



MANUAL DEL PROGRAMADOR

FLORES GARCÍA YAHIR GERARDO 177887

MONTEJANO BRIANO URIEL 177771

PROF. JOSÉ RAMÓN TOLENTINO JIMÉNEZ

PROGRAMACIÓN III – T22A

Contenido

INTRODUCCIÓN	2
INSTALACIÓN	3
Requisitos mínimos:	6
Análisis	7
Requisitos Funcionales:	7
Requisitos No Funcionales:	7
Diseño	8
Diagrama UML.....	8
Diagrama relacional de la Base de Datos.....	8
Codificación.....	9
Clases utilizadas	9
Conexión con la base de datos (Clase Conex).....	9
Clase InterfazLogIn.....	11
Clase InterfazSignIn.....	12
Clase MenuOpcionesVentas	13
Clase InerfazVenta.....	13
Clase HistorialCompras	14
Clase InterazConsultarArticulos	14
Clase InterfazModificarPerfil.....	15
Clase MenuOpcionesAdmin	16
Clase ConsultarVendedores.....	16
Clase AgregarVend.....	17
Clase ConsultarVentas.....	18
Clase GrafVent	19
Clase ConsultarProveedores	19
Clase AgregarProveedor	20
Clase ConsultarClientes	21
Clase ConsultarInventario	22
Clase AgregarArticulo	23
Clase ConsultarSucursales	24
Clase AgregarSucursal.....	25
Pruebas	25
Despedida.....	27

INTRODUCCIÓN

Bienvenido al manual del programador del software de Groove Central, un negocio de comercio minorista especializado en la venta de instrumentos musicales y accesorios de producción musical. Este programa fue diseñado para mejorar la eficiencia en la gestión de inventarios y aumentar la satisfacción del cliente.

En este manual, encontrará información sobre cómo funciona el software y cómo se compone el código para poder hacer las modificaciones necesarias. El programa fue desarrollado utilizando el lenguaje Java y NetBeans como herramienta para programar. Además, se utilizó la Programación Orientada a Objetos para hacer más entendible el funcionamiento del programa y programar distintas características de la vida real.

El software está diseñado para ser utilizado por dos tipos de usuarios: el administrador y el cliente. El cliente podrá registrarse, consultar el catálogo y precios, y realizar compras. El usuario administrador tendrá acceso a toda la información de los clientes y los productos que se encuentran en el inventario, ya sea para agregar productos, modificar precios, entre otras funciones.

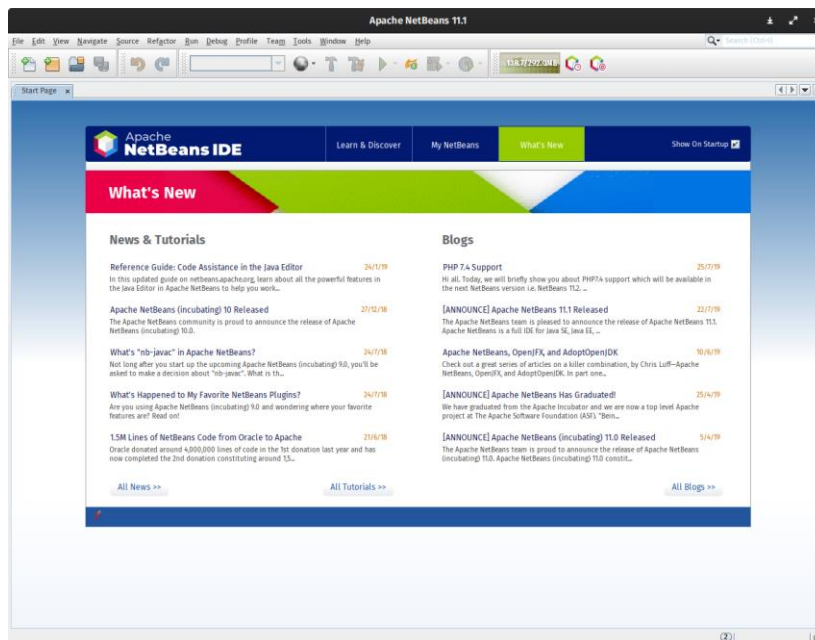
Es importante tener en cuenta que nuestro software se centra exclusivamente en la gestión de inventarios y pedidos, por lo que no incluye funciones de control de calidad o mantenimiento de los instrumentos. Además, el software no tiene pensado un sistema de calendario para identificar las veces que el comprador ha hecho alguna compra y no se implementarán las tablas relacionadas.

En este manual, encontrará información sobre cómo funciona el programa y cómo se compone el código para poder hacer las modificaciones necesarias. Esperamos que este manual le sea de gran ayuda en su tarea como programador y en el desarrollo del software de Groove Central.

INSTALACIÓN

El software de Groove Central está diseñado para funcionar exclusivamente en Windows, sin embargo, necesita de aplicaciones adicionales para que se pueda ejecutar el programa. Para ello siga los siguientes pasos:

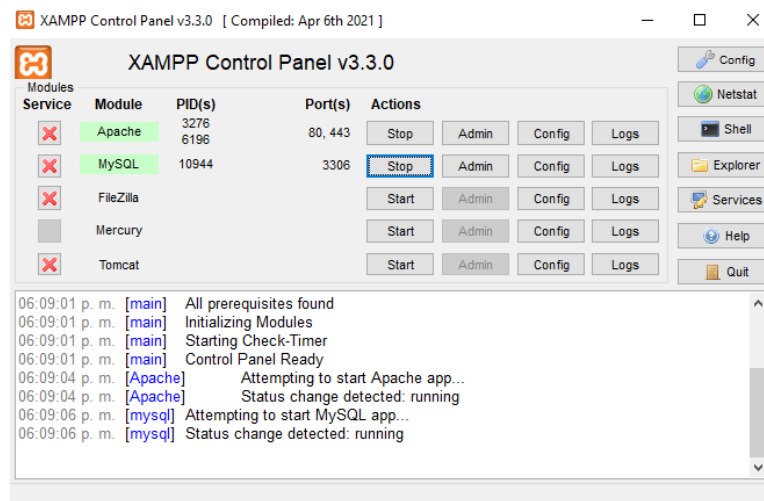
1. Es necesario contar con el software NetBeans. Si ya tiene instalado NetBeans en su equipo, puede saltar esta sección y continuar con la instalación del programa.
 - i. Si no cuenta con el software NetBeans, no se preocupe, es fácil de instalar. En primer lugar, diríjase al sitio web oficial de NetBeans y seleccione la opción de descarga correspondiente a su sistema operativo. Asegúrese de seleccionar la versión más reciente de NetBeans. (<https://netbeans.apache.org/download/index.html>)
 - ii. Una vez descargado el instalador, ejecute el archivo y siga las instrucciones en pantalla para completar la instalación. Es posible que se le solicite seleccionar algunos componentes para instalar, asegúrese de seleccionar "Java SE" para poder ejecutar el programa.



