**SISTEMA DE GESTIÓN DE RESERVAS PARA ESPACIOS DE COWORKING**

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

# Eliú Manuel Viera Lorenzo

# David Liaño Macías

# 2ºDAM - DAD

*Proyecto creado Por Eliú Manuel Viera Lorenzo y David Liaño Macías.*

*2º DAM Semipresencial*

*2024-2025*

*Desarrollo de Interfaces.*

*Proyecto: Aplicación Sistema de Gestión de Reservas para Espacios de Coworking*

*WORKTOPIA*

# Indice

|  |  |  |  |
| --- | --- | --- | --- |
| Descripción |  |  | Pag.1-3 |
|  |  |  |  |
| Idea General |  |  | Pag.1 |
|  |  |  |  |
| Estructura |  |  | Pag.1 |
|  |  |  |  |
|  | Módulos Principales |  | Pag.1 |
|  | Herramientas |  | Pag.1-2 |
|  | Interfaces |  | Pag.2 |
|  | Base de Datos |  | Pag.2 |
|  | Desarrollo |  | Pag.3 |
|  | Pruebas |  | Pag.3 |
|  | Objetivos |  | Pag.3 |
|  |  |  |  |
| Base de datos |  |  | Pag.4-5 |
|  |  |  |  |
|  | Esquema |  | Pag.4 |
|  | MySQL |  | Pag.4-5 |
|  |  |  |  |
| Proyecto |  |  | Pag.6-21 |
|  | Colores y tipografía |  | Pag.6 |
|  | Estructura |  | Pag.6-7 |
|  | Manual Usuario (Interfaz) |  | Pag.8-18 |
|  |  |  |  |
| Código |  |  | Pag.19-123 |
|  |  |  |  |
|  |  |  |  |
| Pom, pruebas y errores |  |  | Pag.124-129 |
|  | Intento de crear documentación |  | Pag.128-129 |
|  |  |  |  |
| Intento de Ejecutable |  |  | Pag.130 |
|  |  |  |  |
| Conclusiones |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Descripción

IDEA GENERAL

Un programa para administrar espacios de coworking o salas de reuniones.

Con características para reservar espacios por horarios y capacidades, interfaz gráfica para reservas. Facturación automática por uso del espacio. Registro de clientes frecuentes y descuentos personalizados. Estadísticas de uso de espacios para optimizar recursos. Aplicación comercial: Diseñado para negocios de coworking en expansión.

Estructura General del Proyecto

Módulos Principales

* Gestión de Clientes:

La aplicación poseerá un registro y edición de clientes. Donde cada uno poseerá un listado de reservas y facturas vinculadas.

* Gestión de Reservas:

Registro de cada reserva, vinculada a un DNI, asientos, periodo de reserva y factura, se vinculará de manera automática al ultima factura del cliente emitida sin abonar o generará una nueva si no posee.

* Sistema de Reservas:

Interfaz visual para seleccionar fechas, horarios o periodos, asientos libres, ocupados o seleccionados además de servicios proporcionados en el espacio como máquinas expendedoras o servicios. Validación de disponibilidad en tiempo real. Generación de facturas al completar reservas.

* Facturación

Cálculo automático de precios basado en el periodo de reserva y descuentos. Emisión de comprobantes (PDF o formato simple).

* Usuarios

Administración de los usuarios (empleados) que usarán la aplicación.

Herramientas y Tecnologías

* IDE:

Se utilizará IntelliJ IDEA.

* Lenguaje

Java.

* Interfaz Gráfica

JavaFX. /CSS.

* Base de Datos

MySQL para crear la base de datos y Dia para el diagrama relacional. Alojada en <https://console.clever-cloud.com/>

* Github

Uso de Github para el desarrollo cooperativo de la aplicación. <https://github.com/denox74/Worktopia>

* Diseño

Figma para el concept art del proyecto.

HomeByMe para el plano 2D (normalmente se usa para la creación de decoración de interiores).

Gimp, herramienta de código libre para manipular imágenes.

ChatGPT junto a su herramienta DALL-E 3 para la creación de imágenes y logos.

Diseño de Interfaces

* Pantalla de Presentación:

1. Imagen corporativa de la aplicación.
2. Pantalla de presentación de aplicación.
3. Login de Usuario con contraseña.
4. Aceptar.

* Barra de navegación.

1. Barra con todos los accesos a las diferentes partes de la aplicación.

* Diseño de clientes:

1. Tablas con listado con la información de todos los clientes.
2. Botón para añadir más clientes.
3. Barra de búsqueda por DNI de los clientes.
4. Acceso a ventana adicional con detalles de los clientes para modificar, eliminar o empezar las reservas del cliente.

* Diseño Reservas

1. Pantalla donde realizar las reservas.
2. Gestión por fecha y escogiendo el asiento.
3. Barra para escribir manualmente el DNI.
4. Selección de horario y disponibilidad mediante colores (Verde: Disponible, Rojo: No Disponible).
5. Coste de la reserva calculada automáticamente.

* Diseño de Reservas.

1. Listado de reservas con los datos de cada ella, precio, DNI vinculado, factura vinculada, asiento, y horario.
2. Menú adicional al hacer doble clic, para modificar, o eliminar la reserva. (No aplicable si está pagada).
3. Barra de búsqueda mediante el DNI del cliente.

* Diseño de facturación.

1. Listado de facturas.
2. Barra de búsqueda mediante el número de factura.
3. Menú adicional de cada factura para aplicar descuentos, borrar (no aplicable si está pagada), medio de pago, pagar factura y emitir en PDF.

* Usuarios.

1. Manejador por tablas de los usuarios, solo accesible para administradores.
2. Menú adicional para modificar o eliminar usuarios.

Desarrollo

* Crear base de datos: Se hará un esquema de entidad/relación para desarrollar la base de datos. Una vez concretada se comenzará a realizar la base de datos en MySQL.
* De idea a boceto: Se crearán maquetas de las interfaces.
* Interfaz: Para una planificación más ágil se realizará primero la parte de interfaz y diseño. Las interfaces o mapas de los planos de los espacios. Una vez realizadas se irán implementando en la interfaz con JavaFX.
* Programación: Una vez diseñado la aplicación, se tendrá más claro los puntos a programar y terminar con el programa.
* Enlace DDBB: Se realizará el enlace con la base de datos.
* Pruebas finales.
* Presentación.

Pruebas

Las pruebas se llevarán a cabo con Junit. Todas las funciones se llevarán a cabo de manera unitaria a medida que se van desarrollando.

Una vez desarrollado el programa sin estar vinculado a la base de datos se llevarán a cabo las pruebas de enlace y funcionamiento general. Probando introducción de usuarios, reservas, ocupaciones, etc.

Tras encontrar problemas con Maven, muchas pruebas y por falta de tiempo no ha podido llevarse a cabo mediante este medio

objetivos

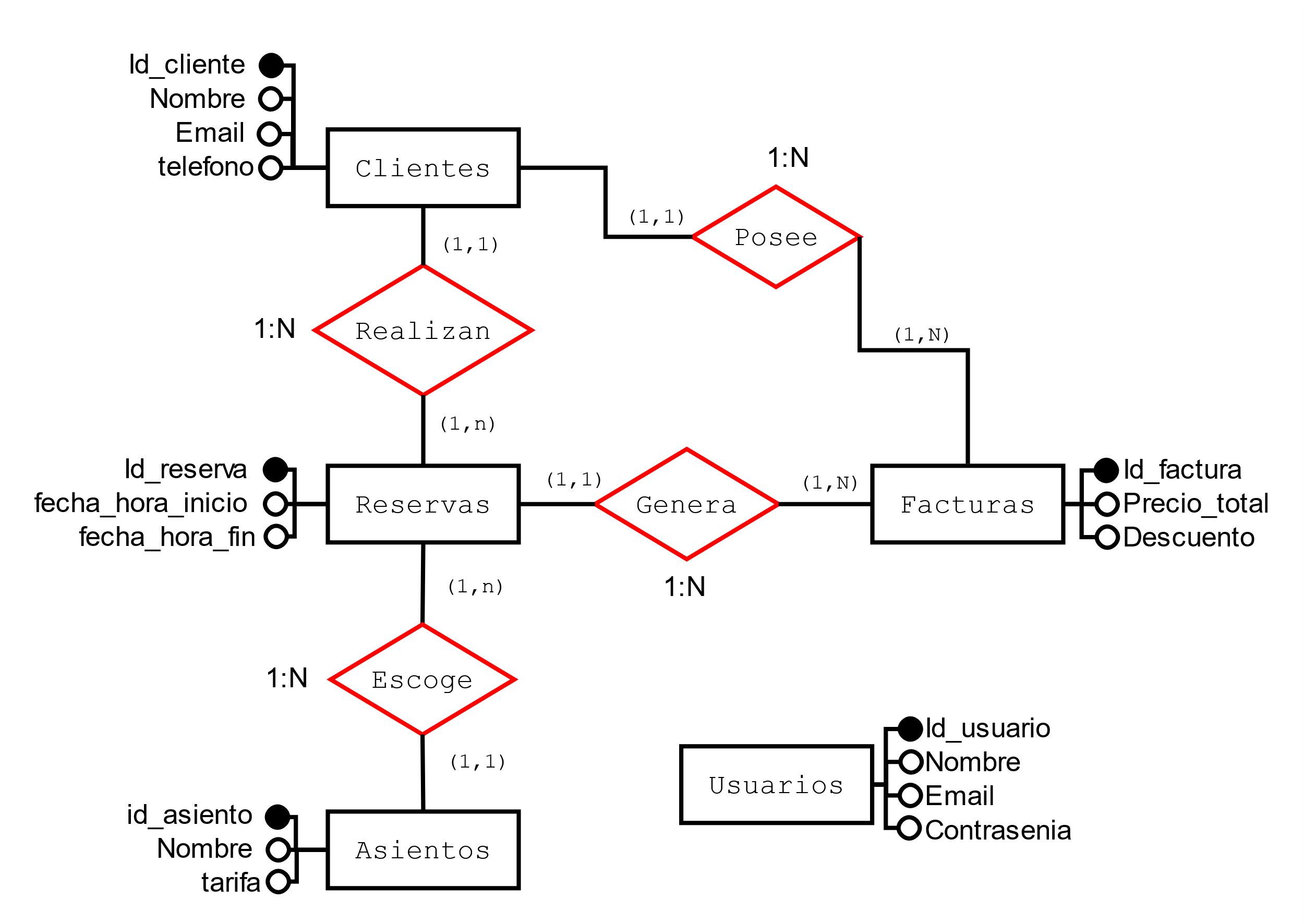
Los objetivos iniciales es crear una aplicación funcional donde una empresa que posea unos locales o espacios acondicionados a una oficina de coworking pueda usarla para gestionar reservas.

La aplicación debe poder gestionar a los usuarios y sus clientes para que puedan realizar sus reservas cómodamente, de manera intuitiva y fácil. Pudiendo ver en todo momento sus reservas actuales, antiguas y facturación.

Debe ser una aplicación visualmente atractiva, minimalista y con colores suaves. Iconos altamente identificativos con sus acciones y localizables.

# Base de datos

Usando el programa Dia para la generación de diagramas y los datos en la descripción, se ha generado el siguiente diagrama:



**CREATE TABLE Clientes (**

**dni VARCHAR(10) NOT NULL,**

**nombre VARCHAR(100) NOT NULL,**

**primerApellido VARCHAR(30) NOT NULL,**

**segundoApellido VARCHAR(30),**

**email VARCHAR(100) NOT NULL UNIQUE,**

**telefono VARCHAR(15),**

**CONSTRAINT pk\_dni\_cliente PRIMARY KEY (dni)**

**);**

**CREATE TABLE Asientos (**

**id\_asiento INT AUTO\_INCREMENT,**

**nombre VARCHAR(255) NOT NULL,**

**tarifa\_hora DECIMAL(10, 2) NOT NULL,**

**CONSTRAINT pk\_id\_asiento PRIMARY KEY (id\_asiento)**

**);**

MySQL

**CREATE TABLE Facturas (**

**id\_factura INT NOT NULL,**

**dni VARCHAR(10) NOT NULL,**

**precio\_total DECIMAL(10, 2) NOT NULL DEFAULT 0.00,**

**descuento DECIMAL(5,2) DEFAULT 0.00,**

**subtotal DECIMAL(10,2) NOT NULL,**

**fecha\_hora\_emision DATETIME DEFAULT CURRENT\_TIMESTAMP,**

**estado ENUM('Pendiente', 'Pagada') NOT NULL DEFAULT 'Pendiente',**

**fecha\_hora\_pago DATETIME,**

**forma\_pago VARCHAR(50),**

**CONSTRAINT pk\_id\_factura PRIMARY KEY (id\_factura),**

**CONSTRAINT fk\_dni\_factura FOREIGN KEY (dni) REFERENCES Clientes(dni)**

**);**

**CREATE TABLE Reservas (**

**id\_reserva INT AUTO\_INCREMENT,**

**dni VARCHAR(10) NOT NULL,**

**id\_asiento INT NOT NULL,**

**id\_factura INT NOT NULL,**

**fecha\_hora\_inicio DATETIME NOT NULL,**

**fecha\_hora\_fin DATETIME NOT NULL,**

**subtotal DECIMAL(10, 2) NOT NULL DEFAULT 0.00,**

**CONSTRAINT pk\_id\_reserva PRIMARY KEY (id\_reserva),**

**CONSTRAINT fk\_dni\_reserva FOREIGN KEY (dni) REFERENCES Clientes(dni),**

**CONSTRAINT fk\_id\_asiento\_reserva FOREIGN KEY (id\_asiento) REFERENCES Asientos(id\_asiento),**

**CONSTRAINT fk\_id\_factura\_reserva FOREIGN KEY (id\_factura) REFERENCES Facturas(id\_factura)**

**);**

**CREATE TABLE Usuarios (**

**id\_usuario INT AUTO\_INCREMENT,**

**nombre VARCHAR(100) NOT NULL,**

**email VARCHAR(100) NOT NULL UNIQUE,**

**contrasenia VARCHAR(255) NOT NULL,**

**categoria ENUM('Admin', 'Empleado') NOT NULL,**

**CONSTRAINT pk\_id\_usuario PRIMARY KEY (id\_usuario)**

**);**

# Proyecto

A continuación, pasamos a los datos más técnicos, tras haber realizado el proyecto. Apartados como colores, estructura del proyecto, iconos, y algo del código.

Colores y tipografía

Principal: #1C7F77

Oscuro: #0F4D48

Claro: #EADDC8

Fondo: #DDDEDE

Integrador: #F4A261

Estos son los colores elegidos que compondrán el proyecto, además de un verde fluorescente para confirmaciones y un rojo para alertas, cancelar o cerrar ventanas. No se ha elegido ninguna Tipografía en concreto, se ha dejado la del sistema por defecto para una mejor integración con la personalización e integración del SO del usuario.

Estructura del proyecto

La estructura del proyecto es la siguiente:

Dado que el proyecto se ha estructurado con Java y Maven ha empeñado una mayor complicación a la hora de integrar *Frameworks*, dependencias y el desarrollo general de la aplicación, se ha intentado mantener un orden por carpetas para un mejor desarrollo y visualización.

Texto

El contenido generado por IA puede ser incorrecto.Carpeta del proyecto **Worktopia\_cod**, a partir de esa carpeta se integrarán todos los archivos para el proyecto.

Sus carpetas principales son el .idea (configuración generada por el *IDE IntelliJ IDEA*), **.nvm** con el **wrapper** de Maven para ejecutarlo sin la necesidad de instalarlo en el sistema.

La otra carpeta importante el **src**, (source) donde almacenará todos los archivos del código del proyecto que son los siguientes.

Texto

El contenido generado por IA puede ser incorrecto.Main contenido dentro del src, tiene dos carpetas altamente importantes, java y resources.

La carpeta java posee varios paquetes, donde cada paquete son archivos .java diferenciados por tipos:

* **Aplicaciones:** donde se encuentra el main que arranca la aplicación “MenuPrincipalApp”.
* **Clases:** aquí se encuentran las clases principales, dichas clases son básicamente las que recoge de la base de datos para una mayor agilidad: “Asientos, Clientes, Facturas, Reservas, SesionUsuario, Usuarios”.
* **ConexionDB**: clase que realizará la conexión de la base de datos.
* **Controlador**: Funciones dedicadas para las clases: “Controlador Clientes, ControladorEmail, ControladorFacturas, ControladorReservas, ControladorUsuarios”.
* **Manejadores\_Reservas\_Facturas**: Finalmente solo se ha usado el archivo “ReservaDAO” para vincular y calcular reservas con las facturas.
* **Modelos**: Esta parte se encarga de gestionar las interfaces de cada ventana; “Facturacion, ListaClientes, ListaReservas, ListaUsuarios, MenuPrincipal, RegistroClientes, RegistroUsuarios, ReservasPanel”.
* Dentro de **resources** tenemos las carpetas **Docs** (que contiene el Excel de formato de factura y las facturas generadas), **Estilos** (estilos css aplicados a los fxml), **Fuente** (finalmente no usada dado que se usa la del sistema), **html** (archivo html que se usa para enviar el email de reservas), **Imágenes** (todas las imágenes del proyecto e iconos) y por último **Menus**.
* **Menus** es la carpeta que contiene los .fxml, los archivos que se encargan del diseño de las ventanas de la aplicación: “(Facturacion, ListaClientes, ListaReservas, Lista Usuarios, MenuPrincipal, RegistroClientes).

Manual Usuario (Interfaz)

Para visualizar el programa se usará el manual de usuario:

Menú de inicio que sería el **MenuPrincipal.fxml** con su modelo **MenuPrincipal.java**, se inicial al iniciar el archivo **MenuPrincipalApp**.

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

Aquí se inicia la sesión de usuario (empleado o admin) indicando su contraseña.

Una vez dentro el programa gestión se conectará a la base de datos y empezaremos con el uso del programa.

Aparecerá directamente en la lista de clientes, que básicamente es una tabla donde carga los datos de todos los clientes, su clave principal es el DNI, los otros datos son su nombre, apellidos, email y teléfono, los clientes pueden ser modificados o eliminados, pero para ello se debe de iniciar su ventana que emerge al hacer doble clic en un cliente. A continuación, un fragmento del manual del usuario para visualizarlo.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

NAVEGACIÓN Y AÑADIR CLIENTES

*Este es el plano general del programa, y la primera vista, que es la lista de clientes. En ella dispones de una barra de búsqueda1 según el DNI del usuario que indiques filtrará y mostrará solo los deseados. Además, en la esquina derecha*

*Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.está situado un botón de* ***+*** *para añadir*

*más clientes a la lista. Añadir Cliente2*

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

1Barra de búsqueda

*Si pulsamos el botón entonces se nos abrirá una nueva pequeña ventana dicha ventana dispondrá de un formulario para introducir los datos de los clientes, todos son obligatorios. Registro*

2 Añadir Cliente

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

3 Registro de Cliente

*Un formulario sencillo donde, indicando todos los datos adecuadamente se procede al registro.*

*DNI/NIF, el nombre del cliente, sus apellidos, email y teléfono.*

*Si todo está correcto vale con dar a* ***Confirmar****, en caso de no querer introducirlo o quitar todos los datos, puede limpiar los datos con el botón de* ***Limpiar Datos****.*

*O si se desea salir simplemente pulse el botón* ***x*** *de la esquina para salir.*

abrir y modificar clientes

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

*Para abrir la ficha de un cliente basta con hacer doble clic sobre el nombre del cliente. Esto abrirá una nueva ventana parecida al formulario, donde puedes modificar sus datos, eliminar el cliente o comenzar a realizar las reservas a su nombre.*

*Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.En dicha ventana se dispone de los datos introducidos de los clientes y tres botones principales:*

*El primer botón es el de modificar, o confirmar la modificación, dado que modificar los datos de un cliente es tan fácil como escribir en sus casillas y presionar ese botón para* ***Modificar****4.*

*El segundo botón es de eliminar, que elimina directamente el cliente, ojo, no recomendable dado que podría afectar a la facturación, un cliente dado de alta con facturas o reservas no debe de eliminarse sin un administrador.* ***Eliminar****5.*

*Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto. Por último, el botón de* ***Reservar****6. Que nos llevará directamente al menú de reservas.*

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

4 Modificar

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

5 Eliminar

6 Reservar

RESERVAR

Ahora, pasamos a la pantalla de reservas, visualmente es la más importante, ya que es la ventana más grafica de todas, donde puedes interactuar directamente con el plano de la “oficina” para proceder con las reservas, seguimos con la parte del manual para explicarlo más detalladamente.

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

*Esta es la pantalla principal para empezar a realizar las reservas de los clientes. Su menú se compone de:*

* *Un cajón donde está el cliente, si está correcto, el borde se coloreará de verde, en caso de que el DNI esté mal, se coloreará de color rojo.*

*Imagen que contiene Tabla

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.*

* *Luego, para comenzar una reserva, se debe de indicar primero de todo, una fecha. Puedes escribirla o elegirla directamente en el desplegable.*

*Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.*

* *Imagen que contiene Diagrama

  El contenido generado por IA puede ser incorrecto.Una vez seleccionada, directamente se puede indicar el lugar donde se quiere realizar la reserva y se desplegará un menú de selección, primero se debe de seleccionar la hora de inicio y luego la hora de fin. Los datos se autocompletarán en el formulario indicando el precio final de la reserva. En caso de que el rango de fechas esté ocupado saldrá marcado en rojo y no podrá elegirse*

*Imagen que contiene Gráfico

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.*

* *Por último, solo quedará confirmar la reserva. Las nuevas*

*Reservas crearán automáticamente una factura, en caso de que*

*el cliente tenga una factura pendiente de pago, esta reserva,*

*se vinculará directamente a esa factura pendiente.*

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

LISTA RESERVAS Y FACTURAS

La mecánica de las facturas y reservas es parecida a la de los clientes, otras tablas que cargan directamente los datos, pueden ser filtrados y manipulados en sus ventanas respectivamente, continua el manual:

*La lista de reservas es un elemento muy importante para controlar todas las reservas que se han realizado, introduciendo el DNI del usuario directamente puedes filtrar para encontrar reservas fácilmente, esto está estrechamente vinculado a las facturas, dado que puedes visualizar el nº de factura a la que está agregada la reserva. Sabiendo el número de factura puedes buscarla luego en el menú de facturas.*

*Interfaz de usuario gráfica, Aplicación, Tabla

El contenido generado por IA puede ser incorrecto.*

*Al igual que los usuarios haciendo doble clic en una reserva se desplegará una ventana nueva con las propiedades de la reserva.*

*Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.Su manejo es más sencillo, puedes modificar el asiento, la fecha y hora de inicio y fin, y el cliente, introduciendo manualmente los datos y luego haciendo clic en modificar.*

*También puedes eliminar directamente la reserva.*

***OJO****, las reservas* ***NO*** *pueden modificarse o eliminarse si la factura a la que están vinculadas está abonada.*

*El menú de facturas es muy parecido al de reservas, pero la búsqueda se hace por el número de facturas, por eso es importante, primero identificar el cliente, sus reservas y luego la factura a la que pertenece, una vez identificada una factura, haciendo doble clic puede visualizarse.*

*Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.*

* *Interfaz de usuario gráfica

  El contenido generado por IA puede ser incorrecto.En la ventana de la factura abierta dispone los datos del cliente, indicará la fecha de la factura y la hora de emisión.*
* *Podrás visualizar directamente las reservas vinculadas y el importe de cada una. Además, puedes modificar un descuento en porcentaje, introduciéndolo se reducirá del precio final de la factura.*

*Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.*

* *Puedes eliminar la factura del todo si esta no ha sido abonada, pero esto eliminará todas las reservas vinculadas a la factura.*

**

* *Para pagar una factura existen dos métodos de pago, en efectivo o por tarjeta, selecciona la forma de pago y por último haz clic en pagar, recuerda que esto no es reversible.*

*Interfaz de usuario gráfica, Tabla

El contenido generado por IA puede ser incorrecto.Imagen de la pantalla de un celular con letras

El contenido generado por IA puede ser incorrecto.*

* *Por último puedes imprimir la factura en formato pdf, para ello presiona el botón y automáticamente se abrirá en pantalla.*

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

La factura se genera a partir de un Excel ya estructurado, la aplicación lo abre, según la factura elegida, encontrará el DNI y los datos del cliente, los integrará en cada campo y luego irá añadiendo cada reserva, también calculará la base e IGIC de la factura y su descuento y fecha de emisión de la factura.

Texto

El contenido generado por IA puede ser incorrecto.

Esto es un ejemplo de la factura una vez emitida, aparecen todas las reservas, los datos de la empresa, y el cliente.

USUARIOS

La administración de usuarios es exclusivamente de los usuarios con acceso de admin (administrador), pueden visualizar los datos de cada usuario, pudiendo crear nuevos o eliminar alguno existente, ojo, si un usuario olvida su contraseña, la manera de recuperar sus datos es borrando su usuario y creándolo de nuevo por un administrador

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

# Código

Se mostrará la estructura del código para un mejor entendimiento, las funciones o códigos están comentados para su documentación y no se explicarán aparte porque sería redundante.

