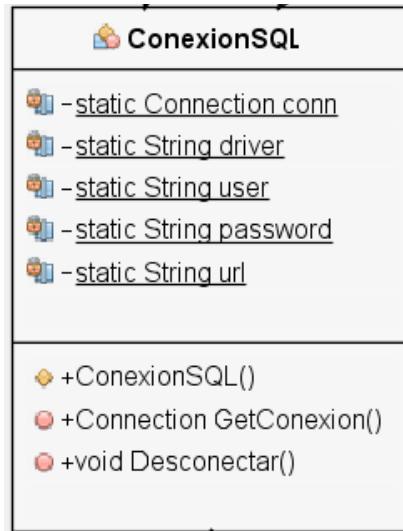
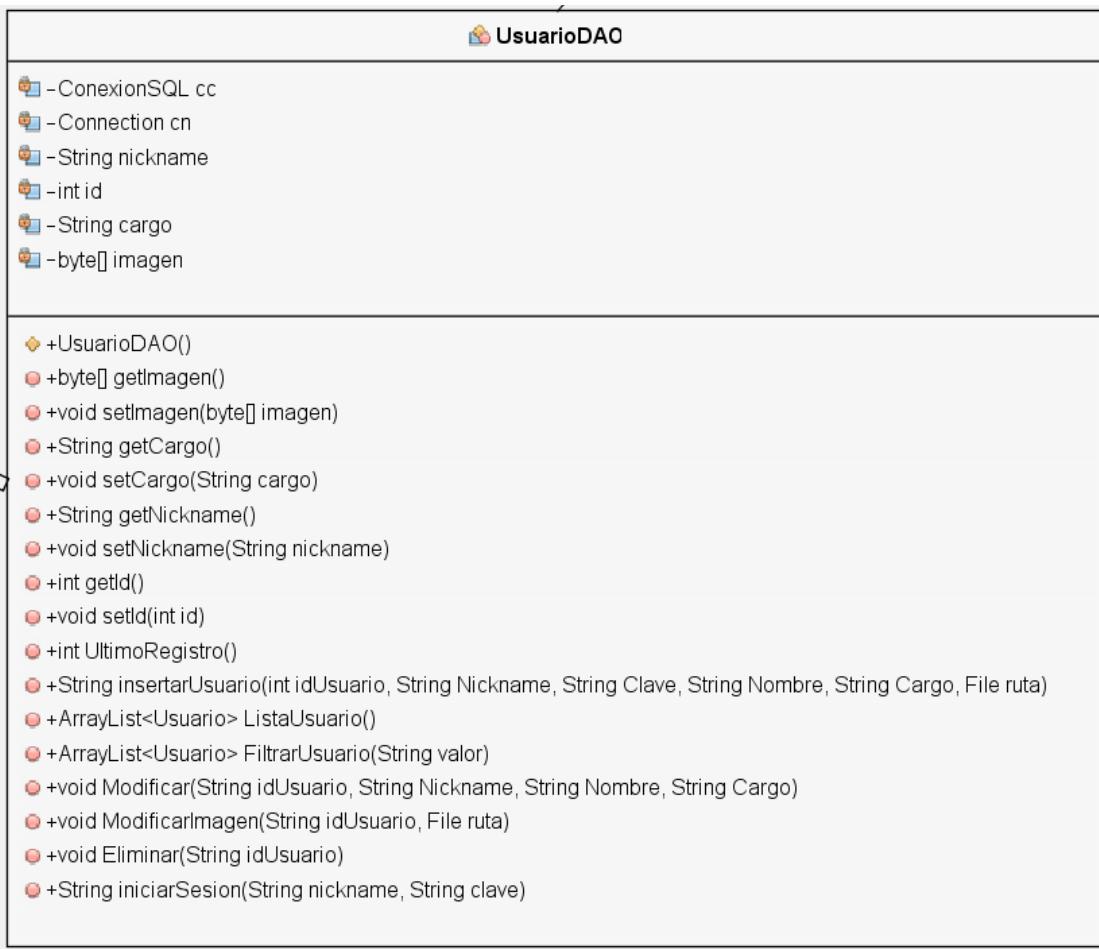
Diagrama de clases