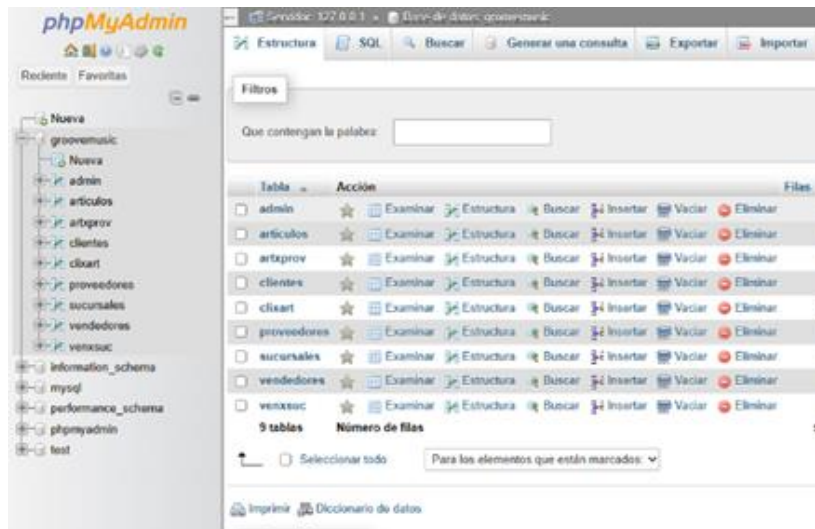
2. También se deberá tener instalado XAMPP en el equipo. XAMPP es un software gratuito que te permitirá instalar un servidor web en tu computadora para que puedas probar y ejecutar aplicaciones web localmente. También incluye un sistema

de gestión de bases de datos llamado phpMyAdmin, el cual permitirá administrar la base de datos que utiliza el programa. Si ya lo tiene instalado, puede saltar esta sección y continuar con la instalación del programa.

- i. Para descargar XAMPP, debes ir a la página oficial de Apache Friends y seleccionar la versión adecuada para tu sistema operativo. Una vez descargado, deberás ejecutar el archivo de instalación y seguir las instrucciones del asistente para completar la instalación. (<https://www.apachefriends.org/es/download.html>)
- ii. Después de haber instalado XAMPP, deberás asegurarte de que los servicios de Apache y MySQL estén iniciados. Para hacer esto, abre el Panel de Control de XAMPP y haz clic en los botones de "Start" junto a los servicios de Apache y MySQL.



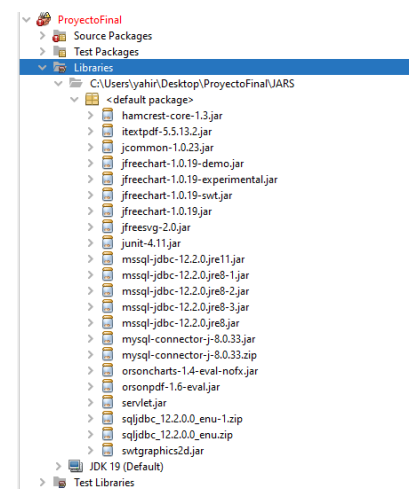
3. Una vez teniendo instalado el XAMPP y el NetBeans, para poder utilizar la base de datos del programa, necesitarás importar el archivo SQL que viene incluido en el proyecto. Para hacer esto, sigue estos pasos:
 - i. Abre el navegador web y escribe "localhost/phpmyadmin" en la barra de direcciones.
 - ii. Inicia sesión en phpMyAdmin con el nombre de usuario y la contraseña que configuraste durante la instalación de XAMPP (opcional).
 - iii. Crea una nueva base de datos con el nombre "groovemusic" (sin comillas).
 - iv. Haz clic en la base de datos recién creada y luego haz clic en el botón "Importar".
 - v. Selecciona el archivo SQL que viene incluido en el proyecto y haz clic en "Continuar".



NOTA: El puerto que se está utilizando puede variar, si se presenta algún problema para acceder se sugiere buscar el número de puerto dentro del panel de control del XAMPP.

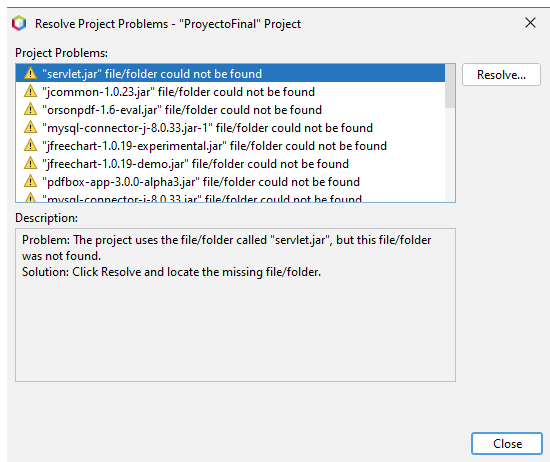
4. Es importante mencionar que para que el programa funcione correctamente, el usuario debe asegurarse de que todas las librerías necesarias estén disponibles en el proyecto de NetBeans. Para ello, es necesario insertar la carpeta "JARS" que ya viene incluida en el proyecto. Esta carpeta contiene todos los archivos JAR necesarios para el correcto funcionamiento del programa. Para insertar la carpeta "JARS", el usuario debe seguir los siguientes pasos:

- i. Abra el proyecto en NetBeans
- ii. Seleccione el apartado "Librerías" en el menú desplegable de la izquierda
- iii. Haga clic en el botón "Agregar JAR/Folder".
- iv. Busque la carpeta "JARS" y selecciónela.
- v. Haga clic en "Aceptar" para agregar la carpeta "JARS" al proyecto.



- vi. Una vez agregadas las librerías, haga clic derecho en el nombre del proyecto y seleccione "Resolver problemas de proyecto".

vii. Seleccione cada librería una por una, haga clic en "Resolver", busque el archivo correspondiente de la carpeta "JARS" y haga clic en Aceptar.



Una vez que se han agregado las librerías necesarias, usted podrá compilar y ejecutar el programa sin problemas. Es importante mencionar que el usuario debe asegurarse de tener todas las librerías necesarias en la carpeta "JARS" para evitar problemas al compilar el programa. Además, como se mencionó anteriormente, el puerto para acceder a la base de datos es variable. Por default se encuentra en el puerto 4306.

