

Manual de proyecto

MenuInicioSesionView



Estructura General

- **AnchorPane:** Este es un contenedor que permite anclar los nodos a posiciones específicas en el espacio. El `prefHeight` y `prefWidth` definen el tamaño preferido de la ventana.

## Componentes

1. **ImageView:** Muestra una imagen de fondo (`MenuInicial.png`) que se adapta al tamaño del `AnchorPane`. Su atributo `preserveRatio` asegura que la relación de aspecto de la imagen se mantenga.
2. **TextField (`tfUsuario`):** Este campo permite al usuario ingresar su nombre de usuario.
  - **promptText:** Muestra un texto indicativo cuando el campo está vacío, que dice "INGRESE SU USUARIO".
  - **Estilo:** Tiene un fondo naranja y bordes redondeados.
3. **PasswordField (`tfContra`):** Similar al `TextField`, pero está diseñado para ingresar contraseñas, ocultando los caracteres que se escriben.
  - **promptText:** Indica "INGRESE SU CONTRASEÑA".
4. **Button (`btnIniciar`):** Este botón permite al usuario iniciar sesión.
  - **text:** Muestra "INICIAR SESION".
  - **onAction="#handleButtonAction":** Este atributo vincula el botón a un método en el controlador (`MenuInicioSesionController`) que maneja la acción del botón cuando se hace clic.
  - **Estilo:** Tiene un fondo oscuro con esquinas redondeadas y el texto es blanco.
5. **Button (`btnRegistrar`):** Este botón está destinado a registrar un nuevo usuario.
  - **text:** Muestra "REGISTAR USUARIO".
  - **disable="true":** Este botón está deshabilitado, lo que significa que no se puede hacer clic en él hasta que se habilite mediante programación.
  - **onAction:** Similar al botón de iniciar sesión, este también está vinculado a un método de acción en el controlador.

## Funciones de los Botones

- **btnIniciar:**

- Cuando el usuario hace clic en este botón, se ejecuta el método `handleButtonAction`, que típicamente se encarga de verificar las credenciales del usuario y permitir el acceso si son correctas.
- `btnRegistrar`:
  - Aunque está deshabilitado en este momento, la intención es que al habilitarse permita al usuario registrarse en el sistema, probablemente abriendo un formulario o ventana para capturar los datos necesarios.

## Resumen

En resumen, este código define una interfaz de inicio de sesión con campos para usuario y contraseña, y botones para iniciar sesión y registrarse. Los botones están diseñados para ejecutar acciones específicas mediante el controlador asociado en JavaFX.

## MenuInstructor

ID	Nombre	Especialidad	Sueldo	Telefono
Tabla sin contenido				

## Estructura General

- **AnchorPane:** Contenedor que permite anclar los nodos a posiciones específicas. Define el espacio de trabajo donde se colocan los elementos de la interfaz.

## Componentes

1. **ImageView:** Muestra una imagen (Instructor.png) que se adapta al tamaño del contenedor y sirve como fondo o elemento decorativo.
2. **Label:** Utilizado para mostrar texto descriptivo al usuario:
  - **ID:** Indica el campo para el ID del instructor.
  - **Nombre:** Indica el campo para el nombre del instructor.
  - **Especialidad:** Indica el campo para la especialidad del instructor.
  - **Sueldo:** Indica el campo para el sueldo del instructor.
  - **Teléfono:** Indica el campo para el número de teléfono del instructor.
  - Cada Label tiene un textFill que define el color del texto.
3. **TextField:** Campos de entrada donde el usuario puede ingresar información:
  - **tfId:** Para ingresar el ID del instructor.
  - **tfNombre:** Para ingresar el nombre del instructor.
  - **tfEspecialidad:** Para ingresar la especialidad del instructor.
  - **tfSueldo:** Para ingresar el sueldo del instructor.
  - **tfTelefono:** Para ingresar el número de teléfono del instructor.
4. **TableView (tblInstructores):** Una tabla que muestra información sobre los instructores. Tiene varias columnas:
  - **colId:** Columna para el ID.
  - **colNombre:** Columna para el nombre.
  - **colEspecialidad:** Columna para la especialidad.
  - **colSueldo:** Columna para el sueldo.
  - **colTelefono:** Columna para el teléfono.
  - Las columnas permiten la visualización de datos de múltiples instructores.
5. **Button (btnGuardar):** Botón para guardar la información del instructor ingresada en los campos de texto.