<p>Album</p> <ul style="list-style-type: none"> ⌚ -int idAlbum ⌚ -String Nombre ⌚ -String Genero ⌚ -String Duracion ⌚ -String Grabacion ⌚ -String Productor ⌚ -int idCantante ⌚ -String nombreCantante ⌚ -int idUsuario <ul style="list-style-type: none"> ❖ +Album() ❖ +String getNombreCantante() ❖ +void setNombreCantante(String nombreCantante) ❖ +int getIdAlbum() ❖ +void setIdAlbum(int idAlbum) ❖ +String getNombre() ❖ +void setNombre(String Nombre) ❖ +String getGenero() ❖ +void setGenero(String Genero) ❖ +String getDuracion() ❖ +void setDuracion(String Duracion) ❖ +String getGrabacion() ❖ +void setGrabacion(String Grabacion) ❖ +String getProductor() ❖ +void setProductor(String Productor) ❖ +int getIdCantante() ❖ +void setIdCantante(int idCantante) ❖ +int getIdUsuario() ❖ +void setIdUsuario(int idUsuario) 	<p>Cancion</p> <ul style="list-style-type: none"> ⌚ -int idCancion ⌚ -String nombre ⌚ -int idAlbum ⌚ -String Duracion ⌚ -String Genero ⌚ -int idUsuario ⌚ -String nombreAlbum <ul style="list-style-type: none"> ❖ +Cancion() ❖ +String getNombre() ❖ +void setNombre(String nombre) ❖ +String getNombreAlbum() ❖ +void setNombreAlbum(String nombreAlbum) ❖ +int getIdCancion() ❖ +void setIdCancion(int idCancion) ❖ +int getIdAlbum() ❖ +void setIdAlbum(int idAlbum) ❖ +String getDuracion() ❖ +void setDuracion(String Duracion) ❖ +String getGenero() ❖ +void setGenero(String Genero) ❖ +int getIdUsuario() ❖ +void setIdUsuario(int idUsuario)
<p>Usuario</p> <ul style="list-style-type: none"> ⌚ -int idUsuario ⌚ -String Nickname ⌚ -String Clave ⌚ -String Nombre ⌚ -String Cargo ⌚ -byte[] imagen <ul style="list-style-type: none"> ❖ +Usuario() ❖ +byte[] getImagen() ❖ +void setImagen(byte[] imagen) ❖ +int getIdUsuario() ❖ +void setIdUsuario(int idUsuario) ❖ +String getNickname() ❖ +void setNickname(String Nickname) ❖ +String getClave() ❖ +void setClave(String Clave) ❖ +String getNombre() ❖ +void setNombre(String Nombre) ❖ +String getCargo() ❖ +void setCargo(String Cargo) 	<p>Cantante</p> <ul style="list-style-type: none"> ⌚ -int idCantante ⌚ -String Nombre ⌚ -String NombreReal ⌚ -String Telefono ⌚ -String Direccion ⌚ -int idUsuario <ul style="list-style-type: none"> ❖ +Cantante() ❖ +int getIdCantante() ❖ +void setIdCantante(int idCantante) ❖ +String getNombre() ❖ +void setNombre(String Nombre) ❖ +String getNombreReal() ❖ +void setNombreReal(String NombreReal) ❖ +String getTelefono() ❖ +void setTelefono(String Telefono) ❖ +String getDireccion() ❖ +void setDireccion(String Direccion) ❖ +int getIdUsuario() ❖ +void setIdUsuario(int idUsuario)

 AlbumDAC	
 -ConexionSQL cc	
 -Connection cn	
	 +AlbumDAO()
	 +int UltimoRegistro()
	 +String insertarAlbum(int idAlbum, String Nombre, String Genero, String Duracion, String Grabacion, String Productor, int idCantante, int idUsuario)
	 +ArrayList<Album> ListaAlbum()
	 +ArrayList<Album> FiltrarAlbum(String valor)
	 +void Modificar(String idAlbum, String nombre, String genero, String duracion, String grabacion, String productor, int idCantante)
	 +void Eliminar(String idAlbum)
	 +ArrayList<String> llenarCombo()
	 +String idAlbum(String Nombre)

 CancionDAC	
 -ConexionSQL cc	
 -Connection cn	
	 +CancionDAO()
	 +int UltimoRegistro()
	 +String insertarCancion(int idCancion, String nombre, int idAlbum, String Duracion, String Genero, int idUsuario)
	 +ArrayList<Cancion> ListaCancion()
	 +ArrayList<Cancion> FiltrarCancion(String valor)
	 +void Modificar(String idCancion, String nombre, int idAlbum, String duracion, String genero)
	 +void Eliminar(String idCancion)