Requisitos mínimos:

- Sistema operativo: El programa está diseñado para funcionar exclusivamente en Windows, pero puede probar en Linux o Mac.
- Memoria RAM: Se recomienda al menos 2 GB de memoria RAM para el correcto funcionamiento del programa.
- Espacio en disco: El programa puede ocupar alrededor de 100 MB en el disco duro aproximadamente.
- Versión de Java: Es necesario tener instalada una versión de Java 8 o superior para que el programa funcione correctamente.
- Conexión a internet: El programa requiere una conexión a internet para acceder a la base de datos y permitir la realización de compras en línea.

Análisis

Requisitos Funcionales:

1. Registro de usuarios:
 - El sistema debe permitir a los usuarios crear una cuenta proporcionando información como nombre, apellido, dirección de correo electrónico y contraseña.
 - El sistema debe validar la información del usuario durante el proceso de registro para garantizar la integridad de los datos.
2. Gestión de productos:
 - El sistema debe permitir a los administradores agregar nuevos productos al inventario, incluyendo información como nombre, descripción, precio y cantidad disponible.
 - El sistema debe permitir a los administradores eliminar o modificar la información de los productos existentes en el inventario.
3. Realización de compras:
 - El sistema debe permitir a los clientes agregar productos al carrito de compras y realizar el pago de manera segura.
 - El sistema debe calcular el monto total de la compra y restar la cantidad de productos comprados del inventario.
4. Gestión de usuarios:
 - El sistema debe permitir a los administradores agregar, eliminar y modificar información de vendedores, proveedores y clientes.

Requisitos No Funcionales:

1. Usabilidad:
 - El sistema debe tener una interfaz intuitiva y fácil de usar para que los usuarios puedan navegar y realizar tareas sin dificultad.
 - El sistema debe proporcionar retroalimentación clara y mensajes de error informativos para guiar a los usuarios en caso de errores.
2. Rendimiento:

- El sistema debe ser capaz de manejar una carga de usuarios concurrentes sin degradar significativamente el rendimiento.
 - El tiempo de respuesta del sistema debe ser rápido, especialmente en tareas críticas como la realización de pagos.
3. Disponibilidad:
- El sistema debe estar disponible para su uso las 24 horas del día, los 7 días de la semana, sin tiempo de inactividad planificado para tareas de mantenimiento y actualizaciones. Cualquier mantenimiento o actualización se realizará sin afectar la disponibilidad del sistema.
4. Portabilidad:
- El sistema debe ser compatible con computadoras de escritorio y laptops con los sistemas operativos Windows, aunque también se puede probar en Mac o Linux.
 - El sistema de bases de datos debe ser compatible con los navegadores web más populares en estos sistemas operativos, como Chrome y Firefox.

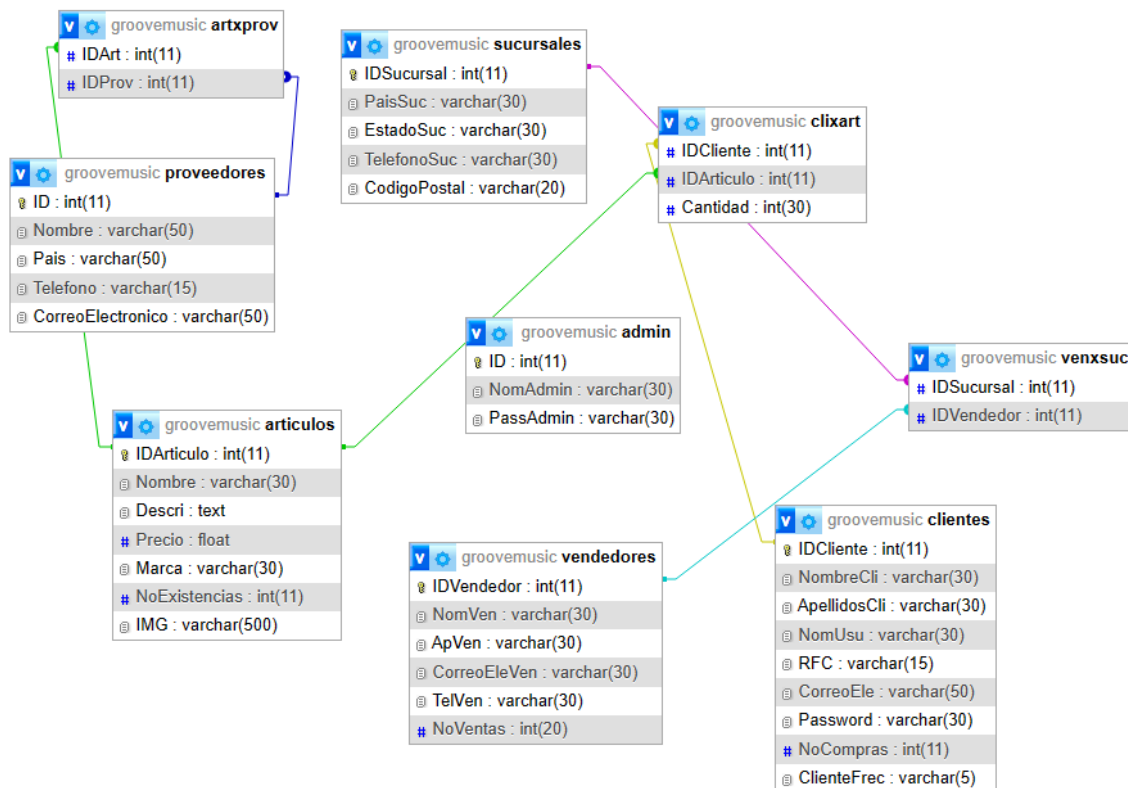
Diseño

Diagrama UML

Para una visualización detallada del diagrama UML completo, incluido en el diseño del sistema, le recomendamos acceder a la versión ampliada en línea. Puede encontrar el diagrama completo en el siguiente enlace: [\[Diagrama\]](#). Al hacer clic en el enlace, se abrirá una ventana nueva o se descargará el archivo con el diagrama para su revisión detallada.

Diagrama relacional de la Base de Datos

El siguiente diagrama representa el diseño relacional de la base de datos utilizada en el sistema. Este diagrama describe la estructura de las tablas y las relaciones entre ellas en la base de datos. Proporciona una visión general de cómo se organiza y almacena la información en el sistema. El diseño relacional de la base de datos es esencial para garantizar la integridad y la eficiencia en la gestión de los datos del sistema. A continuación, se muestra el diagrama relacional de la base de datos:



Este diagrama proporciona una representación visual de las tablas de la base de datos y las relaciones entre ellas, lo cual es fundamental para el correcto funcionamiento del sistema y la gestión de la información.