- `onAction="#handleButtonAction"`: Al hacer clic, ejecuta el método `handleButtonAction` en el controlador, que probablemente guarda la información en una base de datos o en una lista.
- 6. Button (`btnEliminar`): Botón para eliminar la información del instructor seleccionado en la tabla.
  - Similar al botón guardar, ejecuta el mismo método de acción.
- 7. Button (`btnBuscar`): Botón para buscar un instructor específico utilizando la información ingresada.
  - También ejecuta el método `handleButtonAction`.
- 8. Button (`btnVaciar`): Botón que permite limpiar los campos de entrada, facilitando que el usuario comience de nuevo.
  - Este botón también está vinculado al mismo método de acción.
- 9. Button (`btnRegresar`): Botón que probablemente regresa a la pantalla anterior o cierra la ventana actual.
  - Igualmente ejecuta el método `handleButtonAction`.

## Resumen

En resumen, este archivo FXML define una interfaz para gestionar instructores, permitiendo ingresar, guardar, eliminar y buscar información sobre ellos. Utiliza un diseño con etiquetas, campos de texto, y una tabla para presentar los datos. Los botones ejecutan acciones específicas a través del controlador asociado, proporcionando interactividad a la interfaz.

## MenuMembresiaView



## Estructura General

- **AnchorPane:** Un contenedor que permite anclar elementos a posiciones específicas en el panel. Proporciona una estructura flexible para organizar los componentes de la interfaz.

## Componentes

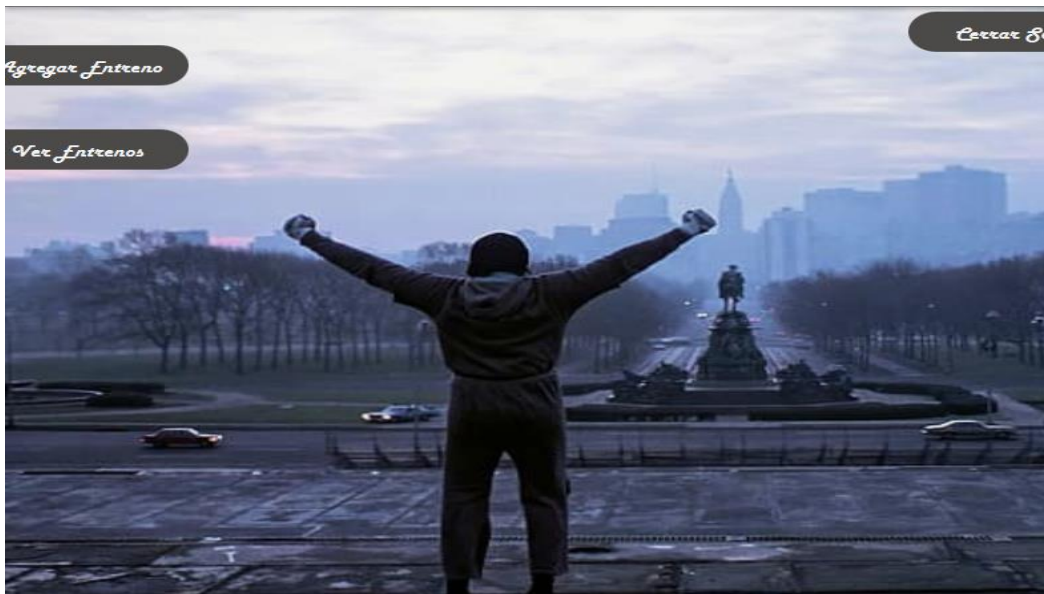
1. **ImageView:** Muestra una imagen de fondo (MenuMembresia.png) que se ajusta al tamaño del contenedor, proporcionando un contexto visual para la interfaz.
2. **Button (btnGuardar):** Botón para guardar la información de la membresía ingresada.
  - `onAction="#handleButtonAction"`: Al hacer clic, ejecuta el método `handleButtonAction` en el controlador, que maneja la lógica para guardar la membresía.
  - `style="-fx-background-color: #ffd700;"`: Aplica un color dorado al botón.
3. **TextField (tfId):** Campo de texto que muestra el ID de la membresía, pero no es editable (su valor se establecerá automáticamente al seleccionar una membresía de la tabla).
4. **TextField (tfDuracion):** Campo de texto donde se puede ingresar la duración de la membresía.