Lo primero la conexión a la base de datos **ConectionDB**:

## ConectionDB.java

package ConexionDB;  
  
import Clases.\*;  
  
import java.math.BigDecimal;  
import java.sql.\*;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
  
*/\*\*  
 \* Clase que establece la conexión con la base de datos.  
 \*/*public class ConectionDB {  
 // private static final String URL = "jdbc:mysql://bd-4free.net";  
 private static final String *URL* = "jdbc:mysql://bt33sWO8fH9pN8LVVDGr@bwj6lss5tgchvux8qwv9-mysql.services.clever-cloud.com:3306/bwj6lss5tgchvux8qwv9";  
 private static final String *PORT* = "3306";  
 //private static final String USER = "worktopia";  
 private static final String *USER* = "uqxj2bxqdgq4e1fp";  
 //private static final String PASSWORD = "iesrincon";  
 private static final String *PASSWORD* = "bt33sWO8fH9pN8LVVDGr";  
 private static final String *DATABASE* = "worktopiaDB";  
 private static Statement *stmt*;  
 private static Connection *conn*;  
 private static ResultSet *rs*;  
  
 */\*\*  
 \* Método que abre la conexión con la base de datos.  
 \*/* public static void openConn() throws ClassNotFoundException {  
 // Hay que asegurar que ejecuta el DRIVER de MySQL  
 Class.*forName*("com.mysql.cj.jdbc.Driver");  
 try {  
  
 } catch (Exception ex) {  
  
 }  
  
 try {  
 String sUrl = *URL* + ":" + *PORT* + "/" + *DATABASE* + "?zeroDateTimeBehavior=convertToNull";  
 *conn* = DriverManager.*getConnection*("jdbc:mysql://uqxj2bxqdgq4e1fp:bt33sWO8fH9pN8LVVDGr@bwj6lss5tgchvux8qwv9-mysql.services.clever-cloud.com:3306/bwj6lss5tgchvux8qwv9", *USER*, *PASSWORD*);  
 System.*out*.println("Connected to: " + sUrl);  
 } catch (SQLException e) {  
 e.printStackTrace();  
 return;  
 }  
  
 try {  
 *stmt* = *conn*.createStatement(ResultSet.*TYPE\_SCROLL\_INSENSITIVE*, ResultSet.*CONCUR\_READ\_ONLY*);  
 } catch (SQLException ex) {  
 ex.printStackTrace();  
 }  
 }  
  
 public static Connection getConn() {  
 return *conn*;  
 }  
  
 */\*\*  
 \* Método que cierra la conexión con la base de datos.  
 \*/* public static void closeConn() {  
 try {  
 if (*conn* != null) {  
 *conn*.close();  
 }  
 } catch (SQLException ex) {  
 Logger.*getLogger*(ConectionDB.class.getName()).log(Level.*SEVERE*, null, ex);  
 }  
 }  
  
  
 public static Statement getStmt() {  
 return *stmt*;  
 }  
  
  
 */\*\*  
 \* Método que adquiere una lista de clientes.  
 \*  
 \* @return Lista de clientes.  
 \*/* public static List<Clientes> getClientes() {  
 List<Clientes> clientes = new ArrayList<>();  
 String query = "SELECT dni, nombre, primerApellido, segundoApellido, email, telefono FROM Clientes";  
 try {  
 if (*conn* != null) {  
 ResultSet rs = *stmt*.executeQuery(query);  
 while (rs.next()) {  
 Clientes cliente = new Clientes(  
 rs.getString("dni"),  
 rs.getString("nombre"),  
 rs.getString("primerApellido"),  
 rs.getString("segundoApellido"),  
 rs.getString("email"),  
 rs.getString("telefono")  
 );  
 clientes.add(cliente);  
 }  
  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return clientes;  
 }  
  
 */\*\*  
 \* Método que adquiere una lista de asientos.  
 \*  
 \* @return Lista de asientos.  
 \*/* public static List<Asientos> getAsientos() {  
 List<Asientos> asientos = new ArrayList<>();  
 String query = "SELECT id\_asiento, nombre, tarifa\_hora FROM Asientos";  
 try {  
 if (*conn* != null) {  
 ResultSet rs = *stmt*.executeQuery(query);  
 while (rs.next()) {  
 Asientos asiento = new Asientos(  
 rs.getInt("id\_asiento"),  
 rs.getString("nombre"),  
 rs.getBigDecimal("tarifa\_hora")  
  
 );  
 asientos.add(asiento);  
 }  
 rs.close();  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return asientos;  
 }  
  
  
 */\*\*  
 \* Método que adquiere una lista de reservas.  
 \* Además, calcula el subtotal de cada reserva.  
 \*  
 \* @return Lista de reservas.  
 \*/* public static List<Reservas> getReservas() {  
 List<Reservas> reservas = new ArrayList<>();  
 String query = "SELECT id\_reserva, dni, id\_asiento, id\_factura, fecha\_hora\_inicio, fecha\_hora\_fin , subtotal FROM Reservas";  
 try (PreparedStatement ps = *conn*.prepareStatement(query);  
 ResultSet rs = ps.executeQuery()) {  
  
 while (rs.next()) {  
 Reservas reserva = new Reservas(  
 rs.getInt("id\_reserva"),  
 rs.getString("dni"),  
 rs.getInt("id\_asiento"),  
 rs.getInt("id\_factura"),  
 rs.getTimestamp("fecha\_hora\_inicio"),  
 rs.getTimestamp("fecha\_hora\_fin"),  
 rs.getBigDecimal("subtotal")  
 );  
 BigDecimal subtotal = reserva.calcularSubtotal();  
 reserva.setSubtotal(subtotal);  
 reservas.add(reserva);  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return reservas;  
 }  
  
  
 */\*\*  
 \* Método que adquiere una lista de facturas.  
 \*  
 \* @return Lista de facturas.  
 \*/* public static List<Facturas> getFacturas() {  
 List<Facturas> facturas = new ArrayList<>();  
 String query = "SELECT id\_factura, dni, precio\_total, descuento, fecha\_hora\_emision, estado, fecha\_hora\_pago, forma\_pago , subtotal FROM Facturas";  
 try {  
 if (*conn* != null) {  
 ResultSet rs = *stmt*.executeQuery(query);  
 while (rs.next()) {  
 Facturas factura = new Facturas(  
 rs.getInt("id\_factura"),  
 rs.getString("dni"),  
 rs.getBigDecimal("precio\_total"),  
 rs.getBigDecimal("descuento"),  
 rs.getString("fecha\_hora\_emision"),  
 rs.getString("estado"),  
 rs.getString("fecha\_hora\_pago"),  
 rs.getString("forma\_pago"),  
 rs.getBigDecimal("subtotal")  
 );  
 facturas.add(factura);  
 }  
 rs.close();  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return facturas;  
 }  
  
  
 */\*\*  
 \* Método que adquiere una lista de usuarios.  
 \*  
 \* @return Lista de usuarios.  
 \*/* public static List<Usuarios> getUsuarios() {  
 List<Usuarios> usuarios = new ArrayList<>();  
 String query = "SELECT id\_usuario, nombre, email, contrasenia, categoria FROM Usuarios";  
 try {  
 if (*conn* != null) {  
 ResultSet rs = *stmt*.executeQuery(query);  
 while (rs.next()) {  
 Usuarios usuario = new Usuarios(  
 rs.getInt("id\_usuario"),  
 rs.getString("nombre"),  
 rs.getString("email"),  
 rs.getString("contrasenia"),  
 rs.getString("categoria")  
 );  
 usuarios.add(usuario);  
 }  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return usuarios;  
 }  
  
}

Este código está diseñado para que se conecte a la base de datos, también posee la función de desconexión de este. Además, adquiere los datos de la base de datos de cada tabla. Cada función está comentada para la documentación y mejor entendimiento.

Ahora pasamos a **MenuPrincipalApp**, este arrancará la aplicación, establecerá la conexión desde la clase **ConectionDB** e iniciará la ventana **MenuPrincipal**. Para un mejor diseño, se ha establecido directamente unas proporciones al programa anulando que se pueda maximizar.

## MenuPrincipalApp.java

*/\*\*  
 \* Worktopia, Aplicación de Coworking  
 \* <p>  
 \* Creadores de la aplicación:  
 \*  
 \* @author David Liaño Macías  
 \* @author Eliu Manuel Viera Lorenzo  
 \* @version 1.0  
 \*/*package Aplicaciones;  
  
  
import ConexionDB.ConectionDB;  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
import javafx.scene.image.Image;  
  
*/\*\*  
 \* Clase principal para la aplicación.  
 \*/*

public class MenuPrincipalApp extends Application {  
 private static final Image *icono* = new Image(MenuPrincipalApp.class.getResourceAsStream("/Imagenes/bannerTopiaC.png"));  
  
 @Override  
 public void start(Stage menu) throws IOException {  
 FXMLLoader menuPrincipal = new FXMLLoader(MenuPrincipalApp.class.getResource("/Menus/MenuPrincipal.fxml"));  
 menu.getIcons().add(*icono*);  
 menu.setTitle("Menu Principal");  
 menu.setScene(new Scene(menuPrincipal.load()));  
 menu.getMaxHeight();  
 menu.setMaxWidth(1300);  
 menu.maximizedProperty().addListener((obs, oldVal, isMaximized) -> {  
 if (isMaximized) {  
 menu.setFullScreen(true);  
 }  
 });  
 menu.show();  
 }  
  
 public static void agregarIcono(Stage ventana) {  
 ventana.getIcons().add(*icono*);  
  
 }  
  
 public static void main(String[] args) throws ClassNotFoundException {  
 ConectionDB.*openConn*();  
 *launch*();  
 Runtime.*getRuntime*().addShutdownHook(new Thread(() -> {  
 ConectionDB.*closeConn*();  
 System.*out*.println("Connection closed.");  
 }));  
 }  
}

Ahora cada clase, además de sus atributos, las clases de reserva y factura tienen funciones para calcular los precios.

## Asientos.java, clientes,java, facturas.java, Reservas.java, Usuarios.java, SesionUsuario.java

package Clases;  
  
import java.math.BigDecimal;  
  
*/\*\*  
 \* Clase Asientos.  
 \*/*public class Asientos {  
  
 private int id\_asiento;  
 private String nombre;  
 private BigDecimal tarifa\_hora;  
  
  
 public Asientos(int id\_asiento, String nombre, BigDecimal tarifa\_hora) {  
 this.id\_asiento = id\_asiento;  
 this.nombre = nombre;  
 this.tarifa\_hora = tarifa\_hora;  
 }  
  
 public int getId\_asiento() {  
 return id\_asiento;  
 }  
  
 public void setId\_asiento(int id\_asiento) {  
 this.id\_asiento = id\_asiento;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public BigDecimal getTarifa\_hora() {  
 return tarifa\_hora;  
 }  
  
 public void setTarifa\_hora(BigDecimal tarifa\_hora) {  
 this.tarifa\_hora = tarifa\_hora;  
 }  
  
 @Override  
 public String toString() {  
 return "Asientos{" +  
 "id\_asiento=" + id\_asiento +  
 ", nombre='" + nombre + '\'' +  
 ", tarifa\_hora=" + tarifa\_hora +  
 '}';  
 }  
}

package Clases;  
  
*/\*\*  
 \* Clase CLientes.  
 \*/*public class Clientes {  
  
  
 private String dni;  
 private String nombre;  
 private String primerApellido;  
 private String segundoApellido;  
 private String email;  
 private String telefono;  
 private String contrasenia;  
  
  
 public Clientes(String dni, String nombre, String primerApellido, String segundoApellido, String email, String telefono) {  
  
 this.dni = dni;  
 this.nombre = nombre;  
 this.primerApellido = primerApellido;  
 this.segundoApellido = segundoApellido;  
 this.email = email;  
 this.telefono = telefono;  
 this.contrasenia = contrasenia;  
  
 }  
  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getTelefono() {  
 return telefono;  
 }  
  
 public void setTelefono(String telefono) {  
 this.telefono = telefono;  
 }  
  
  
 public String getDni() {  
 return dni;  
 }  
  
 public String getPrimerApellido() {  
 return primerApellido;  
 }  
  
 public void setPrimerApellido(String primerApellido) {  
 this.primerApellido = primerApellido;  
 }  
  
 public String getSegundoApellido() {  
 return segundoApellido;  
 }  
  
 public void setSegundoApellido(String segundoApellido) {  
 this.segundoApellido = segundoApellido;  
 }  
  
 public String getContrasenia() {  
 return contrasenia;  
 }  
  
 public void setContrasenia(String contrasenia) {  
 this.contrasenia = contrasenia;  
 }  
  
  
 @Override  
 public String toString() {  
 return dni;  
 }  
}

package Clases;  
  
import java.math.BigDecimal;  
  
*/\*\*  
 \* Clase Facturas.  
 \*/*public class Facturas {  
 private Integer id\_factura;  
 private String dni;  
 private BigDecimal precio\_total;  
 private BigDecimal descuento;  
 private String fecha\_hora\_emision;  
 private String estado;  
 private String fecha\_hora\_pago;  
 private String forma\_pago;  
 private BigDecimal subtotal;  
  
 public Facturas(Integer id\_factura, String dni, BigDecimal precio\_total, BigDecimal descuento, String fecha\_hora\_emision, String estado, String fecha\_hora\_pago, String forma\_pago, BigDecimal subtotal) {  
 this.id\_factura = id\_factura;  
 this.dni = dni;  
 this.precio\_total = precio\_total;  
 this.descuento = descuento;  
 this.fecha\_hora\_emision = fecha\_hora\_emision;  
 this.estado = estado;  
 this.fecha\_hora\_pago = fecha\_hora\_pago;  
 this.forma\_pago = forma\_pago;  
 this.subtotal = subtotal;  
  
 }  
  
 */\*\*  
 \* Clase Otro formato de construcción de Facturas por conveniencia de uso en otras funciones.  
 \*/* public Facturas(int idFactura, String fechaHoraEmision, BigDecimal totalConDescuento, String dni, BigDecimal descuento, BigDecimal totalSubtotales, String estado, String fechaHoraPago, String formaPago) {  
 this.id\_factura = idFactura;  
 this.fecha\_hora\_emision = fechaHoraEmision;  
 this.precio\_total = totalConDescuento;  
 this.dni = dni;  
 this.descuento = descuento;  
 this.subtotal = totalSubtotales;  
 this.estado = estado;  
 this.fecha\_hora\_pago = fechaHoraPago;  
 this.forma\_pago = formaPago;  
 }  
  
 public Integer getId\_factura() {  
 return id\_factura;  
 }  
  
 public void setId\_factura(Integer id\_factura) {  
 this.id\_factura = id\_factura;  
 }  
  
 public BigDecimal getPrecio\_total() {  
 return precio\_total;  
 }  
  
 public void setPrecio\_total(BigDecimal precio\_total) {  
 this.precio\_total = precio\_total;  
 }  
  
 public BigDecimal getdescuento() {  
 return descuento;  
 }  
  
 public void setTiene\_descuento(BigDecimal descuento) {  
 this.descuento = descuento;  
 }  
  
 public String getFecha\_hora\_emision() {  
 return fecha\_hora\_emision;  
 }  
  
 public void setFecha\_hora\_emision(String fecha\_hora\_emision) {  
 this.fecha\_hora\_emision = fecha\_hora\_emision;  
 }  
  
 public String getEstado() {  
 return estado;  
 }  
  
 public String getDni() {  
 return dni;  
 }  
  
 public void setDni(String dni) {  
 this.dni = dni;  
 }  
  
 public BigDecimal getDescuento() {  
 return descuento;  
 }  
  
 public void setDescuento(BigDecimal descuento) {  
 this.descuento = descuento;  
 }  
  
 public void setEstado(String estado) {  
 this.estado = estado;  
 }  
  
 public String getFecha\_hora\_pago() {  
 return fecha\_hora\_pago;  
 }  
  
 public void setFecha\_hora\_pago(String fecha\_hora\_pago) {  
 this.fecha\_hora\_pago = fecha\_hora\_pago;  
 }  
  
 public String getForma\_pago() {  
 return forma\_pago;  
 }  
  
 public void setForma\_pago(String forma\_pago) {  
 this.forma\_pago = forma\_pago;  
 }  
  
 public BigDecimal getSubtotal() {  
 return subtotal;  
 }  
  
 public void setSubtotal(BigDecimal subtotal) {  
 this.subtotal = subtotal;  
 }  
  
 @Override  
 public String toString() {  
 return "Facturas{" +  
 "id\_factura=" + id\_factura +  
 ", dni='" + dni + '\'' +  
 ", precio\_total=" + precio\_total +  
 ", descuento=" + descuento +  
 ", fecha\_hora\_emision='" + fecha\_hora\_emision + '\'' +  
 ", estado='" + estado + '\'' +  
 ", fecha\_pago='" + fecha\_hora\_pago + '\'' +  
 '}';  
 }  
  
 */\*\*  
 \* Función para calcular el total con descuento.  
 \*/* public BigDecimal calcularTotalConDescuento(BigDecimal subtotal, BigDecimal descuento) {  
 if (subtotal == null || descuento == null) {  
 throw new IllegalArgumentException("Subtotal and discount must not be null");  
 }  
 BigDecimal discountAmount = subtotal.multiply(descuento).divide(BigDecimal.*valueOf*(100));  
 BigDecimal totalConDescuento = subtotal.subtract(discountAmount);  
 return totalConDescuento.setScale(2, BigDecimal.*ROUND\_HALF\_UP*);  
 }  
  
}

package Clases;  
  
import ConexionDB.ConectionDB;  
  
import java.math.BigDecimal;  
import java.sql.Timestamp;  
import java.time.Duration;  
import java.time.LocalDateTime;  
import java.util.List;  
  
*/\*\*  
 \* Clase Reservas.  
 \*/*public class Reservas {  
 private int id\_reserva;  
 private String dni;  
 private int id\_asiento;  
 private int id\_factura;  
 private Timestamp fecha\_hora\_inicio;  
 private Timestamp fecha\_hora\_fin;  
 private BigDecimal subtotal;  
  
 public Reservas(int id\_reserva, String dni, int id\_asiento, int id\_factura, Timestamp fecha\_hora\_inicio, Timestamp fecha\_hora\_fin, BigDecimal subtotal) {  
 this.id\_reserva = id\_reserva;  
 this.dni = dni;  
 this.id\_asiento = id\_asiento;  
 this.id\_factura = id\_factura;  
 this.fecha\_hora\_inicio = fecha\_hora\_inicio;  
 this.fecha\_hora\_fin = fecha\_hora\_fin;  
 this.subtotal = subtotal;  
 }  
  
 public int getId\_reserva() {  
 return id\_reserva;  
 }  
  
 public void setId\_reserva(int id\_reserva) {  
 this.id\_reserva = id\_reserva;  
 }  
  
 public String getDni() {  
 return dni;  
 }  
  
 public void setDni(String dni) {  
 this.dni = dni;  
 }  
  
 public int getId\_asiento() {  
 return id\_asiento;  
 }  
  
 public void setId\_asiento(int id\_asiento) {  
 this.id\_asiento = id\_asiento;  
 }  
  
 public int getId\_factura() {  
 return id\_factura;  
 }  
  
 public void setId\_factura(int id\_factura) {  
 this.id\_factura = id\_factura;  
 }  
  
 public BigDecimal getSubtotal() {  
 return subtotal;  
 }  
  
 public void setSubtotal(BigDecimal subtotal) {  
 this.subtotal = subtotal;  
 }  
  
 */\*\*  
 \* Ambs funciones siguientes se usan para calcular el subtotal de la reserva.  
 \*/* public BigDecimal calcularSubtotal() {  
 BigDecimal tarifaHora = getTarifaHoraId(id\_asiento);  
 LocalDateTime inicio = fecha\_hora\_inicio.toLocalDateTime();  
 LocalDateTime fin = fecha\_hora\_fin.toLocalDateTime();  
 long horas = Duration.*between*(inicio, fin).toHours();  
 return tarifaHora.multiply(BigDecimal.*valueOf*(horas));  
 }  
  
 private BigDecimal getTarifaHoraId(int id\_asiento) {  
 List<Asientos> asientos = ConectionDB.*getAsientos*();  
 for (Asientos asiento : asientos) {  
 if (asiento.getId\_asiento() == id\_asiento) {  
 return asiento.getTarifa\_hora();  
 }  
 }  
 return BigDecimal.*ZERO*;  
 }  
  
 public Timestamp getFecha\_hora\_inicio() {  
 return fecha\_hora\_inicio;  
 }  
  
 public void setFecha\_hora\_inicio(Timestamp fecha\_hora\_inicio) {  
 this.fecha\_hora\_inicio = fecha\_hora\_inicio;  
 }  
  
 public Timestamp getFecha\_hora\_fin() {  
 return fecha\_hora\_fin;  
 }  
  
 public void setFecha\_hora\_fin(Timestamp fecha\_hora\_fin) {  
 this.fecha\_hora\_fin = fecha\_hora\_fin;  
 }  
  
 @Override  
 public String toString() {  
 return "Reservas{" +  
 "id\_reserva=" + id\_reserva +  
 ", dni='" + dni + '\'' +  
 ", id\_cliente=" + id\_factura +  
 ", id\_asiento=" + id\_asiento +  
 ", fecha\_hora\_inicio=" + fecha\_hora\_inicio +  
 ", fecha\_hora\_fin=" + fecha\_hora\_fin +  
 ", subtotal=" + subtotal +  
 '}';  
 }  
}

package Clases;  
  
*/\*\*  
 \* Clase Usuarios.  
 \*/*public class Usuarios {  
 private Integer id\_usuario;  
 private String nombre;  
 private String email;  
 private String contrasenia;  
 private String categoria;  
  
 public Usuarios(Integer id\_usuario, String nombre, String email, String contrasenia, String categoria) {  
 this.id\_usuario = id\_usuario;  
 this.nombre = nombre;  
 this.email = email;  
 this.contrasenia = contrasenia;  
 this.categoria = categoria;  
 }  
  
 public Integer getId\_usuario() {  
 return id\_usuario;  
 }  
  
 public void setId\_usuario(Integer id\_usuario) {  
 this.id\_usuario = id\_usuario;  
 }  
  
 public String getNombre() {  
 return nombre;  
 }  
  
 public void setNombre(String nombre) {  
 this.nombre = nombre;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public String getContrasenia() {  
 return contrasenia;  
 }  
  
 public void setContrasenia(String contrasenia) {  
 this.contrasenia = contrasenia;  
 }  
  
 public String getCategoria() {  
 return categoria;  
 }  
  
 public void setCategoria(String categoria) {  
 this.categoria = categoria;  
 }  
  
 @Override  
 public String toString() {  
 return "Usuarios{" +  
 "id\_usuario=" + id\_usuario +  
 ", nombre='" + nombre + '\'' +  
 ", email='" + email + '\'' +  
 ", contrasenia='" + contrasenia + '\'' +  
 ", categoria='" + categoria + '\'' +  
 '}';  
 }  
}

Y por último la clase de **SesionUsuario** para identificar quien inicia la sesión con su categoría.

package Clases;  
  
*/\*\*  
 \* Clase que guarda la categoria del usuario que ha iniciado sesion.  
 \*/*public class SesionUsuario {  
  
 private static String *categoriaUsuario*;  
  
  
 public static void setCategoriaUsuario(String categoria) {  
 *categoriaUsuario* = categoria;  
 }  
  
 public static String getCategoriaUsuario() {  
 return *categoriaUsuario*;  
 }  
  
  
}

## MenuPrincipal.fxml, MenuPrincipal.java

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.\*?>  
<?import javafx.scene.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.image.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.shape.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane id="panel" prefHeight="830.0" prefWidth="1300.0" styleClass="AnchorPane" stylesheets="@../Estilos/EstilosMenuPrincipal.css" xmlns="http://javafx.com/javafx/17.0.12" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Modelos.MenuPrincipal">  
 <children>  
 <ImageView fx:id="imagenPrincipal" fitHeight="449.0" fitWidth="797.0" layoutX="252.0" layoutY="14.0" pickOnBounds="true" smooth="false">  
 <image>  
 <Image url="@../Imagenes/ImagenPrincipalLogin.png" />  
 </image>  
 </ImageView>  
 <Button id="BtnEntrar" fx:id="BtnEntrar" alignment="CENTER" layoutX="595.0" layoutY="618.0" onAction="#logeado" prefHeight="39.0" prefWidth="110.0" styleClass="BtnEntrar" stylesheets="@../Estilos/EstilosMenuPrincipal.css" text="ENTRAR" textAlignment="CENTER" textOverrun="CLIP" defaultButton="true">  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 </Button>  
 <PasswordField id="textfieldPassword" fx:id="password" alignment="TOP\_CENTER" layoutX="527.0" layoutY="548.0" prefHeight="32.0" prefWidth="245.0" promptText="CONTRASEÑA" styleClass="textfielPassword" stylesheets="@../Estilos/EstilosMenuPrincipal.css">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 <opaqueInsets>  
 <Insets />  
 </opaqueInsets>  
 </PasswordField>  
 <TextField id="textFieldUsuario" fx:id="usuario" alignment="TOP\_CENTER" layoutX="527.0" layoutY="472.0" prefHeight="32.0" prefWidth="245.0" promptText="USUARIO" styleClass="textfieldUsuario" stylesheets="@../Estilos/EstilosMenuPrincipal.css" />  
  
 <Label layoutX="560.0" layoutY="715.0" text="Proyecto Worktopia Version 1.0">  
 <font>  
 <Font name="System Bold" size="12.0" />  
 </font></Label>  
 <Label layoutX="472.0" layoutY="740.0" text="Realizado por Eliú Manuel Viera Lorenzo y David Liaño Macias">  
 <font>  
 <Font name="Arial Bold" size="12.0" />  
 </font></Label>  
 <Line endX="405.0" layoutX="645.0" layoutY="520.0" startX="-390.0" strokeWidth="2.0" />  
 <Line endX="392.0" layoutX="657.0" layoutY="595.0" startX="-401.0" strokeWidth="2.0" />  
 <Line endX="-104.0" layoutX="657.0" layoutY="723.0" startX="-400.0" strokeWidth="2.0" />  
 <Line endX="-2.0" layoutX="1049.0" layoutY="723.0" startX="-302.0" strokeWidth="2.0" />  
 <Button layoutX="507.0" layoutY="665.0" mnemonicParsing="false" prefHeight="35.0" prefWidth="285.0" styleClass="btnpassword" stylesheets="@../Estilos/EstilosMenuPrincipal.css" text="¿Se te ha olvidado la contraseña?">  
 <font>  
 <Font name="System Bold" size="16.0" />  
 </font>  
 </Button>  
 </children>  
</AnchorPane>

JavaFX permite una gran codificación para estructurar el diseño de la aplicación, como base se ha ido diseñando desde el “Scene Builder” y luego retocando y codificando en el text, esto ha permitido un mejor nivel para manipular, además se ha aplicado un estilo gracias al CSS para ajustar los tamaños de letra, fondos y colores, incluso cuando el usuario interactúa (pasa el ratón por encima) para cambiar de estilos.

