

SantiShop

Presentado por:
Santiago Vasquez Vega
Julio De La Ossa Madera
Cleyver Espitia Gaitan

Ingeniero
Ruben Baena Navarro

Universidad Cooperativa de Colombia
Ingeniería de Sistemas
Estructura de datos
III Semestre
Campus Montería.
29 de mayo del 2025

TABLA DE CONTENIDO

Resumen.....	3
Objetivos.....	4
Marco teórico.....	5
Desarrollo del proyecto (MANUAL).....	6
Resultados.....	94
Conclusiones	99
Bibliografía.....	100

RESUMEN

El presente informe documenta el desarrollo de SantiShop, una aplicación de escritorio diseñada con JavaFX que simula el funcionamiento de una tienda virtual. El proyecto tiene como objetivo permitir al usuario registrarse, iniciar sesión, visualizar productos, agregarlos a un carrito de compras, marcarlos como favoritos o añadirlos a una lista de deseos. Además, ofrece funcionalidades para gestionar el perfil del usuario, realizar pagos simulados y visualizar un historial de compras que se almacena de forma persistente.

Durante el desarrollo se aplicaron principios de programación orientada a objetos, estructuras de datos personalizadas (listas enlazadas), manejo de archivos de texto para persistencia de información y una interfaz gráfica interactiva usando Scene Builder. Se abordaron funcionalidades clave como el registro y autenticación de usuarios, visualización dinámica de productos, control de flujo de ventanas, y el almacenamiento de información relevante del usuario.

El sistema fue construido con un enfoque modular, asegurando la separación entre lógica de negocio y presentación. Este proyecto no solo permitió afianzar conocimientos técnicos, sino también el trabajo metódico en diseño de interfaces, validación de datos y persistencia. Los resultados obtenidos evidencian la viabilidad de desarrollar una tienda digital funcional y amigable desde cero utilizando tecnologías estándar de Java.

OBJETIVOS

Objetivo General

Desarrollar una aplicación de escritorio con JavaFX que simule una tienda en línea utilizando estructuras de datos para gestionar información de usuarios, productos, compras, deseos y carritos.

Objetivos Específicos

- Implementar un sistema de registro e inicio de sesión para los usuarios.
- Mostrar un catálogo con al menos 10 productos utilizando una interfaz visual.
- Implementar una funcionalidad de carrito de compras que almacene productos seleccionados.
- Desarrollar una lista de deseos para guardar productos favoritos.
- Registrar un historial de compras para cada usuario.
- Utilizar estructuras de datos como listas enlazadas, pilas y colas para la gestión de los diferentes módulos del sistema.

MARCO TEÓRICO

Java y JavaFX

Java es un lenguaje de programación orientado a objetos, ampliamente utilizado en el desarrollo de aplicaciones empresariales. JavaFX es una biblioteca gráfica que permite el desarrollo de interfaces de usuario modernas y dinámicas en Java, siendo una alternativa a Swing.

Scene Builder

Scene Builder es una herramienta visual que permite diseñar interfaces gráficas en JavaFX mediante una estructura tipo drag-and-drop. Genera automáticamente archivo .fxml, los cuales pueden ser cargados dinámicamente en el programa, separando la lógica de presentación de la lógica de negocio.

Estructuras de datos

El proyecto incorpora varias estructuras de datos fundamentales para el manejo interno de la información:

- **Nodo:** Es la unidad básica de una estructura enlazada. Contiene datos y una o más referencias (punteros) a otros nodos. En este proyecto, se utilizan nodos genéricos (`nodo>T>` y `NodoDoble<T>`) para manejar distintos tipos de objetos.
- **Lista simplemente enlazada:** Utiliza nodos conectados de manera unidireccional. Es empleada para manejar usuarios registrados en la clase `ListaUsuarios`, donde cada nodo apunta al siguiente.
- **Lista doblemente enlazada:** Cada nodo tiene referencias tanto al siguiente como al anterior, permitiendo un recorrido bidireccional. Se utiliza en `ListaProductos` para almacenar productos en el carrito o en lista de deseos.
- **Colas con listas doblemente enlazadas:** Aunque no se implementa una cola clásica con operaciones FIFO (first-in, first-out), se emplean listas doblemente enlazadas que pueden adaptarse a dicho comportamiento si se requieren funcionalidades similares como en el flujo de compras o navegación entre productos.
- **Vectores (ArrayList):** Se utilizan estructuras dinámicas de tipo lista como `ArrayList` para almacenar temporalmente productos comprados, historial de compras visuales y otros elementos que requieren recorrido y acceso rápido.

Patrones de Diseño

Se aplicaron principios de programación orientada a objetos como la separación de responsabilidades, uso de controladores (MVC) y encapsulamiento de datos, para facilitar la mantenibilidad y escalabilidad del proyecto.

DESARROLLO DEL PROYECTO (MANUAL)

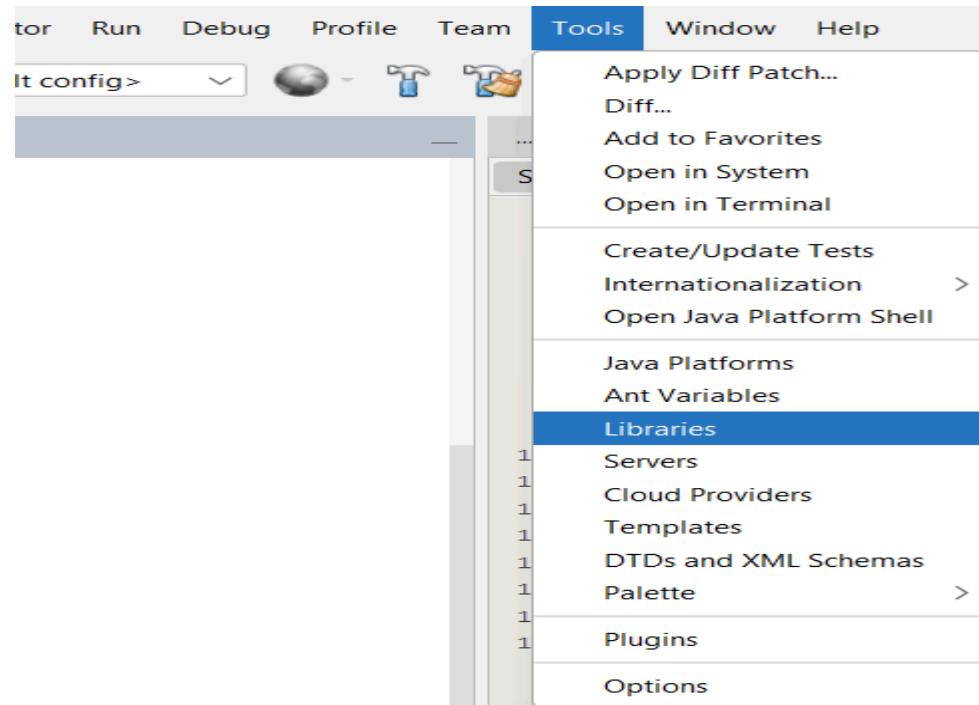
Para realizar este proyecto, primeramente necesitamos tener 2 cosas, primero la librería de javafx y segundo el scenebuilder de javafx.