5. TableView (tblMembresias): Tabla que muestra la información de las membresías disponibles.
  - **Columnas:**
    - colId: Columna que muestra el ID de la membresía.
    - colDuracion: Columna que muestra la duración de la membresía.
    - colVigencia: Columna que muestra la vigencia de la membresía.
  - onMouseClicked="#cargarTextField": Al hacer clic en una fila de la tabla, se llama al método cargarTextField, que probablemente carga los datos de la membresía seleccionada en los campos de texto correspondientes.
6. Button (btnEliminar): Botón para eliminar la membresía seleccionada de la tabla.
  - También ejecuta el método handleButtonAction al hacer clic.
7. Button (btnLimpiar): Botón que permite limpiar todos los campos de entrada, facilitando la entrada de nueva información.
  - Este botón también ejecuta el mismo método de acción que los otros botones.
8. TextField (tfMembresiald): Campo de texto donde el usuario puede ingresar el ID de una membresía para buscarla.
  - promptText="Ingrese el ID para buscar": Texto que aparece como guía en el campo antes de que el usuario ingrese información.
9. Button (btnBuscar): Botón para buscar la membresía basada en el ID ingresado en tfMembresiald.
  - Este botón incluye un ImageView que muestra un icono de lupa para indicar su función de búsqueda.
  - style="-fx-background-color: #ffd700; -fx-border-radius: 50;": Aplica un color dorado y un borde redondeado al botón.
10. ComboBox (cmbVigencia): Un cuadro de combinación donde el usuario puede seleccionar la vigencia de la membresía.
  - style="-fx-background-color: #ffd700;": Aplica un color dorado al cuadro de combinación.

## Resumen

Este archivo FXML crea una interfaz que permite al usuario gestionar membresías en una aplicación de gimnasio. Los componentes incluyen campos para ingresar datos, botones para realizar acciones (guardar, eliminar, limpiar y buscar), y una tabla para visualizar las membresías existentes. La lógica detrás de los botones está manejada en el controlador asociado, facilitando la interacción del usuario con la aplicación.

## MenuPrincipalView



## Estructura General

El archivo FXML crea una interfaz gráfica de usuario para una aplicación JavaFX, utilizando un AnchorPane como contenedor principal. Este contenedor permite anclar los elementos de la interfaz en posiciones fijas.

### Elementos de la Interfaz

1. **ImageView:** Este elemento muestra una imagen de fondo en la interfaz. En este caso, carga MenuPrincipalView.png, que probablemente actúa como fondo visual del menú principal.
2. **MenuBar:** Crea una barra de menú en la parte superior de la ventana, donde se agrupan varias opciones de navegación.
  - **Módulos:** Agrupa opciones relacionadas con la gestión de aspectos del gimnasio:



- Clientes: Permite gestionar la información de los clientes del gimnasio.
- Instructores: Facilita la administración de los instructores.
- Máquinas: Proporciona acceso a la gestión de las máquinas del gimnasio.
- Sesiones: Permite gestionar las sesiones de entrenamiento.
- Sedes: Permite acceder a la gestión de las distintas ubicaciones del gimnasio.
- Membresías: Facilita la gestión de las diferentes opciones de membresía que ofrece el gimnasio.
- Ayuda: Incluye una opción "Acerca De" que brinda información sobre la aplicación, como su versión y desarrollador.
- Editar o Eliminar Entrenos: Menú dedicado a la gestión de entrenamientos, permitiendo al usuario editar o eliminar entrenos existentes.

### **3. Botones:**

- Cerrar Sesión: Permite al usuario cerrar su sesión actual en la aplicación.
- Agregar Entreno: Facilita la adición de nuevos entrenamientos al sistema.
- Ver Entrenos: Permite al usuario acceder y visualizar la lista de entrenamientos existentes.

## **Resumen**

Este archivo FXML define la interfaz principal de un sistema de gestión de gimnasio, con un fondo visual, una barra de menú para acceder a diferentes secciones de la aplicación y botones para realizar acciones clave como cerrar sesión, agregar y ver entrenamientos.

## MenuRegistroUsuario



## Elementos Principales

### 1. AnchorPane

- Es el contenedor principal que organiza los elementos en un espacio de anclaje. Permite posicionar los hijos utilizando coordenadas específicas.

### 2. ImageView

- Muestra una imagen en la interfaz. En este caso, se está utilizando para mostrar un fondo (MenuRegistroUsuario.png).