 Vista

```

+javax.swing.JButton btnCreditos
+javax.swing.JButton btnExit
+javax.swing.JButton btnIngresar
-javax.swing.JCheckBox jCheckBox1
-javax.swing.JLabel jLabel10
-javax.swing.JLabel jLabel3
-javax.swing.JLabel jLabel5
-javax.swing.JLabel jLabel6
-javax.swing.JLabel jLabel7
-javax.swing.JLabel jLabel9
-javax.swing.JPanel jPanel1
-javax.swing.JPanel jPanel2
-javax.swing.JSeparator jSeparator1
-javax.swing.JSeparator jSeparator2
+javax.swing.JTextField txtNick
+javax.swing.JPasswordField txtpassword

```

```

+Vista()
// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents void initComponents()
void btnExitActionPerformed(java.awt.event.ActionEvent evt)
void btnIngresarActionPerformed(java.awt.event.ActionEvent evt)
void btnCreditosActionPerformed(java.awt.event.ActionEvent evt)
+static void main(String args)

```

 VistaCreditos

```

+javax.swing.JButton btnExit
+javax.swing.JButton btnRegresar
-javax.swing.JLabel jLabel3
-javax.swing.JLabel jLabel4
-javax.swing.JLabel jLabel6
-javax.swing.JPanel jPanel1
-javax.swing.JPanel jPanel2
-javax.swing.JLabel lblImagen

```

```

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

```

```
Controlador
+Vista iniciarSesion
+VistaMenu menu
+VistaCreditos creditos
+UsuarioDAO UD
+CantanteDAO CD
+AlbumDAO AD
+CancionDAO CAD
+String Cargo
+int nUsuario
+int nCantante
+int nAlbum
+int nCancion
+int fila
+String UsuarioActual
+int idActual
+byte[] imagen

+Controlador(Vista iniciarSesion, UsuarioDAO DU, CantanteDAO CD, AlbumDAO AD, CancionDAO C)
+void iniciar()
+void llenarCombo()
+void limpiarUsuario()
+void limpiarCantante()
+void limpiarAlbum()
+void limpiarCancion()
+void llenarTabla(JTable tablita)
+void llenarTablaCantante(JTable tablita)
+void llenarTablaAlbum(JTable tablita)
+void llenarTablaCancion(JTable tablita)
+void filtrar(JTable tablita)
+void filtrarCantantes(JTable tablita)
+void filtrarAlbum(JTable tablita)
+void filtrarCancion(JTable tablita)
+void actionPerformed(ActionEvent e)
+void validarUsuario()
+void validarCancion()
+void validarCantante()
+void validarAlbum()
+void keyTyped(KeyEvent e)
+void keyPressed(KeyEvent e)
+void keyReleased(KeyEvent e)
+void mouseClicked(MouseEvent e)
+void mousePressed(MouseEvent e)
+void mouseReleased(MouseEvent e)
+void mouseEntered(MouseEvent e)
+void mouseExited(MouseEvent e)
```

