

Proyecto POStre

Programación Avanzada



Introducción	3
Desarrollo	3
Problemas durante el desarrollo	3
Concesiones y limitaciones	3
Casos de uso	4
Log In	4
Prototipo: No hay	4
Ventana final:	4
Cobrar cuenta	4
Prototipo:	5
Ventana final:	6
Agregar mesa	6
Prototipo: No hay	7
Ventana final:	7
Asignar mesa a cuenta	7
Prototipo:	8
Ventana final:	8
Agregar alimento a cuenta	8
Prototipo:	9
Ventana final:	9
Agregar alimento al sistema	10
Prototipo:	11
Ventana final:	11
Eliminar alimento del sistema	12
Prototipo:	13
Ventana final:	13
Modificar alimento	13
Prototipo:	14
Ventana final:	14
Hacer corte de venta	14
Prototipo: No hay	15
Ventana final:	15
Cerrar sesión	15
Prototipo: No hay	16
Ventana final:	16
Diagrama de casos de uso	17
Diagrama de clases	18
Diagrama de módulos	19
Descripción diagrama de módulos	20
Diagrama ER base de datos	21

Introducción

POStre, un software de punto de venta que simplifica el proceso de ventas al máximo. POSTre se destaca por su enfoque minimalista y eficiente.

Un sistema que elimina todas las acciones innecesarias y se enfoca exclusivamente en lo esencial. POSTre es esa solución,. Con una interfaz intuitiva y limpia, este software les permitirá procesar ventas de manera rápida y sin complicaciones. Adiós a las distracciones y a los pasos redundantes, bienvenida la agilidad y la productividad.

Pero POSTre no solo se trata de velocidad, también es sinónimo de precisión. Cada transacción será registrada con exactitud, sin que los usuarios lo perciban hasta que sea el momento de realizar corte de caja y ver las ventas registradas.

POSTre brinda la libertad de poder enfocarse en atender a sus clientes y hacer crecer su negocio. POSTre se encargará de los detalles técnicos, liberándose de tareas tediosas y permitiéndoles concentrarse en brindar una experiencia excepcional.

Desarrollo

El desarrollo del software comenzó con la creación del proyecto patata, el nombre fue provisional en lo que conseguimos un nombre, pero se seleccionó un alimento por tener relación con un punto de venta para restaurantes.

Problemas durante el desarrollo

Durante el desarrollo del proyecto el mayor problema fue con la compatibilidad de uno de nuestros dispositivos al utilizar el chip M1 de Apple, esto nos generó problemas de compatibilidad que afectaron al desarrollo.

Otro problema es que nuestro otro dispositivo presenta problemas con las rutas absolutas, por lo que nos fue prácticamente imposible utilizar archivos css e importar ciertas librerías para como fue el caso de "com.jfoenix" que deseabamos utilizar para darle una mejor interfaz al proyecto final.

Concesiones y limitaciones

La limitación más grande que tuvimos fue la de la compatibilidad con el chip M1 y bugs de las actualizaciones recientes.

Por el lado de las concesiones fue que no pudimos crear una interfaz más estilizada como deseábamos por los problemas con las rutas, pues incluso no dificultaba mucho usar archivos FXML. Por lo que todo fue programado via java con css introducido localmente.

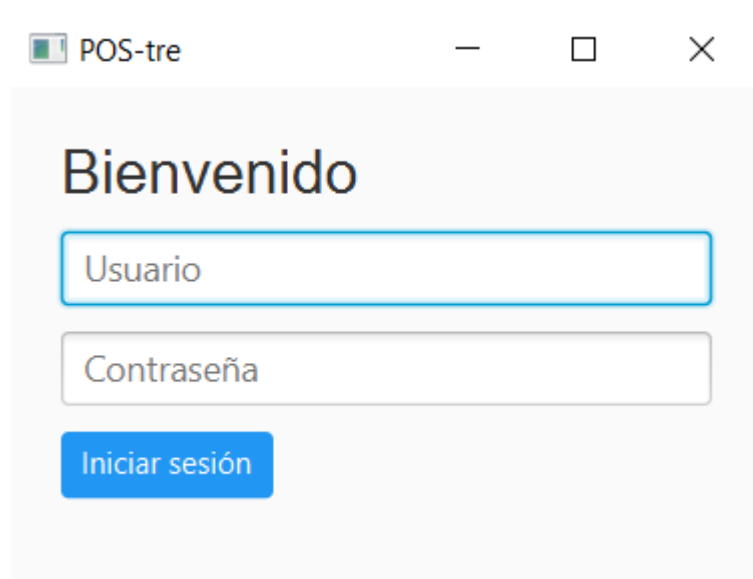
Casos de uso

Log In

Permite a los diferentes tipos de usuario hacer LogIn y los redirige a sus respectivas pestañas, en caso de presentar credenciales incorrectas saldrá una ventana emergente con un mensaje de error.

Prototipo: No hay

Ventana final:



The image shows a window titled "POS-tre" with standard Windows window controls (minimize, maximize, close). The window content is a light gray box with the heading "Bienvenido" in a large, bold, black font. Below the heading are two input fields: the first is labeled "Usuario" and the second is labeled "Contraseña". Both fields have a light blue border. Below the input fields is a blue button with the text "Iniciar sesión" in white. The window is set against a white background.

Cobrar cuenta

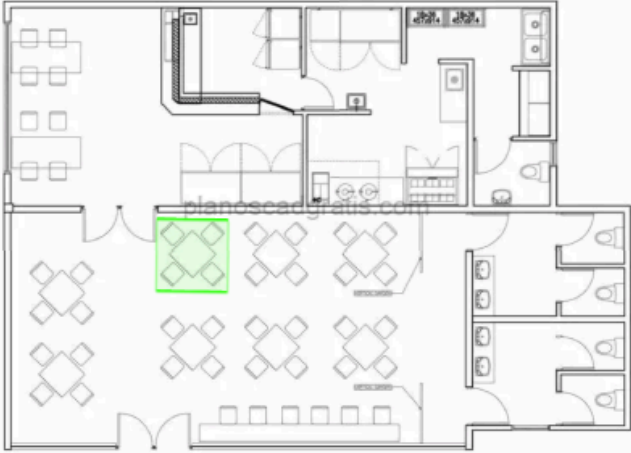
En esta parte se seleccionará primero la cuenta a cobrar y se dará click en cobrar, lo que redirija al usuario a otra ventana, donde poder agregar propina y el pago del cliente. En caso de no contar con dinero suficiente el sistema mandará un mensaje de error y agrega las ventas saldadas al corte.