Codificación

Clases utilizadas

Conexión con la base de datos (Clase Conex)

El código proporcionado más adelante muestra la clase "Conex" que se encarga de establecer una conexión con una base de datos MySQL y realizar consultas a la tabla "admin". A continuación, se explicarán los componentes y el funcionamiento de este código.

La clase "Conex" comienza con un constructor que se encarga de establecer la conexión con la base de datos. Utiliza la biblioteca "com.mysql.jdbc.Driver" para cargar el controlador JDBC necesario para la conexión. Luego, se utiliza el método "getConnection" de la clase

"DriverManager" para establecer la conexión con la base de datos "groovemusic" en el servidor local. Se utiliza el usuario "root" y no se proporciona una contraseña.

El método principal de la clase, denominado "main", se encarga de realizar una consulta a la tabla "admin" y mostrar los resultados por la salida estándar. Comienza instanciando un objeto de la clase "Conex" para obtener una conexión a la base de datos. A continuación, se crea un objeto de tipo "Statement" para ejecutar la consulta SQL y se obtiene un objeto de tipo "ResultSet" que almacena los resultados de la consulta.

Dentro de un bucle while, se recorren los registros del resultado y se imprimen en la salida estándar utilizando el método "getInt", "getString" y el nombre de las columnas correspondientes ("IDAdmin", "NomAdmin" y "PassAdmin" en este caso).

Finalmente, se cierra la conexión con la base de datos mediante el método "close" del objeto "Connection".

```
package Conexion;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.Connection;
import java.sql.DriverManager;

public class Conex {
    Connection con;
    public Conex(){
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost:4306/groovemusic", "root", "");
        } catch (Exception e) {
            System.err.println("Error: " +e);
        }
    }

    public void main(String[] args) {
        Conex cn = new Conex();
        Statement st;
        ResultSet rs;
        try {
            st=cn.con.createStatement();
            rs=st.executeQuery("select * from admin");
            while (rs.next()) {
                System.out.println(rs.getInt("IDAdmin")+ " " +rs.getString("NomAdmin")+ " " +rs.getString("PassAdmin"));
            }
            cn.con.close();
        } catch (Exception e) {
            System.out.println("Error: " +e);
        }
    }
}
```

A continuación, se presentan todas las clases principales del proyecto. Cabe destacar que todas heredan de la clase JFrame que proporciona una ventana en blanco donde se pueden agregar componentes visuales

Clase InterfazLogIn

La clase "InterfazLogIn" es una ventana de inicio de sesión que permite a los usuarios ingresar sus credenciales para acceder a la aplicación.

El código comienza con la declaración de dos variables estáticas "Usuarioe" y "Contrae" para almacenar el nombre de usuario y la contraseña ingresados por el usuario.

El constructor de la clase, llamado "InterfazLogIn", inicializa la interfaz gráfica llamando al método "initComponents()".

El método "initComponents()" se genera automáticamente y contiene el código que define y configura todos los componentes visuales de la ventana. Estos componentes incluyen etiquetas de texto ("lblUsuario" y "lblContra"), campos de texto ("tfUsu" y "pass"), botones de inicio de sesión ("btnLogIn") y de registro ("btnSign"), y otras etiquetas decorativas. El diseño y las propiedades de estos componentes se establecen utilizando el administrador de diseño GroupLayout.

La clase también contiene métodos de eventos para manejar las acciones del usuario. El método "btnSignActionPerformed" se ejecuta cuando se hace clic en el botón "btnSign" de registro. Crea una instancia de la clase "InterfazSignIn" y la muestra en pantalla, luego cierra la ventana de inicio de sesión actual.

El método "btnLogInActionPerformed" se ejecuta cuando se hace clic en el botón "btnLogIn" de inicio de sesión. Obtiene el nombre de usuario y la contraseña ingresados por el usuario, y luego establece una conexión con la base de datos utilizando la clase "Conexion" (definida en el archivo Conex). A continuación, se realizan consultas a las tablas "admin" y "clientes" para verificar si existe una coincidencia entre el nombre de usuario y la contraseña ingresados. Si se encuentra una coincidencia en la tabla "admin", se muestra la ventana "MenuOpcionesAdmin". Si se encuentra una coincidencia en la tabla "clientes", se muestra la ventana "MenuOpcionesVentas". Si no se ingresan datos en los campos de usuario y contraseña, se muestra un mensaje de alerta.

El método "main" es el punto de entrada del programa y crea una instancia de la clase "InterfazLogIn" para mostrar la ventana de inicio de sesión.

Clase InterfazSignIn

La clase "InterfazSignIn" representa una ventana de registro en la aplicación Groove Central.

La clase comienza con la declaración de variables estáticas para almacenar la información ingresada por el usuario durante el proceso de registro, como "NombreUsuario" y "Contraseña".

El constructor de la clase, llamado "InterfazSignIn", inicializa la interfaz gráfica mediante la llamada al método " initComponents()".

El método " initComponents()" es generado automáticamente y contiene el código que define y configura todos los componentes visuales de la ventana de registro. Estos componentes incluyen etiquetas de texto para describir los campos de entrada, como "lblNombre", "lblApellido", "lblRFC", "lblCorreo" y "lblContraseña". Además, se incluyen campos de texto correspondientes, como "tfNombre", "tfApellido", "tfRFC", "tfCorreo" y "passContraseña", que permiten al usuario ingresar los datos necesarios. También se agregan botones, como el botón "btnRegistrar" para registrar la cuenta y el botón "btnCancelar" para cancelar el proceso de registro.

La clase también contiene métodos de eventos para manejar las acciones del usuario. Por ejemplo, el método " btnRegistrarActionPerformed" se ejecuta cuando el usuario hace clic en el botón "btnRegistrar". En este método, se obtienen los datos ingresados por el usuario en los campos de texto y se validan. Luego, se establece una conexión con la base de datos utilizando la clase "Conexion" (definida en el archivo "Conex"), y se insertan los datos en la tabla correspondiente, como "clientes". Si la inserción es exitosa, se muestra un mensaje de confirmación al usuario y se cierra la ventana de registro. En caso de que falten datos en los campos de texto, se muestra un mensaje de alerta al usuario.

El método " btnCancelarActionPerformed" se ejecuta cuando el usuario hace clic en el botón "btnCancelar". Este método simplemente cierra la ventana de registro sin realizar ninguna acción adicional.

No se proporciona un método "main" en la descripción, ya que normalmente se utiliza para iniciar la aplicación y mostrar la ventana de inicio de sesión.

Clase MenuOpcionesVentas

La clase "MenuOpcionesVentas" es una ventana de menú para los clientes de Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen botones para consultar artículos ("btnConArt"), cerrar sesión ("btnLogOut"), realizar una compra ("bntReaVen"), modificar el perfil ("btnModPer") y ver el historial de compras ("btnHistC"). También se define una etiqueta de texto ("lblMenuCli") que muestra el título del menú.