 VistaMenu

- +javax.swing.JButton btnAlbums
- +javax.swing.JButton btnCanciones
- +javax.swing.JButton btnCantantes
- +javax.swing.JButton btnEliminarAlbum
- +javax.swing.JButton btnEliminarCancion
- +javax.swing.JButton btnEliminarCantante
- +javax.swing.JButton btnEliminarUsuario
- +javax.swing.JButton btnExit2
- +javax.swing.JButton btnExit3
- +javax.swing.JButton btnExit4
- +javax.swing.JButton btnExit5
- +javax.swing.JButton btnExit6
- +javax.swing.JButton btnExit7
- +javax.swing.JButton btnExit8
- +javax.swing.JButton btnExit9
- +javax.swing.JButton btnGuardarAlbum
- +javax.swing.JButton btnGuardarCancion
- +javax.swing.JButton btnGuardarCantante
- +javax.swing.JButton btnGuardarUsuario
- +javax.swing.JButton btnImagen
- +javax.swing.JButton btnImagen1
- +javax.swing.JButton btnInsertarAlbum
- +javax.swing.JButton btnInsertarCancion
- +javax.swing.JButton btnInsertarCantante
- +javax.swing.JButton btnInsertarUsuario
- +javax.swing.JButton btnModificarAlbum
- +javax.swing.JButton btnModificarCancion
- +javax.swing.JButton btnModificarCantante
- +javax.swing.JButton btnModificarUsuario
- +javax.swing.JButton btnUsuarios
- +javax.swing.JComboBox cbxAlbumCancion
- +javax.swing.JComboBox cbxCantante
- +javax.swing.JComboBox cbxCargo
- +javax.swing.JComboBox cbxGeneroAlbum
- +javax.swing.JComboBox cbxGeneroCancion
- +com.toedter.calendar.JDate Chooser jDate Grabacion
- javax.swing.JLabel jLabel10
- javax.swing.JLabel jLabel11
- javax.swing.JLabel jLabel12
- javax.swing.JLabel jLabel13
- javax.swing.JLabel jLabel14
- javax.swing.JLabel jLabel15
- javax.swing.JLabel jLabel16
- javax.swing.JLabel jLabel17
- javax.swing.JLabel jLabel18
- javax.swing.JLabel jLabel19

```
↳ -javax.swing.JLabel jLabel20
↳ -javax.swing.JLabel jLabel21
↳ -javax.swing.JLabel jLabel22
↳ -javax.swing.JLabel jLabel23
↳ -javax.swing.JLabel jLabel24
↳ -javax.swing.JLabel jLabel25
↳ -javax.swing.JLabel jLabel26
↳ -javax.swing.JLabel jLabel27
↳ -javax.swing.JLabel jLabel28
↳ -javax.swing.JLabel jLabel29
↳ -javax.swing.JLabel jLabel30
↳ -javax.swing.JLabel jLabel31
↳ -javax.swing.JPanel jPanel3
↳ -javax.swing.JScrollPane jScrollPane1
↳ -javax.swing.JScrollPane jScrollPane2
↳ -javax.swing.JScrollPane jScrollPane3
↳ -javax.swing.JScrollPane jScrollPane4
↳ -javax.swing.JSeparator jSeparator1
↳ -javax.swing.JSeparator jSeparator10
↳ -javax.swing.JSeparator jSeparator11
↳ -javax.swing.JSeparator jSeparator12
↳ -javax.swing.JSeparator jSeparator13
↳ -javax.swing.JSeparator jSeparator15
↳ -javax.swing.JSeparator jSeparator16
↳ -javax.swing.JSeparator jSeparator17
↳ -javax.swing.JSeparator jSeparator2
↳ -javax.swing.JSeparator jSeparator3
↳ -javax.swing.JSeparator jSeparator4
↳ -javax.swing.JSeparator jSeparator5
↳ -javax.swing.JSeparator jSeparator6