Este es el link para instalar javafx: <https://gluonhq.com/products/javafx/>

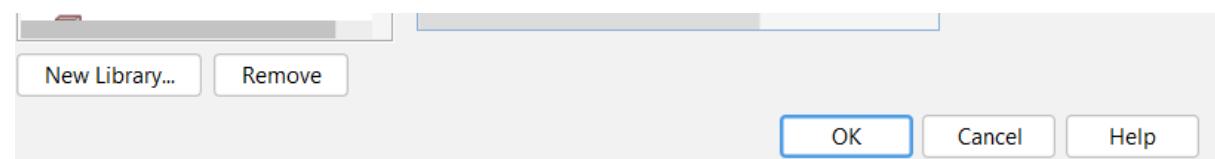
Cómo estamos utilizando el sistema operativo de windows, instalamos el SDK para windows



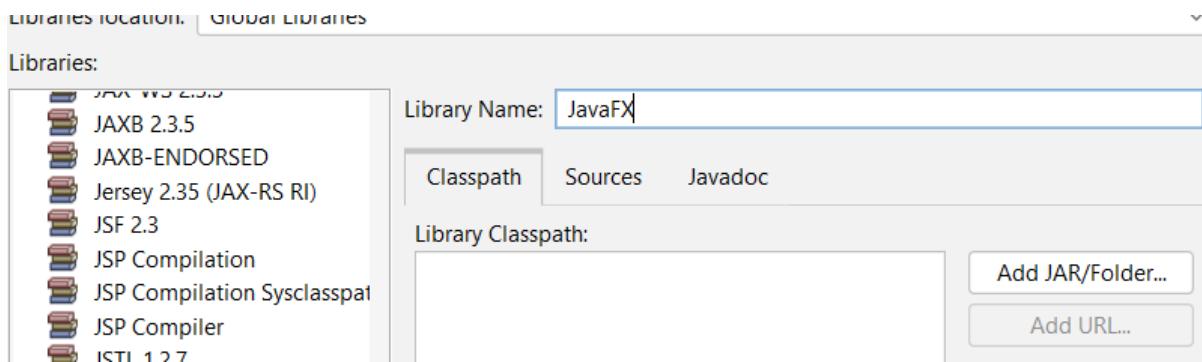
Luego de instalar la carpeta comprimida .zip nos vamos a nuestro explorador de archivos y posteriormente extraemos de .zip a carpeta; ya después de este proceso entramos a netbeans luego,



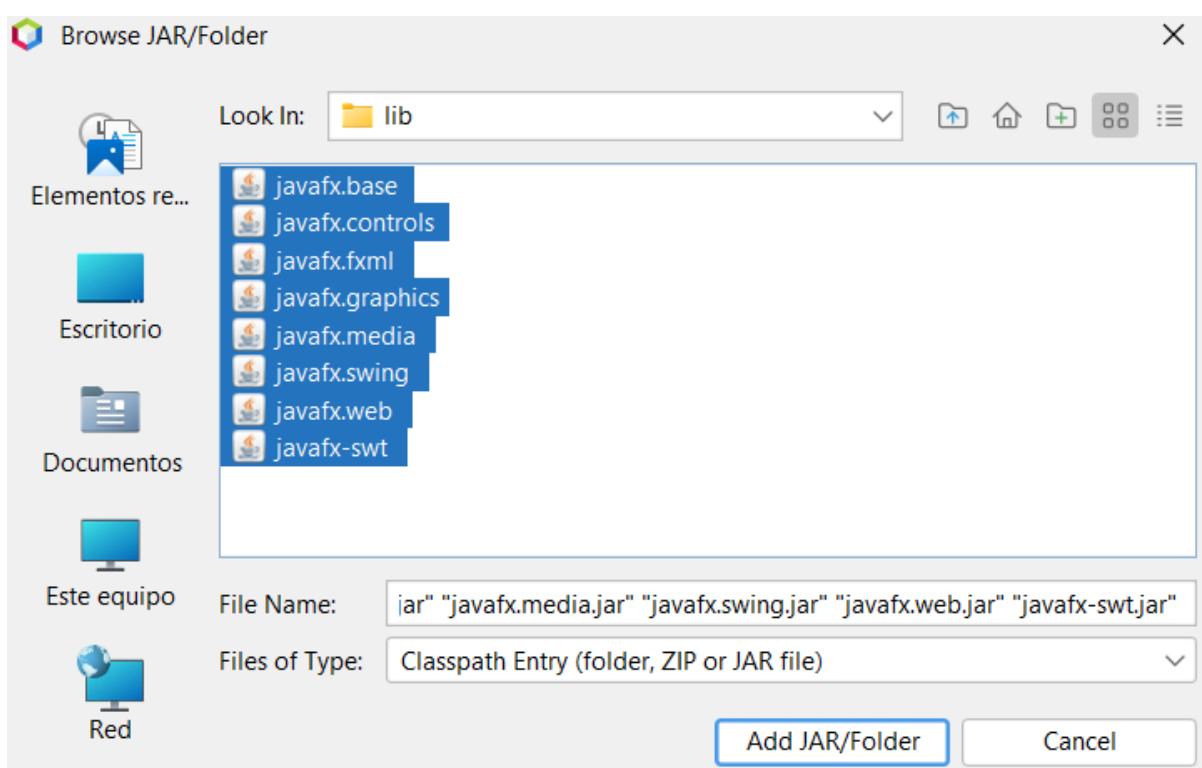
En la opción de Tools nos vamos a donde dice libraries para añadir la librería de javaFx. Nos vamos a New Library..



Después de añadir el nombre de la librería de recomendación “JavaFX(VERSIÓN)”, procedemos a ADD JAR/Folder...