El constructor de la clase, llamado "MenuOpcionesVentas", inicializa la interfaz gráfica llamando al método "initComponents()".

La clase también contiene métodos de eventos para manejar las acciones del usuario. Estos métodos se ejecutan cuando se hace clic en los botones correspondientes. Por ejemplo, el método "bntReaVenActionPerformed" se ejecuta cuando se hace clic en el botón "bntReaVen" de realizar compra. En este caso, se crea una instancia de la clase "InerfazVenta" y se muestra en pantalla, luego se cierra la ventana actual.

Clase InerfazVenta

La clase "InerfazVenta" es una ventana de interfaz de usuario para realizar compras en la aplicación Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen etiquetas de texto ("lblCant", "lblProd", "lblTotal"), campos de texto ("tfCant", "tfTotal"), botones ("btnConfirm", "btnAgCar", "btnBackMenu") y una tabla ("tbAgAr") para mostrar los artículos agregados al carrito de compra. También se define un combo box ("cbProd") para seleccionar el producto a agregar al carrito.

El constructor de la clase, llamado "InerfazVenta", inicializa la interfaz gráfica llamando al método "initComponents()". Dentro del constructor se establece una conexión con la base de datos para obtener los datos de los artículos y cargarlos en el combo box "cbProd". También se inicializa un modelo de tabla ("model") para la tabla "tbAgAr".

La clase también contiene métodos de eventos para manejar las acciones del usuario. Por ejemplo, el método "btnConfirmActionPerformed" se ejecuta cuando se hace clic en el botón

"btnConfirm" para confirmar la compra. Dentro de este método, se realiza una verificación de datos y se muestra un cuadro de diálogo de confirmación al usuario. Si el usuario confirma la compra, se realiza una serie de operaciones, como obtener el ID del cliente, actualizar el número de compras del cliente, actualizar las existencias del producto en la base de datos y registrar la compra en la tabla "clixart".

Clase HistorialCompras

La clase "HistorialCompras" es una ventana de interfaz de usuario que muestra el historial de compras de un cliente en la aplicación Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen una etiqueta de texto ("jLabel1") para el título "Historial de Compras" y una tabla ("tbHistC") para mostrar los detalles de las compras. También se define un botón ("btnBackMenu") para regresar al menú principal.

Dentro del constructor de la clase "HistorialCompras", se establece una conexión con la base de datos utilizando el controlador JDBC para MySQL. Se ejecutan consultas SQL para obtener los datos relevantes de la base de datos y se muestran en la tabla.

El método "btnBackMenuActionPerformed" se ejecuta cuando se hace clic en el botón "Regresar al menu". Dentro de este método, se crea una instancia de la clase "MenuOpcionesVentas" para volver al menú principal y se cierra la ventana actual.

Clase InterazConsultarArticulos

La clase "InterazConsultarArticulos" es una ventana de interfaz de usuario que muestra una tabla con los artículos disponibles en la aplicación Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen una etiqueta de texto ("jLabel1") para el título "Consultar Articulos", una tabla ("tbArticulos") para mostrar los detalles de los artículos y un botón ("btnBackMenu") para regresar al menú principal.

Dentro del constructor de la clase "InterazConsultarArticulos", se establece una conexión con la base de datos utilizando el controlador JDBC para MySQL. Se ejecuta una consulta

SQL para obtener los datos de los artículos y se muestra en la tabla. Se crea un modelo de tabla ("DefaultTableModel") y se agrega a la tabla "tbArticulos".

El método "btnBackMenuActionPerformed" se ejecuta cuando se hace clic en el botón "Regresar al menú". Dentro de este método, se crea una instancia de la clase "MenuOpcionesVentas" para volver al menú principal y se cierra la ventana actual.

Clase InterfazModificarPerfil

La clase "InterfazModificarPerfil" es una ventana de interfaz gráfica para modificar el perfil de un cliente en la aplicación Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen etiquetas de texto para mostrar información del perfil actual del cliente, como nombre, apellidos, nombre de usuario, correo electrónico, contraseña y RFC. También se define un botón para regresar al menú principal ("btnBackMenu"), un combobox para seleccionar el dato a modificar ("cbDatMod"), una etiqueta de texto para indicar qué dato se va a modificar ("lblDatMod"), un campo de texto para ingresar el nuevo valor ("tfValMod") y un botón para realizar la modificación ("btnMod").

El constructor de la clase, llamado "InterfazModificarPerfil", inicializa la interfaz gráfica llamando al método " initComponents() ". Luego, se muestra la información del perfil actual del cliente consultando la base de datos utilizando los valores de usuario y contraseña almacenados en la instancia de la clase "InterfazLogIn" (IL). Se establece el texto de las etiquetas de texto correspondientes con los valores obtenidos de la consulta.

También se añaden elementos al combobox "cbDatMod" para que el cliente pueda seleccionar qué dato desea modificar. Los posibles datos que modificar son: nombre, apellidos, nombre de usuario, RFC, correo electrónico y contraseña.

La clase contiene métodos de eventos para manejar las acciones del usuario. Por ejemplo, el método "btnBackMenuActionPerformed" se ejecuta cuando se hace clic en el botón "btnBackMenu" para regresar al menú principal. En este caso, se crea una instancia de la clase "MenuOpcionesVentas" y se muestra en pantalla, luego se cierra la ventana actual.

El método "btnModActionPerformed" se ejecuta cuando se hace clic en el botón "btnMod" para realizar la modificación del perfil. Se obtiene el índice seleccionado en el combobox

"cbDatMod" y se recupera el nuevo valor ingresado en el campo de texto "tfValMod". Luego, se realiza una actualización en la base de datos según el dato seleccionado y se ejecutan las consultas correspondientes. Dependiendo del dato seleccionado, puede haber acciones adicionales, como cerrar sesión en caso de modificar el nombre de usuario o contraseña. En ese caso, se muestra un mensaje de alerta y se crea una instancia de la clase "InterfazLogIn" para que el cliente pueda iniciar sesión nuevamente.

Clase MenuOpcionesAdmin

La clase "MenuOpcionesAdmin" es una ventana de menú para los administradores de Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen botones para consultar vendedores ("btnConVendedores"), consultar proveedores ("btnConProv"), consultar ventas ("btnConVent"), consultar clientes ("btnConCli"), consultar inventario ("jButton1") y consultar sucursales ("jButton2"). También se define una etiqueta de texto ("lblMenuAdmin") que muestra el título del menú.

El constructor de la clase, llamado "MenuOpcionesAdmin", inicializa la interfaz gráfica llamando al método " initComponents()".