*/\*\*  
 \* Clase MenuPrincipal que se encarga de la lógica de la ventana de inicio de sesión.  
 \* En esta clase se valida el usuario y contraseña ingresados por el usuario y se redirige a la ventana de lista de clientes.  
 \*/*package Modelos;  
  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.event.ActionEvent;  
import javafx.scene.control.Button;  
import javafx.scene.control.PasswordField;  
import javafx.scene.control.TextField;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
public class MenuPrincipal {  
  
 private ListaUsuarios listaUsuarios;  
 @FXML  
 private Button BtnEntrar;  
 @FXML  
 private TextField usuario;  
 @FXML  
 private PasswordField password;  
  
 public MenuPrincipal() {  
 listaUsuarios = new ListaUsuarios(this);  
 }  
  
  
 @FXML  
 public void logeado(ActionEvent event) {  
 String user = usuario.getText();  
 String pass = password.getText();  
 listaUsuarios.loginUsuarios(user, pass);  
 ((Stage) BtnEntrar.getScene().getWindow()).close();  
 }  
  
 public void nuevaVentana() {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource("/Menus/ListaClientes.fxml"));  
 Parent cargaVentana = loader.load();  
 Stage stage = new Stage();  
 stage.setScene(new Scene(cargaVentana));  
 stage.setResizable(false);  
 stage.setMaximized(false);  
 stage.show();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 }  
  
 public void cerrarVentana() {  
 ((Stage) BtnEntrar.getScene().getWindow()).close();  
  
 }  
  
  
}

### Visualización:

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

## ListaClientes.fxml, ListaClientes.java

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.\*?>  
<?import javafx.scene.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.image.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.shape.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="830.0" prefWidth="1300.0" styleClass="AnchorPane" stylesheets="@../Estilos/EstilosListas.css" xmlns="http://javafx.com/javafx/17.0.12" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Modelos.ListaClientes">  
 <children>  
 <ImageView fx:id="portada" fitHeight="117.0" fitWidth="119.0" layoutX="589.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/bannerTopiaC.png" />  
 </image></ImageView>  
 <Label layoutX="562.0" layoutY="116.0" text="Lista De Clientes">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font></Label>  
 <Line endX="707.0" layoutX="515.0" layoutY="650.0" startX="-452.0" strokeWidth="3.0" />  
 <Line endX="137.0" layoutX="403.0" layoutY="130.0" startX="-340.0" strokeWidth="3.0" />  
 <Line endX="74.0" layoutX="1145.0" layoutY="133.0" startX="-373.79998779296875" strokeWidth="3.0" />  
 <TextField id="dniBuscar" fx:id="DNIbuscar" alignment="TOP\_CENTER" layoutX="524.0" layoutY="148.0" prefHeight="32.0" prefWidth="249.0" promptText="Buscar DNI Usuario" style="-fx-border-color: black; -fx-border-width: 3;">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor></TextField>  
 <TableView fx:id="tablaClientes" layoutX="83.0" layoutY="200.0" prefHeight="430.0" prefWidth="1120.0" styleClass="TablaListas" stylesheets="@../Estilos/EstilosListas.css">  
 <columns>  
 <TableColumn fx:id="colDNI" prefWidth="130.0" text="DNI" />  
 <TableColumn fx:id="colNombre" prefWidth="161.0" text="Nombre" />  
 <TableColumn fx:id="colPrimerApellido" prefWidth="177.0" text="Primer Apellido" />  
 <TableColumn fx:id="colSegundoApellido" prefWidth="201.0" text="Segundo Apellido" />  
 <TableColumn fx:id="colEmail" prefWidth="253.0" text="Email" />  
 <TableColumn fx:id="colTelefono" prefWidth="196.0" text="Telefono" />  
  
 </columns>  
 </TableView>  
 <VBox fx:id="vboxIconos" layoutX="522.0" layoutY="250.0" prefHeight="357.0" prefWidth="348.0" spacing="2.0" styleClass="vboxDatos" stylesheets="@../Estilos/EstiloVboxListas.css" visible="false">  
 <children>  
 <Button fx:id="btnSalir" contentDisplay="GRAPHIC\_ONLY" mnemonicParsing="false" onAction="#salirVbox" prefHeight="24.0" prefWidth="28.0" styleClass="btnSalir" stylesheets="@../Estilos/EstiloVboxListas.css">  
 <VBox.margin>  
 <Insets bottom="-10.0" left="330.0" />  
 </VBox.margin>  
 <graphic>  
 <ImageView fitHeight="17.0" fitWidth="17.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/borrar.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 <Label alignment="TOP\_CENTER" contentDisplay="CENTER" prefHeight="32.0" prefWidth="281.0" text="Datos Del Cliente" textAlignment="CENTER">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font>  
 <VBox.margin>  
 <Insets bottom="-20.0" left="40.0" />  
 </VBox.margin>  
 </Label>  
 <TextField fx:id="TextDniGhost" visible="false" />  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Nombre">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextNombre" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Primer Apellido">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextPrimerApellido" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Segundo Apellido">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextSegundoApellido" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Email">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextEmail" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Telefono">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextTelefono" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Button id="eliminar" fx:id="BtnModificar" lineSpacing="3.0" mnemonicParsing="false" onAction="#modificarDatos" prefHeight="34.0" prefWidth="50.0" styleClass="Botones" stylesheets="@../Estilos/EstiloVboxListas.css" text="Modificar">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 <graphic>  
 <ImageView fitHeight="24.0" fitWidth="29.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/editar-perfil.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <VBox.margin>  
 <Insets bottom="15.0" left="240.0" top="-257.0" />  
 </VBox.margin>  
 </Button>  
 <Button id="modificar" fx:id="BtnEliminar" mnemonicParsing="false" onAction="#eliminarDatos" prefHeight="36.0" prefWidth="48.0" styleClass="Botones" stylesheets="@../Estilos/EstiloVboxListas.css" text="Eliminar">  
 <graphic>  
 <ImageView fitHeight="26.0" fitWidth="24.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/papelera-de-reciclaje(1).png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <VBox.margin>  
 <Insets bottom="15.0" left="240.0" />  
 </VBox.margin>  
 </Button>  
 <Button fx:id="BtnGenerarReserva" mnemonicParsing="false" onAction="#generarReserva" prefHeight="26.0" prefWidth="63.0" styleClass="Botones" stylesheets="@../Estilos/EstiloVboxListas.css" text="Reservar">  
 <graphic>  
 <ImageView fitHeight="26.0" fitWidth="24.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/calendario1.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <VBox.margin>  
 <Insets left="240.0" />  
 </VBox.margin>  
 </Button>  
 </children>  
 </VBox>  
 <Button fx:id="BtnAgregarCliente" layoutX="1154.0" layoutY="148.0" mnemonicParsing="false" onAction="#ventanaRegistro" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css">  
 <graphic>  
 <ImageView fitHeight="28.0" fitWidth="45.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/circuloSuma1.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 <VBox layoutX="319.0" layoutY="652.0" prefHeight="178.0" prefWidth="164.0" style="-fx-background-color: #F4A261;" />  
 <GridPane alignment="TOP\_CENTER" layoutX="318.0" layoutY="670.0" prefHeight="123.0" prefWidth="664.0">  
 <columnConstraints>  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 </columnConstraints>  
 <rowConstraints>  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 </rowConstraints>  
 <children>  
 <Button fx:id="BtnUsuarios" alignment="CENTER" contentDisplay="TOP" mnemonicParsing="false" onAction="#ventanaListaUsuarios" prefHeight="78.0" prefWidth="93.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Usuarios" GridPane.columnIndex="3">  
 <graphic>  
 <ImageView fitHeight="73.0" fitWidth="75.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/Mano.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 <Button fx:id="Facturacion" alignment="CENTER" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" onAction="#ventanaFacturaciones" prefHeight="115.0" prefWidth="93.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Facturas" GridPane.columnIndex="2">  
 <graphic>  
 <ImageView fitHeight="78.0" fitWidth="78.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/Facturacion.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="ListaReservas" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" onAction="#ventanaListaReservas" prefHeight="111.0" prefWidth="119.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Lista Reservas" GridPane.columnIndex="1">  
 <graphic>  
 <ImageView fitHeight="79.0" fitWidth="73.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/ListaReservas.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="ListaClientes" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" prefHeight="114.0" prefWidth="119.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Lista Clientes">  
 <graphic>  
 <ImageView fitHeight="79.0" fitWidth="70.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/ListaClientes.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 </Button>  
 </children>  
 </GridPane>  
 </children>  
  
</AnchorPane>

Se aprecia lo destacable de esta parte del código, al igual que las ventanas de ListaClientes y ListaFacturas es la integración de las tablas, que los modelos cargarán.

*/\*\*  
 \* Clase ListaClientes que se encarga de mostrar la lista de clientes en la tabla y de gestionar las acciones de los botones  
 \* de la interfaz de usuario.  
 \*/*package Modelos;  
  
import Aplicaciones.MenuPrincipalApp;  
import Clases.Clientes;  
import Clases.SesionUsuario;  
import ConexionDB.ConectionDB;  
import Controlador.ControladorClientes;  
import javafx.beans.property.SimpleStringProperty;  
import javafx.beans.property.StringProperty;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.collections.transformation.FilteredList;  
import javafx.collections.transformation.SortedList;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.image.Image;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
import javafx.event.ActionEvent;  
import javafx.stage.StageStyle;  
  
import java.io.IOException;  
  
  
public class ListaClientes {  
 private ControladorClientes controladorClientes = new ControladorClientes();  
 @FXML  
 private double xOffset = 0;  
 private double yOffset = 0;  
 @FXML  
 private Button AgregarClientes;  
 @FXML  
 private Button Reservas;  
 @FXML  
 private Button ListaReservas;  
 @FXML  
 private Button Facturacion;  
 @FXML  
 private Button BtnUsuarios;  
 @FXML  
 private TextField DNIbuscar;  
 @FXML  
 private TextField TextNombre;  
 @FXML  
 private TextField TextPrimerApellido;  
 @FXML  
 private TextField TextSegundoApellido;  
 @FXML  
 private TextField TextEmail;  
 @FXML  
 private TextField TextTelefono;  
 @FXML  
 private Button btnSalir;  
  
 private String dni;  
 @FXML  
 private TextField TextDniGhost;  
 @FXML  
 private VBox vboxIconos;  
 @FXML  
 private Button BtnGenerarReserva;  
  
 @FXML  
 private TableView<Clientes> tablaClientes;  
 @FXML  
 private TableColumn<Clientes, String> colDNI;  
 @FXML  
 private TableColumn<Clientes, String> colNombre;  
 @FXML  
 private TableColumn<Clientes, String> colPrimerApellido;  
 @FXML  
 private TableColumn<Clientes, String> colSegundoApellido;  
 @FXML  
 private TableColumn<Clientes, String> colEmail;  
 @FXML  
 private TableColumn<Clientes, String> colTelefono;  
  
 private static final StringProperty *seleccionDni* = new SimpleStringProperty();  
  
 */\*\*  
 \* Método initialize que se encarga de inicializar la tabla de clientes, de cargar los datos de los clientes en la tabla,  
 \* de filtrar los datos de la tabla, de mostrar los datos de un cliente seleccionado en los campos de texto y de gestionar  
 \* la sesión del usuario.  
 \* Además, se encarga de gestionar la acción de los botones de la interfaz de usuario.  
 \*/* @FXML  
 public void initialize() {  
 btnSalir.setStyle(("-fx-background-color: transparent;"));  
 TextDniGhost.setText(dni);  
 Stage ventanaSecundaria = new Stage();  
 ventanaSecundaria.getIcons().add(new Image(getClass().getResourceAsStream("/Imagenes/bannerTopiaC.png")));  
 colDNI.setCellValueFactory(new PropertyValueFactory<>("dni"));  
 colNombre.setCellValueFactory(new PropertyValueFactory<>("nombre"));  
 colPrimerApellido.setCellValueFactory(new PropertyValueFactory<>("primerApellido"));  
 colSegundoApellido.setCellValueFactory(new PropertyValueFactory<>("segundoApellido"));  
 colEmail.setCellValueFactory(new PropertyValueFactory<>("email"));  
 colTelefono.setCellValueFactory(new PropertyValueFactory<>("telefono"));  
  
  
 ObservableList<Clientes> clientesList = FXCollections.*observableArrayList*(ConectionDB.*getClientes*());  
 tablaClientes.setItems(clientesList);  
  
 FilteredList<Clientes> filtroClientes = new FilteredList<Clientes>(clientesList, p -> true);  
 DNIbuscar.textProperty().addListener((observable, oldValue, newValue) -> {  
 filtroClientes.setPredicate(Clientes -> {  
 if (newValue == null || newValue.isEmpty()) {  
 return true;  
 } else {  
 return Clientes.getDni().toLowerCase().contains(newValue.toLowerCase());  
 }  
 });  
 });  
  
 SortedList<Clientes> sortedData = new SortedList<>(filtroClientes);  
 sortedData.comparatorProperty().bind(tablaClientes.comparatorProperty());  
 tablaClientes.setItems(sortedData);  
  
  
 tablaClientes.setOnMouseClicked(event -> {  
 if (event.getClickCount() == 2) {  
 Clientes cliente = tablaClientes.getSelectionModel().getSelectedItem();  
 if (cliente != null) {  
 exportarDatos(cliente);  
 }  
 }  
 });  
  
 vboxIconos.setOnMousePressed(event -> {  
 xOffset = event.getSceneX() - vboxIconos.getLayoutX() + 250;  
 yOffset = event.getSceneY() - vboxIconos.getLayoutY() + 70;  
 });  
 vboxIconos.setOnMouseDragged(event -> {  
 vboxIconos.setLayoutX(event.getScreenX() - xOffset);  
 vboxIconos.setLayoutY(event.getScreenY() - yOffset);  
 });  
 inicioSesion();  
  
 tablaClientes.getSelectionModel().selectedItemProperty().addListener((observable, oldValue, newdni) -> {  
 if (newdni != null) {  
 *seleccionDni*.set(String.*valueOf*(newdni));  
 }  
 });  
 TextDniGhost.textProperty().bind(*seleccionDni*);  
  
  
 }  
  
 public ListaClientes() {  
 }  
  
 public static StringProperty getSelectSeleccionDni() {  
 return *seleccionDni*;  
 }  
  
 */\*\*  
 \* Método que se encarga generar una reserva de un cliente seleccionado en la tabla de clientes.  
 \*/* public void generarReserva(ActionEvent event) {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource("/Menus/Reservas.fxml"));  
 Parent root = loader.load();  
 Stage stage = new Stage();  
 stage.setScene(new Scene(root));  
 stage.setTitle("Registro Usuarios");  
 stage.initStyle(StageStyle.*UNDECORATED*);  
 MenuPrincipalApp.*agregarIcono*(stage);  
  
 stage.show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 }  
  
  
 public void exportarDatos(Clientes clientes) {  
 dni = clientes.getDni();  
 TextNombre.setText(clientes.getNombre());  
 TextPrimerApellido.setText(clientes.getPrimerApellido());  
 TextSegundoApellido.setText(clientes.getSegundoApellido());  
 TextEmail.setText(clientes.getEmail());  
 TextTelefono.setText(clientes.getTelefono());  
 vboxIconos.setVisible(true);  
 }  
  
 public void modificarDatos(ActionEvent event) {  
 controladorClientes.modificarClientes(TextNombre, TextPrimerApellido, TextSegundoApellido, TextEmail, TextTelefono, dni);  
 initialize();  
  
 }  
  
 public void eliminarDatos(ActionEvent event) {  
 controladorClientes.eliminarClientes(dni);  
 initialize();  
 }  
  
 public void inicioSesion() {  
 String categoria = SesionUsuario.*getCategoriaUsuario*();  
  
 if (categoria != null && categoria.equals("Admin")) {  
 BtnUsuarios.setVisible(true);  
 BtnUsuarios.setDisable(false);  
 } else {  
 BtnUsuarios.setVisible(false);  
 }  
 }  
  
 public void abrirVentana(String fxmlPath, String titulo) {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource(fxmlPath));  
 Parent root = loader.load();  
 Stage stage = new Stage();  
 stage.setScene(new Scene(root));  
 stage.setTitle(titulo);  
 stage.setResizable(false);  
 stage.setMaximized(false);  
 MenuPrincipalApp.*agregarIcono*(stage);  
 stage.show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void ventanaRegistro(ActionEvent event) {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource("/Menus/RegistroClientes.fxml"));  
 Parent root = loader.load();  
 Stage stage = new Stage();  
 stage.setScene(new Scene(root));  
 stage.setTitle("Registro Cliente");  
 stage.initStyle(StageStyle.*UNDECORATED*);  
 MenuPrincipalApp.*agregarIcono*(stage);  
 stage.show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
  
 public void ventanaFacturaciones(ActionEvent event) {  
 abrirVentana("/Menus/Facturacion.fxml", "Factuacion");  
 ((Stage) Facturacion.getScene().getWindow()).close();  
 }  
  
 public void ventanaListaReservas(ActionEvent event) {  
 abrirVentana("/Menus/ListaReservas.fxml", "Listas De Reservas");  
 ((Stage) ListaReservas.getScene().getWindow()).close();  
 }  
  
 public void ventanaListaUsuarios(ActionEvent event) {  
 abrirVentana("/Menus/ListaUsuarios.fxml", "Lista De Usuarios");  
 ((Stage) BtnUsuarios.getScene().getWindow()).close();  
 }  
  
 public void salirVbox(ActionEvent event) {  
 vboxIconos.setVisible(false);  
 }  
  
}

### Visualización

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

*/\*\*  
 \* Clase ControladorClientes que se encarga de modificar y eliminar los datos de los clientes  
 \* de la base de datos.  
 \* Esta clase se encarga de modificar y eliminar los datos de los clientes de la base de datos.  
 \* Para ello, se encarga de realizar las consultas necesarias a la base de datos para modificar  
 \* o eliminar los datos de los clientes.  
 \*/*package Controlador;  
  
import ConexionDB.ConectionDB;  
import javafx.scene.control.\*;  
  
  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
import java.util.Optional;  
  
public class ControladorClientes {  
  
 public void modificarClientes(TextField nombre, TextField primerApellido, TextField segundoApellido, TextField email, TextField telefono, String dni) {  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 Alert alert2 = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setContentText("Quiere modificar el Cliente con dni: " + dni);  
  
 ButtonType si = new ButtonType("Sí");  
 ButtonType no = new ButtonType("No", ButtonBar.ButtonData.*CANCEL\_CLOSE*);  
  
 alert.getButtonTypes().setAll(si, no);  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.isPresent() && result.get() == si) {  
 String query = "UPDATE Clientes SET nombre = ?, primerApellido = ?, segundoApellido = ?, email = ?, telefono = ? WHERE dni = ?";  
 try (PreparedStatement update = ConectionDB.*getConn*().prepareStatement(query)) {  
 update.setString(1, nombre.getText());  
 update.setString(2, primerApellido.getText());  
 update.setString(3, segundoApellido.getText());  
 update.setString(4, email.getText());  
 update.setString(5, telefono.getText());  
 update.setString(6, dni);  
 int actualizado = update.executeUpdate();  
 if (actualizado > 0) {  
 alert2.setContentText("Datos actualizados correctamente");  
 alert2.show();  
 }  
  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
  
 public void eliminarClientes(String dni) {  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 Alert alert2 = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setContentText("Quiere eliminar el Cliente con dni: " + dni);  
 ButtonType si = new ButtonType("Sí");  
 ButtonType no = new ButtonType("No", ButtonBar.ButtonData.*CANCEL\_CLOSE*);  
  
 alert.getButtonTypes().setAll(si, no);  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.isPresent() && result.get() == si) {  
 String query = "DELETE FROM Clientes WHERE dni = ?";  
 try (PreparedStatement delete = ConectionDB.*getConn*().prepareStatement(query)) {  
 delete.setString(1, dni);  
 int actualizado = delete.executeUpdate();  
 if (actualizado > 0) {  
 alert2.setContentText("Datos Eliminados correctamente");  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
}

## RegistroClientes.fxml, ControladorClientes.java

Este apartado se centra en la ventana emergente al hacer doble clic en un cliente.

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.\*?>  
<?import javafx.scene.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.image.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane maxHeight="460.0" maxWidth="305.0" prefHeight="450.0" prefWidth="297.0" style="-fx-background-color: #DDDEDE;" xmlns="http://javafx.com/javafx/17.0.12" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Modelos.RegistroClientes">  
 <children>  
 <VBox layoutY="2.0" prefHeight="472.0" prefWidth="299.0" spacing="3.0" styleClass="vboxDatos" stylesheets="@../Estilos/EstiloRegistroClientes.css">  
 <children>  
 <Button fx:id="btnSalir" contentDisplay="GRAPHIC\_ONLY" mnemonicParsing="false" onAction="#salirVentana" styleClass="btnSalir" stylesheets="@../Estilos/EstiloVboxListas.css">  
 <graphic>  
 <ImageView fitHeight="28.0" fitWidth="20.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/borrar.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <VBox.margin>  
 <Insets left="256.0" top="10.0" />  
 </VBox.margin>  
 </Button>  
 <Label text="Registrar Nuevo Cliente">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font>  
 <VBox.margin>  
 <Insets bottom="5.0" left="20.0" top="10.0" />  
 </VBox.margin>  
 </Label>  
 <Label text="DNI / NIF">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <TextField id="dni" fx:id="dni" alignment="CENTER" prefHeight="40.0" prefWidth="400.0" styleClass="textFieldRegistro" stylesheets="@../Estilos/EstiloRegistroClientes.css">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 <VBox.margin>  
 <Insets left="20.0" right="20.0" />  
 </VBox.margin>  
 </TextField>  
 <Label text="Nombre">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <TextField id="nombre" fx:id="nombre" alignment="CENTER" prefHeight="40.0" prefWidth="400.0" styleClass="textFieldRegistro" stylesheets="@../Estilos/EstiloRegistroClientes.css">  
 <VBox.margin>  
 <Insets left="20.0" right="20.0" />  
 </VBox.margin></TextField>  
 <Label layoutX="32.0" layoutY="126.0" text="Primer Apellido">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <TextField id="primerApellido" fx:id="primerApellido" alignment="CENTER" prefHeight="40.0" prefWidth="400.0" styleClass="textFieldRegistro" stylesheets="@../Estilos/EstiloRegistroClientes.css">  
 <VBox.margin>  
 <Insets left="20.0" right="20.0" />  
 </VBox.margin></TextField>  
 <Label text="Segundo Apellido">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <TextField id="segundoApellido" fx:id="segundoApellido" alignment="CENTER" prefHeight="40.0" prefWidth="400.0" styleClass="textFieldRegistro" stylesheets="@../Estilos/EstiloRegistroClientes.css">  
 <VBox.margin>  
 <Insets left="20.0" right="20.0" />  
 </VBox.margin></TextField>  
 <Label text="Email">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <TextField id="eMail" fx:id="eMail" alignment="CENTER" prefHeight="40.0" prefWidth="400.0" styleClass="textFieldRegistro" stylesheets="@../Estilos/EstiloRegistroClientes.css">  
 <VBox.margin>  
 <Insets left="20.0" right="20.0" />  
 </VBox.margin></TextField>  
 <Label text="Telefono Contacto">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <TextField id="telefono" fx:id="telefono" alignment="CENTER" prefHeight="40.0" prefWidth="400.0" styleClass="textFieldRegistro" stylesheets="@../Estilos/EstiloRegistroClientes.css">  
 <VBox.margin>  
 <Insets bottom="5.0" left="20.0" right="20.0" />  
 </VBox.margin></TextField>  
 <Button id="btnConfirmar" fx:id="btnConfirmar" alignment="CENTER" onAction="#guardarCliente" prefHeight="39.0" prefWidth="110.0" styleClass="Botones" stylesheets="@../Estilos/EstiloRegistroClientes.css" text="Confirmar" textAlignment="CENTER" textFill="WHITE" textOverrun="CLIP">  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 <VBox.margin>  
 <Insets bottom="5.0" left="90.0" />  
 </VBox.margin>  
 </Button>  
 <Button id="btnCancelar" fx:id="btnCancelar" alignment="CENTER" prefHeight="39.0" prefWidth="127.0" styleClass="Botones" stylesheets="@../Estilos/EstiloRegistroClientes.css" text="Limpiar Datos" textAlignment="CENTER" textFill="#fcfcfc" textOverrun="CLIP">  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 <VBox.margin>  
 <Insets bottom="10.0" left="80.0" />  
 </VBox.margin>  
 </Button>  
 </children>  
 </VBox>  
 </children>  
  
  
</AnchorPane>

*/\*\*  
 \* Clase ControladorClientes que se encarga de modificar y eliminar los datos de los clientes  
 \* de la base de datos.  
 \* Esta clase se encarga de modificar y eliminar los datos de los clientes de la base de datos.  
 \* Para ello, se encarga de realizar las consultas necesarias a la base de datos para modificar  
 \* o eliminar los datos de los clientes.  
 \*/*package Controlador;  
  
import ConexionDB.ConectionDB;  
import javafx.scene.control.\*;  
  
  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
import java.util.Optional;  
  
public class ControladorClientes {  
  
 public void modificarClientes(TextField nombre, TextField primerApellido, TextField segundoApellido, TextField email, TextField telefono, String dni) {  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 Alert alert2 = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setContentText("Quiere modificar el Cliente con dni: " + dni);  
  
 ButtonType si = new ButtonType("Sí");  
 ButtonType no = new ButtonType("No", ButtonBar.ButtonData.*CANCEL\_CLOSE*);  
  
 alert.getButtonTypes().setAll(si, no);  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.isPresent() && result.get() == si) {  
 String query = "UPDATE Clientes SET nombre = ?, primerApellido = ?, segundoApellido = ?, email = ?, telefono = ? WHERE dni = ?";  
 try (PreparedStatement update = ConectionDB.*getConn*().prepareStatement(query)) {  
 update.setString(1, nombre.getText());  
 update.setString(2, primerApellido.getText());  
 update.setString(3, segundoApellido.getText());  
 update.setString(4, email.getText());  
 update.setString(5, telefono.getText());  
 update.setString(6, dni);  
 int actualizado = update.executeUpdate();  
 if (actualizado > 0) {  
 alert2.setContentText("Datos actualizados correctamente");  
 alert2.show();  
 }  
  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
  
 public void eliminarClientes(String dni) {  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 Alert alert2 = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setContentText("Quiere eliminar el Cliente con dni: " + dni);  
 ButtonType si = new ButtonType("Sí");  
 ButtonType no = new ButtonType("No", ButtonBar.ButtonData.*CANCEL\_CLOSE*);  
  
 alert.getButtonTypes().setAll(si, no);  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.isPresent() && result.get() == si) {  
 String query = "DELETE FROM Clientes WHERE dni = ?";  
 try (PreparedStatement delete = ConectionDB.*getConn*().prepareStatement(query)) {  
 delete.setString(1, dni);  
 int actualizado = delete.executeUpdate();  
 if (actualizado > 0) {  
 alert2.setContentText("Datos Eliminados correctamente");  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
}

### Visualización

Captura de pantalla de un computador

El contenido generado por IA puede ser incorrecto.