### Campos de Texto (TextFields)

#### 1. tfUsuario

- Función: Permite al usuario ingresar su nombre de usuario.
- Propiedades:
  - promptText: Indica al usuario que debe ingresar su usuario.

- style: Estilo visual del campo, con un fondo color naranja y bordes redondeados.

## 2. tfContra

- Función: Permite al usuario ingresar su contraseña.
- **Propiedades:**
  - promptText: Indica al usuario que debe ingresar su contraseña.
  - style: Estilo visual similar al de tfUsuario, con un fondo color naranja.

## Botones (Buttons)

### 1. btnRegistrar

- Función: Se utiliza para registrar un nuevo usuario. Al hacer clic, se ejecuta la acción definida en `onAction="#handleButtonAction"`.
- **Propiedades:**
  - text: Muestra "Registrar" en el botón.
  - style: Estilo visual con un fondo gris oscuro y bordes redondeados.
  - textFill: Color del texto en el botón, que es blanco.

### 2. btnRegresar

- Función: Permite al usuario regresar a la pantalla anterior o salir del proceso de registro. También ejecuta la misma acción de manejo de eventos que btnRegistrar.
- **Propiedades:**
  - text: Muestra "Regresar" en el botón.
  - style: Similar al de btnRegistrar, con un fondo gris oscuro y bordes redondeados.
  - textFill: Texto en color blanco.

### 3. btnInstructor

- Función: Permite agregar un instructor a la base de datos o lista. También ejecuta la acción definida en `onAction="#handleButtonAction"`.
- Propiedades:
  - text: Muestra "Agregar Instructor".

- style: Estilo visual similar a los otros botones, con un fondo gris oscuro y bordes redondeados.
- textFill: Color blanco para el texto.

## Comboboxes

### 1. cmbInstructores

- Función: Permite seleccionar un instructor de una lista desplegable.
- Propiedades:
  - style: Estilo visual que utiliza un fondo naranja y bordes redondeados.

### 2. cmbNivelAcceso

- Función: Permite seleccionar el nivel de acceso del usuario de una lista desplegable.
- Propiedades:
  - style: Estilo visual similar al anterior.

## MenuClienteView

Id:

Dpi:

Nombre:

Telefono:

Membresia:

Ingrese el ID para buscar

Id	dpi	nombre
Tabla sin contenido		

Guardar Eliminar

## Estructura General

- **AnchorPane:** Este es el contenedor principal que organiza los elementos hijos en un diseño de anclaje, permitiendo que cada componente se posicione de manera flexible dentro de su área.

## 2. Imágenes

- **ImageView:** Utilizado para mostrar una imagen de fondo (MenuCliente.png) que ocupa todo el espacio del AnchorPane. Se asegura de mantener la proporción de la imagen mientras se ajusta al tamaño del contenedor.

## 3. Tabla de Clientes

- **TableView:** Muestra una tabla que contiene información sobre los clientes. Incluye varias columnas:
  - **TableColumn (colId):** Muestra el ID del cliente.
  - **TableColumn (colDpi):** Muestra el DPI del cliente.
  - **TableColumn (colNombre):** Muestra el nombre del cliente.
  - **TableColumn (colTelefono):** Muestra el teléfono del cliente.
  - **TableColumn (colMembresia):** Muestra la membresía del cliente.

## 4. Campos de Texto

- **TextField:** Varias instancias para ingresar información del cliente:
  - **tfId:** Muestra el ID del cliente y no es editable.
  - **tfDpi:** Campo para ingresar el DPI del cliente.
  - **tfNombre:** Campo para ingresar el nombre del cliente.
  - **tfTelefono:** Campo para ingresar el teléfono del cliente.
  - **tfClientesId:** Campo para buscar clientes por ID, con un texto de marcador de posición.

## 5. Etiquetas de Texto

- **Text:** Utilizado para mostrar etiquetas descriptivas junto a los campos de texto. Ejemplos incluyen:
  - "Id:"
  - "Dpi:"
  - "Nombre:"

- "Telefono:"
- "Membresia:"

## 6. ComboBox

- ComboBox (cmbMembresia): Permite seleccionar la membresía del cliente de una lista desplegable.