La clase también contiene métodos de eventos para manejar las acciones del usuario. Estos métodos se ejecutan cuando se hace clic en los botones correspondientes. Por ejemplo, el método "btnLogoutActionPerformed" se ejecuta cuando se hace clic en el botón "btnLogout" de cerrar sesión. En este caso, se crea una instancia de la clase "InterfazLogIn" y se muestra en pantalla, luego se cierra la ventana actual.

Clase ConsultarVendedores

La clase "ConsultarVendedores" es una ventana de consulta de vendedores en la aplicación Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen una tabla para mostrar los vendedores ("tbVendedores"), una etiqueta de título ("lblConVen"), botones para regresar al menú principal ("btnBM") y agregar un nuevo vendedor

("btnAgregarVendedor"), y campos de texto y etiquetas para modificar y eliminar registros de vendedores.

El constructor de la clase, llamado "ConsultarVendedores", inicializa la interfaz gráfica llamando al método " initComponents() ". También realiza una consulta a la base de datos para obtener la información de los vendedores y mostrarla en la tabla.

La clase también contiene métodos de eventos para manejar las acciones del usuario. Por ejemplo, el método " btnBMActionPerformed " se ejecuta cuando se hace clic en el botón " btnBM " de regresar al menú principal. En este caso, se crea una instancia de la clase " MenuOpcionesAdmin " y se muestra en pantalla, luego se cierra la ventana actual.

El método " btnAgregarVendedorActionPerformed " se ejecuta cuando se hace clic en el botón " btnAgregarVendedor " de agregar un nuevo vendedor. En este caso, se crea una instancia de la clase " AgregarVend " y se muestra en pantalla, luego se cierra la ventana actual.

El método " btnModActionPerformed " se ejecuta cuando se hace clic en el botón " btnMod " de modificar un registro. Realiza una consulta a la base de datos para actualizar el valor del campo seleccionado del vendedor especificado.

El método " btnDelActionPerformed " se ejecuta cuando se hace clic en el botón " btnDel " de eliminar un registro. Realiza una consulta a la base de datos para eliminar el vendedor especificado y su asignación a una sucursal.

El método " main " es el punto de entrada de la aplicación. No realiza ninguna operación adicional y se puede utilizar para ejecutar la clase " ConsultarVendedores " de forma independiente.

Clase AgregarVend

La clase " AgregarVend " es una ventana para agregar vendedores en la aplicación Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen etiquetas de texto (" lblAgVen ", " jLabel1 ", " jLabel2 ", " jLabel3 ", " jLabel4 ", " jLabel5 "), campos de texto (" tfNom ", " tfAp ", " tfCE ", " tfTel ", " tfNV "), botones (" btnBackVend ", " btnAgVen ", " btnVendSuc ") y una tabla para mostrar los vendedores (" tbVend ").

El constructor de la clase, llamado "AgregarVend", inicializa la interfaz gráfica llamando al método "initComponents()". También realiza una consulta a la base de datos para obtener la información de los vendedores y mostrarla en la tabla.

La clase también contiene métodos de eventos para manejar las acciones del usuario. El método "btnAgVenActionPerformed" se ejecuta cuando se hace clic en el botón "btnAgVen" para agregar un nuevo vendedor. En este caso, se obtienen los valores de los campos de texto y se realiza una consulta a la base de datos para insertar un nuevo registro de vendedor.

El método "btnBackVendActionPerformed" se ejecuta cuando se hace clic en el botón "btnBackVend" para regresar a la ventana de consulta de vendedores. En este caso, se crea una instancia de la clase "ConsultarVendedores" y se muestra en pantalla, luego se cierra la ventana actual.

El método "btnVendSucActionPerformed" se ejecuta cuando se hace clic en el botón "btnVendSuc" para asignar un vendedor a una sucursal. En este caso, se crea una instancia de la clase "VendSuc" y se muestra en pantalla, luego se cierra la ventana actual.

El método "main" es el punto de entrada de la aplicación. No realiza ninguna operación adicional y se puede utilizar para ejecutar la clase "AgregarVend" de forma independiente.

Clase ConsultarVentas

La clase "ConsultarVentas" es una ventana que permite consultar las ventas en el sistema Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen etiquetas de texto ("lblConVen" y "lblGT"), botones para regresar al menú principal ("btnBM") y mostrar una gráfica ("btnGraf"), y dos tablas ("tbConVen" y "tbGan") para mostrar la información de las ventas.

El constructor de la clase, llamado "ConsultarVentas", inicializa la interfaz gráfica llamando al método "initComponents()". Además, realiza consultas a la base de datos para obtener información sobre las ventas y actualiza las tablas correspondientes con los resultados.

La clase también contiene métodos de eventos para manejar las acciones del usuario. El método "btnBMActionPerformed" se ejecuta cuando se hace clic en el botón "btnBM" para regresar al menú principal. En este caso, se crea una instancia de la clase "MenuOpcionesAdmin" y se muestra en pantalla, luego se cierra la ventana actual.

El método "btnGrafActionPerformed" se ejecuta cuando se hace clic en el botón "btnGraf" para mostrar una gráfica de las ventas. En este caso, se crea una instancia de la clase "GrafVent" y se muestra en pantalla, luego se cierra la ventana actual.

El método "main" crea una instancia de la clase "ConsultarVentas" y la muestra en pantalla.

Clase GrafVent

La clase "GrafVent" es una ventana que muestra un gráfico de las ventas en el sistema Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen una etiqueta de texto ("jLabel1"), un botón para regresar a la ventana de consultas de ventas ("btnBV"), y un panel ("PanGraf") donde se muestra el gráfico.

El constructor de la clase, llamado "GrafVent", inicializa la interfaz gráfica llamando al método "initComponents()". Además, realiza consultas a la base de datos para obtener información sobre las ventas y crea un gráfico circular utilizando la biblioteca JFreeChart.

El gráfico muestra la cantidad de ventas para cada artículo en forma de porcentaje. Los datos se obtienen mediante una consulta a la base de datos, y se van agregando al gráfico utilizando el objeto DefaultPieDataset.

El método "btnBVActionPerformed" se ejecuta cuando se hace clic en el botón "btnBV" para regresar a la ventana de consultas de ventas. En este caso, se crea una instancia de la clase "ConsultarVentas" y se muestra en pantalla, luego se cierra la ventana actual.

El método "main" crea una instancia de la clase "GrafVent" y la muestra en pantalla.

Clase ConsultarProveedores

La clase "ConsultarProveedores" es una ventana de consulta de proveedores en Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen una etiqueta de título ("lblConProv"), un botón para regresar al menú ("btnBM"), una tabla para mostrar los proveedores ("tbProv"), un botón para añadir proveedores ("jButton1"), campos de texto y etiquetas para realizar modificaciones ("cbValMod", "tfID", "tfMod", "btnMod", "tfIDE", "btnDel").