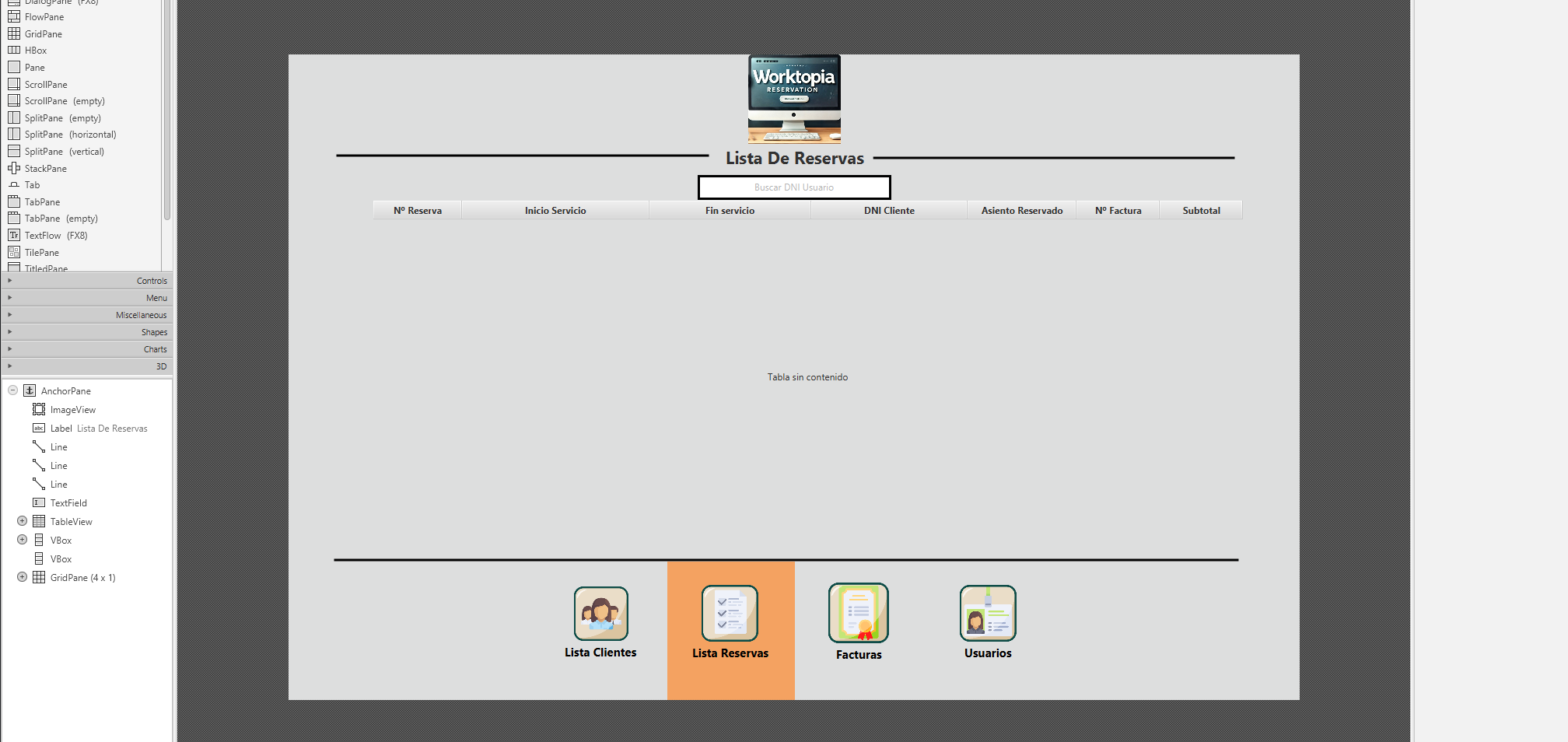
## Listareserva.fxml, listaReservas.java, controladorreservas.java

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import java.net.\*?>  
<?import javafx.geometry.\*?>  
<?import javafx.scene.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.image.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.shape.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="830.0" prefWidth="1300.0" styleClass="TablaListas" stylesheets="@../Estilos/EstilosListas.css" xmlns="http://javafx.com/javafx/17.0.12" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Modelos.ListaReservas">  
 <children>  
 <ImageView id="portada" fx:id="portadaListaReservas" fitHeight="117.0" fitWidth="119.0" layoutX="591.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/bannerTopiaC.png" />  
 </image></ImageView>  
 <Label layoutX="562.0" layoutY="117.0" text="Lista De Reservas">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font></Label>  
 <Line endX="708.0" layoutX="512.0" layoutY="650.0" startX="-452.0" strokeWidth="3.0" />  
 <Line endX="136.0" layoutX="403.0" layoutY="130.0" startX="-340.0" strokeWidth="3.0" />  
 <Line endX="88.0" layoutX="1127.0" layoutY="133.0" startX="-373.79998779296875" strokeWidth="3.0" />  
 <TextField id="dnibuscar" fx:id="DNIBuscar" alignment="TOP\_CENTER" layoutX="526.0" layoutY="155.0" prefHeight="32.0" prefWidth="249.0" promptText="Buscar DNI Usuario" style="-fx-border-color: black; -fx-border-width: 3;">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor></TextField>  
 <TableView fx:id="tablaReservas" layoutX="108.0" layoutY="187.0" prefHeight="430.0" prefWidth="1120.0" styleClass="TablaListas" stylesheets="@../Estilos/EstilosListas.css">  
 <columns>  
 <TableColumn fx:id="colNReserva" prefWidth="114.0" text="Nº Reserva" />  
 <TableColumn fx:id="colFechaInicio" prefWidth="241.0" text="Inicio Servicio" />  
 <TableColumn fx:id="colFechaFin" minWidth="8.0" prefWidth="208.0" text="Fin servicio" />  
 <TableColumn fx:id="colDNI" minWidth="0.0" prefWidth="201.0" text="DNI Cliente" />  
 <TableColumn fx:id="colAsiento" prefWidth="140.0" text="Asiento Reservado" />  
 <TableColumn fx:id="colFactura" minWidth="0.0" prefWidth="107.0" text="Nº Factura" />  
 <TableColumn fx:id="colSubtotal" maxWidth="418.0" prefWidth="107.0" text="Subtotal" />  
 </columns>  
 </TableView>  
 <VBox fx:id="vboxReservas" layoutX="522.0" layoutY="250.0" prefHeight="294.0" prefWidth="340.0" spacing="2.0" styleClass="vboxDatos" stylesheets="@../Estilos/EstiloVboxListas.css" visible="false">  
 <children>  
 <Button fx:id="btnSalir" contentDisplay="GRAPHIC\_ONLY" mnemonicParsing="false" onAction="#salirVbox" prefHeight="24.0" prefWidth="28.0" styleClass="btnSalir" stylesheets="@../Estilos/EstiloVboxListas.css">  
 <VBox.margin>  
 <Insets bottom="-10.0" left="302.0" />  
 </VBox.margin>  
 <graphic>  
 <ImageView fitHeight="17.0" fitWidth="17.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/borrar.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 <Label alignment="TOP\_CENTER" contentDisplay="CENTER" prefHeight="32.0" prefWidth="269.0" text="Datos De Reserva" textAlignment="CENTER">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font>  
 <VBox.margin>  
 <Insets left="40.0" />  
 </VBox.margin>  
 </Label>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Inicio Servicio">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextInicio" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Fin Servicio">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextFin" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Dni Cliente">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextDni" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Asiento Reservado">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextAsiento" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets bottom="20.0" left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Button id="eliminar" fx:id="BtnModificar" mnemonicParsing="false" onAction="#modificarReservas" prefHeight="36.0" prefWidth="75.0" styleClass="Botones" text="Modificar">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 <graphic>  
 <ImageView fitHeight="24.0" fitWidth="29.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/editar-perfil.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <stylesheets>  
 <URL value="@../Estilos/EstilosListas.css" />  
 <URL value="@../Estilos/EstiloVboxListas.css" />  
 </stylesheets>  
 <VBox.margin>  
 <Insets bottom="15.0" left="210.0" top="-225.0" />  
 </VBox.margin>  
 </Button>  
 <Button id="modificar" fx:id="BtnEliminar" mnemonicParsing="false" onAction="#eliminarReservas" prefHeight="36.0" prefWidth="75.0" styleClass="Botones" stylesheets="@../Estilos/EstiloVboxListas.css" text="Eliminar">  
 <graphic>  
 <ImageView fitHeight="26.0" fitWidth="24.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/papelera-de-reciclaje(1).png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <VBox.margin>  
 <Insets left="210.0" />  
 </VBox.margin>  
 </Button>  
 </children>  
 </VBox>  
 <VBox layoutX="487.0" layoutY="652.0" prefHeight="178.0" prefWidth="164.0" style="-fx-background-color: #F4A261;" />  
 <GridPane alignment="TOP\_CENTER" layoutX="318.0" layoutY="669.0" prefHeight="123.0" prefWidth="664.0">  
 <columnConstraints>  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 </columnConstraints>  
 <rowConstraints>  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 </rowConstraints>  
 <children>  
 <Button fx:id="BtnUsuarios" alignment="CENTER" contentDisplay="TOP" mnemonicParsing="false" onAction="#ventanaListaUsuarios" prefHeight="78.0" prefWidth="93.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Usuarios" GridPane.columnIndex="3">  
 <graphic>  
 <ImageView fitHeight="73.0" fitWidth="75.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/Mano.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 <Button fx:id="Facturacion" alignment="CENTER" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" onAction="#ventanaFacturaciones" prefHeight="115.0" prefWidth="93.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Facturas" GridPane.columnIndex="2">  
 <graphic>  
 <ImageView fitHeight="78.0" fitWidth="78.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/Facturacion.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="ListaReservas" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" prefHeight="111.0" prefWidth="119.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Lista Reservas" GridPane.columnIndex="1">  
 <graphic>  
 <ImageView fitHeight="79.0" fitWidth="73.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/ListaReservas.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="ListaClientes" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" onAction="#ventanaListaClientes" prefHeight="114.0" prefWidth="119.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Lista Clientes">  
 <graphic>  
 <ImageView fitHeight="79.0" fitWidth="70.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/ListaClientes.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 </Button>  
 </children>  
 </GridPane>  
 </children>  
  
  
</AnchorPane>

*/\*\*  
 \* Clase para listar las reservas  
 \*/*package Modelos;  
  
  
import Aplicaciones.MenuPrincipalApp;  
import Clases.Reservas;  
import Clases.SesionUsuario;  
import ConexionDB.ConectionDB;  
import Controlador.ControladorReservas;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.collections.transformation.FilteredList;  
import javafx.collections.transformation.SortedList;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
import java.sql.SQLException;  
import java.util.List;  
  
public class ListaReservas {  
 private ControladorReservas controladorReservas = new ControladorReservas();  
 private double xOffset = 0;  
 private double yOffset = 0;  
 private int idReserva;  
 @FXML  
 private Button AgregarClientes;  
 @FXML  
 private Button Reservas;  
 @FXML  
 private Button ListaClientes;  
 @FXML  
 private Button Facturacion;  
 @FXML  
 private Button btnModificar;  
 @FXML  
 private Button btnEliminar;  
 @FXML  
 private Button BtnUsuarios;  
 @FXML  
 private TextField DNIBuscar;  
 @FXML  
 private TextField TextInicio;  
 @FXML  
 private TextField TextFin;  
 @FXML  
 private TextField TextDni;  
 @FXML  
 private TextField TextAsiento;  
 @FXML  
 private VBox vboxReservas;  
 @FXML  
 private Button btnSalir;  
 @FXML  
 private TableView<Clases.Reservas> tablaReservas;  
 @FXML  
 private TableColumn<Clases.Reservas, Integer> colNReserva;  
 @FXML  
 private TableColumn<Clases.Reservas, String> colDNI;  
 @FXML  
 private TableColumn<Clases.Reservas, Integer> colAsiento;  
 @FXML  
 private TableColumn<Clases.Reservas, Integer> colFactura;  
 @FXML  
 private TableColumn<Clases.Reservas, java.sql.Timestamp> colFechaInicio;  
 @FXML  
 private TableColumn<Clases.Reservas, java.sql.Timestamp> colFechaFin;  
 @FXML  
 private TableColumn<Clases.Reservas, java.math.BigDecimal> colSubtotal;  
  
 */\*\*  
 \* Metodo para inicializar la ventana  
 \*/* @FXML  
 public void initialize() {  
 btnSalir.setStyle(("-fx-background-color: transparent;"));  
 colNReserva.setCellValueFactory(new PropertyValueFactory<>("id\_reserva"));  
 colDNI.setCellValueFactory(new PropertyValueFactory<>("dni"));  
 colAsiento.setCellValueFactory(new PropertyValueFactory<>("id\_asiento"));  
 colFactura.setCellValueFactory(new PropertyValueFactory<>("id\_factura"));  
 colFechaInicio.setCellValueFactory(new PropertyValueFactory<>("fecha\_hora\_inicio"));  
 colFechaFin.setCellValueFactory(new PropertyValueFactory<>("fecha\_hora\_fin"));  
 colSubtotal.setCellValueFactory(new PropertyValueFactory<>("subtotal"));  
  
  
 tablaReservas.setOnMouseClicked(event -> {  
 if (event.getClickCount() == 2) {  
 Reservas reserva = tablaReservas.getSelectionModel().getSelectedItem();  
 if (reserva != null) {  
 exportarDatos(reserva);  
 }  
 }  
 });  
  
 vboxReservas.setOnMousePressed(event -> {  
 xOffset = event.getSceneX() - vboxReservas.getLayoutX() + 250;  
 yOffset = event.getSceneY() - vboxReservas.getLayoutY() + 70;  
 });  
 vboxReservas.setOnMouseDragged(event -> {  
 vboxReservas.setLayoutX(event.getScreenX() - xOffset);  
 vboxReservas.setLayoutY(event.getScreenY() - yOffset);  
 });  
  
 try {  
 loadReservasFromDatabase();  
 } catch (SQLException |  
 ClassNotFoundException e) {  
 e.printStackTrace();  
 } finally {  
  
 }  
 inicioSesion();  
  
 }  
  
 public ListaReservas() {  
 }  
  
 */\*\*  
 \* Metodo para cargar las reservas de la base de datos  
 \*/* private void loadReservasFromDatabase() throws SQLException, ClassNotFoundException {  
 List<Reservas> reservas = ConectionDB.*getReservas*();  
 ObservableList<Clases.Reservas> reservasList = FXCollections.*observableArrayList*(reservas);  
 tablaReservas.setItems(reservasList);  
  
 FilteredList<Reservas> filtroReservas = new FilteredList<Reservas>(reservasList, p -> true);  
 DNIBuscar.textProperty().addListener((observable, oldValue, newValue) -> {  
 filtroReservas.setPredicate(Reservas -> {  
 if (newValue == null || newValue.isEmpty()) {  
 return true;  
 } else {  
 return Reservas.getDni().toLowerCase().contains(newValue.toLowerCase());  
 }  
 });  
 });  
  
 SortedList<Reservas> sortedData = new SortedList<>(filtroReservas);  
 sortedData.comparatorProperty().bind(tablaReservas.comparatorProperty());  
 tablaReservas.setItems(sortedData);  
  
  
 }  
  
 public void inicioSesion() {  
 String categoria = SesionUsuario.*getCategoriaUsuario*();  
  
 if (categoria != null && categoria.equals("Admin")) {  
 BtnUsuarios.setVisible(true);  
 BtnUsuarios.setDisable(false);  
 } else {  
 BtnUsuarios.setVisible(false);  
 }  
 }  
  
 public void exportarDatos(Reservas reservas) {  
 idReserva = reservas.getId\_reserva();  
 TextInicio.setText(String.*valueOf*(reservas.getFecha\_hora\_inicio()));  
 TextFin.setText(String.*valueOf*(reservas.getFecha\_hora\_fin()));  
 TextDni.setText(reservas.getDni());  
 TextAsiento.setText(String.*valueOf*(reservas.getId\_asiento()));  
 vboxReservas.setVisible(true);  
 }  
  
 public void modificarReservas(ActionEvent event) {  
 String forma = controladorReservas.obtenerForma(TextDni.getText());  
 if (forma.equals("Pendiente")) {  
 controladorReservas.modificarReservas(TextDni, TextInicio, TextFin, TextAsiento, idReserva);  
 } else {  
 showAlert("Aviso", "La reserva esta pagada");  
  
 }  
 System.*out*.println("forma" + forma);  
  
 initialize();  
 }  
  
 */\*\*  
 \* Metodo para eliminar reservas  
 \*/* public void eliminarReservas(ActionEvent event) {  
 String forma = controladorReservas.obtenerForma(TextDni.getText());  
 if (forma.equals("Pendiente")) {  
 controladorReservas.eliminarReservas(TextDni, idReserva);  
 } else {  
 showAlert("Aviso", "La reserva esta pagada");  
 }  
 System.*out*.println("forma" + forma);  
 initialize();  
  
 }  
  
 public void abrirVentana(String fxmlPath, String titulo) {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource(fxmlPath));  
 Parent root = loader.load();  
 Stage stage = new Stage();  
 stage.setScene(new Scene(root));  
 stage.setTitle(titulo);  
 stage.setResizable(false);  
 stage.setMaximized(false);  
 MenuPrincipalApp.*agregarIcono*(stage);  
 stage.show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void showAlert(String title, String message) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
  
 public void ventanaListaClientes(ActionEvent event) {  
 abrirVentana("/Menus/ListaClientes.fxml", "Lista de Cliente");  
 ((Stage) ListaClientes.getScene().getWindow()).close();  
 }  
  
 public void ventanaFacturaciones(ActionEvent event) {  
 abrirVentana("/Menus/Facturacion.fxml", "Factuacion");  
 ((Stage) Facturacion.getScene().getWindow()).close();  
 }  
  
 public void ventanaListaUsuarios(ActionEvent event) {  
 abrirVentana("/Menus/ListaUsuarios.fxml", "Usuarios");  
 ((Stage) BtnUsuarios.getScene().getWindow()).close();  
 }  
  
 public void salirVbox(ActionEvent event) {  
 vboxReservas.setVisible(false);  
 }  
  
}

*/\*\*  
 \* Clase para listar las reservas  
 \*/*package Modelos;  
  
  
import Aplicaciones.MenuPrincipalApp;  
import Clases.Reservas;  
import Clases.SesionUsuario;  
import ConexionDB.ConectionDB;  
import Controlador.ControladorReservas;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.collections.transformation.FilteredList;  
import javafx.collections.transformation.SortedList;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
import java.sql.SQLException;  
import java.util.List;  
  
public class ListaReservas {  
 private ControladorReservas controladorReservas = new ControladorReservas();  
 private double xOffset = 0;  
 private double yOffset = 0;  
 private int idReserva;  
 @FXML  
 private Button AgregarClientes;  
 @FXML  
 private Button Reservas;  
 @FXML  
 private Button ListaClientes;  
 @FXML  
 private Button Facturacion;  
 @FXML  
 private Button btnModificar;  
 @FXML  
 private Button btnEliminar;  
 @FXML  
 private Button BtnUsuarios;  
 @FXML  
 private TextField DNIBuscar;  
 @FXML  
 private TextField TextInicio;  
 @FXML  
 private TextField TextFin;  
 @FXML  
 private TextField TextDni;  
 @FXML  
 private TextField TextAsiento;  
 @FXML  
 private VBox vboxReservas;  
 @FXML  
 private Button btnSalir;  
 @FXML  
 private TableView<Clases.Reservas> tablaReservas;  
 @FXML  
 private TableColumn<Clases.Reservas, Integer> colNReserva;  
 @FXML  
 private TableColumn<Clases.Reservas, String> colDNI;  
 @FXML  
 private TableColumn<Clases.Reservas, Integer> colAsiento;  
 @FXML  
 private TableColumn<Clases.Reservas, Integer> colFactura;  
 @FXML  
 private TableColumn<Clases.Reservas, java.sql.Timestamp> colFechaInicio;  
 @FXML  
 private TableColumn<Clases.Reservas, java.sql.Timestamp> colFechaFin;  
 @FXML  
 private TableColumn<Clases.Reservas, java.math.BigDecimal> colSubtotal;  
  
 */\*\*  
 \* Metodo para inicializar la ventana  
 \*/* @FXML  
 public void initialize() {  
 btnSalir.setStyle(("-fx-background-color: transparent;"));  
 colNReserva.setCellValueFactory(new PropertyValueFactory<>("id\_reserva"));  
 colDNI.setCellValueFactory(new PropertyValueFactory<>("dni"));  
 colAsiento.setCellValueFactory(new PropertyValueFactory<>("id\_asiento"));  
 colFactura.setCellValueFactory(new PropertyValueFactory<>("id\_factura"));  
 colFechaInicio.setCellValueFactory(new PropertyValueFactory<>("fecha\_hora\_inicio"));  
 colFechaFin.setCellValueFactory(new PropertyValueFactory<>("fecha\_hora\_fin"));  
 colSubtotal.setCellValueFactory(new PropertyValueFactory<>("subtotal"));  
  
  
 tablaReservas.setOnMouseClicked(event -> {  
 if (event.getClickCount() == 2) {  
 Reservas reserva = tablaReservas.getSelectionModel().getSelectedItem();  
 if (reserva != null) {  
 exportarDatos(reserva);  
 }  
 }  
 });  
  
 vboxReservas.setOnMousePressed(event -> {  
 xOffset = event.getSceneX() - vboxReservas.getLayoutX() + 250;  
 yOffset = event.getSceneY() - vboxReservas.getLayoutY() + 70;  
 });  
 vboxReservas.setOnMouseDragged(event -> {  
 vboxReservas.setLayoutX(event.getScreenX() - xOffset);  
 vboxReservas.setLayoutY(event.getScreenY() - yOffset);  
 });  
  
 try {  
 loadReservasFromDatabase();  
 } catch (SQLException |  
 ClassNotFoundException e) {  
 e.printStackTrace();  
 } finally {  
  
 }  
 inicioSesion();  
  
 }  
  
 public ListaReservas() {  
 }  
  
 */\*\*  
 \* Metodo para cargar las reservas de la base de datos  
 \*/* private void loadReservasFromDatabase() throws SQLException, ClassNotFoundException {  
 List<Reservas> reservas = ConectionDB.*getReservas*();  
 ObservableList<Clases.Reservas> reservasList = FXCollections.*observableArrayList*(reservas);  
 tablaReservas.setItems(reservasList);  
  
 FilteredList<Reservas> filtroReservas = new FilteredList<Reservas>(reservasList, p -> true);  
 DNIBuscar.textProperty().addListener((observable, oldValue, newValue) -> {  
 filtroReservas.setPredicate(Reservas -> {  
 if (newValue == null || newValue.isEmpty()) {  
 return true;  
 } else {  
 return Reservas.getDni().toLowerCase().contains(newValue.toLowerCase());  
 }  
 });  
 });  
  
 SortedList<Reservas> sortedData = new SortedList<>(filtroReservas);  
 sortedData.comparatorProperty().bind(tablaReservas.comparatorProperty());  
 tablaReservas.setItems(sortedData);  
  
  
 }  
  
 public void inicioSesion() {  
 String categoria = SesionUsuario.*getCategoriaUsuario*();  
  
 if (categoria != null && categoria.equals("Admin")) {  
 BtnUsuarios.setVisible(true);  
 BtnUsuarios.setDisable(false);  
 } else {  
 BtnUsuarios.setVisible(false);  
 }  
 }  
  
 public void exportarDatos(Reservas reservas) {  
 idReserva = reservas.getId\_reserva();  
 TextInicio.setText(String.*valueOf*(reservas.getFecha\_hora\_inicio()));  
 TextFin.setText(String.*valueOf*(reservas.getFecha\_hora\_fin()));  
 TextDni.setText(reservas.getDni());  
 TextAsiento.setText(String.*valueOf*(reservas.getId\_asiento()));  
 vboxReservas.setVisible(true);  
 }  
  
 public void modificarReservas(ActionEvent event) {  
 String forma = controladorReservas.obtenerForma(TextDni.getText());  
 if (forma.equals("Pendiente")) {  
 controladorReservas.modificarReservas(TextDni, TextInicio, TextFin, TextAsiento, idReserva);  
 } else {  
 showAlert("Aviso", "La reserva esta pagada");  
  
 }  
 System.*out*.println("forma" + forma);  
  
 initialize();  
 }  
  
 */\*\*  
 \* Metodo para eliminar reservas  
 \*/* public void eliminarReservas(ActionEvent event) {  
 String forma = controladorReservas.obtenerForma(TextDni.getText());  
 if (forma.equals("Pendiente")) {  
 controladorReservas.eliminarReservas(TextDni, idReserva);  
 } else {  
 showAlert("Aviso", "La reserva esta pagada");  
 }  
 System.*out*.println("forma" + forma);  
 initialize();  
  
 }  
  
 public void abrirVentana(String fxmlPath, String titulo) {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource(fxmlPath));  
 Parent root = loader.load();  
 Stage stage = new Stage();  
 stage.setScene(new Scene(root));  
 stage.setTitle(titulo);  
 stage.setResizable(false);  
 stage.setMaximized(false);  
 MenuPrincipalApp.*agregarIcono*(stage);  
 stage.show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void showAlert(String title, String message) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
  
 public void ventanaListaClientes(ActionEvent event) {  
 abrirVentana("/Menus/ListaClientes.fxml", "Lista de Cliente");  
 ((Stage) ListaClientes.getScene().getWindow()).close();  
 }  
  
 public void ventanaFacturaciones(ActionEvent event) {  
 abrirVentana("/Menus/Facturacion.fxml", "Factuacion");  
 ((Stage) Facturacion.getScene().getWindow()).close();  
 }  
  
 public void ventanaListaUsuarios(ActionEvent event) {  
 abrirVentana("/Menus/ListaUsuarios.fxml", "Usuarios");  
 ((Stage) BtnUsuarios.getScene().getWindow()).close();  
 }  
  
 public void salirVbox(ActionEvent event) {  
 vboxReservas.setVisible(false);  
 }  
  