## 7. Botones

- Button (btnGuardar): Al hacer clic, ejecuta una acción para guardar la información del cliente ingresada en los campos.
- Button (btnLimpiar): Al hacer clic, limpia los campos de texto para facilitar la entrada de nuevos datos.
- Button (btnRegresar): Permite regresar a la pantalla anterior o a un menú principal.
- Button (btnBuscar): Incluye una imagen (lupa) que, al hacer clic, ejecuta una acción para buscar al cliente según el ID ingresado.
- Button (btnEliminar): Permite eliminar al cliente seleccionado de la tabla.

## 8. Interactividad

- onAction: Los botones están vinculados a un controlador que maneja las acciones, lo que significa que al hacer clic en cada botón, se ejecutará un método específico definido en el controlador (#handleButtonAction).

# MenuEntrenoView



## Código XML

1. Estructura Principal:
  - Se utiliza un AnchorPane como contenedor principal, con dimensiones prefijadas de 1000x600.
2. Imágenes:
  - Se incluye una imagen de fondo (MenúEntrenos.png) que ocupa toda la ventana.
3. Tablas:
  - Tres TableView son definidas:
    - tblEntrenos: Muestra información sobre los entrenamientos (ID, instructor, sesión).

- tblMaquinas: Muestra detalles sobre las máquinas (ID, marca, especialidad).
- tblClientes: Muestra información sobre los clientes (ID).

#### 4. Campos de Texto y Botones:

- Un TextField (tfEntrenold) permite la búsqueda de entrenamientos.
- Botones para realizar acciones como buscar (btnBuscar), guardar (btnGuardar), eliminar (btnEliminar), limpiar (btnLimpiar), y regresar (btnRegresar).
- Dos ComboBox (cmbInstructores y cmbSesiones) para seleccionar instructores y sesiones.

#### 5. Estilo:

- Los botones tienen un estilo personalizado, con un color de fondo específico y texto en blanco.
- Se utilizan fuentes específicas para el texto de los botones.

#### 6. Interactividad:

- Se incluyen métodos de controlador asociados (onAction), permitiendo que cada botón ejecute una acción específica cuando se hace clic.

#### 7. Imagen de Lupa:

- Se incluye un icono de lupa para la funcionalidad de búsqueda, con un evento de detección de arrastre.



# MenuMaquinasView



## Estructura General

### 1. Root Element: AnchorPane

- Propiedades: Establece dimensiones máximas, mínimas y preferidas para la interfaz.
- Controller: Vinculado a MenuMaquinasController, que maneja la lógica de la interfaz.

### 2. Elementos Visuales

- ImageView: Muestra una imagen de fondo (en este caso, MenúMáquinas.png), que ocupa todo el panel.
- TextField: Hay varios campos de texto que permiten la entrada de datos, como el ID, la marca y la especialidad de las máquinas.

- TableView: Muestra una tabla con las máquinas disponibles, incluyendo columnas para ID, marca y especialidad.

## Función de Cada Componente

### Botones

1. btnGuardar:
  - Función: Al hacer clic, este botón ejecuta la acción definida en el controlador (handleButtonAction). Se utiliza para guardar la información ingresada en los campos de texto (ID, marca, especialidad) en algún tipo de almacenamiento (base de datos, lista, etc.).
2. btnVaciar:
  - Función: Este botón se utiliza para limpiar todos los campos de texto. Es útil para borrar la información ingresada previamente.
3. btnRegresar:
  - Función: Este botón permite regresar a la pantalla anterior o al menú principal. La lógica de esta acción también se define en el controlador.
4. btnEliminar:
  - Función: Al hacer clic, este botón eliminará la máquina seleccionada de la tabla. Esto podría implicar la eliminación de un registro en la base de datos.
5. btnBuscar:
  - Función: Este botón permite buscar una máquina específica. Se puede usar el campo de texto tfBuscar para ingresar un ID o cualquier criterio de búsqueda. La lógica de búsqueda se maneja en el controlador.

### Campos de Texto

1. tfId:
  - Función: Muestra el ID de la máquina seleccionada en la tabla. Este campo es de solo lectura (editable="false").
2. tfMarca:
  - Función: Permite al usuario ingresar la marca de la máquina.
3. tfEspecialidad:
  - Función: Permite al usuario ingresar la especialidad de la máquina (por ejemplo, cardio, fuerza, etc.).

#### 4. tfBuscar:

- Función: Permite al usuario ingresar un término de búsqueda para encontrar una máquina en la tabla.

#### Tabla

- tblMaquinas:
  - Función: Muestra la lista de máquinas disponibles en la aplicación. Se compone de columnas para ID, marca y especialidad, lo que permite visualizar las máquinas registradas.