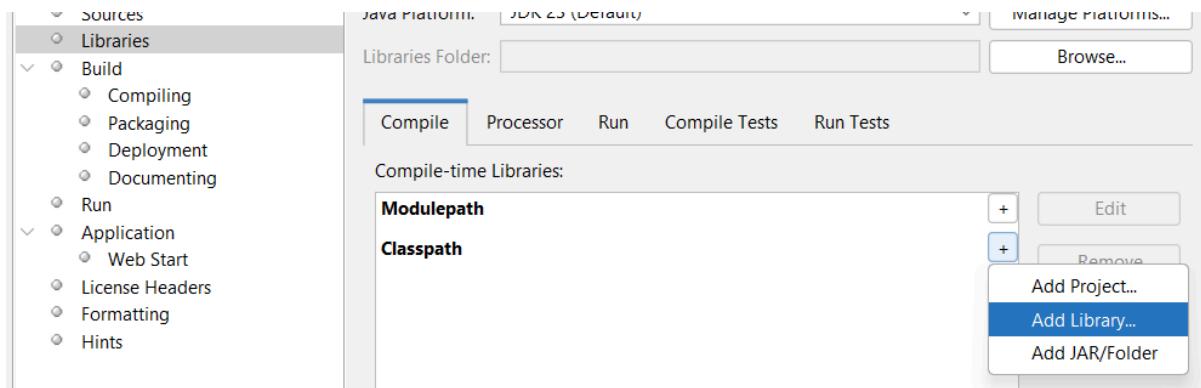


luego buscamos la carpeta posteriormente exportada y buscamos dentro de ella la carpeta con el nombre de “lib”



Seleccionamos todas las opciones y añadimos.. ya luego de eso damos ok y la librería de javaFx quedará configurada correctamente dentro de NetBeans

Luego para configurar esta librería dentro de un proyecto, seleccionamos el proyecto y nos vamos en el apartado de propiedades y luego en libraries



le damos Add Library.. y seleccionamos la librería con el nombre que posteriormente le pusimos, ya como consejo si se desea visitar los pasos de la pagina oficial aqui se encuentra el link:

<https://openjfx.io/openjfx-docs/>

y seleccionamos en nuestro caso JavaFX and NetBeans, o si ustedes tienen otro IDE diferente siguen los pasos de acuerdo a ello, pero nosotros lo estamos realizando en netbeans.

Introduction

Install Java

Run HelloWorld using JavaFX

Run HelloWorld via Maven

Run HelloWorld via Gradle

Runtime images

JavaFX and IntelliJ

JavaFX and NetBeans

JavaFX and Eclipse

JavaFX and Visual Studio Code

Next Steps

Ya después de instalar correctamente el JavaFx necesitamos la vista para la tienda que en nuestro caso utilizamos la interfaz de Scene Builder como vista.

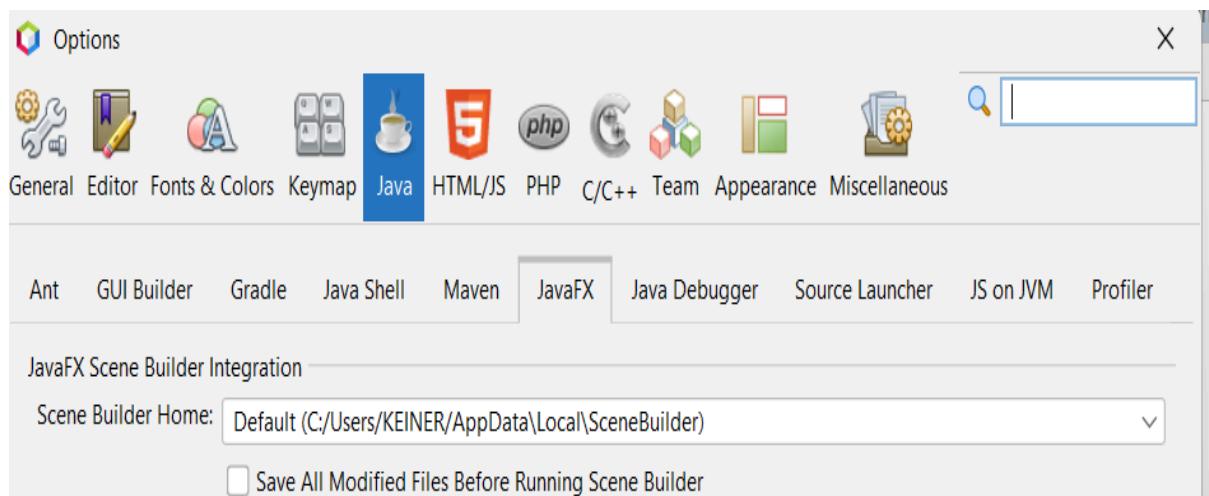
Para descargar este apartado nos vamos al siguiente link:

<https://gluonhq.com/products/scene-builder/n>

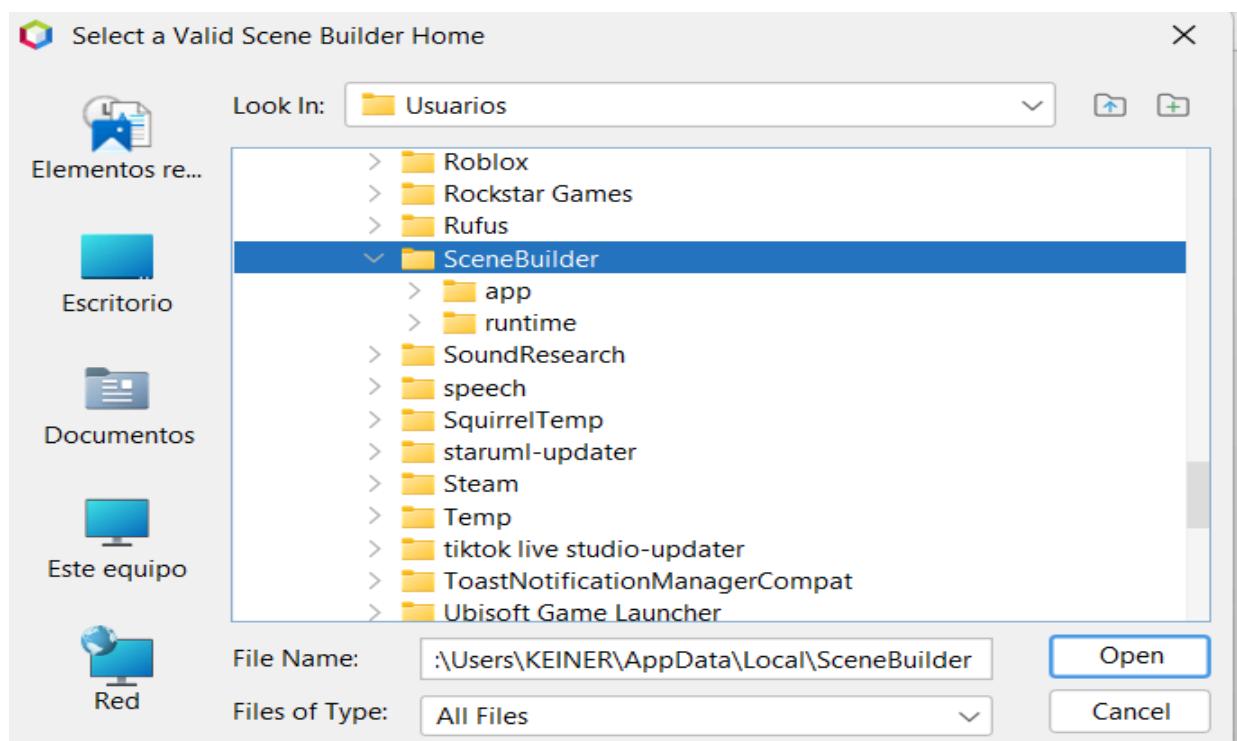
Y como cualquier programa luego de descargarlo en su buscador web nos abrirá un instalador y luego de seguir los procesos de instalación como cualquier programa.

Luego dentro de NetBeans hacemos el proceso para implementarlo y que a la hora de abrir los archivos FXML se nos dirija al scene builder

Para esto dentro de netbeans nos vamos a las opciones de arriba y seguido donde dice tools presionamos y bajamos hasta donde dice options al abrir options nos saldrá lo siguiente:



Luego en la opción Java nos dirigimos donde dice JavaFx en el ComboBox nos debe de decir Browse. presionamos allí y añadimos la ruta de instalación donde se nos descargó el scene builder en nuestro caso:



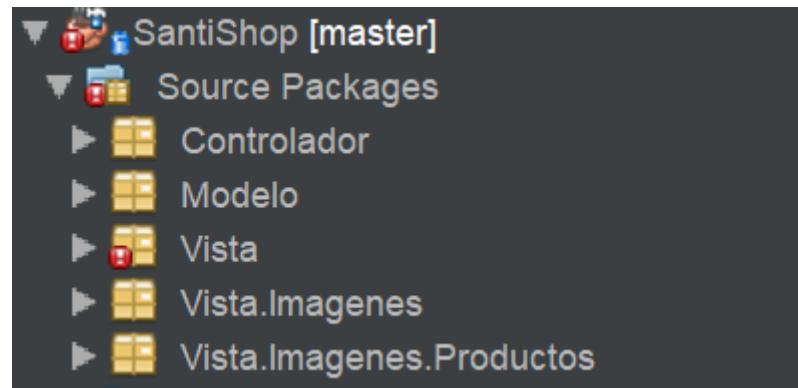
Le damos open y apply para que se guarden los cambios y por precaución o recomendación reiniciamos NetBeans y ya tendremos el Scene Builder listo para crear los fxml.

Nota: para abrir los FXML en scene builder se le tiene que dar doble click al archivo fxml.
A continuación, primero se muestra el diagrama de clases UML del proyecto, luego de ello procedemos a ya a la creación del proyecto, este es nuestro diagrama UML completo de la actividad:

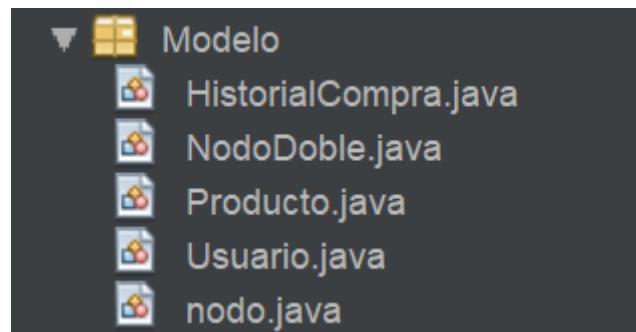


Luego de tener todo lo necesario, las librerías y el scene builder, creamos un nuevo proyecto, y en source package, new package, creamos 3 paquetes para organizar nuestro proyecto Modelo-Vista-Controlador. Además, agregaremos dos paquetes más para las imágenes.

debería de quedarnos así(cabe aclarar que nuestro proyecto no tiene ningun error, solo que se hizo una eliminación incorrecta y quedó bugueado la carpeta vista con el error):



Luego de tener los 3 paquetes, nos vamos al modelo, y creamos las siguientes clase, dandole click derecho al paquete modelo, new, java class, y creamos las 5 clases llamadas: HistorialCompra, Nododoble, Nodo, Producto y Usuario.



Ahora miraremos a fondo las clases como tal, en la clase nodo agregaremos el siguiente código:

```

package Modelo;
/**
 *
 * @author santi
 */
public class nodo<T>{

    public T dato;
    public nodo <T> sig;

    public nodo(T dato) {
        this.dato = dato;
        sig = null;
    }
}

```

Y en la clase NodoDoble:

```
package Modelo;
/**
 *
 * @author santi
 */
public class NodoDoble<T> {

    public T dato;
    public NodoDoble sig;
    public NodoDoble ant;

    public NodoDoble(T dato) {
        this.dato = dato;
        this.sig = null;
        this.ant = null;
    }
}
```

Los anteriores códigos, nos permiten guardar de manera dinámica la distinta información de nuestro proyecto.