}

### Visualización



## RESERVAS.fxml, RESERVASPANEL.java

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.image.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.shape.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="837.0" prefWidth="1300.0" xmlns="http://javafx.com/javafx/17.0.12" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Modelos.ReservasPanel">  
 <children>  
 <ImageView fx:id="portada" fitHeight="117.0" fitWidth="119.0" layoutX="589.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/bannerTopiaC.png" />  
 </image>  
 </ImageView>  
 <Label layoutX="605.0" layoutY="115.0" text="Reservas">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font>  
 </Label>  
 <Line endX="180.0" layoutX="404.0" layoutY="130.0" startX="-340.0" strokeWidth="3.0" />  
 <Line endX="6.0" layoutX="1207.0" layoutY="131.0" startX="-490.0" strokeWidth="3.0" />  
 <Label layoutX="998.0" layoutY="267.0" text="Seleccionar Fecha">  
 <font>  
 <Font name="System Bold" size="18.0" />  
 </font>  
 </Label>  
 <Line endX="-33.0" layoutX="1249.0" layoutY="295.0" startX="-324.0" strokeWidth="3.0" />  
 <Label layoutX="1018.0" layoutY="349.0" text="Confirmación">  
 <font>  
 <Font name="System Bold" size="18.0" />  
 </font>  
 </Label>  
 <Line endX="-33.0" layoutX="1249.0" layoutY="376.0" startX="-324.0" strokeWidth="3.0" />  
 <ImageView fitHeight="604.0" fitWidth="833.0" layoutX="62.0" layoutY="146.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/salasCoworkingFondo1.png" />  
 </image>  
 </ImageView>  
 <Label layoutX="923.0" layoutY="389.0" style="-fx-font-weight: bold;" text="Espacio Reservado:">  
 <font>  
 <Font size="14.0" />  
 </font>  
 </Label>  
 <Label layoutX="973.0" layoutY="420.0" style="-fx-font-weight: bold;" text="Hora inicio:">  
 <font>  
 <Font size="14.0" />  
 </font>  
 </Label>  
 <Label layoutX="1004.0" layoutY="487.0" style="-fx-font-weight: bold;" text="Precio:">  
 <font>  
 <Font size="14.0" />  
 </font>  
 </Label>  
 <Button fx:id="btnConfirmar" alignment="CENTER" layoutX="1016.0" layoutY="549.0" onAction="#insertarReserva" prefHeight="39.0" prefWidth="110.0" styleClass="BtnRegistro" stylesheets="@../Estilos/EstiloReservas.css" text="Confirmar" textAlignment="CENTER" textOverrun="CLIP">  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 </Button>  
 <TextField id="espacio" fx:id="espacio" layoutX="1065.0" layoutY="387.0" />  
 <TextField id="horario" fx:id="horaInicio" layoutX="1065.0" layoutY="417.0" />  
 <TextField id="precio" fx:id="precio" layoutX="1066.0" layoutY="488.0" />  
 <DatePicker id="Fecha" fx:id="fecha" layoutX="960.0" layoutY="311.0" promptText="Fechas" styleClass="fechas" stylesheets="@../Estilos/EstiloReservas.css" />  
 <Button fx:id="BtnOficina1" alignment="TOP\_CENTER" layoutX="521.0" layoutY="528.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="218.0" prefWidth="193.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="Oficina 1">  
 <font>  
 <Font name="System Bold" size="18.0" />  
 </font>  
 </Button>  
 <Button id="oficina2" fx:id="BtnOficina2" alignment="TOP\_CENTER" layoutX="715.0" layoutY="528.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="218.0" prefWidth="182.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="Oficina 2">  
 <font>  
 <Font name="System Bold" size="18.0" />  
 </font>  
 </Button>  
 <Button id="BtnConferencias1" fx:id="BtnConferencias1" alignment="BOTTOM\_CENTER" contentDisplay="RIGHT" layoutX="62.0" layoutY="149.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="324.0" prefWidth="182.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="Sala de Conferencias 1" textAlignment="CENTER">  
 <font>  
 <Font name="System Bold" size="13.0" />  
 </font>  
 </Button>  
 <Button fx:id="BtnConferencias2" alignment="BOTTOM\_CENTER" layoutX="365.0" layoutY="150.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="210.0" prefWidth="203.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="Sala de Conferencias 2">  
 <font>  
 <Font name="System Bold" size="13.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa1" layoutX="108.0" layoutY="548.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A1">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa2" layoutX="143.0" layoutY="548.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="39.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A2">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa3" layoutX="108.0" layoutY="618.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A3">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa4" layoutX="143.0" layoutY="618.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="39.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A4">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa5" layoutX="108.0" layoutY="673.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A5">  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa6" layoutX="143.0" layoutY="673.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="39.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A6">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa7" layoutX="234.0" layoutY="548.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A7">  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa8" layoutX="269.0" layoutY="548.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="39.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A8">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa9" layoutX="234.0" layoutY="618.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A9">  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa10" layoutX="266.0" layoutY="618.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A10">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa11" layoutX="234.0" layoutY="673.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A11">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa12" layoutX="273.0" layoutY="673.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A12">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa13" layoutX="361.0" layoutY="548.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A13">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa14" layoutX="396.0" layoutY="548.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A14">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa15" layoutX="361.0" layoutY="618.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A15">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa16" layoutX="396.0" layoutY="618.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A16">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa17" layoutX="361.0" layoutY="673.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A17">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <Button fx:id="Mesa18" layoutX="396.0" layoutY="673.0" mnemonicParsing="false" onMouseClicked="#ventanaHorarios" prefHeight="39.0" prefWidth="45.0" styleClass="BtnOficina" stylesheets="@../Estilos/EstiloReservas.css" text="A18">  
 <font>  
 <Font name="System Bold" size="14.0" />  
 </font>  
 </Button>  
 <VBox id="contenedorHorarios" fx:id="contenedorHorarios" layoutX="561.0" layoutY="282.0" prefHeight="176.0" prefWidth="137.0" visible="false">  
 <children>  
 <GridPane prefHeight="175.0" prefWidth="137.0">  
 <columnConstraints>  
 <ColumnConstraints halignment="LEFT" hgrow="ALWAYS" maxWidth="1.7976931348623157E308" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 </columnConstraints>  
 <rowConstraints>  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 </rowConstraints>  
 <children>  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="08:00" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="09:00" GridPane.rowIndex="1" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="10:00" GridPane.rowIndex="2" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="11:00" GridPane.rowIndex="3" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="12:00" GridPane.rowIndex="4" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="13:00" GridPane.rowIndex="5" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="14:00" GridPane.rowIndex="6" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="15:00" GridPane.columnIndex="1" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="16:00" GridPane.columnIndex="1" GridPane.rowIndex="1" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="17:00" GridPane.columnIndex="1" GridPane.rowIndex="2" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="18:00" GridPane.columnIndex="1" GridPane.rowIndex="3" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="19:00" GridPane.columnIndex="1" GridPane.rowIndex="4" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="20:00" GridPane.columnIndex="1" GridPane.rowIndex="5" />  
 <Button mnemonicParsing="false" onAction="#datosTextFieldMesa" prefHeight="30.0" prefWidth="65.0" styleClass="BtnHoras" stylesheets="@../Estilos/EstiloHorario.css" text="21:00" GridPane.columnIndex="1" GridPane.rowIndex="6" />  
 </children>  
 <VBox.margin>  
 <Insets />  
 </VBox.margin>  
 </GridPane>  
 </children>  
 </VBox>  
 <Label layoutX="940.0" layoutY="455.0" style="-fx-font-weight: bold;" text="Hora finalizacion:">  
 <font>  
 <Font size="14.0" />  
 </font>  
 </Label>  
 <TextField id="horaFin" fx:id="horaFin" layoutX="1065.0" layoutY="452.0" onAction="#datosTextFieldMesa" />  
 <Label layoutX="996.0" layoutY="168.0" text="Seleccionar Cliente">  
 <font>  
 <Font name="System Bold" size="18.0" />  
 </font>  
 </Label>  
 <Line endX="-33.0" layoutX="1249.0" layoutY="197.0" startX="-324.0" strokeWidth="3.0" />  
 <TextField fx:id="dniCliente" alignment="CENTER" layoutX="949.0" layoutY="212.0" prefHeight="32.0" prefWidth="251.0" />  
 <Button fx:id="btnSalir" layoutX="1261.0" layoutY="14.0" mnemonicParsing="false" onAction="#salir" styleClass="BtnMenu" stylesheets="@../Estilos/EstiloReservas.css">  
 <graphic>  
 <ImageView fitHeight="16.0" fitWidth="16.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/borrar.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 </children>  
  
  
</AnchorPane>

*/\*\*  
 \* Clase ReservasPanel que se encarga de gestionar las reservas de los clientes  
 \* y de enviar un correo de confirmación de la reserva.  
 \* Implementa la interfaz Initializable.  
 \*/*package Modelos;  
  
import Clases.Reservas;  
import Controlador.ControladorEmail;  
import Controlador.ControladorReservas;  
import Manejadores\_Reservas\_Facturas.ReservaDAO;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.input.MouseEvent;  
import javafx.scene.control.Alert;  
import javafx.scene.control.Button;  
import javafx.scene.control.DatePicker;  
import javafx.scene.control.TextField;  
import javafx.scene.image.Image;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
  
import java.math.BigDecimal;  
import java.sql.\*;  
import java.time.LocalDate;  
import java.time.LocalDateTime;  
import java.time.LocalTime;  
import java.time.format.DateTimeFormatter;  
  
import java.util.Set;  
  
public class ReservasPanel {  
 private ControladorReservas controladorReservas = new ControladorReservas();  
 private ControladorEmail controladorEmail = new ControladorEmail();  
 private ListaClientes listaClientes = new ListaClientes();  
 @FXML  
 private Button btnSalir;  
 @FXML  
 private TextField espacio;  
 @FXML  
 private TextField horaInicio;  
 @FXML  
 private TextField horaFin;  
 @FXML  
 private TextField precio;  
 @FXML  
 private DatePicker fecha;  
 @FXML  
 private VBox contenedorHorarios;  
 @FXML  
 private TextField dniCliente;  
  
 private Button BtnSeleccionado;  
 private double precioTotal;  
 private BigDecimal subtotal;  
  
 @FXML  
 public void initialize() {  
 Stage ventanaSecundaria = new Stage();  
 ventanaSecundaria.getIcons().add(new Image(getClass().getResourceAsStream("/Imagenes/bannerTopiaC.png")));  
 dniCliente.textProperty().addListener((observableValue, s, t1) -> {  
 controladorReservas.buscarDni(dniCliente.getText(), dniCliente);  
 });  
 dniCliente.textProperty().bindBidirectional(listaClientes.*getSelectSeleccionDni*());  
  
 }  
  
 public ReservasPanel() {  
 }  
  
 */\*\*  
 \* Inserta una reserva en la base de datos y envía un correo de confirmación al cliente.  
 \*/* public void insertarReserva(ActionEvent event) {  
  
 String emailCliente = controladorReservas.obtenerEmailCliente(dniCliente.getText());  
 String nombreCliente = controladorReservas.obtenerNombreCliente(dniCliente.getText());  
 //Formateo de la fecha a fecha europea  
 LocalDate fechas = fecha.getValue();  
 DateTimeFormatter formateo = DateTimeFormatter.*ofPattern*("dd/MM/yyyy");  
 String fechaFormateada = fechas.format(formateo);  
  
 if (emailCliente != null && !emailCliente.isEmpty()) {  
 String htmlPlantilla = "src/main/resources/html/reservasHTML.html";  
 String mensaje = controladorEmail.cargarPlantilla(  
 htmlPlantilla,  
 nombreCliente,  
 fechaFormateada,  
 horaInicio.getText().toString(),  
 horaFin.getText().toString(),  
 espacio.getText().toString(),  
 subtotal.toString()  
 );  
 controladorEmail.*enviarCorreo*(emailCliente, "Confirmación de reserva", mensaje);  
 }  
 System.*out*.println("idreserva" + fecha);  
  
 insertarDatos();  
  
 }  
  
 */\*\*  
 \* Muestra los datos de la mesa seleccionada en los campos de texto.  
 \*/* public void datosTextFieldMesa(ActionEvent event) {  
 Button BtnSeleccionHora = (Button) event.getSource();  
  
 if (BtnSeleccionHora != null) {  
 espacio.setText(BtnSeleccionado.getText());  
 if (horaInicio.getText().isEmpty()) {  
 horaInicio.setText(BtnSeleccionHora.getText());  
 } else if (horaFin.getText().isEmpty()) {  
 horaFin.setText(BtnSeleccionHora.getText());  
 cierreVentanaHorarios();  
 }  
 }  
 precioTotal = controladorReservas.facturaPrecioText(espacio, horaInicio, horaFin);  
 precio.setText(String.*format*("%.2f €", precioTotal));  
 subtotal = BigDecimal.*valueOf*(precioTotal);  
  
  
 }  
 */\*\*  
 \* Muestra la ventana de horarios al hacer clic en un botón de espacio.  
 \*/* public void ventanaHorarios(MouseEvent event) {  
 Button BtnSeleccion = (Button) event.getSource();  
  
 if (contenedorHorarios.isVisible() && BtnSeleccionado == BtnSeleccion) {  
 cierreVentanaHorarios();  
 return;  
 }  
  
 if (BtnSeleccionado != null && BtnSeleccionado != BtnSeleccion) {  
 cierreVentanaHorarios();  
 }  
  
 BtnSeleccionado = BtnSeleccion;  
 limpiarCamposHorarios();  
 contenedorHorarios.setVisible(true);  
  
 int espacioId = controladorReservas.obtenerIdEspacio(BtnSeleccionado.getText());  
 actualizarColoresHorarios(espacioId);  
  
 posicionarVentanaEnRaton(event);  
 }  
  
 private void limpiarCamposHorarios() {  
 horaInicio.setText("");  
 horaFin.setText("");  
 precio.setText("");  
 }  
  
 */\*\*  
 \* Posiciona la ventana de horarios en la posición del ratón.  
 \*/* private void posicionarVentanaEnRaton(MouseEvent event) {

// Obtener coordenadas del ratón relativas al layout donde está la imagen  
double x = event.getSceneX() - imagenLayout.localToScene(0, 0).getX();  
double y = event.getSceneY() - imagenLayout.localToScene(0, 0).getY();  
  
// Asegurar que el contenedorHorarios no salga de los límites de la imagen  
double maxX = 833 - contenedorHorarios.getWidth();  
double maxY = 604 - contenedorHorarios.getHeight();  
  
// Limitar coordenadas para que no se salgan del área de la imagen  
x = Math.*max*(0, Math.*min*(x, maxX));  
y = Math.*max*(0, Math.*min*(y, maxY));  
  
// Ajustar la posición dentro del layout que contiene la imagen  
contenedorHorarios.setLayoutX(62 + x);  
contenedorHorarios.setLayoutY(146 + y);

}  
  
 public void cierreVentanaHorarios() {  
 contenedorHorarios.setVisible(false);  
 BtnSeleccionado = null;  
 }  
  
 */\*\*  
 \* Actualiza los colores de los botones de horarios según si están ocupados o no.  
 \*  
 \* @param idEspacio ID del espacio seleccionado.  
 \*/* public void actualizarColoresHorarios(int idEspacio) {  
 LocalDate fechaSeleccionada = fecha.getValue();  
 if (fechaSeleccionada == null) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle("Error");  
 alert.setHeaderText(null);  
 alert.setContentText("Debe seleccionar una fecha.");  
 alert.showAndWait();  
 contenedorHorarios.setVisible(false);  
 return;  
 }  
  
 Set<String> horariosOcupados = controladorReservas.obtenerHorariosReservados(idEspacio, fechaSeleccionada);  
  
 for (javafx.scene.Node node : contenedorHorarios.lookupAll(".button")) {  
 if (node instanceof Button) {  
 Button btnHorario = (Button) node;  
 String horario = btnHorario.getText();  
  
 if (horariosOcupados.contains(horario)) {  
 btnHorario.setStyle("-fx-background-color: #e60415; -fx-text-fill: white;");  
 btnHorario.setDisable(true);  
 } else {  
 btnHorario.setStyle("-fx-background-color: #047f07; -fx-text-fill: black;");  
 btnHorario.setDisable(false);  
 }  
 }  
 }  
 }  
  
 */\*\*  
 \* Inserta los datos de la reserva en la base de datos.  
 \* Muestra un mensaje de error si no se han completado todos los campos.  
 \*/* public void insertarDatos() {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
  
 String dniText = dniCliente.getText();  
 String espacioText = espacio.getText();  
 String horaInicioText = horaInicio.getText();  
 String horaFinText = horaFin.getText();  
 LocalDate fechaReserva = fecha.getValue();  
 BigDecimal subTotal = subtotal;  
 int idAsiento = controladorReservas.obtenerIdEspacio(espacioText);  
  
 if (dniText.isEmpty() || espacioText.isEmpty() || horaInicioText.isEmpty() || horaFinText.isEmpty()) {  
 alert.setContentText("Debe completar todos los campos.");  
 alert.show();  
 return;  
 }  
  
 LocalDateTime fechaHoraInicio = LocalDateTime.*of*(fechaReserva, LocalTime.*parse*(horaInicioText));  
 LocalDateTime fechaHoraFin = LocalDateTime.*of*(fechaReserva, LocalTime.*parse*(horaFinText));  
  
 try {  
 int idFactura = ReservaDAO.*obtenerFacturaPendiente*(dniText);  
 if (idFactura == -1) {  
 idFactura = ReservaDAO.*crearFactura*(dniText, subTotal);  
 }  
  
 Reservas reserva = new Reservas(0, dniText, idAsiento, idFactura, Timestamp.*valueOf*(fechaHoraInicio), Timestamp.*valueOf*(fechaHoraFin), subTotal);  
 int idReserva = ReservaDAO.*insertarReserva*(reserva);  
  
 alert.setContentText("Reserva y factura creadas correctamente. ID Reserva: " + idReserva);  
 alert.show();  
  
 } catch (SQLException e) {  
 alert.setContentText("Error en la base de datos: " + e.getMessage());  
 alert.show();  
 }  
 }  
  
  
 public void salir(ActionEvent event) {  
 ((Stage) btnSalir.getScene().getWindow()).close();  
 }  
  
}

### visualización

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

## Facturación.fxml, facturación.java, controladorFacturas.java, ReservaDAO.java

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import java.net.\*?>  
<?import javafx.geometry.\*?>  
<?import javafx.scene.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.image.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.shape.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="830.0" prefWidth="1300.0" styleClass="AnchorPane" stylesheets="@../Estilos/EstilosListas.css" xmlns="http://javafx.com/javafx/17.0.12" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Modelos.Facturacion">  
 <children>  
 <ImageView id="portada" fx:id="portadaListaReservas" fitHeight="117.0" fitWidth="119.0" layoutX="589.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/bannerTopiaC.png" />  
 </image>  
 </ImageView>  
 <Label layoutX="578.0" layoutY="113.0" text="Facturaciones">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font>  
 </Label>  
 <Line endX="690.0" layoutX="520.0" layoutY="650.0" startX="-452.0" strokeWidth="3.0" />  
 <Line endX="155.0" layoutX="403.0" layoutY="130.0" startX="-340.0" strokeWidth="3.0" />  
 <Line endX="-5.0" layoutX="1209.0" layoutY="131.0" startX="-470.0" strokeWidth="3.0" />  
 <TextField fx:id="facturaBuscar" alignment="TOP\_CENTER" layoutX="515.0" layoutY="153.0" prefHeight="32.0" prefWidth="249.0" promptText="Nº Factura" style="-fx-border-color: black; -fx-border-width: 3;">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 </TextField>  
 <VBox fx:id="vboxFacturas" layoutX="440.0" layoutY="229.0" prefHeight="401.0" prefWidth="627.0" spacing="10.0" styleClass="vboxDatos" stylesheets="@../Estilos/EstiloVboxListas.css" visible="false">  
 <padding>  
 <Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />  
 </padding>  
 <children>  
 <Label alignment="TOP\_CENTER" contentDisplay="CENTER" prefHeight="32.0" prefWidth="522.0" text="Datos De Factura" textAlignment="CENTER">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font>  
 <VBox.margin>  
 <Insets bottom="-40.0" left="40.0" />  
 </VBox.margin>  
 </Label>  
 <VBox prefHeight="233.0" prefWidth="128.0" spacing="30.0">  
 <children>  
 <Button fx:id="BtnModificar" mnemonicParsing="false" onAction="#modificarFactura" prefHeight="36.0" prefWidth="75.0" styleClass="Botones" text="Modificar">  
 <graphic>  
 <ImageView fitHeight="24.0" fitWidth="29.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/editar-perfil.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <stylesheets>  
 <URL value="@../Estilos/EstilosListas.css" />  
 <URL value="@../Estilos/EstiloVboxListas.css" />  
 </stylesheets>  
 </Button>  
 <Button fx:id="BtnEliminar" mnemonicParsing="false" onAction="#eliminarFactura" prefHeight="36.0" prefWidth="75.0" styleClass="Botones" stylesheets="@../Estilos/EstiloVboxListas.css" text="Eliminar">  
 <graphic>  
 <ImageView fitHeight="26.0" fitWidth="24.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/papelera-de-reciclaje(1).png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 <Button fx:id="BtnPagarFactura" mnemonicParsing="false" onAction="#pagarFactura" prefHeight="36.0" prefWidth="75.0" styleClass="Botones" stylesheets="@../Estilos/EstiloVboxListas.css" text="Pagar">  
 <graphic>  
 <ImageView fitHeight="30.0" fitWidth="24.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/dolar1.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 <Button fx:id="BtnGenerarFactura" mnemonicParsing="false" onAction="#generarFactura" prefHeight="36.0" prefWidth="75.0" styleClass="Botones" stylesheets="@../Estilos/EstiloVboxListas.css" text="Generar PDF">  
 <graphic>  
 <ImageView fitHeight="51.0" fitWidth="25.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/factura1.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 </children>  
 <VBox.margin>  
 <Insets left="470.0" top="79.0" />  
 </VBox.margin>  
 </VBox>  
 <Button fx:id="btnSalir" contentDisplay="GRAPHIC\_ONLY" mnemonicParsing="false" onAction="#salirVbox" prefHeight="24.0" prefWidth="28.0" styleClass="btnSalir" stylesheets="@../Estilos/EstiloVboxListas.css">  
 <VBox.margin>  
 <Insets left="560.0" right="10.0" top="-315.0" />  
 </VBox.margin>  
 <graphic>  
 <ImageView fitHeight="17.0" fitWidth="17.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/borrar.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 <HBox prefHeight="669.0" prefWidth="615.0" spacing="3.0">  
 <children>  
 <VBox spacing="10.0" HBox.hgrow="ALWAYS">  
 <padding>  
 <Insets bottom="10.0" left="10.0" right="5.0" top="16.0" />  
 </padding>  
 <children>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Fecha Factura">  
 <VBox.margin>  
 <Insets bottom="0.0" left="10.0" right="0.0" top="0.0" />  
 </VBox.margin>  
 </Label>  
 <TextField fx:id="TextFechaFactura" prefHeight="35.0" prefWidth="180.0">  
 <VBox.margin>  
 <Insets left="10.0" right="20.0" />  
 </VBox.margin>  
 </TextField>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Dni Cliente">  
 <VBox.margin>  
 <Insets bottom="0.0" left="10.0" right="0.0" top="0.0" />  
 </VBox.margin>  
 </Label>  
 <TextField fx:id="TextDniCliente" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="10.0" right="20.0" />  
 </VBox.margin>  
 </TextField>  
 <HBox spacing="10.0">  
 <padding>  
 <Insets bottom="0.0" left="10.0" right="0.0" top="0.0" />  
 </padding>  
 <children>  
 <VBox spacing="5.0">  
 <children>  
 <Label alignment="CENTER" styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Subtotal" />  
 <TextField fx:id="TextSubtotal" editable="false" prefHeight="30.0" prefWidth="100.0" />  
 </children>  
 </VBox>  
 <VBox spacing="5.0">  
 <children>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Descuento" />  
 <TextField fx:id="TextDescuento" prefHeight="30.0" prefWidth="122.0">  
 <VBox.margin>  
 <Insets right="20.0" />  
 </VBox.margin>  
 </TextField>  
 </children>  
 </VBox>  
 </children>  
 </HBox>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Total">  
 <VBox.margin>  
 <Insets left="10.0" />  
 </VBox.margin>  
 </Label>  
 <TextField fx:id="TextTotal" editable="false" prefHeight="30.0" prefWidth="165.0">  
 <VBox.margin>  
 <Insets left="10.0" right="100.0" />  
 </VBox.margin>  
 </TextField>  
 <Label contentDisplay="CENTER" prefHeight="17.0" prefWidth="120.0" styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Forma de pago">  
 <VBox.margin>  
 <Insets bottom="0.0" left="10.0" right="0.0" top="0.0" />  
 </VBox.margin>  
 </Label>  
 <ComboBox fx:id="comboFormaPago" onAction="#llenarComboBox" prefHeight="30.0" prefWidth="150.0" visibleRowCount="2">  
 <VBox.margin>  
 <Insets bottom="0.0" left="10.0" right="100.0" top="0.0" />  
 </VBox.margin>  
 </ComboBox>  
 </children>  
 <HBox.margin>  
 <Insets top="42.0" />  
 </HBox.margin>  
 </VBox>  
 <VBox alignment="TOP\_CENTER" maxHeight="346.0" prefHeight="337.0" prefWidth="321.0" spacing="10.0">  
 <padding>  
 <Insets bottom="10.0" left="10.0" right="10.0" top="10.0" />  
 </padding>  
 <children>  
 <Label styleClass="labels" stylesheets="@../Estilos/EstiloVboxListas.css" text="Reservas de Factura">  
 <VBox.margin>  
 <Insets bottom="0.0" left="10.0" right="0.0" top="5.0" />  
 </VBox.margin>  
 </Label>  
 <ListView fx:id="listaReservas" prefHeight="287.0" prefWidth="213.0" styleClass="ListaReservas" stylesheets="@../Estilos/EstilosListas.css">  
 <VBox.margin>  
 <Insets />  
 </VBox.margin>  
 </ListView>  
 </children>  
 <opaqueInsets>  
 <Insets />  
 </opaqueInsets>  
 <HBox.margin>  
 <Insets top="42.0" />  
 </HBox.margin>  
 </VBox>  
 </children>  
 <VBox.margin>  
 <Insets right="157.0" top="-57.0" />  
 </VBox.margin>  
 </HBox>  
 </children>  
 </VBox>  
 <VBox layoutX="651.0" layoutY="652.0" prefHeight="178.0" prefWidth="164.0" style="-fx-background-color: #F4A261;" />  
 <GridPane alignment="TOP\_CENTER" layoutX="318.0" layoutY="669.0" prefHeight="123.0" prefWidth="664.0">  
 <columnConstraints>  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 </columnConstraints>  
 <rowConstraints>  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 </rowConstraints>  
 <children>  
 <Button fx:id="BtnUsuarios" alignment="CENTER" contentDisplay="TOP" mnemonicParsing="false" onAction="#ventanaUsuarios" prefHeight="78.0" prefWidth="93.0" styleClass="BtnMenu" text="Usuarios" GridPane.columnIndex="3">  
 <graphic>  
 <ImageView fitHeight="73.0" fitWidth="75.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/Mano.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <stylesheets>  
 <URL value="@../Estilos/EstiloReservas.css" />  
 <URL value="@../Estilos/EstilosListas.css" />  
 </stylesheets>  
 </Button>  
 <Button fx:id="Facturacion" alignment="CENTER" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" prefHeight="115.0" prefWidth="93.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Facturas" GridPane.columnIndex="2">  
 <graphic>  
 <ImageView fitHeight="78.0" fitWidth="78.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/Facturacion.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="ListaReservas" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" onAction="#ventanaListaReservas" prefHeight="111.0" prefWidth="119.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Lista Reservas" GridPane.columnIndex="1">  
 <graphic>  
 <ImageView fitHeight="79.0" fitWidth="73.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/ListaReservas.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="ListaClientes" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" onAction="#ventanaListaClientes" prefHeight="114.0" prefWidth="119.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Lista Clientes">  
 <graphic>  
 <ImageView fitHeight="79.0" fitWidth="70.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/ListaClientes.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 </Button>  
 </children>  
 </GridPane>  
 <TableView fx:id="tablaFacturas" layoutX="75.0" layoutY="196.0" prefHeight="439.0" prefWidth="1126.0" styleClass="TablaListas" stylesheets="@../Estilos/EstilosListas.css">  
 <columns>  
 <TableColumn fx:id="colNFactura" prefWidth="87.0" text="Nº Factura" />  
 <TableColumn fx:id="colFechaFactura" prefWidth="187.0" text="Fecha Factura" />  
 <TableColumn fx:id="colDni" prefWidth="166.0" text="DNI Cliente" />  
 <TableColumn fx:id="colSubtotal" prefWidth="85.0" text="Subtotal" />  
 <TableColumn fx:id="colDescuento" prefWidth="101.0" text="Descuento" />  
 <TableColumn fx:id="colTotal" prefWidth="85.0" text="Total" />  
 <TableColumn fx:id="colEstadoPago" prefWidth="123.0" text="Estado del pago" />  
 <TableColumn fx:id="colFechaPago" prefWidth="174.0" text="Fecha Pago" />  
 <TableColumn fx:id="colFormaPago" prefWidth="117.0" text="Forma de pago" />  
 </columns>  
 </TableView>  
 </children>  
 <opaqueInsets>  
 <Insets />  
 </opaqueInsets>  
  