Prototipo:

Base de datos Joshua

← Menu principal/Meseros

Sucursal Av.Camacho



Mesa 1

Mesa 2

Mesa 3

Mesa 4

Mesa 5

Mesa 6

Mesa 7

Mesa 8

Barra 1

Reservar

Ver

Cobrar

Base de datos Joshua

Pago

Efectivo

Tarjeta

Propina

%

Monto

Subtotal

Total

Cambio

Cancelar

Cobrar

Ventana final:

The screenshot shows a window titled "POS-tre" with a blue header bar. The header bar contains a back arrow and the text "Mesas" on the left, and a "Cerrar sesión" button on the right. Below the header, there is a list of tables labeled "Mesa 1" through "Mesa 7". "Mesa 1" is highlighted in blue, while the others are in light gray. Below the list, there are several empty light gray bars. In the bottom right corner, there is a blue button labeled "Cobrar".

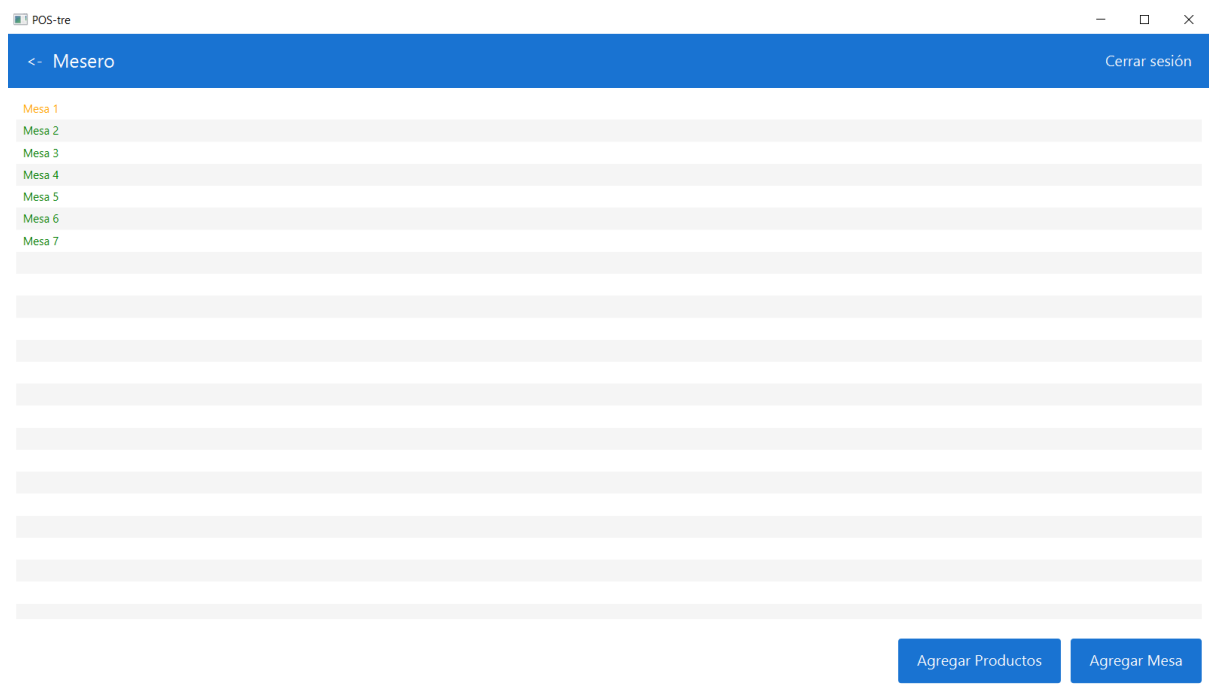
The screenshot shows a window titled "POS-tre" for the payment process. It has a light gray background. On the left, there are input fields for "Pago: \$" (containing "100"), "Propina: \$" (containing "23"), and a "Cancelar" button at the bottom left. In the center, there is a purple button labeled "Añadir Pago" and another purple button labeled "Rescindir Propina". On the right, there is a summary of the payment: "Subtotal: \$130.0", "Total: \$153.0", and "Cambio: \$-53.0". Below this summary is a purple button labeled "Pagar".

Agregar mesa

Esta función es para crear nuevas mesas al restaurante, simplemente es dando click al botón "Agregar mesa".

Prototipo: No hay

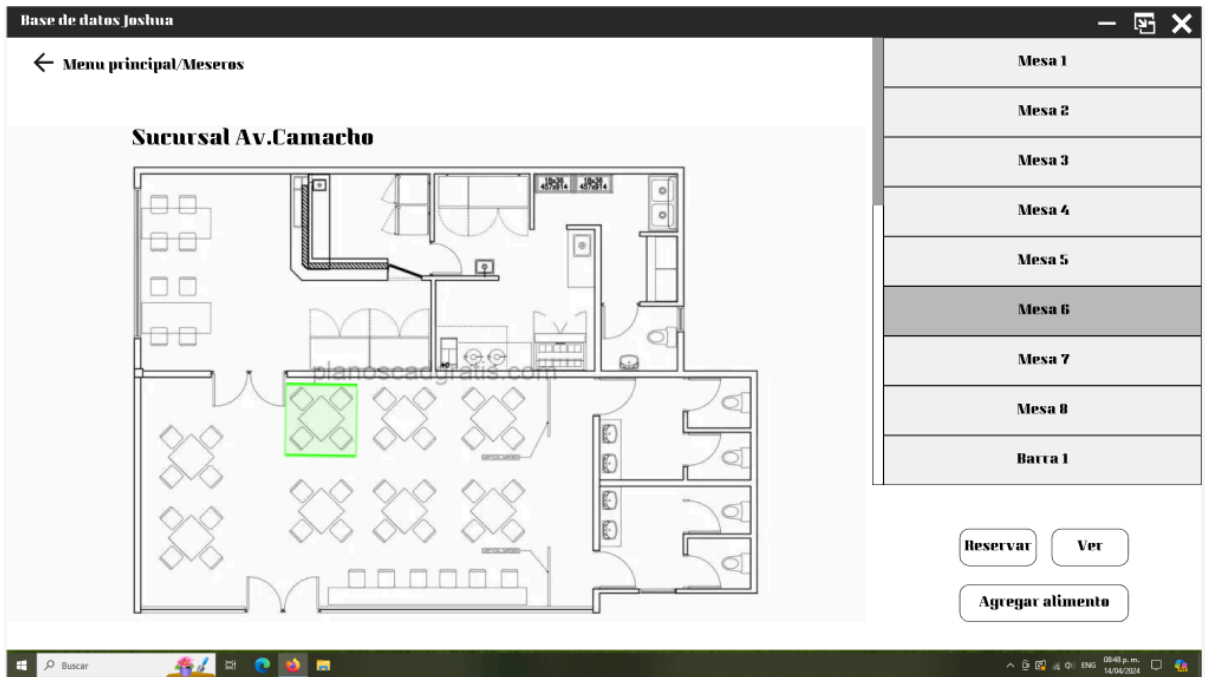
Ventana final:



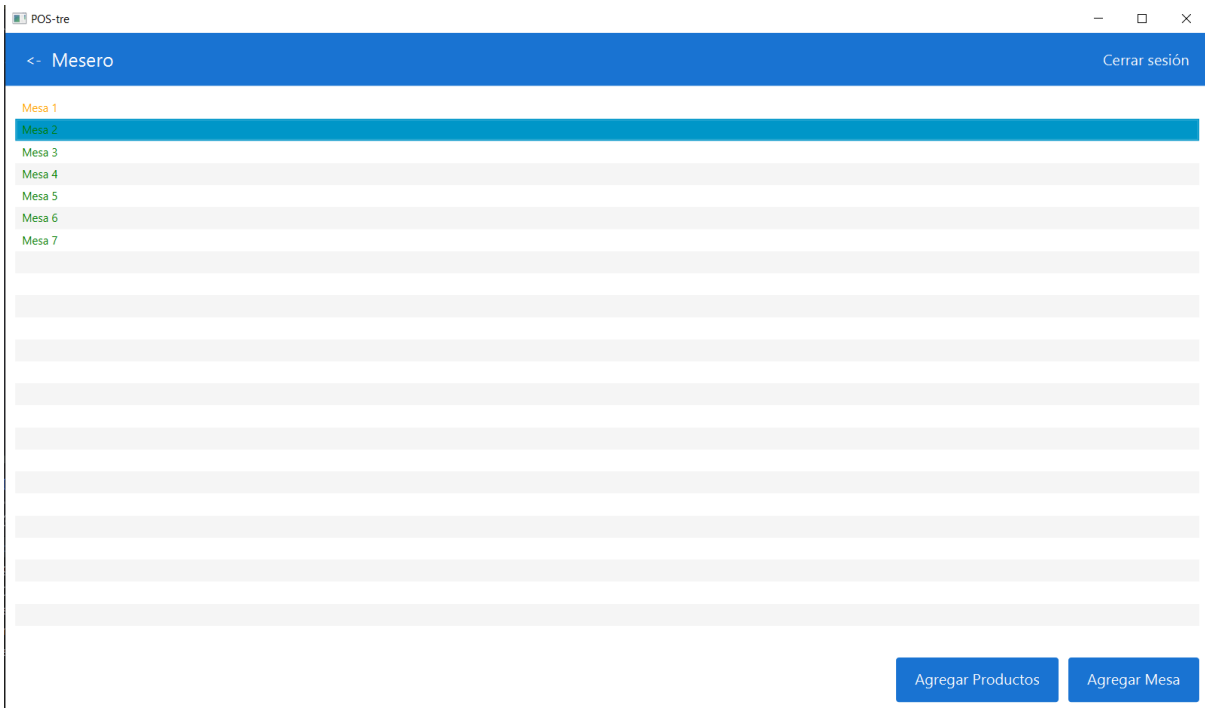
Asignar mesa a cuenta

Permite agregar cuentas a una mesa, su funcionamiento es seleccionar una mesa disponible (color verde) o ocupada (color naranja) y darle al botón de agregar producto, se redirigirá a las siguiente ventana.

Prototipo:



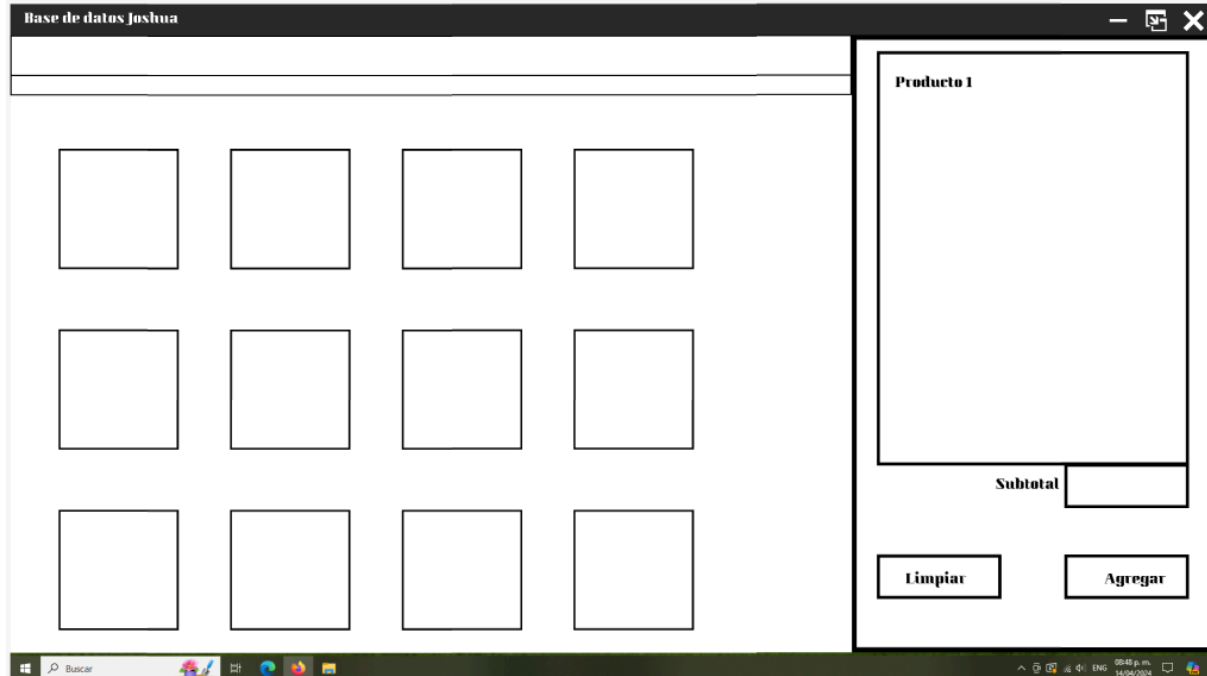
Ventana final:



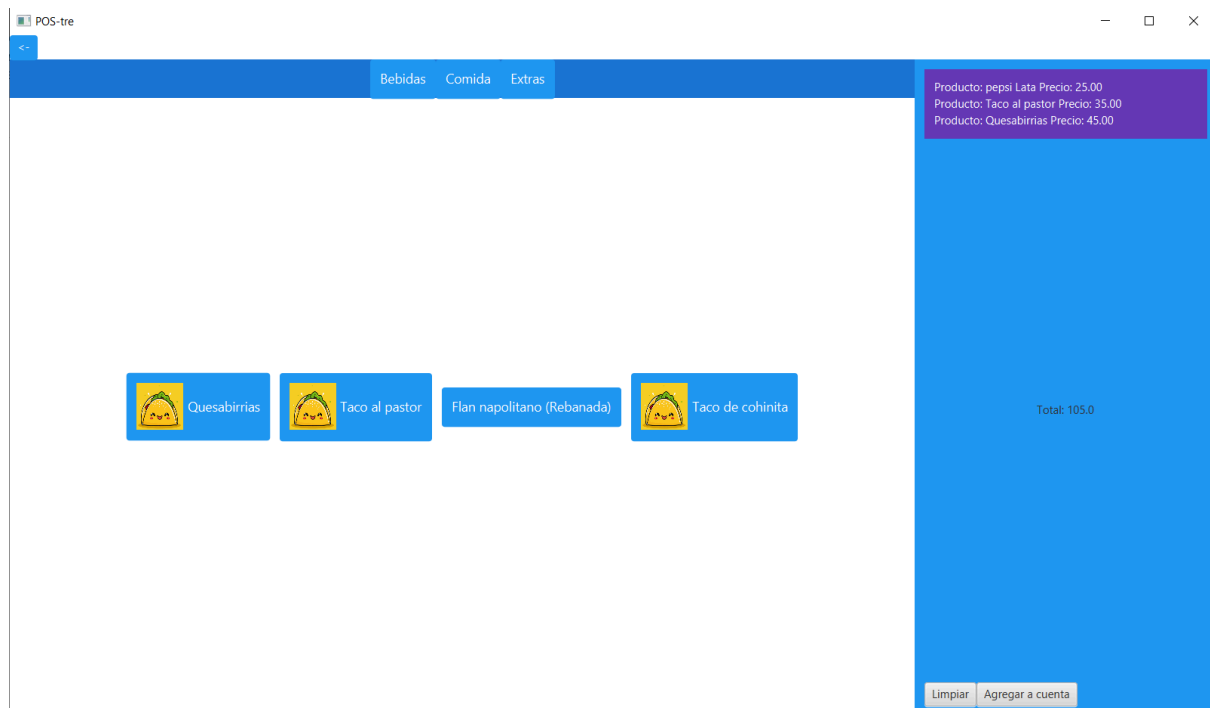
Agregar alimento a cuenta

Habr  una barra de categor as en la parte superior que permitir n seleccionar el tipo de producto a agregar, al hacer clic se desplegar n los productos de dicha categor a y podremos agregarlos a la cuenta o limpiar la comanda en caso de error.

Prototipo:



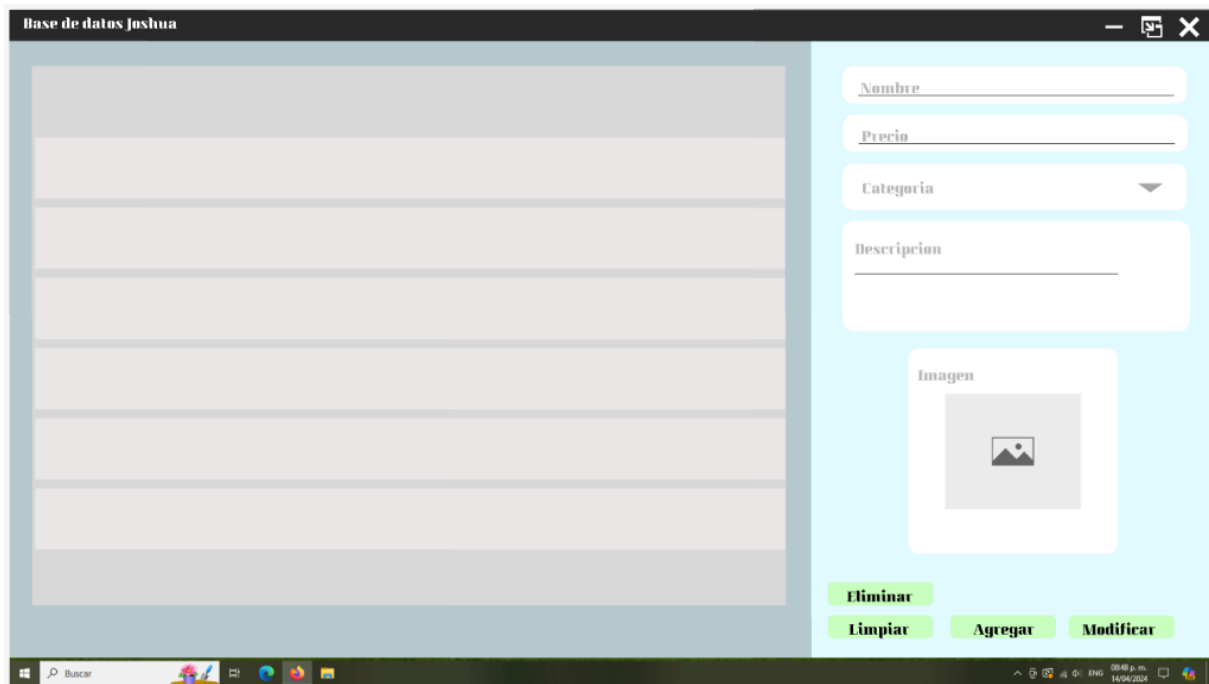
Ventana final:



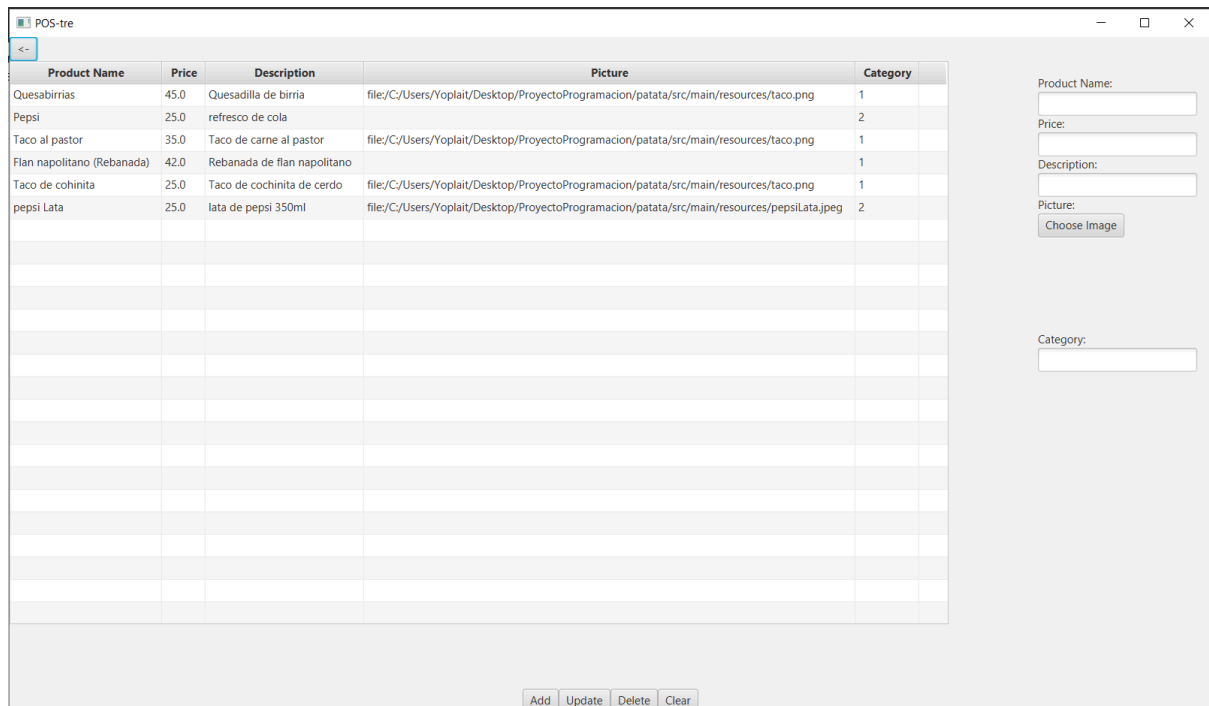
Agregar alimento al sistema

El administrador y solo el administrador podrá modificar la base de datos de alimento para agregar alimentos con las siguientes características: Nombre, categoría, precio, descripción y una imagen;

Prototipo:



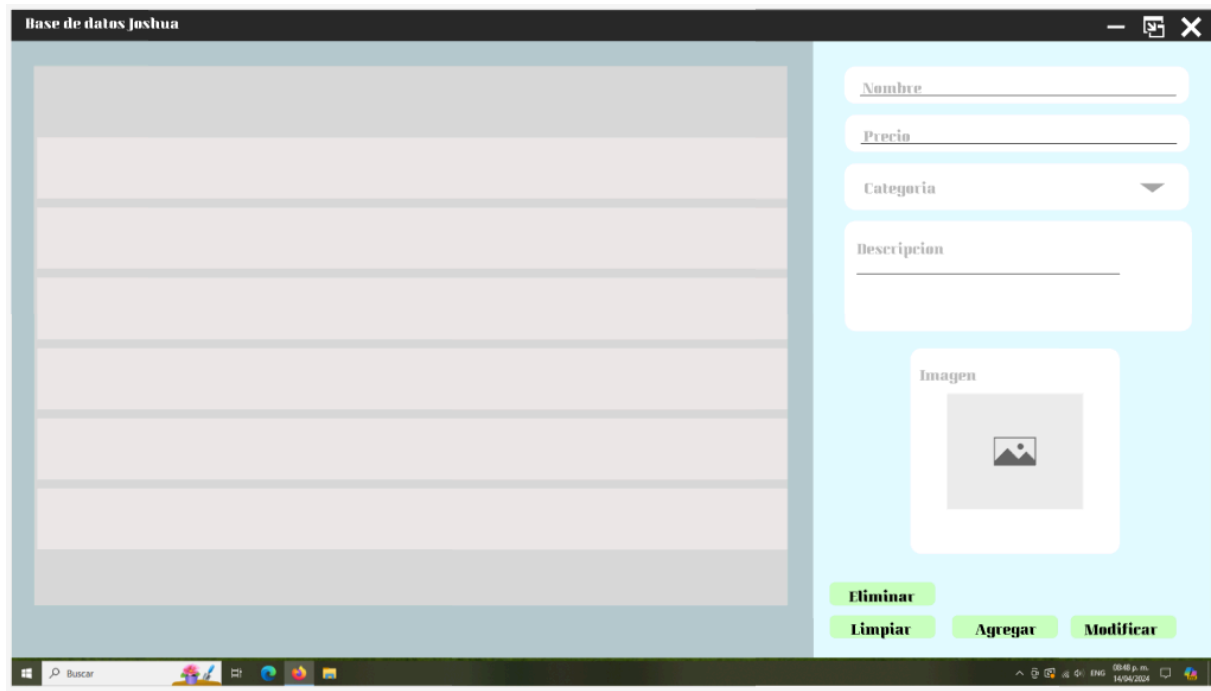
Ventana final:



Eliminar alimento del sistema

El administrador y solo el administrador podrá modificar la base datos de alimento para eliminar alimentos, primero deberá seleccionar uno y darle al botón de eliminar.

Prototipo:



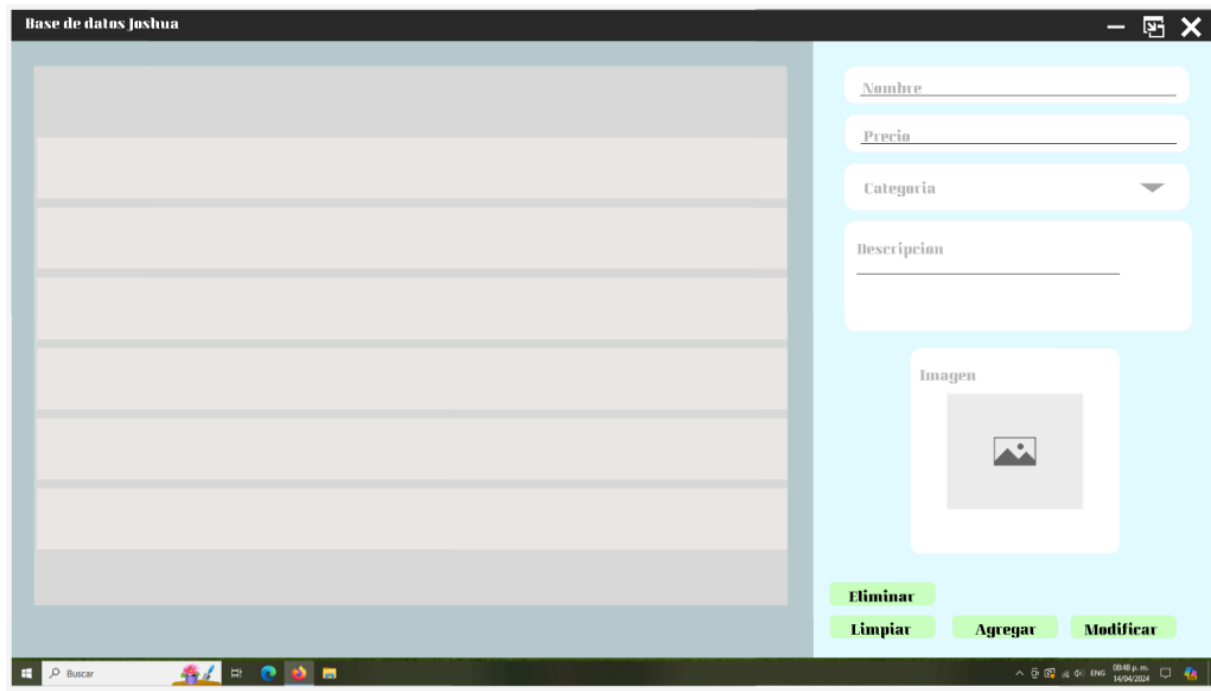
Ventana final:

[illegible]

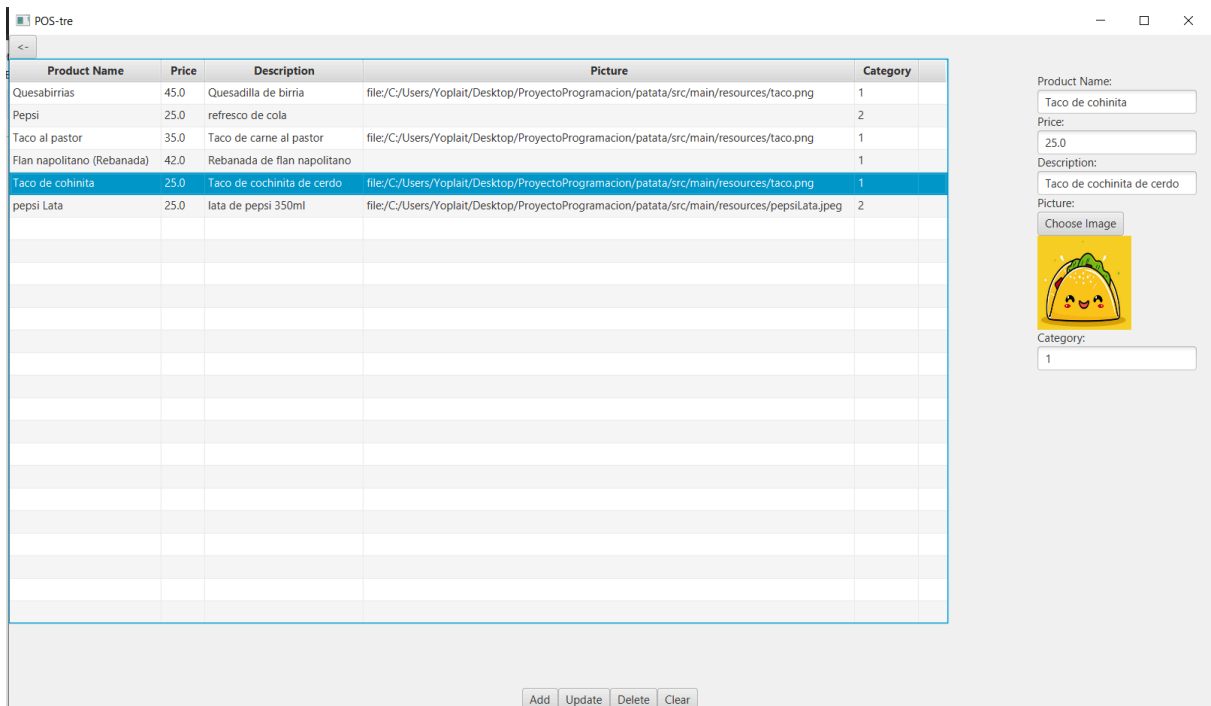
Modificar alimento

El administrador y solo el administrador podrá modificar la base datos de alimento para modificar las siguientes características: Nombre, categoría, precio, descripción y una imagen de los alimentos;

Prototipo:



Ventana final:

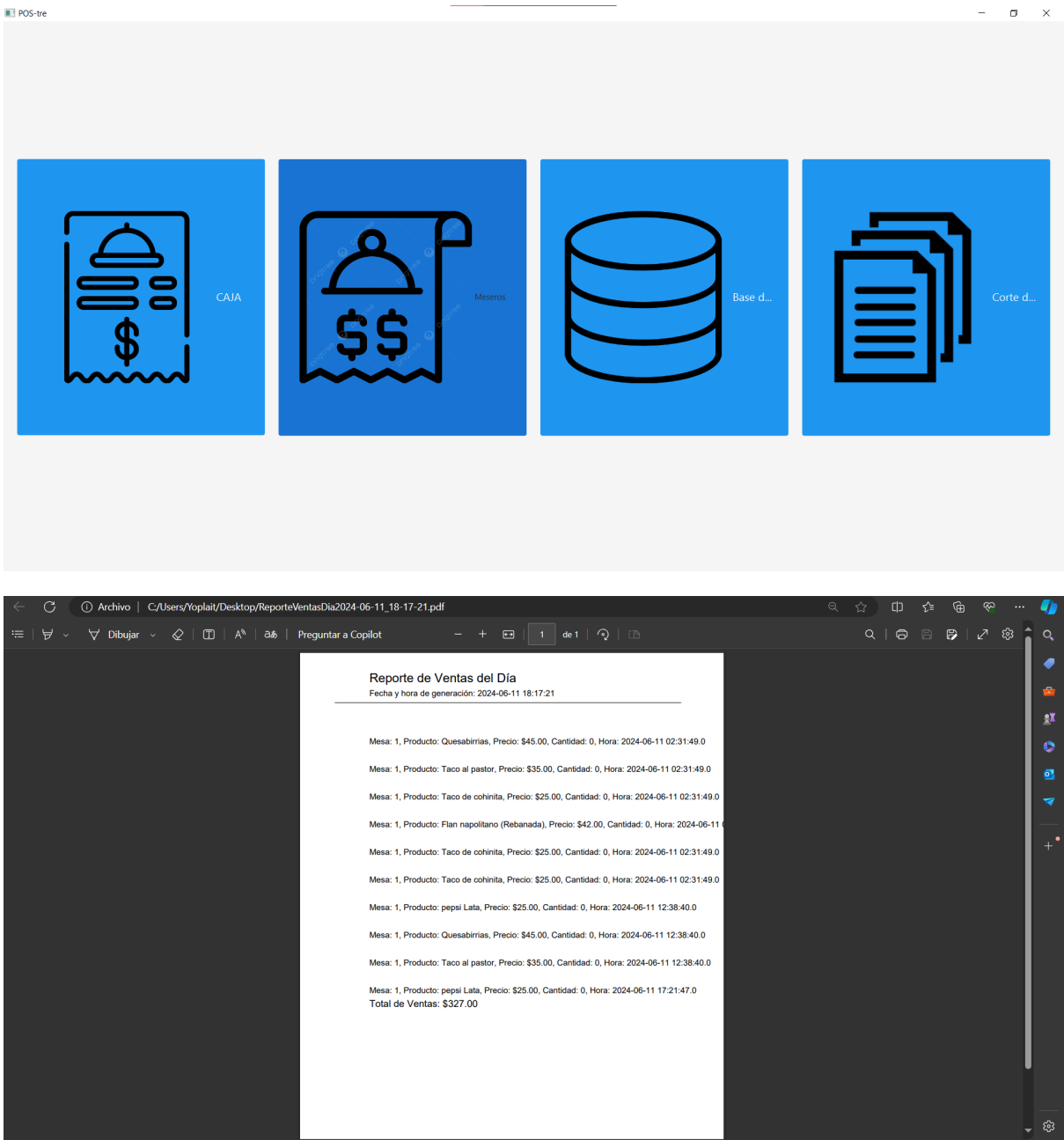


Hacer corte de venta

El administrador será el único en poder hacer el corte de venta al dar clic en el botón de corte, este devolverá un pdf en el escritorio del dispositivo.

Prototipo: No hay

Ventana final:



Cerrar sesión

Una forma sencilla par poder cerrar sesion en caso de que se requiera, algunos menus, cuentan con esta opcion en la esquina superior derecha.

Prototipo: No hay

Ventana final:

POS-tre

<- Mesas

Cerrar sesión

Mesa 1

Mesa 2

Mesa 3

Mesa 4

Mesa 5

Mesa 6

Mesa 7

Cobrar

Diagrama de casos de uso



Diagrama de clases

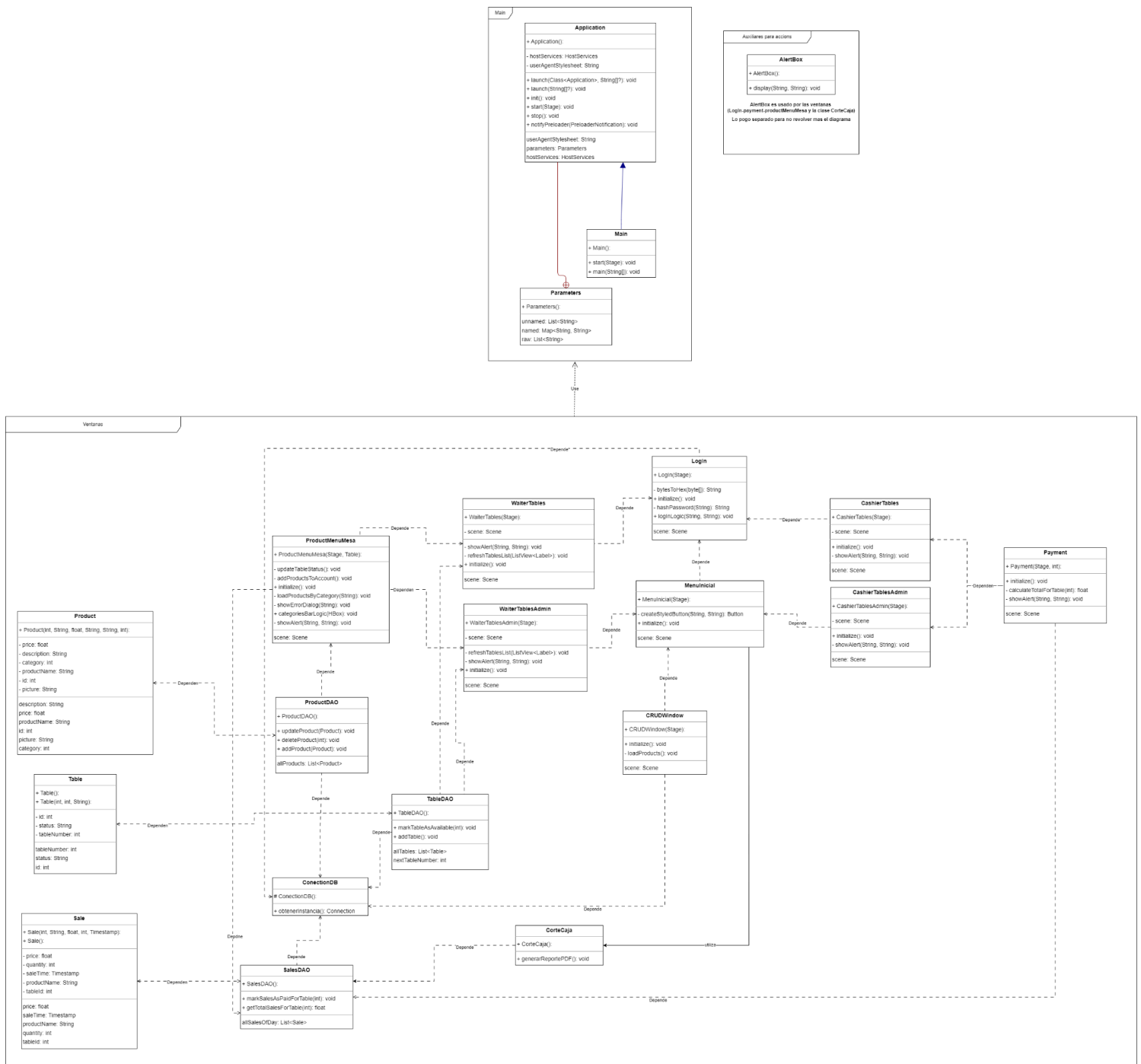
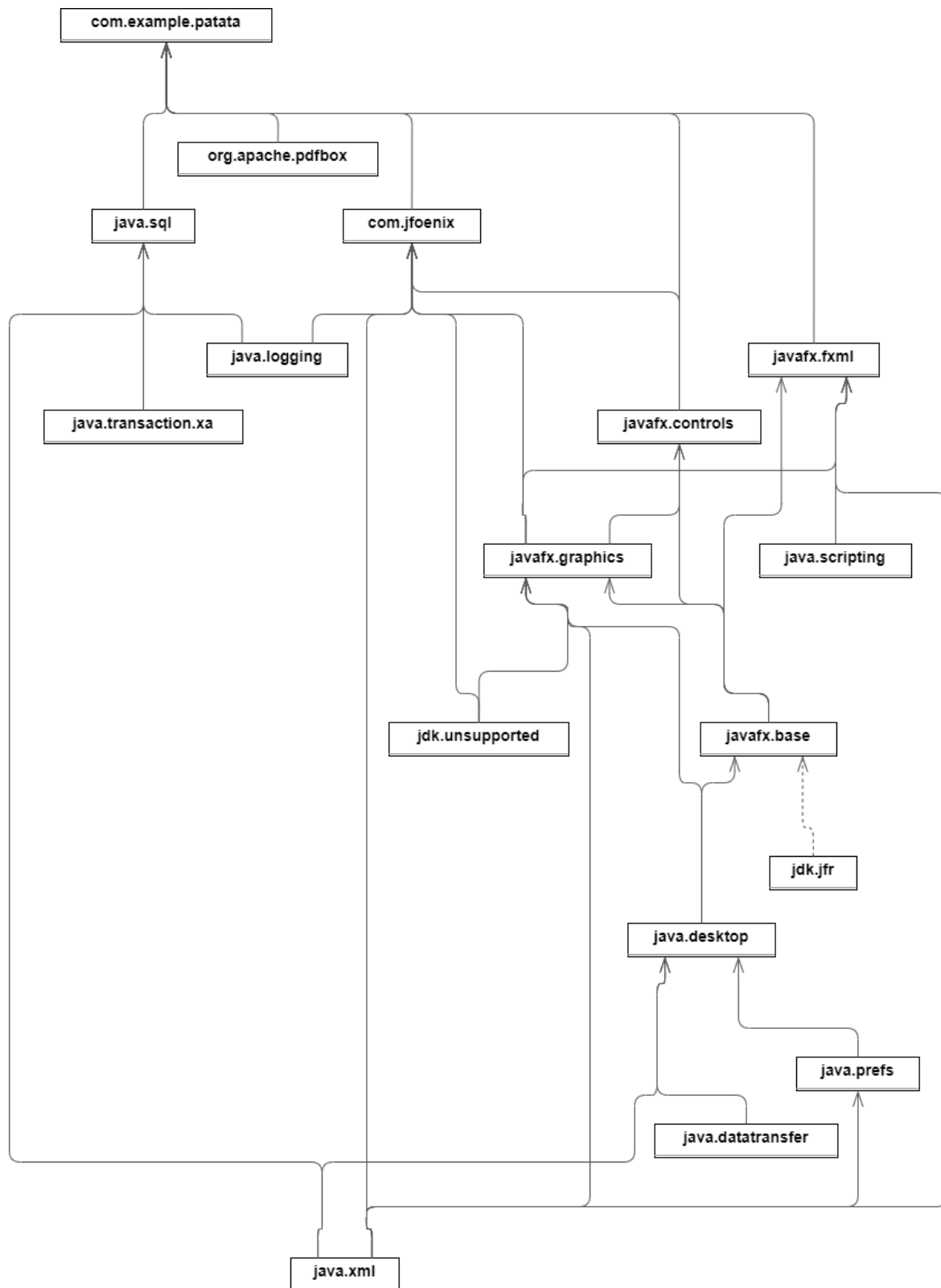


Diagrama de módulos



Descripción diagrama de módulos

El diagrama representa la estructura de paquetes y clases del proyecto POSTre, mostrando las dependencias y relaciones entre ellos. En la parte superior, vemos el nombre del paquete raíz "com.example.patata".

Debajo de este paquete raíz, encontramos "org.apache.pdfbox", que es una biblioteca de código abierto ampliamente utilizada en Java para trabajar con archivos PDF. Esta biblioteca permite crear, modificar, renderizar y extraer contenido de documentos PDF.

A continuación, se ramifican tres secciones principales: "java.sql", "com.jfoenix" y "javafx". La rama "java.sql" contiene clases y paquetes relacionados con la conectividad y el acceso a bases de datos SQL desde aplicaciones Java. Aquí encontramos la clase "java.sql" que proporciona métodos para ejecutar consultas y manipular datos en bases de datos relacionales.

La rama "com.jfoenix" parece ser una biblioteca o marco de trabajo externo, posiblemente utilizado para mejorar la interfaz de usuario o agregar funcionalidades específicas a la aplicación. (Sin embargo por problemas de compatibilidad no fue usado).

La rama "javafx" representa la biblioteca JavaFX, que es un conjunto de paquetes y clases utilizados para crear interfaces de usuario modernas y aplicaciones de escritorio en Java. Debajo de "javafx" se muestran varios subpaquetes y clases, como "javafx.controls", que contiene componentes de interfaz de usuario como botones, campos de texto y listas; "javafx.graphics", que maneja gráficos y renderizado; "javafx.base", que proporciona clases base para JavaFX; "java.scripting", que permite la integración de lenguajes de scripting en Java; y "jdk.unsupported", que contiene clases no respaldadas oficialmente pero que fueron necesarias.

Además, el diagrama incluye otros paquetes y clases estándar de Java, como "java.logging" para el registro de eventos, "java.transaction.xa" para transacciones distribuidas, "java.desktop" para el desarrollo de aplicaciones de escritorio, "java.prefs" para el manejo de preferencias de usuario y "java.datatransfer" para la transferencia de datos entre aplicaciones.

En resumen, este diagrama proporciona una visión general de la estructura de una aplicación Java, mostrando las dependencias entre los diferentes paquetes y bibliotecas, tanto estándar como externas, que se utilizan en el desarrollo del proyecto POSTre.

Diagrama ER base de datos