### MenuRegistroUsuario

The screenshot displays the 'MenuRegistroUsuario' interface. At the top right, the title 'Menú Usuarios:' is prominently displayed. Below it, there are two input fields for user registration: 'INGRESE SU USUARIO' and 'INGRESE SU CONTRASEÑA'. A dropdown menu is positioned below these fields. To the right of the dropdown, there is a button labeled 'Agregar Instructor'. At the bottom of the interface, there are two buttons: 'Regresar' and 'Registrar'. The background of the interface is a blurred image of a gym with dumbbells.

#### Estructura General

## 1. Root Element: AnchorPane

- Propiedades: Define el tamaño preferido del panel (500.0 x 650.0) y se vincula a un controlador (MenuRegistroUsuarioController) que manejará la lógica de la interfaz.

## 2. Elementos Visuales

- ImageView: Se utiliza para mostrar una imagen de fondo (MenuRegistroUsuario.png), que ocupa todo el panel.
- TextField: Permite la entrada de datos del usuario, como el nombre de usuario y la contraseña.
- ComboBox: Se utiliza para seleccionar opciones, en este caso, instructores y niveles de acceso.
- Botones: Se utilizan para ejecutar acciones como registrar al usuario y regresar al menú anterior.

## Función de Cada Componente

### Botones

#### 1. btnRegistrar

- Función: Este botón se utiliza para registrar un nuevo usuario. Cuando se hace clic, se invoca la acción definida en el controlador (handleButtonAction), donde se espera que se maneje la lógica para guardar los datos del nuevo usuario (nombre de usuario, contraseña, instructor y nivel de acceso).

#### 2. btnRegresar

- Función: Este botón permite regresar a la pantalla anterior o al menú principal. La acción también se maneja en el controlador, donde se puede definir la lógica para cerrar la pantalla actual y abrir la anterior.

#### 3. btnInstructor

- Función: Al hacer clic, este botón permite agregar un instructor a la lista de instructores. La lógica de esta acción también se maneja en el controlador y puede incluir agregar un nuevo instructor a la base de datos o a la lista de opciones del ComboBox correspondiente.

## Campos de Texto

#### 1. tfUsuario

- Función: Campo de texto donde el usuario debe ingresar su nombre de usuario. Tiene un texto de sugerencia ("INGRESE SU USUARIO") que se muestra cuando el campo está vacío.

## 2. tfContra

- Función: Campo de texto para ingresar la contraseña del usuario. También tiene un texto de sugerencia ("INGRESE SU CONTRASEÑA") y puede ser configurado para ocultar el texto ingresado, por ejemplo, utilizando `setPromptText` o configuraciones adicionales en el controlador.

## ComboBoxes

### 1. cmbInstructores

- Función: Permite seleccionar un instructor de una lista. Los instructores deben ser agregados a este ComboBox desde el controlador, probablemente cargando los datos desde una base de datos o lista de instructores disponibles.

### 2. cmbNivelAcceso

- Función: Permite seleccionar un nivel de acceso para el usuario, que puede incluir opciones como "Administrador", "Instructor" o "Miembro". Las opciones deben ser agregadas en el controlador, posiblemente dependiendo del rol del usuario.

## MenuSedesView



### Estructura General

#### 1. Elemento Raíz: AnchorPane

- Propiedades: Define un panel con dimensiones preferidas de 600x400 píxeles. Las propiedades de máximo y mínimo tamaño están configuradas para ser infinitas, lo que permite que el panel se redimensione si es necesario. Está vinculado al controlador MenuSedeController, donde se manejará la lógica de la interfaz.

#### 2. Elementos Visuales

- ImageView: Muestra una imagen de fondo (MenúSedes.png), que ocupa todo el panel.
- TextField: Se utiliza para ingresar y mostrar información sobre la sede, como su ID y dirección.
- TableView: Muestra una tabla con las sedes registradas, permitiendo la selección de filas para editar o eliminar.
- Botones: Ejecutan acciones específicas como guardar, eliminar, limpiar, buscar y regresar.

## Función de Cada Componente

### Botones

#### 1. btnGuardar

- Función: Este botón se utiliza para guardar los cambios realizados en la información de la sede. Cuando se hace clic, invoca la acción definida en el controlador (handleButtonAction), donde se espera que se maneje la lógica para guardar la nueva sede o actualizar la información de una existente.