</AnchorPane>

*/\*\*  
 \* Clase Facturacion  
 \* Clase que se encarga de la lógica de la ventana de facturación.  
 \* Se encarga de mostrar las facturas, generar facturas en Excel y PDF, y de abonar facturas.  
 \*/*package Modelos;  
  
import Aplicaciones.MenuPrincipalApp;  
import Clases.\*;  
import Controlador.ControladorFacturas;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.collections.transformation.FilteredList;  
import javafx.collections.transformation.SortedList;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.control.Button;  
import javafx.scene.control.TextField;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.image.Image;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
import ConexionDB.ConectionDB;  
import org.apache.poi.ss.usermodel.\*;  
import org.apache.poi.ss.usermodel.Cell;  
import org.apache.poi.xssf.usermodel.XSSFWorkbook;  
  
import java.awt.\*;  
import java.io.\*;  
import java.math.BigDecimal;  
import java.sql.\*;  
import java.io.IOException;  
import java.time.Duration;  
import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.\*;  
import java.util.List;  
  
  
public class Facturacion {  
 private double xOffset = 0;  
 private double yOffset = 0;  
 private ControladorFacturas controladorFacturas = new ControladorFacturas();  
 private Reservas reservas;  
 @FXML  
 private TextField TextDniCliente;  
 @FXML  
 private TextField TextSubtotal;  
 @FXML  
 private TextField TextDescuento;  
 @FXML  
 private TextField TextTotal;  
 @FXML  
 private TextField TextFechaFactura;  
 @FXML  
 private Button btnSalir;  
 @FXML  
 private Button ListaReservas;  
 @FXML  
 private Button ListaClientes;  
 @FXML  
 private Button BtnUsuarios;  
 @FXML  
 private TextField facturaBuscar;  
 @FXML  
 private VBox vboxFacturas;  
 @FXML  
 private ComboBox<String> comboFormaPago;  
  
 ObservableList<String> formasPago = FXCollections.*observableArrayList*("Tarjeta", "Efectivo");  
  
 private int idFactura;  
 @FXML  
 private TableView<Facturas> tablaFacturas;  
 @FXML  
 private TableColumn<Facturas, String> colNFactura;  
 @FXML  
 private TableColumn<Facturas, String> colFechaFactura;  
 @FXML  
 private TableColumn<Facturas, String> colTotal;  
 @FXML  
 private TableColumn<Facturas, String> colDni;  
 @FXML  
 private TableColumn<Facturas, String> colDescuento;  
 @FXML  
 private TableColumn<Facturas, String> colEstadoPago;  
 @FXML  
 private TableColumn<Facturas, String> colFechaPago;  
 @FXML  
 private TableColumn<Facturas, String> colFormaPago;  
 @FXML  
 private TableColumn<Facturas, String> colSubtotal;  
 @FXML  
 private ListView<String> listaReservas;  
  
  
 @FXML  
 public void generarFactura(ActionEvent event) {  
 if (idFactura == 0) {  
 showAlert("Error", "Debe seleccionar una factura.");  
 return;  
 }  
  
 try {  
 generarFacturaExcel();  
 showAlert("Éxito", "Factura generada correctamente.");  
 } catch (Exception e) {  
 showAlert("Error", "Error al generar la factura: " + e.getMessage());  
 }  
 }  
  
 */\*\*  
 \* Método que se ejecuta al iniciar la ventana de facturación.  
 \* Se encarga de cargar las facturas en la tabla de facturas.  
 \* Además, se encarga de filtrar las facturas por su ID.  
 \* También se encarga de abrir la ventana de facturación al hacer doble clic en una factura.  
 \* Por último, se encarga de mover la ventana de facturación.  
 \*/* @FXML  
 public void initialize() {  
 llenarComboBoxInicio();  
 btnSalir.setStyle(("-fx-background-color: transparent;"));  
 Stage ventanaSecundaria = new Stage();  
 ventanaSecundaria.getIcons().add(new Image(getClass().getResourceAsStream("/Imagenes/bannerTopiaC.png")));  
 colNFactura.setCellValueFactory(new PropertyValueFactory<>("id\_factura"));  
 colFechaFactura.setCellValueFactory(new PropertyValueFactory<>("fecha\_hora\_emision"));  
 colTotal.setCellValueFactory(new PropertyValueFactory<>("precio\_total"));  
 colDni.setCellValueFactory(new PropertyValueFactory<>("dni"));  
 colDescuento.setCellValueFactory(new PropertyValueFactory<>("descuento"));  
 colSubtotal.setCellValueFactory(new PropertyValueFactory<>("subtotal"));  
 colEstadoPago.setCellValueFactory(new PropertyValueFactory<>("estado"));  
 colFechaPago.setCellValueFactory(new PropertyValueFactory<>("fecha\_hora\_pago"));  
 colFormaPago.setCellValueFactory(new PropertyValueFactory<>("forma\_pago"));  
  
 ObservableList<Facturas> facturasList = FXCollections.*observableArrayList*();  
  
 Connection conn = null;  
 try {  
 conn = ConectionDB.*getConn*();  
 String queryFacturas = "SELECT \* FROM Facturas";  
  
 try (PreparedStatement psFacturas = conn.prepareStatement(queryFacturas);  
 ResultSet rsFacturas = psFacturas.executeQuery()) {  
  
 while (rsFacturas.next()) {  
 int idFactura = rsFacturas.getInt("id\_factura");  
 BigDecimal descuento = rsFacturas.getBigDecimal("descuento");  
  
 BigDecimal totalSubtotales = cargarReservasEnFactura(conn, idFactura);  
 BigDecimal totalConDescuento = totalSubtotales.subtract(totalSubtotales.multiply(descuento.divide(BigDecimal.*valueOf*(100))));  
 totalConDescuento = totalConDescuento.setScale(2, BigDecimal.*ROUND\_HALF\_UP*);  
  
 Facturas factura = new Facturas(  
 idFactura,  
 rsFacturas.getString("fecha\_hora\_emision"),  
 totalConDescuento,  
 rsFacturas.getString("dni"),  
 descuento,  
 totalSubtotales,  
 rsFacturas.getString("estado"),  
 rsFacturas.getString("fecha\_hora\_pago"),  
 rsFacturas.getString("forma\_pago")  
 );  
 facturasList.add(factura);  
 }  
 }  
 } catch (SQLException | ClassNotFoundException e) {  
 showAlert("Error", "No se pudieron cargar las facturas: " + e.getMessage());  
 } finally {  
  
 }  
  
 tablaFacturas.setItems(facturasList);  
 inicioSesion();  
  
 FilteredList<Facturas> filtroFacturas = new FilteredList<>(facturasList, p -> true);  
 facturaBuscar.textProperty().addListener((observable, oldValue, newValue) -> {  
 filtroFacturas.setPredicate(Facturas -> {  
 if (newValue == null || newValue.isEmpty()) {  
 return true;  
 } else {  
 return Facturas.getId\_factura().toString().contains(newValue.toLowerCase());  
 }  
 });  
 });  
  
 SortedList<Facturas> sortedData = new SortedList<>(filtroFacturas);  
 sortedData.comparatorProperty().bind(tablaFacturas.comparatorProperty());  
 tablaFacturas.setItems(sortedData);  
  
 tablaFacturas.setOnMouseClicked(event -> {  
 if (event.getClickCount() == 2) {  
 Facturas factura = tablaFacturas.getSelectionModel().getSelectedItem();  
 if (factura != null) {  
 try {  
 exportarFacturas(factura);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
 });  
  
 TextDescuento.textProperty().addListener((observableValue, s, t1) -> actualizarTotal());  
  
 vboxFacturas.setOnMousePressed(event -> {  
 xOffset = event.getSceneX() - vboxFacturas.getLayoutX() + 250;  
 yOffset = event.getSceneY() - vboxFacturas.getLayoutY() + 70;  
 });  
 vboxFacturas.setOnMouseDragged(event -> {  
 vboxFacturas.setLayoutX(event.getScreenX() - xOffset);  
 vboxFacturas.setLayoutY(event.getScreenY() - yOffset);  
 });  
  
 tablaFacturas.setOnMouseClicked(event -> {  
 if (event.getClickCount() == 2) {  
 Facturas factura = tablaFacturas.getSelectionModel().getSelectedItem();  
 if (factura != null) {  
 idFactura = factura.getId\_factura(); // Almacena la ID de la factura  
 try {  
 exportarFacturas(factura);  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
 });  
 }  
  
 public Facturacion() {  
 }  
  
 public BigDecimal cargarReservasEnFactura(Connection conn, int idFactura) throws SQLException, ClassNotFoundException {  
 List<Reservas> reservas = new ArrayList<>();  
 String queryReservas = "SELECT \* FROM Reservas WHERE id\_factura = ?";  
 BigDecimal totalSubtotales = BigDecimal.*ZERO*;  
  
 try (PreparedStatement ps = conn.prepareStatement(queryReservas)) {  
 ps.setInt(1, idFactura); // Establecemos el idFactura en el PreparedStatement  
  
 try (ResultSet rs = ps.executeQuery()) {  
 while (rs.next()) {  
 int idReserva = rs.getInt("id\_reserva");  
 String dni = rs.getString("dni");  
 int idAsiento = rs.getInt("id\_asiento");  
 Timestamp fechaInicio = rs.getTimestamp("fecha\_hora\_inicio");  
 Timestamp fechaFin = rs.getTimestamp("fecha\_hora\_fin");  
  
 BigDecimal subtotal = calcularSubtotal(idAsiento, fechaInicio, fechaFin);  
 totalSubtotales = totalSubtotales.add(subtotal);  
  
 Reservas reserva = new Reservas(  
 idReserva,  
 dni,  
 idAsiento,  
 idFactura,  
 fechaInicio,  
 fechaFin,  
 subtotal  
 );  
 reservas.add(reserva);  
 }  
 }  
 }  
  
 ObservableList<String> reservasList = FXCollections.*observableArrayList*();  
 for (Reservas r : reservas) {  
 reservasList.add("Nº Reserva: " + r.getId\_reserva() + " Precio: " + r.getSubtotal());  
 }  
  
 listaReservas.setItems(reservasList);  
 return totalSubtotales;  
 }  
  
 @FXML  
 public void salirVbox() {  
 System.*out*.println("Limpiando facturaBuscar");  
 facturaBuscar.clear();  
 vboxFacturas.setVisible(false);  
 }  
  
  
 public static void rellenarCombo(ComboBox<String> comboBox, ObservableList<String> forma) {  
 comboBox.setItems(forma);  
 }  
  
 public void llenarComboBox(ActionEvent event) {  
 *rellenarCombo*(comboFormaPago, formasPago);  
 }  
  
 public void llenarComboBoxInicio() {  
 *rellenarCombo*(comboFormaPago, formasPago);  
 }  
  
  
 public void inicioSesion() {  
 String categoria = SesionUsuario.*getCategoriaUsuario*();  
  
 if (categoria != null && categoria.equals("Admin")) {  
 BtnUsuarios.setVisible(true);  
 BtnUsuarios.setDisable(false);  
 } else {  
 BtnUsuarios.setVisible(false);  
 }  
 }  
  
 */\*\*  
 \* Método que se encarga de exportar las facturas a la ventana de facturación.  
 \* Se encarga de mostrar la información de la factura seleccionada.  
 \*  
 \* @param facturas Factura seleccionada  
 \* @throws ClassNotFoundException Excepción de clase no encontrada  
 \*/* public void exportarFacturas(Facturas facturas) throws ClassNotFoundException {  
 idFactura = facturas.getId\_factura();  
 TextFechaFactura.setText(facturas.getFecha\_hora\_emision());  
 TextDniCliente.setText(facturas.getDni());  
 TextDescuento.setText(facturas.getDescuento().toString());  
 comboFormaPago.getSelectionModel().select(facturas.getForma\_pago());  
  
 try {  
 BigDecimal totalSubtotales = cargarReservasEnFactura(idFactura);  
 TextSubtotal.setText(totalSubtotales.toString());  
 double descuento = Double.*parseDouble*(TextDescuento.getText());  
 double subTotal = totalSubtotales.doubleValue();  
 TextTotal.setText(String.*valueOf*(descuento(descuento, subTotal)));  
 } catch (SQLException | ClassNotFoundException e) {  
 showAlert("Error", "No se pudieron cargar las reservas: " + e.getMessage());  
 }  
  
 vboxFacturas.setVisible(true);  
 }  
  
 public double descuento(double descuentoText, double subtotal) {  
 if (descuentoText > 0) {  
 return subtotal - (subtotal \* descuentoText / 100);  
 }  
 return subtotal;  
 }  
  
 private void actualizarTotal() {  
 try {  
 double descuentoValor = validarNumero(TextDescuento.getText(), 0.00);  
 double subtotal = validarNumero(TextSubtotal.getText(), 0.00);  
 double total = descuento(descuentoValor, subtotal);  
  
 TextTotal.setText(String.*valueOf*(total));  
 } catch (Exception e) {  
 TextTotal.setText("Error");  
 }  
 }  
  
 private double validarNumero(String texto, double valorPorDefecto) {  
 if (texto == null || texto.trim().isEmpty()) {  
 return valorPorDefecto;  
 }  
 try {  
 return Double.*parseDouble*(texto);  
 } catch (NumberFormatException e) {  
 return valorPorDefecto;  
 }  
 }  
  
 */\*\*  
 \* Método que se encarga cargar las reservas en la factura.  
 \*  
 \*/* public BigDecimal cargarReservasEnFactura(int idFactura) throws SQLException, ClassNotFoundException {  
 List<Reservas> reservas = new ArrayList<>();  
 String queryReservas = "SELECT \* FROM Reservas WHERE id\_factura = ?";  
 BigDecimal totalSubtotales = BigDecimal.*ZERO*;  
  
 Connection conn = ConectionDB.*getConn*();  
 try (PreparedStatement ps = conn.prepareStatement(queryReservas)) {  
 ps.setInt(1, idFactura);  
  
 try (ResultSet rs = ps.executeQuery()) {  
 while (rs.next()) {  
 int idReserva = rs.getInt("id\_reserva");  
 String dni = rs.getString("dni");  
 int idAsiento = rs.getInt("id\_asiento");  
 Timestamp fechaInicio = rs.getTimestamp("fecha\_hora\_inicio");  
 Timestamp fechaFin = rs.getTimestamp("fecha\_hora\_fin");  
  
 BigDecimal subtotal = calcularSubtotal(idAsiento, fechaInicio, fechaFin);  
 totalSubtotales = totalSubtotales.add(subtotal);  
  
 Reservas reserva = new Reservas(  
 idReserva,  
 dni,  
 idAsiento,  
 idFactura,  
 fechaInicio,  
 fechaFin,  
 subtotal  
 );  
 reservas.add(reserva);  
 }  
 }  
 }  
  
 ObservableList<String> reservasList = FXCollections.*observableArrayList*();  
 for (Reservas r : reservas) {  
 reservasList.add("Nº Reserva: " + r.getId\_reserva() + " Precio: " + r.getSubtotal());  
 }  
  
 listaReservas.setItems(reservasList);  
 return totalSubtotales;  
 }  
  
 public void modificarFactura(ActionEvent event) {  
 String forma = controladorFacturas.obtenerForma(idFactura).toLowerCase();  
 if (forma.equals("pendiente")) {  
 controladorFacturas.modificarFactura(TextFechaFactura, TextDescuento, TextTotal, TextSubtotal, idFactura);  
 } else {  
 showAlert("aviso", "La factura se encuentra pagada");  
 }  
 System.*out*.println("forma" + forma);  
 initialize();  
 }  
  
 public void eliminarFactura(ActionEvent event) {  
 String forma = controladorFacturas.obtenerForma(idFactura).toLowerCase();  
 if (forma.equals("pendiente")) {  
 controladorFacturas.eliminarFactura(idFactura);  
 } else {  
 showAlert("aviso", "La factura se encuentra pagada");  
 }  
 System.*out*.println("forma" + forma);  
 initialize();  
  
  
 }  
  
 public void pagarFactura(ActionEvent event) {  
 controladorFacturas.abonarFactura(TextFechaFactura, TextDescuento, TextTotal, TextSubtotal, comboFormaPago, idFactura);  
 initialize();  
  
  
 }  
  
 */\*\*  
 \* Método que se encarga de generar la factura en Excel y PDF.  
 \* Se encarga de obtener la información de la factura, cliente y reservas.  
 \* Además, se encarga de calcular el subtotal de las reservas.  
 \* Por último, se encarga de convertir el archivo Excel a PDF.  
 \* @throws SQLException  
 \* @throws IOException  
 \* @throws ClassNotFoundException  
 \*/* private void generarFacturaExcel() throws SQLException, IOException, ClassNotFoundException {  
 String queryFactura = "SELECT \* FROM Facturas WHERE id\_factura = ?";  
 String queryCliente = "SELECT \* FROM Clientes WHERE dni = ?";  
 String queryReservas = "SELECT \* FROM Reservas WHERE id\_factura = ?";  
  
 Connection conn = ConectionDB.*getConn*(); // Obtén la conexión sin cerrarla  
 try (PreparedStatement psFactura = conn.prepareStatement(queryFactura);  
 PreparedStatement psCliente = conn.prepareStatement(queryCliente);  
 PreparedStatement psReservas = conn.prepareStatement(queryReservas)) {  
  
 psFactura.setInt(1, idFactura);  
 try (ResultSet rsFactura = psFactura.executeQuery()) {  
 if (!rsFactura.next()) {  
 throw new SQLException("Factura no encontrada.");  
 }  
  
 String dniCliente = rsFactura.getString("dni");  
 LocalDateTime fechaEmision = rsFactura.getTimestamp("fecha\_hora\_emision").toLocalDateTime();  
 BigDecimal totalFactura = rsFactura.getBigDecimal("precio\_total");  
 BigDecimal descuento = rsFactura.getBigDecimal("descuento");  
  
 psCliente.setString(1, dniCliente);  
 try (ResultSet rsCliente = psCliente.executeQuery()) {  
 if (!rsCliente.next()) {  
 throw new SQLException("Cliente no encontrado.");  
 }  
  
 String nombreCliente = rsCliente.getString("nombre");  
 String telefonoCliente = rsCliente.getString("telefono");  
 String emailCliente = rsCliente.getString("email");  
  
 psReservas.setInt(1, idFactura);  
 try (ResultSet rsReservas = psReservas.executeQuery()) {  
 List<String> reservas = new ArrayList<>();  
 List<BigDecimal> subtotales = new ArrayList<>();  
 BigDecimal totalSubtotales = BigDecimal.*ZERO*;  
  
 while (rsReservas.next()) {  
 int idReserva = rsReservas.getInt("id\_reserva");  
 int idAsiento = rsReservas.getInt("id\_asiento");  
 Timestamp inicio = rsReservas.getTimestamp("fecha\_hora\_inicio");  
 Timestamp fin = rsReservas.getTimestamp("fecha\_hora\_fin");  
  
 BigDecimal subtotal = calcularSubtotal(idAsiento, inicio, fin);  
 totalSubtotales = totalSubtotales.add(subtotal);  
  
 String reserva = String.*format*(  
 "Reserva ID: %d Asiento ID: %d fecha y hora de inicio %s y fin %s",  
 idReserva, idAsiento, inicio.toLocalDateTime(), fin.toLocalDateTime()  
 );  
 reservas.add(reserva);  
 subtotales.add(subtotal);  
 }  
  
 try (FileInputStream fis = new FileInputStream("src/main/resources/Docs/Modelo\_factura\_excel.xlsx");  
 Workbook workbook = new XSSFWorkbook(fis)) {  
  
 Sheet sheet = workbook.getSheetAt(0);  
  
 setCellValue(sheet, 9, 1, String.*valueOf*(idFactura));  
 setCellValue(sheet, 15, 1, dniCliente);  
 setCellValue(sheet, 16, 1, fechaEmision.format(DateTimeFormatter.*ofPattern*("yyyy-MM-dd HH:mm:ss")));  
 setCellValue(sheet, 15, 5, nombreCliente);  
 setCellValue(sheet, 16, 5, telefonoCliente);  
 setCellValue(sheet, 14, 5, emailCliente);  
  
 int rowIndex = 17;  
 for (int i = 0; i < reservas.size(); i++) {  
 Row row = sheet.getRow(rowIndex);  
 if (row == null) {  
 row = sheet.createRow(rowIndex);  
 }  
  
 Cell cell0 = row.getCell(0);  
 if (cell0 == null) {  
 cell0 = row.createCell(0);  
 }  
 cell0.setCellValue(reservas.get(i));  
  
 Cell cell8 = row.getCell(8);  
 if (cell8 == null) {  
 cell8 = row.createCell(8);  
 }  
 cell8.setCellValue(subtotales.get(i).doubleValue());  
  
 rowIndex++;  
 }  
  
 BigDecimal totalConDescuento = totalSubtotales.subtract(totalSubtotales.multiply(descuento.divide(BigDecimal.*valueOf*(100))));  
 BigDecimal baseImponible = totalConDescuento.divide(BigDecimal.*valueOf*(1.07), BigDecimal.*ROUND\_HALF\_UP*);  
 BigDecimal impuestos = baseImponible.multiply(BigDecimal.*valueOf*(0.07));  
  
 setCellValue(sheet, 45, 6, descuento.doubleValue() / 100);  
 BigDecimal valorDescuento = totalSubtotales.multiply(descuento.divide(BigDecimal.*valueOf*(100)));  
 setCellValue(sheet, 45, 7, valorDescuento.doubleValue());  
 setCellValue(sheet, 46, 7, totalConDescuento.doubleValue());  
 setCellValue(sheet, 43, 7, baseImponible.doubleValue());  
 setCellValue(sheet, 44, 7, impuestos.doubleValue());  
 setCellValue(sheet, 47, 7, fechaEmision.format(DateTimeFormatter.*ofPattern*("yyyy-MM-dd HH:mm:ss")));  
  
 String excelFilePath = "src/main/resources/Docs/Factura\_" + idFactura + ".xlsx";  
 try (FileOutputStream fos = new FileOutputStream(excelFilePath)) {  
 workbook.write(fos);  
 }  
  
 convertirApdf(excelFilePath);  
 }  
 }  
 }  
 }  
 }  
 }  
  
 private void convertirApdf(String archivoExcel) throws IOException {  
 String libreOfficePath = "C:\\Program Files\\LibreOffice\\program\\soffice.exe";  
 String comando = libreOfficePath + " --headless --convert-to pdf " + archivoExcel + " --outdir " + new File(archivoExcel).getParent();  
  
 try {  
 Process process = Runtime.*getRuntime*().exec(comando);  
 process.waitFor();  
 System.*out*.println("Archivo convertido exitosamente a PDF.");  
  
 String pdfFilePath = archivoExcel.replace(".xlsx", ".pdf");  
  
 File pdfFile = new File(pdfFilePath);  
 if (pdfFile.exists()) {  
 if (System.*getProperty*("os.name").toLowerCase().contains("win")) {  
 Desktop.*getDesktop*().open(pdfFile);  
 } else if (System.*getProperty*("os.name").toLowerCase().contains("mac")) {  
 Runtime.*getRuntime*().exec("open " + pdfFilePath);  
 } else {  
 Runtime.*getRuntime*().exec("xdg-open " + pdfFilePath);  
 }  
 } else {  
 System.*out*.println("El archivo PDF no se ha generado correctamente.");  
 }  
 } catch (IOException | InterruptedException e) {  
 e.printStackTrace();  
 throw new IOException("Error al convertir el archivo a PDF.");  
 }  
 }  
  
  
 public BigDecimal calcularSubtotal(int idAsiento, Timestamp fechaInicio, Timestamp fechaFin) {  
 BigDecimal tarifaHora = getTarifaHoraId(idAsiento);  
 LocalDateTime inicio = fechaInicio.toLocalDateTime();  
 LocalDateTime fin = fechaFin.toLocalDateTime();  
 long horas = Duration.*between*(inicio, fin).toHours();  
 return tarifaHora.multiply(BigDecimal.*valueOf*(horas));  
 }  
  
 private static Map<Integer, BigDecimal> *tarifaHoraCache* = new HashMap<>();  
  
 private BigDecimal getTarifaHoraId(int id\_asiento) {  
 if (!*tarifaHoraCache*.containsKey(id\_asiento)) {  
 List<Asientos> asientos = ConectionDB.*getAsientos*();  
 for (Asientos asiento : asientos) {  
 if (asiento.getId\_asiento() == id\_asiento) {  
 *tarifaHoraCache*.put(id\_asiento, asiento.getTarifa\_hora());  
 return asiento.getTarifa\_hora();  
 }  
 }  
 }  
 return *tarifaHoraCache*.getOrDefault(id\_asiento, BigDecimal.*ZERO*);  
 }  
  
 private void setCellValue(Sheet sheet, int rowIndex, int colIndex, Object value) {  
 Row row = sheet.getRow(rowIndex);  
 if (row == null) {  
 row = sheet.createRow(rowIndex);  
 }  
 Cell cell = row.getCell(colIndex);  
 if (cell == null) {  
 cell = row.createCell(colIndex);  
 }  
 if (value instanceof String) {  
 cell.setCellValue((String) value);  
 } else if (value instanceof Double) {  
 cell.setCellValue((Double) value);  
 } else if (value instanceof BigDecimal) {  
 cell.setCellValue(((BigDecimal) value).doubleValue());  
 }  
 }  
  
  
 private void showAlert(String title, String message) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
  
 public void abrirVentana(String fxmlPath, String titulo) {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource(fxmlPath));  
 Parent root = loader.load();  
 Stage stage = new Stage();  
 stage.setScene(new Scene(root));  
 stage.setTitle(titulo);  
 stage.setResizable(false);  
 stage.setMaximized(false);  
 MenuPrincipalApp.*agregarIcono*(stage);  
 stage.show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void ventanaListaClientes(ActionEvent event) {  
 abrirVentana("/Menus/ListaClientes.fxml", "Lista de Cliente");  
 ((Stage) ListaClientes.getScene().getWindow()).close();  
 }  
  
 public void ventanaListaReservas(ActionEvent event) {  
 abrirVentana("/Menus/ListaReservas.fxml", "Listas De Reservas");  
 ((Stage) ListaReservas.getScene().getWindow()).close();  
 }  
  
 public void ventanaUsuarios(ActionEvent event) {  
 abrirVentana("/Menus/ListaUsuarios.fxml", "Usuarios");  
 ((Stage) BtnUsuarios.getScene().getWindow()).close();  
 }  
  
 public void salirVbox(ActionEvent event) {  
 vboxFacturas.setVisible(false);  
 }  
}