El constructor de la clase, llamado "ConsultarProveedores", inicializa la interfaz gráfica llamando al método "initComponents()". Dentro del constructor, se configura el ComboBox "cbValMod" con opciones de selección para modificar proveedores.

La clase también contiene métodos de eventos para manejar las acciones del usuario. Por ejemplo, el método "btnBMActionPerformed" se ejecuta cuando se hace clic en el botón "btnBM" para regresar al menú principal. En este caso, se crea una instancia de la clase "MenuOpcionesAdmin" y se muestra en pantalla, luego se cierra la ventana actual.

El método "jButton1ActionPerformed" se ejecuta cuando se hace clic en el botón "jButton1" para añadir un proveedor. En este caso, se crea una instancia de la clase "AgregarProveedor" y se muestra en pantalla, luego se cierra la ventana actual.

El método "btnModActionPerformed" se ejecuta cuando se hace clic en el botón "btnMod" para realizar una modificación en los datos de un proveedor. Se obtienen los valores seleccionados y los datos ingresados por el usuario, se establece la conexión a la base de datos y se ejecuta una consulta de actualización para modificar los datos del proveedor seleccionado.

El método "btnDelActionPerformed" se ejecuta cuando se hace clic en el botón "btnDel" para eliminar un proveedor. Se obtiene el ID del proveedor a eliminar, se establece la conexión a la base de datos y se ejecutan consultas de eliminación en las tablas correspondientes.

El método "main" se encarga de iniciar la aplicación y mostrar la ventana de consulta de proveedores al ejecutar el programa.

Clase AgregarProveedor

La clase "AgregarProveedor" es una ventana para agregar proveedores en Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen una etiqueta de título ("lblAgreProv"), un botón para regresar a la ventana de consulta de proveedores ("jButton1"), etiquetas y campos de texto para ingresar los datos del proveedor ("lblNomP", "lblPais", "lblTel", "lblCorrEle", "tfNomP", "tfPa", "tfTel", "tfCoEl"), y un botón para registrar el proveedor ("btnRegPro").

El constructor de la clase, llamado "AgregarProveedor", inicializa la interfaz gráfica llamando al método " initComponents()".

La clase también contiene métodos de eventos para manejar las acciones del usuario. Por ejemplo, el método "btnRegProActionPerformed" se ejecuta cuando se hace clic en el botón "btnRegPro" para registrar un proveedor. En este caso, se obtienen los valores ingresados por el usuario en los campos de texto, se establece la conexión a la base de datos y se ejecuta una consulta de inserción para agregar el proveedor.

El método "jButton1ActionPerformed" se ejecuta cuando se hace clic en el botón "jButton1" para regresar a la ventana de consulta de proveedores. En este caso, se crea una instancia de la clase "ConsultarProovedores" y se muestra en pantalla, luego se cierra la ventana actual.

El método "main" se encarga de iniciar la aplicación y mostrar la ventana para agregar proveedores al ejecutar el programa.

Clase ConsultarClientes

La clase "ConsultarClientes" es una ventana de consulta de clientes en la aplicación Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen una etiqueta de título ("lblConCli"), un botón para regresar al menú ("btnBm"), una tabla para mostrar los clientes ("tbConCli"), una etiqueta y un campo de texto para eliminar un cliente ("jLabel1" y "tfIDE", respectivamente), y un botón para realizar la eliminación ("jButton1").

El constructor de la clase, llamado "ConsultarClientes", inicializa la interfaz gráfica llamando al método " initComponents()". Dentro del constructor, se establece una conexión a la base de datos "groovemusic" en MySQL y se ejecuta una consulta para obtener todos los clientes.

Los resultados de la consulta se muestran en la tabla utilizando un modelo de tabla predeterminado.

La clase también contiene métodos de eventos para manejar las acciones del usuario. El método "btnBmActionPerformed" se ejecuta cuando se hace clic en el botón "btnBm" para regresar al menú de opciones de administrador. En este caso, se crea una instancia de la clase "MenuOpcionesAdmin" y se muestra en pantalla, luego se cierra la ventana actual.

El método "jButton1ActionPerformed" se ejecuta cuando se hace clic en el botón "jButton1" para eliminar un cliente. El ID del cliente a eliminar se obtiene del campo de texto "tfIDE". Se establece una conexión a la base de datos y se ejecutan las consultas necesarias para eliminar el cliente tanto de la tabla "clixart" como de la tabla "clientes". Si la eliminación se realiza correctamente, se muestra un mensaje de confirmación.

El método "main" crea una instancia de la clase "ConsultarClientes" y la muestra en pantalla cuando se ejecuta el programa.

Clase ConsultarInventario

La clase "ConsultarInventario" es una ventana de consulta de inventario en el sistema Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen una etiqueta de título ("jLabel1"), un botón para regresar al menú ("jButton1"), una tabla para mostrar los artículos del inventario ("tbArticulos"), un botón para agregar un nuevo artículo ("btnAgArt"), un combo box para seleccionar un artículo ("cbArt"), una etiqueta para mostrar el nombre del artículo seleccionado ("lblArt"), una etiqueta para ingresar la cantidad de existencias a agregar ("lblCant"), un campo de texto para ingresar la cantidad de existencias a agregar ("tfEx"), un botón para agregar las existencias ("btnEx"), una etiqueta de título para eliminar registros ("jLabel4"), un campo de texto para ingresar el ID del registro a eliminar ("tfIDE"), y un botón para eliminar el registro ("btnDel").

El constructor de la clase, llamado "ConsultarInventario", inicializa la interfaz gráfica llamando al método "initComponents()". En este constructor también se realiza la conexión a la base de datos y se ejecutan consultas para obtener los datos de los artículos del

inventario y los proveedores asociados a cada artículo. Estos datos se muestran en la tabla "tbArticulos" utilizando un objeto "DefaultTableModel".

La clase también contiene métodos de eventos para manejar las acciones del usuario. Por ejemplo, el método "jButton1ActionPerformed" se ejecuta cuando se hace clic en el botón "jButton1" de regresar al menú. En este caso, se crea una instancia de la clase "MenuOpcionesAdmin" y se muestra en pantalla, luego se cierra la ventana actual.

Clase AgregarArticulo

La clase "AgregarArticulo" es una ventana para agregar un nuevo artículo al sistema Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen una etiqueta de título ("jLabel1"), etiquetas y campos de texto para ingresar los detalles del artículo como nombre, descripción, precio, marca, número de existencias y URL de la imagen. También hay un botón para agregar el artículo ("btnRA"), un botón para regresar a la lista de artículos ("btnBA"), y un combo box para seleccionar el proveedor del artículo ("cbProv").