↳ -javax.swing.JSeparator jSeparator7
↳ -javax.swing.JSeparator jSeparator8
↳ -javax.swing.JSeparator jSeparator9
↳ +com.bolivia.label.CLabel lblImagen
↳ +javax.swing.JLabel lblUser
↳ +javax.swing.JPanel pnlAlbum
↳ +javax.swing.JPanel pnlCanciones
↳ +javax.swing.JPanel pnlCantantes
↳ +javax.swing.JPanel pnlFormAlbum
↳ +javax.swing.JPanel pnlFormCanciones
↳ +javax.swing.JPanel pnlFormCantantes
↳ +javax.swing.JPanel pnlFormUsuario
↳ +javax.swing.JPanel pnlMenu
↳ +javax.swing.JPanel pnlUsuarios
↳ +rojerusan.RSPanels Slider rSPanels Slider1
↳ +javax.swing.JTable tblAlbum
↳ +javax.swing.JTable tblCanciones
↳ +javax.swing.JTable tblCantantes
↳ +javax.swing.JTable tblUsuarios
```

```

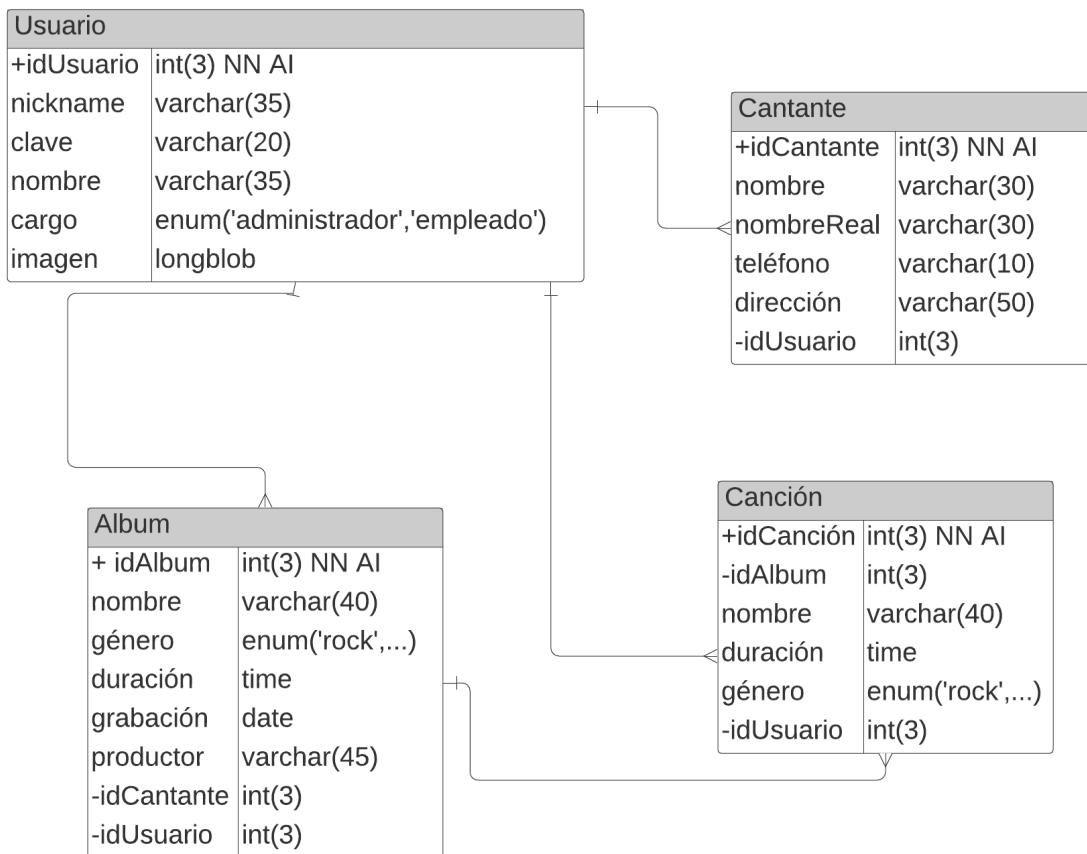
■+javax.swing.JTextField txtBuscarAlbum
■+javax.swing.JTextField txtBuscarCancion
■+javax.swing.JTextField txtBuscarCantante
■+javax.swing.JTextField txtBuscarUsuario
■+javax.swing.JTextField txtClave
■+javax.swing.JTextField txtDireccion
■+javax.swing.JTextField txtDuracion
■+javax.swing.JTextField txtDuracionCancion
■+javax.swing.JTextField txtIdAlbum
■+javax.swing.JTextField txtIdCancion
■+javax.swing.JTextField txtIdCantante
■+javax.swing.JTextField txtNickname
■+javax.swing.JTextField txtNombre
■+javax.swing.JTextField txtNombreAlbum
■+javax.swing.JTextField txtNombreArt
■+javax.swing.JTextField txtNombreCancion
■+javax.swing.JTextField txtNombreProductor
■+javax.swing.JTextField txtNombreReal
■+javax.swing.JTextField txtProductor
■+javax.swing.JTextField txtRuta
■+javax.swing.JTextField txtRutaI
■+javax.swing.JTextField txtTelefono
■+javax.swing.JLabel txtUser1
■+javax.swing.JLabel txtUser10
■+javax.swing.JLabel txtUser11
■+javax.swing.JLabel txtUser12
■+javax.swing.JLabel txtUser2
■+javax.swing.JLabel txtUser3
■+javax.swing.JLabel txtUser4
■+javax.swing.JLabel txtUser5
■+javax.swing.JLabel txtUser6
■+javax.swing.JLabel txtUser7
■+javax.swing.JLabel txtUser8
■+javax.swing.JLabel txtUser9
■+javax.swing.JTextField txtUsuario
■+javax.swing.JTextField txtUsuario2

♦+VistaMenu()
❷-// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents void initComponents()
❸-void btnUsuariosActionPerformed(java.awt.event.ActionEvent evt)
❸-void btnCantantesActionPerformed(java.awt.event.ActionEvent evt)
❸-void btnAlbumsActionPerformed(java.awt.event.ActionEvent evt)
❸-void btnCancionesActionPerformed(java.awt.event.ActionEvent evt)
❸-void btnExit2ActionPerformed(java.awt.event.ActionEvent evt)
❸-void btnExit3ActionPerformed(java.awt.event.ActionEvent evt)
❸-void btnExit4ActionPerformed(java.awt.event.ActionEvent evt)
❸-void btnExit5ActionPerformed(java.awt.event.ActionEvent evt)
❸-void btnInsertarCancionActionPerformed(java.awt.event.ActionEvent evt)
❸-void btnEliminarCancionActionPerformed(java.awt.event.ActionEvent evt)