*/\*\*  
 \* Clase ControladorFacturas que se encarga de gestionar las facturas de la aplicación.  
 \* Se encarga de modificar, eliminar y abonar facturas.  
 \* También se encarga de obtener la forma de pago de una factura.  
 \*/*package Controlador;  
  
import ConexionDB.ConectionDB;  
import javafx.scene.control.\*;  
  
import java.sql.\*;  
import java.time.LocalDateTime;  
import java.util.Optional;  
  
public class ControladorFacturas {  
  
 public void modificarFactura(TextField fechaFactura, TextField descuento, TextField total, TextField subtotal, int idFactura) {  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 alert.setContentText("¿Quiere modificar la factura? :" + idFactura);  
  
 ButtonType si = new ButtonType("Sí");  
 ButtonType no = new ButtonType("No", ButtonBar.ButtonData.*CANCEL\_CLOSE*);  
  
 alert.getButtonTypes().setAll(si, no);  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.isPresent() && result.get() == si) {  
 String queryUrl = "UPDATE Facturas SET fecha\_hora\_emision = ?,descuento = ?, precio\_total = ? , subtotal = ? WHERE id\_factura = ?";  
 try {  
 PreparedStatement stmt = ConectionDB.*getConn*().prepareStatement(queryUrl);  
 stmt.setString(1, fechaFactura.getText());  
 stmt.setString(2, descuento.getText());  
 stmt.setString(3, total.getText());  
 stmt.setString(4, subtotal.getText());  
 stmt.setInt(5, idFactura);  
 int confirmacion = stmt.executeUpdate();  
 if (confirmacion == 1) {  
 showAlert("Modificar Factura", "La factura se ha modificado Correctamente");  
 } else {  
 showAlert("Modificar Factura", "No se ha modificado la factura");  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
  
 public void eliminarFactura(int idFactura) {  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 alert.setContentText("¿Quiere eliminar la factura? :" + idFactura);  
  
 ButtonType si = new ButtonType("Sí");  
 ButtonType no = new ButtonType("No", ButtonBar.ButtonData.*CANCEL\_CLOSE*);  
  
 alert.getButtonTypes().setAll(si, no);  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.isPresent() && result.get() == si) {  
 String queryUrl = "DELETE FROM Facturas WHERE id\_factura = ?";  
 String subquery = "DELETE FROM Reservas WHERE id\_factura = ?";  
 try {  
 //Eliminamos la reserva  
 PreparedStatement stmt = ConectionDB.*getConn*().prepareStatement(subquery);  
 stmt.setInt(1, idFactura);  
 stmt.executeUpdate();  
 //Eliminamos la factura  
 PreparedStatement stmt2 = ConectionDB.*getConn*().prepareStatement(queryUrl);  
 stmt2.setInt(1, idFactura);  
  
 int confirmacion2 = stmt2.executeUpdate();  
 if (confirmacion2 == 1) {  
 showAlert("Eliminar Factura", "La factura se ha eliminado Correctamente");  
 } else {  
 showAlert("Eliminar Factura", "No se ha elimado la factura");  
 }  
  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
  
 public void abonarFactura(TextField fechaFactura, TextField descuento, TextField total, TextField subtotal, ComboBox formaPago, int idFactura) {  
 String estadoPago = "Pagada";  
 LocalDateTime fecha = LocalDateTime.*now*();  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 alert.setContentText("¿Quiere abonar la factura? :" + idFactura);  
  
 ButtonType si = new ButtonType("Sí");  
 ButtonType no = new ButtonType("No", ButtonBar.ButtonData.*CANCEL\_CLOSE*);  
  
 alert.getButtonTypes().setAll(si, no);  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.isPresent() && result.get() == si) {  
 String queryUrl = "UPDATE Facturas SET fecha\_hora\_emision = ?,descuento = ?, precio\_total = ?, fecha\_hora\_pago = ? ,estado = ? , subtotal = ? , forma\_pago = ? WHERE id\_factura = ?";  
 try {  
 PreparedStatement stmt = ConectionDB.*getConn*().prepareStatement(queryUrl);  
 stmt.setString(1, fechaFactura.getText());  
 stmt.setString(2, descuento.getText());  
 stmt.setString(3, subtotal.getText());  
 stmt.setString(4, fecha.toString());  
 stmt.setString(5, estadoPago);  
 stmt.setString(6, total.getText());  
 stmt.setString(7, formaPago.getSelectionModel().getSelectedItem().toString());  
 stmt.setInt(8, idFactura);  
 int confirmacion = stmt.executeUpdate();  
 if (confirmacion == 1) {  
 showAlert("Pagar Factura", "La factura se ha Pagado Correctamente");  
 } else {  
 showAlert("Pagar Factura", "No se ha Pagado la factura");  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
  
 public String obtenerForma(int idFactura) {  
 String estado = "";  
 String query = "SELECT estado FROM Facturas WHERE id\_factura = ?";  
 try (PreparedStatement ps = ConectionDB.*getConn*().prepareStatement(query)) {  
 ps.setInt(1, idFactura);  
 ResultSet rs = ps.executeQuery();  
 if (rs.next()) {  
 estado = rs.getString("estado");  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 return estado;  
 }  
  
 private void showAlert(String title, String message) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
}

*/\*\*  
 \* Clase que se encarga de realizar las operaciones de la base de datos relacionadas con las reservas.  
 \* Se encarga de obtener las reservas, insertar una nueva reserva, obtener la factura pendiente de un cliente y crear una nueva factura.  
 \*/*package Manejadores\_Reservas\_Facturas;  
  
import Clases.Reservas;  
import ConexionDB.ConectionDB;  
  
import java.math.BigDecimal;  
import java.sql.\*;  
import java.util.ArrayList;  
import java.util.List;  
  
public class ReservaDAO {  
  
 public static List<Reservas> getReservas() throws SQLException {  
 List<Reservas> reservas = new ArrayList<>();  
 String query = "SELECT id\_reserva, dni, id\_asiento, id\_factura, fecha\_hora\_inicio, fecha\_hora\_fin, subtotal FROM Reservas";  
 try (PreparedStatement ps = ConectionDB.*getConn*().prepareStatement(query);  
 ResultSet rs = ps.executeQuery()) {  
  
 while (rs.next()) {  
 Reservas reserva = new Reservas(  
 rs.getInt("id\_reserva"),  
 rs.getString("dni"),  
 rs.getInt("id\_asiento"),  
 rs.getInt("id\_factura"),  
 rs.getTimestamp("fecha\_hora\_inicio"),  
 rs.getTimestamp("fecha\_hora\_fin"),  
 rs.getBigDecimal("subtotal")  
 );  
 BigDecimal subtotal = reserva.calcularSubtotal();  
 reserva.setSubtotal(subtotal);  
 reservas.add(reserva);  
 }  
 }  
 return reservas;  
 }  
  
 public static int insertarReserva(Reservas reserva) throws SQLException {  
 String query = "INSERT INTO Reservas (dni, id\_asiento, id\_factura, fecha\_hora\_inicio, fecha\_hora\_fin, subtotal) VALUES (?, ?, ?, ?, ?, ?)";  
 try (PreparedStatement ps = ConectionDB.*getConn*().prepareStatement(query, Statement.*RETURN\_GENERATED\_KEYS*)) {  
 ps.setString(1, reserva.getDni());  
 ps.setInt(2, reserva.getId\_asiento());  
 ps.setInt(3, reserva.getId\_factura());  
 ps.setTimestamp(4, reserva.getFecha\_hora\_inicio());  
 ps.setTimestamp(5, reserva.getFecha\_hora\_fin());  
 ps.setBigDecimal(6, reserva.getSubtotal());  
  
 int rowsInserted = ps.executeUpdate();  
 if (rowsInserted == 0) {  
 throw new SQLException("No se pudo crear la reserva.");  
 }  
  
 ResultSet generatedKeys = ps.getGeneratedKeys();  
 if (generatedKeys.next()) {  
 return generatedKeys.getInt(1);  
 } else {  
 throw new SQLException("No se pudo obtener el ID de la reserva.");  
 }  
 }  
 }  
  
 public static int obtenerFacturaPendiente(String dni) throws SQLException {  
 String query = "SELECT id\_factura FROM Facturas WHERE dni = ? AND estado = 'Pendiente' ORDER BY fecha\_hora\_emision DESC LIMIT 1";  
 try (PreparedStatement ps = ConectionDB.*getConn*().prepareStatement(query)) {  
 ps.setString(1, dni);  
 ResultSet rs = ps.executeQuery();  
 if (rs.next()) {  
 return rs.getInt("id\_factura");  
 }  
 }  
 return -1;  
 }  
  
 public static int crearFactura(String dni, BigDecimal precioTotal) throws SQLException {  
 int nuevoIdFactura = *obtenerMaxIdFactura*() + 1;  
 String query = "INSERT INTO Facturas (id\_factura, dni, precio\_total, descuento, fecha\_hora\_emision, estado, subtotal) VALUES (?, ?, ?, 0.00, NOW(), 'Pendiente',?)";  
 try (PreparedStatement ps = ConectionDB.*getConn*().prepareStatement(query)) {  
 ps.setInt(1, nuevoIdFactura);  
 ps.setString(2, dni);  
 ps.setBigDecimal(3, precioTotal);  
 ps.setBigDecimal(4, precioTotal);  
 ps.executeUpdate();  
 return nuevoIdFactura;  
 }  
 }  
  
 private static int obtenerMaxIdFactura() throws SQLException {  
 String query = "SELECT MAX(id\_factura) AS max\_id FROM Facturas";  
 try (PreparedStatement ps = ConectionDB.*getConn*().prepareStatement(query);  
 ResultSet rs = ps.executeQuery()) {  
 if (rs.next()) {  
 return rs.getInt("max\_id");  
 }  
 }  
 return 0; // Return 0 if there are no records  
 }  
}

### visualización

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

## Listausuarios.fxml, listausuarios.java, registrousuarios.fxml, controladorusuarios.java

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import java.net.\*?>  
<?import javafx.geometry.\*?>  
<?import javafx.scene.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.image.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.shape.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="830.0" prefWidth="1300.0" styleClass="AnchorPane" stylesheets="@../Estilos/EstilosListas.css" xmlns="http://javafx.com/javafx/17.0.12" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Modelos.ListaUsuarios">  
 <children>  
 <ImageView fx:id="portada" fitHeight="117.0" fitWidth="119.0" layoutX="589.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/bannerTopiaC.png" />  
 </image></ImageView>  
 <Label layoutX="562.0" layoutY="116.0" text="Lista De Usuarios">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font></Label>  
 <Line endX="707.0" layoutX="515.0" layoutY="650.0" startX="-452.0" strokeWidth="3.0" />  
 <Line endX="137.0" layoutX="403.0" layoutY="130.0" startX="-340.0" strokeWidth="3.0" />  
 <Line endX="74.0" layoutX="1145.0" layoutY="133.0" startX="-373.79998779296875" strokeWidth="3.0" />  
 <TextField id="dniBuscar" fx:id="empleado" alignment="TOP\_CENTER" layoutX="518.0" layoutY="149.0" prefHeight="32.0" prefWidth="249.0" promptText="Buscar Nombre" style="-fx-border-color: black; -fx-border-width: 3;">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor></TextField>  
 <VBox fx:id="vboxUsuarios" layoutX="522.0" layoutY="250.0" prefHeight="237.0" prefWidth="357.0" spacing="2.0" styleClass="vboxDatos" stylesheets="@../Estilos/EstiloVboxListas.css" visible="false">  
 <children>  
 <Button fx:id="btnSalir" contentDisplay="GRAPHIC\_ONLY" mnemonicParsing="false" onAction="#salirVbox" prefHeight="24.0" prefWidth="28.0" styleClass="btnSalir" stylesheets="@../Estilos/EstiloVboxListas.css">  
 <VBox.margin>  
 <Insets bottom="-10.0" left="325.0" />  
 </VBox.margin>  
 <graphic>  
 <ImageView fitHeight="17.0" fitWidth="17.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/borrar.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
 <Label alignment="TOP\_CENTER" contentDisplay="CENTER" prefHeight="32.0" prefWidth="191.0" text="Datos Del Usuario" textAlignment="CENTER">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font>  
 <VBox.margin>  
 <Insets left="40.0" />  
 </VBox.margin>  
 </Label>  
 <Label text="Nombre">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextNombre" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Label text="Email">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <TextField fx:id="TextEmail" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="140.0" />  
 </VBox.margin></TextField>  
 <Label text="Categoria">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin></Label>  
 <ComboBox fx:id="enumCategoria" onAction="#llenarComboBox" prefHeight="31.0" prefWidth="194.0" promptText="........." visibleRowCount="2">  
 <VBox.margin>  
 <Insets bottom="20.0" left="20.0" />  
 </VBox.margin></ComboBox>  
 <Button id="eliminar" fx:id="BtnModificar" mnemonicParsing="false" onAction="#modificarUsuarios" prefHeight="36.0" prefWidth="75.0" styleClass="Botones" text="Modificar">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 <graphic>  
 <ImageView fitHeight="24.0" fitWidth="29.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/editar-perfil.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <stylesheets>  
 <URL value="@../Estilos/EstilosListas.css" />  
 <URL value="@../Estilos/EstiloVboxListas.css" />  
 </stylesheets>  
 <VBox.margin>  
 <Insets bottom="15.0" left="223.0" top="-163.0" />  
 </VBox.margin>  
 </Button>  
 <Button id="modificar" fx:id="BtnEliminar" mnemonicParsing="false" onAction="#eliminarUsuarios" prefHeight="36.0" prefWidth="75.0" styleClass="Botones" stylesheets="@../Estilos/EstiloVboxListas.css" text="Eliminar">  
 <graphic>  
 <ImageView fitHeight="26.0" fitWidth="24.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/papelera-de-reciclaje(1).png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <VBox.margin>  
 <Insets left="223.0" />  
 </VBox.margin>  
 </Button>  
 </children>  
 </VBox>  
 <VBox layoutX="817.0" layoutY="652.0" prefHeight="178.0" prefWidth="165.0" style="-fx-background-color: #F4A261;" />  
 <GridPane alignment="TOP\_CENTER" layoutX="318.0" layoutY="669.0" prefHeight="123.0" prefWidth="664.0">  
 <columnConstraints>  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 <ColumnConstraints halignment="CENTER" hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />  
 </columnConstraints>  
 <rowConstraints>  
 <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />  
 </rowConstraints>  
 <children>  
 <Button fx:id="BtnUsuarios" alignment="CENTER" contentDisplay="TOP" mnemonicParsing="false" prefHeight="78.0" prefWidth="93.0" styleClass="BtnMenu" text="Usuarios" GridPane.columnIndex="3">  
 <graphic>  
 <ImageView fitHeight="73.0" fitWidth="75.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/Mano.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <stylesheets>  
 <URL value="@../Estilos/EstiloReservas.css" />  
 <URL value="@../Estilos/EstilosListas.css" />  
 </stylesheets>  
 </Button>  
 <Button fx:id="Facturacion" alignment="CENTER" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" onAction="#ventanaFacturaciones" prefHeight="115.0" prefWidth="93.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Facturas" GridPane.columnIndex="2">  
 <graphic>  
 <ImageView fitHeight="78.0" fitWidth="78.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/Facturacion.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="ListaReservas" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" onAction="#ventanaListaReservas" prefHeight="111.0" prefWidth="119.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Lista Reservas" GridPane.columnIndex="1">  
 <graphic>  
 <ImageView fitHeight="79.0" fitWidth="73.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/ListaReservas.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font size="15.0" />  
 </font>  
 </Button>  
 <Button fx:id="ListaClientes" contentDisplay="TOP" defaultButton="true" mnemonicParsing="false" onAction="#ventanaListaClientes" prefHeight="114.0" prefWidth="119.0" styleClass="BtnMenu" stylesheets="@../Estilos/EstilosListas.css" text="Lista Clientes">  
 <graphic>  
 <ImageView fitHeight="79.0" fitWidth="70.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/ListaClientes.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <font>  
 <Font name="System Bold" size="15.0" />  
 </font>  
 </Button>  
 </children>  
 </GridPane>  
 <TableView fx:id="tablaUsuarios" layoutX="82.0" layoutY="201.0" prefHeight="430.0" prefWidth="1120.0" styleClass="TablaListas" stylesheets="@../Estilos/EstilosListas.css">  
 <columns>  
 <TableColumn fx:id="colEmpleado" prefWidth="151.0" text="Nº Empleado" />  
 <TableColumn fx:id="colNombre" prefWidth="192.0" text="Nombre" />  
 <TableColumn fx:id="colEmail" prefWidth="386.0" text="Email" />  
 <TableColumn fx:id="colPassword" prefWidth="244.0" text="Contraseña" />  
 <TableColumn fx:id="colCategoria" prefWidth="146.0" text="Categoria" />  
  
 </columns>  
 </TableView>  
 <Button fx:id="ponerUsuario" layoutX="1162.0" layoutY="146.0" mnemonicParsing="false" onAction="#ventanaAgregarUsuarios" styleClass="BtnAcciones" stylesheets="@../Estilos/EstilosListas.css">  
 <graphic>  
 <ImageView fitHeight="50.0" fitWidth="29.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/circuloSuma1.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 </Button>  
  
 </children>  
  
</AnchorPane>

*/\*\*  
 \* Clase ListaUsuarios que se encarga de mostrar la lista de usuarios en la tabla de la interfaz gráfica  
 \* y de realizar las acciones de modificar y eliminar usuarios.  
 \* También se encarga de realizar la acción de login de los usuarios.  
 \*/*package Modelos;  
  
import Aplicaciones.MenuPrincipalApp;  
import Clases.SesionUsuario;  
import Clases.Usuarios;  
import ConexionDB.ConectionDB;  
import Controlador.ControladorUsuarios;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.collections.transformation.FilteredList;  
import javafx.collections.transformation.SortedList;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.control.Button;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.image.Image;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
import javafx.scene.control.TextField;  
import javafx.stage.StageStyle;  
  
import java.io.IOException;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.List;  
  
public class ListaUsuarios {  
 private double xOffset = 0;  
 private double yOffset = 0;  
 private ControladorUsuarios controladorUsuarios = new ControladorUsuarios();  
 private MenuPrincipal menuPrincipal;  
 private int idusuario;  
 @FXML  
 private Button AgregarClientes;  
 @FXML  
 private Button ListaReservas;  
 @FXML  
 private Button ListaClientes;  
 @FXML  
 private Button Facturacion;  
 @FXML  
 private Button btnSalir;  
 @FXML  
 private Button ponerUsuario;  
 @FXML  
 private TextField empleado;  
 @FXML  
 private VBox vboxUsuarios;  
 @FXML  
 private TextField TextNombre;  
 @FXML  
 private TextField TextEmail;  
 @FXML  
 private ComboBox enumCategoria;  
  
 private MenuPrincipalApp menuPrincipalapp = new MenuPrincipalApp();  
  
  
 ObservableList<String> categorias = FXCollections.*observableArrayList*("Admin", "Empleado");  
 @FXML  
 private TableView<Clases.Usuarios> tablaUsuarios;  
 @FXML  
 private TableColumn<Usuarios, Integer> colEmpleado;  
 @FXML  
 private TableColumn<Usuarios, String> colNombre;  
 @FXML  
 private TableColumn<Usuarios, String> colEmail;  
 @FXML  
 private TableColumn<Usuarios, String> colPassword;  
 @FXML  
 private TableColumn<Usuarios, String> colCategoria;  
  
 */\*\*  
 \* Método initialize que se encarga de cargar los datos de los usuarios en la tabla de la interfaz gráfica  
 \* y de realizar la acción de doble click en la tabla para exportar los datos del usuario seleccionado.  
 \*/* @FXML  
 public void initialize() {  
 btnSalir.setStyle(("-fx-background-color: transparent;"));  
 Stage ventanaSecundaria = new Stage();  
 ventanaSecundaria.getIcons().add(new Image(getClass().getResourceAsStream("/Imagenes/bannerTopiaC.png")));  
 colEmpleado.setCellValueFactory(new PropertyValueFactory<>("id\_usuario"));  
 colNombre.setCellValueFactory(new PropertyValueFactory<>("nombre"));  
 colEmail.setCellValueFactory(new PropertyValueFactory<>("email"));  
 colPassword.setCellValueFactory(new PropertyValueFactory<>("contrasenia"));  
 colCategoria.setCellValueFactory(new PropertyValueFactory<>("categoria"));  
  
  
 colPassword.setCellFactory(column -> new TableCell<Usuarios, String>() {  
 @Override  
 protected void updateItem(String password, boolean empty) {  
 super.updateItem(password, empty);  
 if (empty || password == null) {  
 setText(null);  
 } else {  
 setText("•".repeat(password.length()));  
 }  
 }  
 });  
  
  
 tablaUsuarios.setOnMouseClicked(event -> {  
 if (event.getClickCount() == 2) {  
 Usuarios usuario = tablaUsuarios.getSelectionModel().getSelectedItem();  
 if (usuario != null) {  
 exportarDatos(usuario);  
 }  
 }  
 });  
  
 vboxUsuarios.setOnMousePressed(event -> {  
 xOffset = event.getSceneX() - vboxUsuarios.getLayoutX() + 250;  
 yOffset = event.getSceneY() - vboxUsuarios.getLayoutY() + 70;  
 });  
 vboxUsuarios.setOnMouseDragged(event -> {  
 vboxUsuarios.setLayoutX(event.getScreenX() - xOffset);  
 vboxUsuarios.setLayoutY(event.getScreenY() - yOffset);  
 });  
  
 try {  
 loadUsuariosFromDatabase();  
 } catch (SQLException |  
 ClassNotFoundException e) {  
 e.printStackTrace();  
 } finally {  
  
 }  
 }  
  
 */\*\*ºº  
 \* Método loadUsuariosFromDatabase que se encarga de cargar los datos de los usuarios desde la base de datos.  
 \*/* private void loadUsuariosFromDatabase() throws SQLException, ClassNotFoundException {  
 List<Usuarios> usuarios = ConectionDB.*getUsuarios*();  
 ObservableList<Usuarios> usuariosList = FXCollections.*observableArrayList*(usuarios);  
 tablaUsuarios.setItems(usuariosList);  
  
 FilteredList<Usuarios> filtroUsuarios = new FilteredList<>(usuariosList, u -> true);  
 empleado.textProperty().addListener((observable, oldValue, newValue) -> {  
 filtroUsuarios.setPredicate(usuario -> {  
 if (newValue == null || newValue.isEmpty()) {  
 return true;  
 } else {  
 return usuario.getNombre().toLowerCase().contains(newValue.toLowerCase());  
 }  
 });  
 });  
  
 SortedList<Usuarios> sortedData = new SortedList<>(filtroUsuarios);  
 sortedData.comparatorProperty().bind(tablaUsuarios.comparatorProperty());  
 tablaUsuarios.setItems(sortedData);  
 }  
  
 public void exportarDatos(Usuarios usuarios) {  
 idusuario = usuarios.getId\_usuario();  
 TextNombre.setText(usuarios.getNombre());  
 TextEmail.setText(usuarios.getEmail());  
 enumCategoria.getSelectionModel().select(usuarios.getCategoria());  
 vboxUsuarios.setVisible(true);  
 }  
  
 public ListaUsuarios() {  
  
 }  
  
 public ListaUsuarios(MenuPrincipal menuPrincipal) {  
 this.menuPrincipal = menuPrincipal;  
  
 }  
  
  
 public void modificarUsuarios(ActionEvent event) {  
 controladorUsuarios.modificarUsuario(TextNombre, TextEmail, enumCategoria, idusuario);  
 initialize();  
  
 }  
  
 public void eliminarUsuarios(ActionEvent event) {  
 controladorUsuarios.eliminarUsuario(idusuario);  
 initialize();  
  
 }  
  
 */\*\*  
 \* Método loginUsuarios que se encarga de realizar la acción de login de los usuarios.  
 \* Si el usuario y la contraseña son correctos, se muestra un mensaje de login correcto y se abre la ventana del menú principal.  
 \*/* public void loginUsuarios(String usuario, String pass) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 String query = "SELECT nombre, contrasenia, categoria FROM Usuarios WHERE nombre = ? AND contrasenia = ?";  
  
 try (PreparedStatement ps = ConectionDB.*getConn*().prepareStatement(query)) {  
 ps.setString(1, usuario);  
 ps.setString(2, pass);  
 ResultSet rs = ps.executeQuery();  
  
 if (rs.next()) {  
 String categoria = rs.getString("categoria");  
 String user = rs.getString("nombre");  
 String password = rs.getString("contrasenia");  
 if (user.equals(usuario) && password.equals(pass)) {  
 }  
 SesionUsuario.*setCategoriaUsuario*(categoria); // Guardamos la categoría en la sesión  
 alert.setContentText("Login correcto");  
 alert.showAndWait();  
 menuPrincipal.nuevaVentana();  
  
 } else {  
 alert.setContentText("Login incorrecto");  
 menuPrincipal.cerrarVentana();  
 menuPrincipalapp.start(new Stage());  
 alert.show();  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
 }  
  
  
 public static void rellenarCombo(ComboBox<String> comboBox, ObservableList<String> categorias) {  
 comboBox.setItems(categorias);  
 }  
  
 public void llenarComboBox(ActionEvent event) {  
 *rellenarCombo*(enumCategoria, categorias);  
 }  
  
 public void abrirVentana(String fxmlPath, String titulo) {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource(fxmlPath));  
 Parent root = loader.load();  
 Stage stage = new Stage();  
 stage.setScene(new Scene(root));  
 stage.setTitle(titulo);  
 stage.setResizable(false);  
 stage.setMaximized(false);  
 MenuPrincipalApp.*agregarIcono*(stage);  
 stage.show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
  
 public void ventanaListaClientes(ActionEvent event) {  
 abrirVentana("/Menus/ListaClientes.fxml", "Lista de Cliente");  
 ((Stage) ListaClientes.getScene().getWindow()).close();  
 }  
  
 public void ventanaFacturaciones(ActionEvent event) {  
 abrirVentana("/Menus/Facturacion.fxml", "Factuacion");  
 ((Stage) Facturacion.getScene().getWindow()).close();  
 }  
  
 public void ventanaListaReservas(ActionEvent event) {  
 abrirVentana("/Menus/ListaReservas.fxml", "Listas De Reservas");  
 ((Stage) ListaReservas.getScene().getWindow()).close();  
 }  
  