El constructor de la clase, llamado "AgregarArticulo", inicializa la interfaz gráfica llamando al método " initComponents() ". Dentro del constructor, se establece una conexión a la base de datos y se ejecuta una consulta para obtener los nombres de los proveedores, que se agregan al combo box "cbProv".

La clase también contiene métodos de eventos para manejar las acciones del usuario. Por ejemplo, el método "btnBAActionPerformed" se ejecuta cuando se hace clic en el botón "btnBA" para regresar a la lista de artículos. En este caso, se crea una instancia de la clase "ConsultarInventario" y se muestra en pantalla, luego se cierra la ventana actual.

El método "btnRAActionPerformed" se ejecuta cuando se hace clic en el botón "btnRA" para agregar el artículo. En este método, se obtienen los valores ingresados por el usuario en los campos de texto y se realiza una inserción en la tabla "articulos" de la base de datos con esos valores. También se obtiene el ID del artículo recién insertado y se realiza otra inserción en la tabla "artxprov" para asociar el artículo con el proveedor seleccionado.

El método "cbProvActionPerformed" se ejecuta cuando se selecciona un proveedor en el combo box "cbProv". En este método, se obtiene el índice y el nombre del proveedor seleccionado, y se realiza una consulta en la base de datos para obtener el ID del proveedor correspondiente. El ID se guarda en la variable "IDProvE" para su posterior uso.

El método "main" crea una instancia de la clase "AgregarArticulo" y la muestra en pantalla cuando se ejecuta el programa.

Clase ConsultarSucursales

La clase "ConsultarSucursales" es una ventana que permite consultar y modificar información de las sucursales en Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen botones para agregar una sucursal ("btnAgSuc"), regresar al menú principal ("btnBackMenu"), modificar datos de una sucursal ("btnMod"), y eliminar una sucursal ("btnDel"). También se definen etiquetas de texto y campos de texto para ingresar y mostrar información relacionada con las sucursales.

El constructor de la clase, llamado "ConsultarSucursales", inicializa la interfaz gráfica llamando al método "initComponents()". En este constructor también se configura un ComboBox ("cbSuc") para seleccionar el tipo de dato a modificar.

La clase contiene métodos de eventos para manejar las acciones del usuario. Por ejemplo, el método "btnAgSucActionPerformed" se ejecuta cuando se hace clic en el botón "btnAgSuc" para agregar una sucursal. En este caso, se crea una instancia de la clase "AgregarSucursal" y se muestra en pantalla, luego se cierra la ventana actual.

También se incluyen métodos para modificar y eliminar datos de las sucursales. Estos métodos utilizan consultas SQL para actualizar la información en la base de datos.

Al final del código, se encuentra el método "main" que inicia la aplicación creando una instancia de la clase "ConsultarSucursales" y mostrándola en pantalla.

Clase AgregarSucursal

La clase "AgregarSucursal" es una ventana que permite agregar una sucursal en Groove Central.

El código comienza con la declaración de los componentes visuales, que incluyen etiquetas de texto y campos de texto para ingresar información relacionada con la sucursal, y un botón para agregar la sucursal ("btnAgSucu") y regresar a la ventana de consultar sucursales ("btnBackSuc").

El constructor de la clase, llamado "AgregarSucursal", inicializa la interfaz gráfica llamando al método "initComponents()".

La clase contiene un método de evento para manejar la acción del usuario al hacer clic en el botón "btnAgSucu" para agregar una sucursal. En este método se obtiene la información ingresada en los campos de texto y se valida que no estén vacíos. Luego, se establece una conexión con la base de datos y se ejecuta una consulta SQL para insertar los datos de la sucursal. Si la inserción es exitosa, se muestra un mensaje de éxito; de lo contrario, se muestra un mensaje de error.

También se incluye un método de evento para manejar la acción del usuario al hacer clic en el botón "btnBackSuc" para regresar a la ventana de consultar sucursales. En este método se crea una instancia de la clase "ConsultarSucursales" y se muestra en pantalla, luego se cierra la ventana actual.

Al final del código, se encuentra el método "main" que inicia la aplicación creando una instancia de la clase "AgregarSucursal" y mostrándola en pantalla.

Pruebas

A lo largo del proyecto se realizaron múltiples versiones del programa:

Versión 1

En la versión 1 del programa Groove Central, se desarrolló la interfaz gráfica de usuario. Se crearon las ventanas y componentes visuales necesarios para consultar y modificar información de las sucursales, clientes, artículos, proveedores y vendedores. Sin embargo,

en esta versión inicial, la funcionalidad de la base de datos no estaba integrada y no se habían implementado características adicionales.

Versión 1.1

En la versión 1.1, se añadió la integración con una base de datos MySQL. Se implementaron consultas SQL para realizar operaciones de agregar, modificar y eliminar datos de las sucursales, clientes, proveedores, artículos y vendedores en la base de datos. Esto permitió almacenar y recuperar la información de forma persistente, mejorando la eficiencia y la integridad de los datos.

Versión 1.2

En la versión 1.2, se realizaron mejoras en la presentación visual del programa. Se agregaron gráficos y elementos visuales adicionales para mejorar la experiencia del usuario. Además, se implementó la funcionalidad de generar archivos PDF con información relevante del sistema, como tickets de compras de los productos. Esto facilitó la generación de informes y el intercambio de información en formato PDF.

Versión 1.3

La versión 1.3 representa la versión final y completa del programa Groove Central. Se agregaron componentes visuales y consultas SQL adicionales para manejar la información de los productos de manera eficiente. Además, se llevaron a cabo pruebas exhaustivas para garantizar el correcto funcionamiento del programa, se corrigieron errores y se implementaron mejoras sugeridas.

Despedida

Enhorabuena, has llegado al final de este Manual del Programador de Groove Central. Esperamos que la información proporcionada te haya sido de gran ayuda para comprender y utilizar el programa de manera efectiva.

Durante este recorrido, hemos cubierto desde la creación de la interfaz gráfica hasta la implementación de la base de datos, pasando por la generación de gráficos y archivos PDF. Cada versión del programa ha agregado nuevas funcionalidades y mejoras significativas.

Te animamos a que sigas explorando y aprendiendo en el campo de la programación, manteniéndote actualizado sobre las últimas tecnologías y practicando tus habilidades. Esperamos que Groove Central sea una herramienta exitosa en tus proyectos futuros.

Agradecemos tu tiempo y dedicación al leer este manual. ¡Te deseamos mucho éxito en tus aventuras como programador!

Hasta pronto,

El equipo de Groove Central