```

```
⑨ -void btnInsertarAlbumActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void txtBuscarAlbumActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void btnEliminarUsuarioActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void btnInsertarUsuarioActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void txtBuscarUsuarioActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void btnGuardarUsuarioActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void btnGuardarCantanteActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void txtTelefonoActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void btnGuardarAlbumActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void btnGuardarCancionActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void txtNombreRealActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void txtRutaActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void btnImagenActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void txtRuta1ActionPerformed(java.awt.event.ActionEvent evt)
⑨ -void btnImagen1ActionPerformed(java.awt.event.ActionEvent evt)
⑨ +static void main(String args)
```

Modelo Relacional



Análisis y diseño del sistema de información

ESPECIFICACIÓN DE REQUERIMIENTOS

FUNCIONALES

- **Registro de usuarios:** La aplicación debe permitir ingresar usuarios, así también como el nombre de usuario, el nickname, contraseña y tipo de empleado.
- **Registro de canciones:** La aplicación debe permitir ingresar toda la información relevante acerca de las canciones, como el título de ésta, el álbum en el que se encuentra, el artista, la duración, el género y el año de lanzamiento.
- **Registro de artista:** La aplicación debe permitir ingresar toda la información relevante acerca del artista, como el nombre de éste, el nombre real, teléfono y dirección.
- **Registro de album:** La aplicación debe permitir ingresar toda la información relevante acerca de los álbumes, como el título de éste, productor/es, la duración, el género y el año de lanzamiento.
- **Guardar registros en una base de datos:** La aplicación debe permitir guardar todos los registros en una base datos.
- **Lectura de registros:** La aplicación debe permitir la consulta de todos los registros realizados.
- **Modificación de registros:** La aplicación debe permitir modificar los registros.
- **Eliminación de registros:** La aplicación debe permitir eliminar el registro seleccionado.
- **Restricción de usuarios:** La aplicación debe restringir vistas dependiendo del usuario con el que se inició sesión.

NO FUNCIONALES

- De interfaz
 - Estándar de GUI:
 - El sistema debe poseer interfaces gráficas bien formadas.
 - Distribución de la pantalla.

- De seguridad
 - Protección de la información que se manejará:
 - Grado de protección de los datos, software y plataforma de tecnología de posibles pérdidas, actividades no permitidas o uso para propósitos no establecidos previamente.
 - Definir autorización del ingreso de los usuarios:
 - La información debe estar protegida contra accesos no autorizados utilizando mecanismos de validación que puedan garantizar el cumplimiento de esto.
- De calidad de software
 - Eficiencia en el manejo de recursos:
 - Los datos modificados deben ser actualizados para todos los usuarios que acceden en menos de 5 segundos.
 - Mantenibilidad:
 - Flexibilidad para añadir requisitos futuros:
 - Desarrollar nuevos requerimientos
 - Aislar los defectos y sus causas
 - Corregir estos defectos y atender las demandas del entorno cambiante.
 - Disponibilidad:
 - El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.
 - Integridad:
 - Realiza validación con base en los riesgos.
 - Archiva periódicamente
 - Usabilidad:
 - El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
 - La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.