Ahora trabajaremos sobre la clase Producto, agregando los métodos necesarios para su ejecución, con su constructor y sus getters.

```
package Modelo;
/**
 *
 * @author santi
 */
public class Producto {

    private String nombre;
    private double precio;
    private double descuento;
    private String marca;
    private String imagenUrl;

    public Producto(String nombre, double precio, double descuento, String marca, String imagenUrl)
    {
        this.nombre = nombre;
        this.precio = precio;
        this.descuento = descuento;
        this.marca = marca;
        this.imagenUrl = imagenUrl;
    }

    public String getNombre() { return nombre; }
    public double getPrecio() { return precio; }
    public double getDescuento() { return descuento; }
    public String getMarca() { return marca; }
    public String getImagenUrl() { return imagenUrl; }
}
```

Luego de agregar la clase productos, agregaremos la clase Usuarios, importante para registrar y manejar los datos de los usuarios. En este caso, además de los getters también agregaremos los setters.

```
package Modelo;
/**
 *
 * @author santi
 */
public class Usuario {

    private String Usuario;
    private String Contrasena;
    private String Nombre;
    private String Apellido;
    private String Sexo;
    private String Edad;
    private String Pais;
    private String Direccion;

    public Usuario(String Usuario, String Contrasena, String Nombre, String Apellido, String Sexo,
String Edad, String Pais) {
        this.Usuario = Usuario;
        this.Contrasena = Contrasena;
        this.Nombre = Nombre;
        this.Apellido = Apellido;
        this.Sexo = Sexo;
        this.Edad = Edad;
        this.Pais = Pais;
        this.Direccion = Direccion;
    }

    public String getUsuario() { return Usuario; }
    public String getContrasena() { return Contrasena; }
    public String getNombres() { return Nombre; }
    public String getApellidos() { return Apellido; }
    public String getSexo() { return Sexo; }
    public String getEdad() { return Edad; }
    public String getPais() { return Pais; }
    public String getDireccion() { return Direccion; }

    public void setUsuario(String Usuario) {
        this.Usuario = Usuario;
    }
    public void setContrasena(String nuevaContrasena) {
        this.Contrasena = nuevaContrasena;
    }
    public void setNombre(String Nombre) {
        this.Nombre = Nombre;
    }
    public void setApellido(String Apellido) {
        this.Apellido = Apellido;
    }
    public void setSexo(String Sexo) {
        this.Sexo = Sexo;
    }
}
```

```

    }
    public void setEdad(String Edad) {
        this.Edad = Edad;
    }
    public void setPais(String Pais) {
        this.Pais = Pais;
    }
    public void setDireccion(String Direccion) {
        this.Direccion = Direccion;
    }
}

```

Y para finalizar con el contenido del empaquetado de modelo, agregaremos la clase HistorialCompra, la cual permite hacer manejo del historial de los productos que el usuario compro.

```

package Modelo;
/**
 *
 * @author santi
 */
public class HistorialCompra {
    private Producto producto;
    private String fechaCompra;
    private double precioCompra;

    public HistorialCompra(Producto producto, String fechaCompra, double precioCompra) {
        this.producto = producto;
        this.fechaCompra = fechaCompra;
        this.precioCompra = precioCompra;
    }

    public Producto getProducto() { return producto; }
    public String getFechaCompra() { return fechaCompra; }
    public double getPrecioCompra() { return precioCompra; }
}

```

Ya teniendo nuestro paquete Modelo listo, nos dirigimos al paquete de Controlador y creamos una serie de nuevas clases: ListaProducto, ListaUsuarios, RegisterControlador, LoginControlador, PaginaPrincipalControlador y PaginaPagosControlador, mediante las cuales podremos controlar todo lo que tiene que ver con el manejo de datos y todos los métodos necesarios que vamos a usar. Estas clases también nos ayudan a manejar el control de la interfaz de la aplicación. El paquete Controlador nos debería de quedar así:



Empezamos editando la Clase ListaProductos, la cual está enlazada con el NodoDoble (cola doblemente enlazada). Esta clase permite hacer el manejo de los datos de los 10 productos de nuestro aplicativo. Agregaremos las siguientes importaciones y métodos (cada método tiene una linea de código fuera de el que explica para qué sirve) que nos permitirán trabajar con los distintos componentes:

```
package Controlador;

import Modelo.NodoDoble;
import Modelo.Producto;
import java.util.ArrayList;
import java.util.List;
/**
 *
 * @author santi
 */
public class ListaProductos {

    //declaramos la cabecera del nodo
    private NodoDoble<Producto> cab;
    //declaramos la cola del nodo
    private NodoDoble<Producto> cola;

    //inicializamos ambas variables como nulas
    public ListaProductos() {
        cab = null;
        cola = null;
    }

    //metodo para agregar un producto al nodo
    public void agregarProducto(Producto p) {
        NodoDoble nuevo = new NodoDoble(p);
        if (cab == null) {
            cab = cola = nuevo;
        } else {
            cola.sig = nuevo;
            nuevo.ant = cola;
            cola = nuevo;
        }
    }

    //metodo para obtener el primer elemento del nodo
    public NodoDoble getCabeza() {
        return cab;
    }

    //metodo para ver si el nodo está vacio
    public boolean estaVacia() {
        return cab == null;
    }
}
```

```

//metodo para buscar un producto por su nombre
public Producto buscarPorNombre(String nombre) {
    NodoDoble<Producto> actual = cab;
    while (actual != null) {
        if (actual.dato.getNombre().equals(nombre)) {
            return actual.dato;
        }
        actual = actual.sig;
    }
    return null;
}

//metodo para vaciar el carrito de compra
public void vaciarCarrito() {
    // Liberar referencias para permitir la recolección de basura
    NodoDoble<Producto> actual = cab;
    while (actual != null) {
        NodoDoble<Producto> siguiente = actual.sig;
        actual.ant = null;
        actual.sig = null;
        actual = siguiente;
    }

    // Resetear la cabeza y cola de la lista
    cab = null;
    cola = null;
}

//metodo para obtener los productos que están en la lista
public List<Producto> getProductos() {
    List<Producto> lista = new ArrayList<>();
    NodoDoble<Producto> actual = cab;

    while (actual != null) {
        lista.add(actual.dato);
        actual = actual.sig;
    }

    return lista;
}
}

```

Luego de agregar el código a nuestra clase ListaProductos haremos lo mismo con ListaUsuarios, la cual está enlazada con el Nodo(lista sencilla). Esta clase permite hacer el manejo de los datos de los usuarios que se registran en nuestro aplicativo. Agregaremos las siguientes importaciones y métodos (cada método tiene una linea de código fuera de el que explica para qué sirve) que nos permitirán trabajar con los distintos componentes:

```

package Controlador;

import Modelo.Usuario;
import Modelo.nodo;
/**
 *