 */\*\*  
 \* Método ventanaAgregarUsuarios que se encarga de abrir la ventana de registro de usuarios.  
 \*/* public void ventanaAgregarUsuarios(ActionEvent event) {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource("/Menus/RegistroUsuarios.fxml"));  
 Parent root = loader.load();  
 Stage stage = new Stage();  
 stage.setScene(new Scene(root));  
 stage.setTitle("Registro Usuarios");  
 stage.initStyle(StageStyle.*UNDECORATED*);  
 MenuPrincipalApp.*agregarIcono*(stage);  
  
 stage.show();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void salirVbox(ActionEvent event) {  
 vboxUsuarios.setVisible(false);  
 }  
  
}

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.geometry.\*?>  
<?import javafx.scene.\*?>  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.image.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane prefHeight="366.0" prefWidth="216.0" xmlns="http://javafx.com/javafx/17.0.12" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Modelos.RegistroUsuarios">  
 <VBox fx:id="vboxAgregarUsuario" prefHeight="366.0" prefWidth="280.0" spacing="2.0" styleClass="vboxDatos" stylesheets="@../Estilos/EstiloRegistroClientes.css">  
 <children>  
 <Button fx:id="btnSalir" contentDisplay="GRAPHIC\_ONLY" mnemonicParsing="false" onAction="#salir" styleClass="btnSalir" stylesheets="@../Estilos/EstiloVboxListas.css">  
 <graphic>  
 <ImageView fitHeight="43.0" fitWidth="20.0" pickOnBounds="true" preserveRatio="true">  
 <image>  
 <Image url="@../Imagenes/borrar.png" />  
 </image>  
 </ImageView>  
 </graphic>  
 <VBox.margin>  
 <Insets bottom="-29.0" left="240.0" top="2.0" />  
 </VBox.margin>  
 </Button>  
 <Label alignment="TOP\_CENTER" contentDisplay="CENTER" prefHeight="32.0" prefWidth="191.0" text="Añadir Usuario" textAlignment="CENTER">  
 <font>  
 <Font name="System Bold" size="22.0" />  
 </font>  
 <VBox.margin>  
 <Insets bottom="3.0" left="40.0" top="20.0" />  
 </VBox.margin>  
 </Label>  
 <Label text="Nombre">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <TextField fx:id="TextAgregarNombre" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="20.0" />  
 </VBox.margin>  
 </TextField>  
 <Label text="Email">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <TextField fx:id="TextAgregarEmail" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="20.0" />  
 </VBox.margin>  
 </TextField>  
 <Label layoutX="32.0" layoutY="119.0" text="Contraseña">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <TextField fx:id="TextAgregarPassword" layoutX="32.0" layoutY="138.0" prefHeight="35.0" prefWidth="220.0">  
 <VBox.margin>  
 <Insets left="20.0" right="20.0" />  
 </VBox.margin>  
 </TextField>  
 <Label text="Categoria">  
 <VBox.margin>  
 <Insets left="20.0" />  
 </VBox.margin>  
 </Label>  
 <ComboBox fx:id="enumAgregarCategoria" onAction="#llenarComboBox" prefHeight="40.0" prefWidth="233.0" promptText="........." visibleRowCount="2">  
 <VBox.margin>  
 <Insets bottom="20.0" left="20.0" />  
 </VBox.margin>  
 </ComboBox>  
 <Button id="eliminar" fx:id="BtnAgregarUsuario" mnemonicParsing="false" onAction="#agregarUsuario" prefHeight="33.0" prefWidth="109.0" styleClass="Botones" stylesheets="@../Estilos/EstiloRegistroClientes.css" text="Confirmar">  
 <cursor>  
 <Cursor fx:constant="DEFAULT" />  
 </cursor>  
 <VBox.margin>  
 <Insets bottom="10.0" left="83.0" />  
 </VBox.margin>  
 </Button>  
 </children>  
 </VBox>  
  
</AnchorPane>

*/\*\*  
 \* Controlador de la vista de usuarios  
 \* En este controlador se encuentran los métodos para modificar y eliminar usuarios  
 \* de la base de datos.  
 \*/*package Controlador;  
  
import ConexionDB.ConectionDB;  
import javafx.scene.control.\*;  
  
import java.sql.PreparedStatement;  
import java.sql.SQLException;  
import java.util.Optional;  
  
public class ControladorUsuarios {  
  
 public void modificarUsuario(TextField nombre, TextField email, ComboBox categoria, int idusuario) {  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 Alert alert2 = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setContentText("Quiere modificar El Usuario :" + idusuario);  
  
 ButtonType si = new ButtonType("Sí");  
 ButtonType no = new ButtonType("No", ButtonBar.ButtonData.*CANCEL\_CLOSE*);  
  
 alert.getButtonTypes().setAll(si, no);  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.isPresent() && result.get() == si) {  
 String query = "UPDATE Usuarios SET nombre = ?, email = ?, categoria = ? WHERE id\_usuario = ?";  
 try (PreparedStatement update = ConectionDB.*getConn*().prepareStatement(query)) {  
 update.setString(1, nombre.getText());  
 update.setString(2, email.getText());  
 update.setString(3, categoria.getSelectionModel().getSelectedItem().toString());  
 update.setInt(4, idusuario);  
 int actualizado = update.executeUpdate();  
 if (actualizado > 0) {  
 alert2.setContentText("Datos actualizados correctamente");  
 alert2.show();  
 }  
  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
  
 }  
  
 public void eliminarUsuario(int idusuario) {  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 Alert alert2 = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setContentText("Quiere eliminar el usuario :" + idusuario);  
 ButtonType si = new ButtonType("Sí");  
 ButtonType no = new ButtonType("No", ButtonBar.ButtonData.*CANCEL\_CLOSE*);  
  
 alert.getButtonTypes().setAll(si, no);  
  
 Optional<ButtonType> result = alert.showAndWait();  
 if (result.isPresent() && result.get() == si) {  
 String query = "DELETE FROM Usuarios WHERE id\_usuario = ?";  
 try (PreparedStatement delete = ConectionDB.*getConn*().prepareStatement(query)) {  
 delete.setInt(1, idusuario);  
 int actualizado = delete.executeUpdate();  
 if (actualizado > 0) {  
 alert2.setContentText("Datos Eliminados correctamente");  
 alert2.show();  
 }  
 } catch (SQLException e) {  
 throw new RuntimeException(e);  
 }  
 }  
 }  
}

### visualización

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

## Controladoremail.java

*/\*\*  
 \* Clase ControladorEmail que se encarga de enviar correos electrónicos a los clientes  
 \* con la información de la reserva realizada.  
 \*/*package Controlador;  
  
import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.Properties;  
import javax.mail.\*;  
import javax.mail.internet.\*;  
import javax.mail.Session;  
import javax.mail.PasswordAuthentication;  
import javax.mail.Authenticator;  
  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
  
  
public class ControladorEmail {  
 private static final String *correo* = "woktopiacoworking@gmail.com";  
 private static final String *contrasena* = "zywzibuvkyhqzmvk";  
  
 private static final ExecutorService *executor* = Executors.*newFixedThreadPool*(3);  
 private static final Session *session*;  
 private static Transport *transport*;  
  
 static {  
 Properties propiedades = new Properties();  
 propiedades.put("mail.smtp.host", "smtp.gmail.com");  
 propiedades.put("mail.smtp.ssl.trust", "smtp.gmail.com");  
 propiedades.put("mail.smtp.charset", "utf-8");  
 propiedades.put("mail.smtp.auth", "true");  
 propiedades.put("mail.smtp.starttls.enable", "true");  
 propiedades.put("mail.smtp.ssl.protocols", "TLSv1.2");  
 propiedades.put("mail.smtp.port", "587");  
  
 *session* = Session.*getInstance*(propiedades, new Authenticator() {  
 @Override  
 protected PasswordAuthentication getPasswordAuthentication() {  
 return new PasswordAuthentication(*correo*, *contrasena*);  
 }  
 });  
  
 try {  
  
 *transport* = *session*.getTransport("smtp");  
 *transport*.connect("smtp.gmail.com", *correo*, *contrasena*);  
 } catch (MessagingException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public static void enviarCorreo(String destinatario, String asunto, String mensajeHtml) {  
 *executor*.submit(() -> {  
 try {  
 Message email = new MimeMessage(*session*);  
 email.setFrom(new InternetAddress(*correo*));  
 email.addRecipient(Message.RecipientType.*TO*, new InternetAddress(destinatario));  
 email.setSubject(asunto);  
 email.setContent(mensajeHtml, "text/html; charset=utf-8");  
  
 synchronized (*transport*) {  
 *transport*.sendMessage(email, email.getAllRecipients());  
 }  
  
 System.*out*.println("Correo enviado correctamente");  
 } catch (MessagingException e) {  
 e.printStackTrace();  
 System.*out*.println("Error al enviar el correo.");  
 }  
 });  
 }  
  
 public String cargarPlantilla(String ruta, String cliente, String fecha, String horaInicio, String horaFin, String espacio, String subtotal) {  
 StringBuilder contenido = new StringBuilder();  
 try (BufferedReader br = new BufferedReader(new FileReader(ruta))) {  
 String linea;  
 while ((linea = br.readLine()) != null) {  
 contenido.append(linea).append("\n");  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 return "Error cargando la plantilla";  
 }  
  
 return contenido.toString()  
  
 .replace("{{CLIENTE}}", cliente)  
 .replace("{{FECHA}}", fecha)  
 .replace("{{HORA\_INICIO}}", horaInicio)  
 .replace("{{HORA\_FIN}}", horaFin)  
 .replace("{{ESPACIO}}", espacio)  
 .replace("{{SUBTOTAL}}", subtotal);  
 }  
  
  
}

Por último los estilos

## Estilohorario.css, estiloregistroclientes.css, estiloreservas.css, estiloslistas.css, estilosmenuprincipal.css, estilovboclistas.css

.BtnHoras{  
 -fx-font-weight: bold;  
}

.vboxDatos{  
 -fx-background-color: #DDDEDE;  
 -fx-border-color: black;  
 -fx-border-width: 2;  
}  
.Botones{ /\*Botones de Confirmar\*/  
 -fx-background-color: #4CAF50;  
 -fx-border-color: #FFA500FF;  
 -fx-text-fill: #FFFFFF;  
 -fx-font-size: 15px;  
 -fx-font-weight: bold;  
}  
.Botones:hover{  
 -fx-background-color: #046107;  
 -fx-scale-x: 1.2;  
 -fx-scale-y: 1.2 ;  
}  
  
.Botones:pressed{  
 -fx-background-color: #047f07;  
 -fx-scale-y: 1;  
 -fx-scale-x: 1;  
}

.AnchorPane{  
 -fx-background-color: #DDDEDE;  
}  
.BtnMenu{  
 -fx-background-color: Transparent;  
}  
  
.BtnMenu:hover{  
 -fx-scale-x: 1.2;  
 -fx-scale-y: 1.2 ;  
}  
.BtnRegistro{  
 -fx-background-color: #4CAF50;  
 -fx-border-color: #FFA500FF;  
 -fx-text-fill: #FFFFFF;  
 -fx-font-size: 15px;  
 -fx-font-weight: bold;  
}  
.BtnRegistro:hover{  
 -fx-background-color: #046107;  
 -fx-scale-x: 1.2;  
 -fx-scale-y: 1.2 ;  
}  
.BtnRegistro:pressed{  
 -fx-background-color: Transparent;  
 -fx-scale-y: 0.98;  
 -fx-scale-x: 0.98;  
}  
.fechas{  
 -fx-background-color: #FFFFFF;  
 -fx-font-size: 15px;  
 -fx-font-weight: bold;  
 -fx-text-fill: orange;  
}  
.BtnOficina{  
 -fx-background-color: Transparent;  
}  
.BtnOficina:hover{  
 -fx-background-color: Transparent;  
 -fx-border-color: #1eff00;  
 -fx-border-width: 3;  
}  
.BtnOficina:pressed{  
 -fx-background-color: Transparent;  
 -fx-scale-y: 0.98;  
 -fx-scale-x: 0.98;  
}

.AnchorPane {  
 -fx-background-color: #DDDEDE;  
}  
  
.BtnMenu {  
 -fx-background-color: Transparent;  
 -fx-text-fill: rgb(0, 0, 0);  
 -fx-font-size: 15px;  
 -fx-font-weight: bold;  
}  
  
.BtnMenu:hover {  
 -fx-scale-x: 1.2;  
 -fx-scale-y: 1.2;  
}  
  
.BtnMenu:pressed {  
 -fx-background-color: transparent;  
 -fx-scale-y: 1;  
 -fx-scale-x: 1;  
}  
  
.BtnAcciones {  
 -fx-background-color: transparent;  
 -fx-font-weight: bold;  
 -fx-text-fill: #FFA500FF;  
}  
  
.BtnAcciones:hover {  
 -fx-scale-x: 1.2;  
 -fx-scale-y: 1.2;  
}  
  
.TablaListas {  
 -fx-background-color: #DDDEDE;  
  
}  
.TablaListas .column-header-background{  
 -fx-background-color: #EADDC8;  
}  
  
.btnVbox {  
 -fx-background-color: #4CAF50;  
 -fx-border-color: #FFA500FF;  
 -fx-text-fill: #FFFFFF;  
 -fx-font-size: 15px;  
 -fx-font-weight: bold;  
 -fx-border-width: 4;  
}  
  
.btnVbox:hover {  
 -fx-scale-x: 1.2;  
 -fx-scale-y: 1.2;  
}

.AnchorPane{  
 -fx-background-color: #DDDEDE;  
}  
.BtnEntrar{  
 -fx-background-color: #4CAF50;  
 -fx-text-fill: rgb(245, 242, 242);  
 -fx-font-size: 15px;  
 -fx-font-weight: bold;  
 -fx-border-color: orange;  
}  
  
.BtnEntrar:hover{  
 -fx-background-color:#046107;  
 -fx-scale-x: 1.2;  
 -fx-scale-y: 1.2 ;  
}  
.BtnEntrar:pressed{  
 -fx-background-color: #047f07;  
 -fx-scale-y: 1;  
 -fx-scale-x: 1;  
}  
.textfieldUsuario{  
 -fx-border-color: #000000;  
 -fx-border-width: 3px;  
}  
.textfielPassword{  
 -fx-border-color: #000000;  
 -fx-border-width: 3px;  
}  
.btnpassword{  
 -fx-background-color: Transparent;  
 -fx-text-fill: orange;  
}  
.btnpassword:hover{  
 -fx-text-fill: #bf7b16;  
 -fx-scale-y: 1.1;  
 -fx-scale-x: 1.1;  
}  
.btnpassword:pressed{  
 -fx-text-fill: #efa32d;  
 -fx-scale-y: 1;  
 -fx-scale-x: 1;  
}

.vboxDatos{  
 -fx-background-color: #DDDEDE;  
 -fx-border-color: #FFA500FF;  
 -fx-border-width: 2;  
}  
  
.Botones { /\*Botones de Confirmar\*/  
 -fx-background-color: #4CAF50;  
 -fx-border-color: #FFA500FF;  
 -fx-text-fill: transparent;  
 -fx-pref-width: 60px;  
 -fx-transition: -fx-pref-width 0.3s ease-in-out, -fx-text-fill 0.3s ease-in-out;  
}  
.Botones:hover {  
 -fx-opacity: 1;  
 -fx-background-color: #046107;  
 -fx-pref-width: 120px;  
 -fx-text-fill: #ffffff;  
 -fx-font-weight: bold;  
 -fx-font-size: 12px;  
  
}  
.labels{  
 -fx-font-family: "Tahoma";  
}  
  
.btnSalir{  
 -fx-background-color:trasparent;  
}  
.btnSalir:hover{  
 -fx-background-color:trasparent;  
 -fx-scale-x: 1.2;  
 -fx-scale-y: 1.2;  
}  
.btnSalir:pressed{  
 -fx-background-color:trasparent;  
 -fx-scale-x: 1.2;  
 -fx-scale-y: 1.2;  
}

# Pom y pruebas

A pesar de intentar utilizar Junit en Maven, nos fue imposible implementarlo, tuvimos varias dificultades con la integración de dependencias.

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example</groupId>  
 <artifactId>Worktopia\_cod</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>jar</packaging>  
 <name>Worktopia\_cod</name>  
  
 <properties>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 <junit.version>5.10.0</junit.version>  
 </properties>  
  
 <repositories>  
 <repository>  
 <id>central</id>  
 <url>https://repo.maven.apache.org/maven2</url>  
 </repository>  
 </repositories>  
  
 <dependencies>  
  
 <dependency>  
 <groupId>com.sun.mail</groupId>  
 <artifactId>javax.mail</artifactId>  
 <version>1.6.2</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.apache.logging.log4j</groupId>  
 <artifactId>log4j-api</artifactId>  
 <version>2.20.0</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.apache.logging.log4j</groupId>  
 <artifactId>log4j-core</artifactId>  
 <version>2.20.0</version>  
 </dependency>  
  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>8.0.29</version>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-controls</artifactId>  
 <version>21</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-fxml</artifactId>  
 <version>21</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-web</artifactId>  
 <version>21</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-swing</artifactId>  
 <version>21</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-media</artifactId>  
 <version>21</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.controlsfx</groupId>  
 <artifactId>controlsfx</artifactId>  
 <version>11.1.2</version>  
 </dependency>  
  
 <dependency>  
 <groupId>com.dlsc.formsfx</groupId>  
 <artifactId>formsfx-core</artifactId>  
 <version>11.6.0</version>  
 <exclusions>  
 <exclusion>  
 <groupId>org.openjfx</groupId>  
 <artifactId>\*</artifactId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
  
 <dependency>  
 <groupId>net.synedra</groupId>  
 <artifactId>validatorfx</artifactId>  
 <version>0.4.0</version>  
 <exclusions>  
 <exclusion>  
 <groupId>org.openjfx</groupId>  
 <artifactId>\*</artifactId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
  
 <dependency>  
 <groupId>org.kordamp.ikonli</groupId>  
 <artifactId>ikonli-javafx</artifactId>  
 <version>12.3.1</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.kordamp.bootstrapfx</groupId>  
 <artifactId>bootstrapfx-core</artifactId>  
 <version>0.4.0</version>  
 </dependency>  
  
 <dependency>  
 <groupId>eu.hansolo</groupId>  
 <artifactId>tilesfx</artifactId>  
 <version>11.48</version>  
 <exclusions>  
 <exclusion>  
 <groupId>org.openjfx</groupId>  
 <artifactId>\*</artifactId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
  
 <dependency>  
 <groupId>com.github.almasb</groupId>  
 <artifactId>fxgl</artifactId>  
 <version>17.3</version>  
 <exclusions>  
 <exclusion>  
 <groupId>org.openjfx</groupId>  
 <artifactId>\*</artifactId>  
 </exclusion>  
 </exclusions>  
 </dependency>  
  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>5.8.2</version>  
 <scope>test</scope>  
 </dependency>  
  
 <dependency>  
 <groupId>org.junit.platform</groupId>  
 <artifactId>junit-platform-engine</artifactId>  
 <version>1.8.2</version>  
 <scope>test</scope>  
 </dependency>  
  
  
 <dependency>  
 <groupId>org.apache.poi</groupId>  
 <artifactId>poi-ooxml</artifactId>  
 <version>5.2.3</version>  
 </dependency>  
  
 <dependency>  
 <groupId>org.testng</groupId>  
 <artifactId>testng</artifactId>  
 <version>RELEASE</version>  
 <scope>compile</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.junit.jupiter</groupId>  
 <artifactId>junit-jupiter-api</artifactId>  
 <version>5.10.0</version>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.11.0</version>  
 <configuration>  
 <source>21</source>  
 <target>21</target>  
 </configuration>  
 </plugin>  
  
 <plugin>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-maven-plugin</artifactId>  
 <version>0.0.8</version>  
 <executions>  
 <execution>  
 <id>default-cli</id>  
 <configuration>  
 <mainClass>Aplicaciones.MenuPrincipalApp</mainClass>  
 <jvmArgs>  
 --module-path ${java.home}/lib --add-modules javafx.controls,javafx.fxml  
 </jvmArgs>  
 </configuration>  
 </execution>  
 </executions>  
 </plugin>  
  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-jar-plugin</artifactId>  
 <version>3.1.0</version>  
 <configuration>  
 <archive>  
 <manifest>  
 <mainClass>Aplicaciones.MenuPrincipalApp</mainClass>  
 </manifest>  
 </archive>  
 </configuration>  
 </plugin>  
  
  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-surefire-plugin</artifactId>  
 <version>3.0.0-M9</version>  
 <configuration>  
 <argLine>--add-opens org.junit.platform.commons/org.junit.platform.commons.util=ALL-UNNAMED</argLine>  
 </configuration>  
 </plugin>  
  
 <plugin>  
 <groupId>org.panteleyev</groupId>  
 <artifactId>jpackage-maven-plugin</artifactId>  
 <version>1.6.0</version>  
 <executions>  
 <execution>  
 <goals>  
 <goal>jpackage</goal>  
 </goals>  
 </execution>  
 </executions>  
 <configuration>  
 <name>Worktopia</name>  
 <appVersion>1.0</appVersion>  
 <type>image</type>  
 <mainClass>Aplicaciones.MenuPrincipalApp</mainClass>  
 <mainJar>${project.build.finalName}.jar</mainJar>  
 <winShortcut>true</winShortcut>  
 <winMenu>true</winMenu>  
 <verbose>true</verbose>  
 </configuration>  
 </plugin>  
  
 </plugins>  
 </build>  
</project>

A pesar de estar bien “supuestamente” integrado, no paran de surgir errores tras errores y finalmente tuvimos que probar la aplicación en uso real en diferentes situaciones para ir corrigiendo.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

intento de crear documentación.

Ha pesar de que la codificación del proyecto con respecto a la documentación está bien distribuida y no hay errores en la configuración ni el pom, tampoco fuimos capaz de crear el archivo de documentación directamente con el IDE.

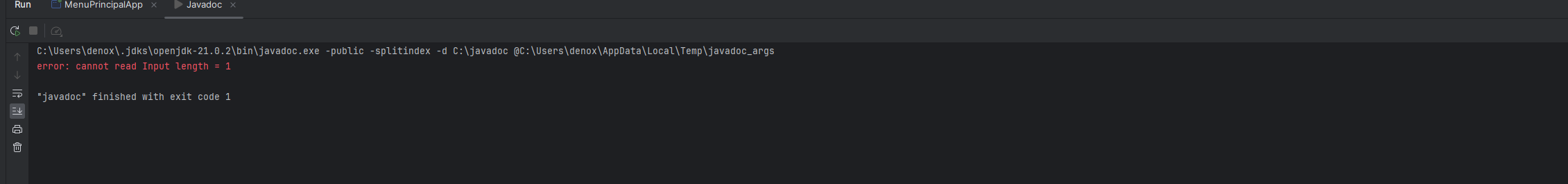
module org.example.worktopia\_cod {  
  
 requires javafx.controls;  
 requires javafx.fxml;  
 requires javafx.web;  
 requires javafx.graphics;  
  
  
 requires org.controlsfx.controls;  
 requires com.dlsc.formsfx;  
 requires net.synedra.validatorfx;  
 requires org.kordamp.ikonli.javafx;  
 requires org.kordamp.bootstrapfx.core;  
 requires eu.hansolo.tilesfx;  
 requires com.almasb.fxgl.all;  
 requires org.apache.poi.ooxml;  
 requires com.fasterxml.jackson.databind;  
 requires org.apache.logging.log4j.core;  
 requires org.apache.logging.log4j;  
 requires mysql.connector.java;  
 requires com.google.protobuf;  
 requires jdk.httpserver;  
 requires java.mail;  
 requires java.sql;  
 requires java.desktop;  
  
 requires org.junit.jupiter.api;  
 requires org.testng;  
  
 opens Modelos to javafx.fxml, org.junit.jupiter.api;  
 opens Aplicaciones to javafx.graphics, org.junit.jupiter.api;  
 opens Clases to javafx.base, org.junit.jupiter.api;  
  
 exports Aplicaciones;  
 exports Modelos;  
 exports Clases;  
 exports Prueba\_errores to org.junit.platform.commons;  
}

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Sitio web

El contenido generado por IA puede ser incorrecto.



Este error fue continuo y a pesar de buscar documentación y varias pruebas no conseguimos resolverlo.

# Intento de ejecutable

Se ha intentado realizar un ejecutable del programa, pero sin éxito, conseguimos crear el instalador, incluimos los módulos de JavaFX, pero nada ha funcionado hasta ahora.

Ejecutando el jar directamente pero aún no llegamos a entender por que el ejecutable no.

java --module-path "C:\Program Files\Microsoft\javafx-sdk-23.0.2\lib" --add-modules javafx.controls,javafx.fxml,javafx.base,javafx.graphics -jar "C:\Users\denox\OneDrive\Estudios DAM\DAM2do\Diseño de Interfaces\Worktopia\Worktopia\_cod\target\Worktopia\_cod-1.0-SNAPSHOT.jar"

Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

Usamos este comando para PowerShell para crear el instalador

& "C:\Program Files\Microsoft\jdk-21.0.2\bin\jpackage.exe" --name "Worktopia" `

--input . `

--main-jar "C:\Users\denox\OneDrive\Estudios DAM\DAM2do\Diseño de Interfaces\Worktopia\Worktopia\_cod\target\Worktopia\_cod-1.0-SNAPSHOT.jar" `

--main-class "Aplicaciones.MenuPrincipalApp" `

--type "msi" `

--win-shortcut `

--win-menu `

--icon "C:\Users\denox\OneDrive\Estudios DAM\DAM2do\Diseño de Interfaces\Worktopia\Worktopia\_cod\src\main\resources\Imagenes\worktopiaICO.ico" `

--module-path "C:\Program Files\Microsoft\javafx-sdk-23.0.2\lib" `

--add-modules javafx.controls,javafx.fxml,javafx.base,javafx.graphics

Imagen que contiene Icono

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Una lástima porque sentimos que nos faltaba poco para poder lograrlo, pero nos quedamos atascados.

# conclusiones

La aplicación Worktopia, ha sido nuestro primer reto para intentar realizar una aplicación completa en java, no ha sido fácil, dado que, nunca habíamos programado a este nivel, y algunas cosas han sido frustrantes.

Si bien es cierto que JavaFX es fácil de manejar gracias a su sencillez, su aspecto gráfico era un tanto tosco y en ocasiones no funcionaba del todo bien, forzando a usar código para que volviera a funcionar correctamente. Pero gracias al css y estilos estamos contentos con el resultado. Pero al depender de Maven ha supuesto un nuevo grado de dificultad.

Integrar la base de datos, bibliotecas externas para crear archivos, creación de ejecutables, pruebas de errores, todo dependía de la buena configuración del archivo pom de Maven, la mayoría lo logramos, pero hay algunas que nos derrotaron por completo por nuestra falta de experiencia.

Sobre todo con la parte del ejecutable, que pensamos que está todo correcto, pero tan solo hay módulos o similares que por algún motivo no implementamos, o tal vez sea algún error en la instalación del propio Windows..

Pero la gran conclusión es que ha sido un avance en nuestro aprendizaje y muchas más cosas que se pueden mejorar.