Metodología

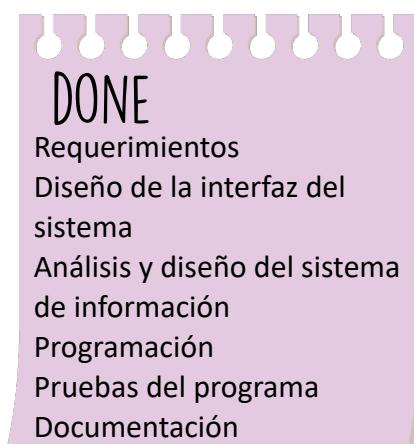
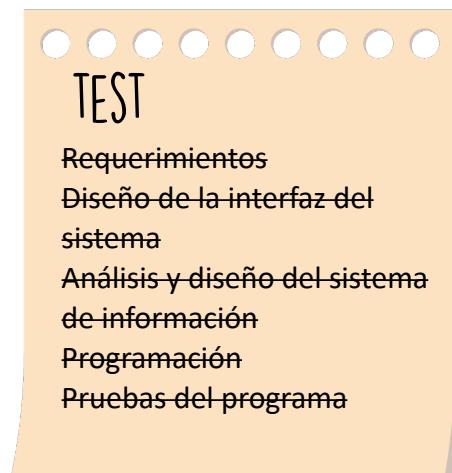
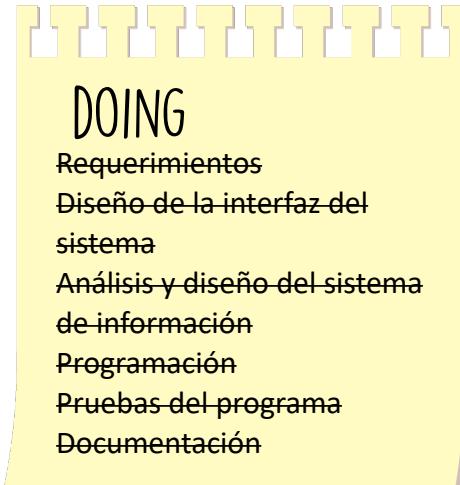
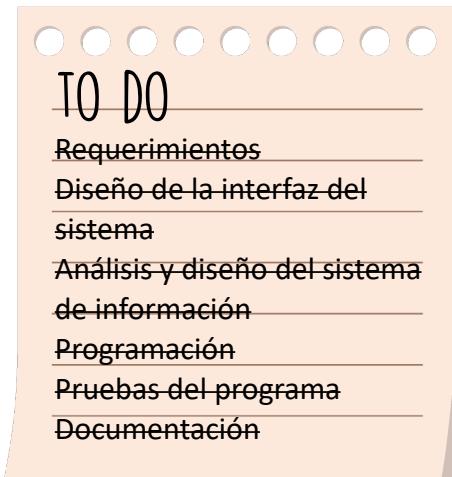
Como metodología se eligió “KANBAN”, la cual es una metodología ágil que consiste en gestionar el trabajo intelectual, con énfasis en la entrega justo a tiempo, se eligió porque en este enfoque, el proceso, desde la definición de una tarea hasta su entrega al cliente, se muestra para que los participantes lo vean y de esta manera se puede limitar el trabajo en curso.

- Lista de tareas o <>to do>>: engloba la lista de tareas que están pendientes y que se pueden afrontar inmediatamente. Para destacar la prioridad de las tareas, se colocan arriba las de mayor prioridad y orden descendente el resto, posteriormente, se puede pasar a otras columnas.
- En desarrollo o <>doing>>: se sitúan aquí las tareas hasta que estas sean completadas. Si algo falla, la tarea debe regresar a la columna previa.
- Pruebas o <>test>>: se realizan las comprobaciones o tests necesarios para ver si la tarea se ha realizado con éxito o no. Si está todo correcto pasa a la siguiente fase, si no, volvería a la fase de desarrollo.
- Despliegue o <>deployment>>: una vez validado el código se añaden a esta columna para su subida a producción en el sistema. Si todo funciona correctamente se pasaría a la siguiente fase.
- Terminado o <>done>>: aquí se incluyen las tareas que están finalizadas por completo.

Las tareas ha realizar son: reuniones con los clientes, identificar los requerimientos, diseño, diagramas, programación y pruebas.

Cada tarea debe pasar por cada fase de la metodología, con excepción de reuniones con los clientes, identificar los requerimientos, diseño y diagramas, ya que estás no pasan por la fase de despliegue o <>deployment>>.

Vista previa

A small image of a Trello board with a yellow sticky note pinned to it. The note has the text "Click [aquí](#) para ver el tablero en Trello."

Click [aquí](#) para ver el tablero en Trello.