```

```

* @author santi
*/
public class ListaUsuarios {

    //Creamos la cabecera del nodo
    public nodo<Usuario> cabeza;

    //La inicializamos como nula
    public ListaUsuarios() {
        cabeza=null;
    }

    //Método para ver si la lista es vacía
    public boolean ListaEsVacia(){
        return cabeza ==null?true:false;
    }

    //Método para agregar un usuario
    public void agregarUsuario(Usuario nuevoUsuario) {
        nodo<Usuario> p = new nodo<>(nuevoUsuario);

        if (ListaEsVacia()) {
            cabeza = p;
        } else {
            nodo<Usuario> actual = cabeza;
            while (actual.sig != null) {
                actual = actual.sig;
            }
            actual.sig = p;
        }
    }

    //Método para buscar un usuario por usuario y contraseña
    public Usuario buscarUsuario(String usuario, String contrasena) {
        nodo<Usuario> actual = cabeza;

        while (actual != null) {
            if (actual.dato.getUsuario().equalsIgnoreCase(usuario) &&
actual.dato.getContrasena().equals(contrasena)) {
                return actual.dato; // Usuario encontrado
            }
            actual = actual.sig;
        }

        return null; // No encontrado
    }

    //Método para verificar si ya existe un usuario registrado
    public boolean existeUsuario(String usuario) {
        nodo<Usuario> actual = cabeza;

        while (actual != null) {
            if (actual.dato.getUsuario().equalsIgnoreCase(usuario)) {
                return true;
            }
        }
    }
}

```

```
        actual = actual.sig;
    }

    return false;
}

//metodo para actualizar la contraseña del usuario
public void actualizarContrasena(String usuario, String nuevaContrasena) {
    nodo<Usuario> actual = cabeza;

    while (actual != null) {
        if (actual.dato.getUsuario().equalsIgnoreCase(usuario)) {
            actual.dato.setContrasena(nuevaContrasena);
            break;
        }
        actual = actual.sig;
    }
}
```

Una vez definidas nuestras 2 clases de manejos de datos de Productos y Usuarios, codificaremos nuestra clase RegisterControlador, la cual está enlazada con la clase ListaUsuarios, el archivo FXML del registro (más adelante lo trabajaremos) y es la que permite el guardado de los datos en el archivo.txt

A partir de esta clase, empezaremos a manejar una serie de métodos más robustos y extensos, que son los que permiten la correcta ejecución de nuestro proyecto. (como lo manifesté anteriormente, cada método tendrá su respectiva explicación de funcionamiento).

```
package Controlador;

import static Controlador.LoginControlador.cargarUsuariosDesdeArchivo;
import Modelo.Usuario;
import java.io.BufferedReader;
import java.io.FileWriter;
import java.io.IOException;
import java.net.URL;
import java.util.ResourceBundle;
import javafx.animation.PauseTransition;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.image.Image;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;
import javafx.util.Duration;
```

```

/**
 *
 * @author santi
 */
public class RegisterControlador implements Initializable {

    //declaramos las variables a usar
    public static ListaUsuarios listaUsuarios = new ListaUsuarios();
    @FXML
    private TextField txtUsuario;
    @FXML
    private PasswordField txtContrasena;
    @FXML
    private TextField txtNombres;
    @FXML
    private TextField txtApellidos;
    @FXML
    private ComboBox<String> comboSexo;
    @FXML
    private TextField txtEdad;
    @FXML
    private TextField txtPais;

    @Override
    public void initialize(URL url, ResourceBundle rb){
        comboSexo.getItems().addAll("Masculino", "Femenino", "Otro", "Prefiero no decirlo");
    }

    //metodo para Regresar al login
    @FXML
    private void irAlLogin() {
        try {
            FXMLLoader loader = new FXMLLoader(getClass().getResource("/Vista/Login.fxml"));
            Parent root = loader.load();

            Stage stage = new Stage();
            stage.setScene(new Scene(root));
            stage.setTitle("SantiShop - Login");
            stage.getIcons().add(new Image(getClass().getResourceAsStream("/Vista/Imagenes/LOGO
FONDO BLANCO SANTISHOP.png")));
            stage.setResizable(false);
            stage.centerOnScreen();
            stage.show();

            // Cerrar la ventana actual
            Stage ventanaActual = (Stage) txtUsuario.getScene().getWindow();
            ventanaActual.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    //metodo para regresar al login al hacer click en un label
    @FXML

```

```

private void volverLogin(MouseEvent event) {
    irAlLogin();
}

//metodo para regresar al login al hacer click en un botón
private void volverLoginBoton() {
    irAlLogin();
}

//metodo para registrar un usuario en un nodo
@FXML
private void registrarUsuario() {

    String usuario = txtUsuario.getText();
    String contrasena = txtContrasena.getText();
    String nombres = txtNombres.getText();
    String apellidos = txtApellidos.getText();
    String sexo = comboSexo.getValue();
    String edad = txtEdad.getText();
    String pais = txtPais.getText();

    //validamos que los campos no estén vacios
    if (usuario.isEmpty() || contrasena.isEmpty() || nombres.isEmpty() || apellidos.isEmpty() || sexo
    == null || edad.isEmpty() || pais.isEmpty()) {
        Alert alerta = new Alert(Alert.AlertType.WARNING);
        alerta.setTitle("Campos incompletos");
        alerta.setHeaderText(null);
        alerta.setContentText("Por favor, completa todos los campos.");
        alerta.show();
        txtUsuario.requestFocus();
        return;
    }else{

        // Validar que la edad sea un valor aceptado
        int edad1;
        try {
            edad1 = Integer.parseInt(edad);
            if (edad1 <= 0) {
                Alert alerta = new Alert(Alert.AlertType.WARNING);
                alerta.setTitle("Edad no válida");
                alerta.setHeaderText(null);
                alerta.setContentText("La edad debe ser un número mayor a cero.");
                alerta.show();
                txtEdad.setText("");
                txtEdad.requestFocus();
                return;
            }
        }catch (NumberFormatException e) {
            Alert alerta = new Alert(Alert.AlertType.ERROR);
            alerta.setTitle("Error en la edad");
            alerta.setHeaderText(null);
            alerta.setContentText("Debes ingresar un número válido para la edad.");
            alerta.show();
            txtEdad.setText("");
            txtEdad.requestFocus();
        }
    }
}