#### 2. btnEliminar

- Función: Este botón permite eliminar una sede seleccionada de la tabla. Al hacer clic, también se invoca el método handleButtonAction en el controlador, que debería manejar la lógica para eliminar la sede seleccionada.

#### 3. btnLimpiar

- Función: Este botón se utiliza para limpiar los campos de entrada (ID y dirección). Al hacer clic, se invoca la acción en el controlador que restablece los campos de texto a su estado inicial, facilitando la entrada de nuevos datos sin la interferencia de información previa.

#### 4. btnBuscar

- Función: Permite buscar una sede en la tabla introduciendo un ID en el campo correspondiente (tfSedeld). Al hacer clic, se ejecuta la acción en el controlador para filtrar las sedes mostradas en la tabla según el ID proporcionado. El botón incluye un ícono de lupa para indicar su función de búsqueda.

#### 5. btnRegresar

- Función: Este botón permite regresar a la pantalla anterior o al menú principal. Al hacer clic, se invoca la acción en el controlador para cerrar la ventana actual y volver a la interfaz anterior, facilitando la navegación dentro de la aplicación.

### Campos de Texto

#### 1. tfId

- Función: Campo de texto que muestra el ID de la sede. Está configurado como no editable (editable="false"), lo que significa que el usuario no puede modificar este campo directamente. Este ID probablemente se establece al seleccionar una sede en la tabla.

## 2. tfDireccion

- Función: Este campo de texto permite ingresar o modificar la dirección de la sede. Es editable y debe contener la dirección correspondiente a la sede seleccionada.

## 3. tfSedeId

- Función: Campo de texto donde el usuario puede ingresar el ID de una sede que desea buscar. Tiene un texto de sugerencia que indica al usuario que debe "Ingresar el ID para buscar".

## Tabla

### 1. tblSedes

- Función: Muestra una lista de sedes registradas en la aplicación. Permite al usuario seleccionar una fila para cargar la información de la sede en los campos de texto (ID y dirección). La selección de una fila activa el método cargarTextField, que probablemente llena los campos de texto con los datos de la sede seleccionada.

## MenuSessionView

**MENÚ SESIONES**

Ingrese el ID para buscar

ID

ESPECIALIDAD

Guardar Eliminar Limpiar

Id	Especialidad
Tabla sin contenido	

## Estructura General

- AnchorPane: Es un contenedor que permite posicionar sus hijos en posiciones específicas (ancladas) dentro de él, lo que facilita el diseño de la interfaz.



## Componentes y Funcionalidades

### 1. ImageView:

- Muestra una imagen de fondo que en este caso es MenúSesiones.png.
- Se ajusta a las dimensiones del AnchorPane (400x600).

### 2. TextField tfId:

- Campo de texto no editable que probablemente se utiliza para mostrar el ID de la sesión seleccionada.

### 3. TextField tfEspecialidad:

- Campo de texto donde se ingresará o mostrará la especialidad de la sesión.

### 4. TableView tblSesiones:

- Tabla que muestra las sesiones disponibles.
- Contiene dos columnas:
  - colId: Para mostrar el ID de cada sesión.
  - colEspecialidad: Para mostrar la especialidad de cada sesión.
- onMouseClicked: Al hacer clic en una fila, probablemente se cargará la información en los campos de texto (tfId y tfEspecialidad).

## Botones

### 1. Button btnGuardar:

- Funcionalidad: Guarda la información ingresada en los campos de texto.
- Estilo: Tiene un fondo oscuro y texto en blanco.

### 2. Button btnEliminar:

- Funcionalidad: Elimina la sesión seleccionada de la tabla.
- Estilo: También tiene un fondo oscuro y texto en blanco.

### 3. Button btnLimpiar:

- Funcionalidad: Limpia los campos de texto para permitir la entrada de nueva información.
- Estilo: Igual que los otros botones, con un fondo oscuro.

### 4. TextField tfSesionId:

- Campo de texto donde el usuario puede ingresar el ID de una sesión para buscarla en la tabla.

5. Button btnBuscar:

- Funcionalidad: Busca la sesión basada en el ID ingresado en tfSesionId y, si existe, carga la información en los otros campos.
- Gráfico: Incluye un icono de lupa, indicando su función de búsqueda.

6. Button btnRegresar:

- Funcionalidad: Permite al usuario regresar a la pantalla anterior o salir de la vista actual.
- Estilo: Similar a los otros botones.