```

```

        return;
    }

    // Verificar que el usuario no esté ya registrado
    if (listaUsuarios.existeUsuario(usuario)) {
        Alert alerta = new Alert(Alert.AlertType.ERROR);
        alerta.setTitle("Error de Registro");
        alerta.setHeaderText(null);
        alerta.setContentText("El usuario ya existe. Intenta con otro nombre de usuario.");
        alerta.showAndWait();
        txtUsuario.requestFocus();
    } else {
        // Se crea el usuario
        Usuario nuevoUsuario = new Usuario(usuario, contrasena, nombres, apellidos, sexo, edad,
        pais);

        listaUsuarios.agregarUsuario(nuevoUsuario);
        guardarUsuarioEnArchivo(nuevoUsuario);

        Alert alerta = new Alert(Alert.AlertType.INFORMATION);
        alerta.setTitle("Registro Exitoso");
        alerta.setHeaderText(null);
        alerta.setContentText("Usuario registrado exitosamente.\nCargando login...");
        alerta.show();

        //una pequeña pausa luego de crear el usuario para volver al login
        PauseTransition pausa = new PauseTransition(Duration.seconds(3));
        pausa.setOnFinished(event -> {
            alerta.close();
            volverLoginBoton();
        });
        pausa.play();
    }
}

//metodo para guardar el usuario en un archivo txt
private void guardarUsuarioEnArchivo(Usuario usuario) {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter("usuarios.txt", true))) {
        // Guardar datos separados por punto y coma
        writer.write(usuario.getUsuario() + ";" +
                    usuario.getContrasena() + ";" +
                    usuario.getNombres() + ";" +
                    usuario.getApellidos() + ";" +
                    usuario.getSexo() + ";" +
                    usuario.getEdad() + ";" +
                    usuario.getPais() + ";" +
                    (usuario.getDireccion() != null ? usuario.getDireccion() : ""));
        writer.newLine();
    } catch (IOException e) {
        e.printStackTrace();
        System.out.println("Error al guardar el usuario.");
    }
}

```

Continuaremos editando nuestra clase LoginControlador, que es la que valida los datos guardados en el RegisterControlador, y permite o no el acceso del usuario a la pagina principal.

```

package Controlador;

import Modelo.Usuario;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.net.URL;
import java.util.Optional;
import java.util.ResourceBundle;
import javafx.animation.PauseTransition;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.scene.control.TextInputDialog;
import javafx.scene.image.Image;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;
import javafx.stage.StageStyle;
import javafx.util.Duration;

/**
 *
 * @author santi
 */
public class LoginControlador implements Initializable{

    //declaramos las variables a usar
    @FXML
    private TextField txtUsuario;
    @FXML
    private PasswordField txtContrasena;

    private Usuario usuarioCargado;

    @Override
    public void initialize(URL url, ResourceBundle rb){
        RegisterControlador.listaUsuarios = cargarUsuariosDesdeArchivo();
    }

    //metodo para abrir el registro mediante un label
    @FXML
    private void openRegister(MouseEvent event){

```

```

try{
    FXMLLoader loader = new FXMLLoader(getClass().getResource("/Vista/Register.fxml"));
    Parent root = loader.load();

    Stage stage = new Stage();
    stage.setScene(new Scene(root));
    stage.setTitle("SantiShop - Registro");
    stage.getIcons().add(new Image(getClass().getResourceAsStream("/Vista/Imagenes/LOGO
FONDO BLANCO SANTISHOP.png")));
    stage.setResizable(false);
    stage.centerOnScreen();
    stage.show();

    ((Stage)((Label)event.getSource()).getScene().getWindow())).close();

} catch (IOException e) {
    e.printStackTrace();
}
}

//metodo para abrir la pagina principal en un boton siempre y cuando exista el usuario
private void abrirPaginaPrincipal(ActionEvent event, Usuario usuarioEncontrado) {
    try {
        FXMLLoader loader = new
FXMLLoader(getClass().getResource("/Vista/paginaPrincipal.fxml"));
        Parent root = loader.load();

        // Obtener el controlador de la página principal
        PaginaPrincipalControlador controladorPagina = loader.getController();

        // Pasar el usuario al controlador de la página principal
        controladorPagina.setUsuarioActivo(usuarioEncontrado);

        // CARGA EL HISTORIAL DEL ARCHIVO (DESPUÉS DE ASIGNAR EL
        // USUARIO)
        controladorPagina.cargarHistorialDesdeArchivo(usuarioEncontrado.getUsuario());

        Stage stage = new Stage();
        stage.setScene(new Scene(root));
        stage.setTitle("SantiShop - Página Principal");
        stage.getIcons().add(new Image(getClass().getResourceAsStream("/Vista/Imagenes/LOGO
FONDO BLANCO SANTISHOP.png")));
        stage.setResizable(false);
        stage.centerOnScreen();
        stage.show();

        // Cerrar la ventana de login
        ((Stage)((Button)event.getSource()).getScene().getWindow())).close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

//metodo para iniciar sesión
@FXML

```

```

private void iniciarSesion(ActionEvent event) {
    String usuarioIngresado = txtUsuario.getText();
    String contrasenaIngresada = txtContrasena.getText();

    // Primero verificar si el usuario existe
    boolean usuarioExiste =
        Controlador.RegisterControlador.listaUsuarios.existeUsuario(usuarioIngresado);

    if (!usuarioExiste) {
        // Usuario no encontrado
        Alert alerta = new Alert(Alert.AlertType.ERROR);
        alerta.setTitle("Error de inicio de sesión");
        alerta.setHeaderText(null);
        alerta.setContentText("El usuario no existe. Regístrate primero.");
        alerta.show();
        txtUsuario.setText("");
        txtContrasena.setText("");
        txtUsuario.requestFocus();
        return;
    }

    // Si el usuario existe, ahora validar la contraseña
    Usuario usuarioEncontrado =
        Controlador.RegisterControlador.listaUsuarios.buscarUsuario(usuarioIngresado,
        contrasenaIngresada);

    if (usuarioEncontrado == null) {
        // Contraseña incorrecta
        Alert alerta = new Alert(Alert.AlertType.ERROR);
        alerta.setTitle("Error de inicio de sesión");
        alerta.setHeaderText(null);
        alerta.setContentText("La contraseña es incorrecta.");
        alerta.show();
        txtContrasena.setText("");
        txtContrasena.requestFocus();
    } else {
        // Usuario y contraseña correctos
        Alert alerta = new Alert(Alert.AlertType.INFORMATION);
        alerta.setTitle("Inicio de sesión exitoso");
        alerta.setHeaderText(null);
        alerta.setContentText("¡Bienvenido a SantiShop, " + usuarioEncontrado.getNombres() + "!");
        alerta.show();

        // Pausa de 2 segundos antes de abrir el form de página principal
        PauseTransition pausa = new PauseTransition(Duration.seconds(2));
        pausa.setOnFinished(e -> {
            alerta.close();
            abrirPaginaPrincipal(event, usuarioEncontrado); // Llamamos método que abre la página
            principal
        });
        pausa.play();
    }
}

//método para cambiar la contraseña si se olvidó

```

```

@FXML
private void olvidarContrasena() {
    // Solicitar nombre de usuario
    TextInputDialog dialogoUsuario = new TextInputDialog();
    dialogoUsuario.setTitle("Olvidé mi contraseña");
    dialogoUsuario.setHeaderText(null);
    dialogoUsuario.setContentText("Ingresa tu nombre de usuario:");

    Optional<String> resultadoUsuario = dialogoUsuario.showAndWait();

    if (resultadoUsuario.isPresent()) {
        String usuarioIngresado = resultadoUsuario.get();

        // Verificar si el usuario existe
        boolean usuarioExiste =
Controlador.RegisterControlador.listaUsuarios.existeUsuario(usuarioIngresado);

        if (!usuarioExiste) {
            Alert alerta = new Alert(Alert.AlertType.ERROR);
            alerta.setTitle("Error");
            alerta.setHeaderText(null);
            alerta.setContentText("El usuario no existe.");
            alerta.show();
            return;
        }

        // Si existe, pedir nueva contraseña
        TextInputDialog dialogoNuevaClave = new TextInputDialog();
        dialogoNuevaClave.setTitle("Nueva contraseña");
        dialogoNuevaClave.setHeaderText(null);
        dialogoNuevaClave.setContentText("Ingresa tu nueva contraseña:");

        Optional<String> resultadoClave = dialogoNuevaClave.showAndWait();

        if (resultadoClave.isPresent()) {
            String nuevaContrasena = resultadoClave.get();

            // Actualizar contraseña
            Controlador.RegisterControlador.listaUsuarios.actualizarContrasena(usuarioIngresado,
nuevaContrasena);

            Alert alerta = new Alert(Alert.AlertType.INFORMATION);
            alerta.setTitle("Contraseña actualizada");
            alerta.setHeaderText(null);
            alerta.setContentText("Tu contraseña ha sido actualizada exitosamente.");
            alerta.show();
        }
    }

    //metodo para cargar un usuario desde el archivo txt
    public static ListaUsuarios cargarUsuariosDesdeArchivo() {
        ListaUsuarios lista = new ListaUsuarios();
        File archivo = new File("usuarios.txt");
        if (!archivo.exists()) return lista;
    }
}

```

```
try (BufferedReader reader = new BufferedReader(new FileReader(archivo))) {
    String linea;
    while ((linea = reader.readLine()) != null) {
        String[] datos = linea.split(";");
        if (datos.length >= 7) { // 7 campos mínimos requeridos
            Usuario u = new Usuario(
                datos[0], // usuario
                datos[1], // contraseña
                datos[2], // nombres
                datos[3], // apellidos
                datos[4], // sexo
                datos[5], // edad (String)
                datos[6] // país
            );
            // Establecer dirección si existe y no está vacía
            if (datos.length >= 8 && !datos[7].isEmpty()) {
                u.setDireccion(datos[7]);
            }
            lista.agregarUsuario(u);
        }
    }
} catch (IOException e) {
    e.printStackTrace();
    System.out.println("Error al leer el archivo de usuarios.");
}
return lista;
}
```

Una vez editadas estas dos clases tan importantes para la clase Usuario, seguiremos con las esenciales para la Pagina Principal y el manejo de pagos.

Iniciaremos editando nuestra clase PaginaPrincipalControlador, esta clase es la que se encarga de controlar el archivo FXML de PaginaPrincipal, y es donde estan la mayoria de metodos y importaciones de otra clase(esta clase está relacionada con muchas otras clases de nuestro proyecto, puesto que es la que se encarga del funcionamiento de la tienda). agregaremos el siguiente código a nuestra clase:

```
package Controlador;  
  
import Modelo.HistorialCompra;  
import Modelo.Producto;  
import Modelo.Usuario;  
import Modelo.nodo;  
import java.io.BufferedReader;  
import java.io.BufferedWriter;  
import java.io.File;  
import java.io.FileReader;  
import java.io.FileWriter;  
import java.io.IOException;
```

```
import java.net.URL;
import java.time.LocalDate;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.ResourceBundle;
import java.util.Scanner;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.animation.PauseTransition;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.fxml.Initializable;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import static javafx.scene.Cursor.cursor;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.scene.layout.Priority;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import javafx.util.Duration;

/**
 * FXML Controller class
 *
 * @author santi
 */
public class PaginaPrincipalControlador implements Initializable {

    //declaramos las variables asignadas a los productos
    @FXML private ImageView imgProducto1;
    @FXML private Label lblNombre1;
    @FXML private Label lblPrecioOriginal1;
    @FXML private Label lblPrecioDescuento1;
    @FXML private Label lblPorcentaje1;
    @FXML private Label lblMarca1;
    @FXML private ImageView imgProducto2;
    @FXML private Label lblNombre2;
```

```
@FXML private Label lblPrecioOriginal2;
@FXML private Label lblPrecioDescuento2;
@FXML private Label lblPorcentaje2;
@FXML private Label lblMarca2;
@FXML private ImageView imgProducto3;
@FXML private Label lblNombre3;
@FXML private Label lblPrecioOriginal3;
@FXML private Label lblPrecioDescuento3;
@FXML private Label lblPorcentaje3;
@FXML private Label lblMarca3;
@FXML private ImageView imgProducto4;
@FXML private Label lblNombre4;
@FXML private Label lblPrecioOriginal4;
@FXML private Label lblPrecioDescuento4;
@FXML private Label lblPorcentaje4;
@FXML private Label lblMarca4;
@FXML private ImageView imgProducto5;
@FXML private Label lblNombre5;
@FXML private Label lblPrecioOriginal5;
@FXML private Label lblPrecioDescuento5;
@FXML private Label lblPorcentaje5;
@FXML private Label lblMarca5;
@FXML private ImageView imgProducto6;
@FXML private Label lblNombre6;
@FXML private Label lblPrecioOriginal6;
@FXML private Label lblPrecioDescuento6;
@FXML private Label lblPorcentaje6;
@FXML private Label lblMarca6;
@FXML private ImageView imgProducto7;
@FXML private Label lblNombre7;
@FXML private Label lblPrecioOriginal7;
@FXML private Label lblPrecioDescuento7;
@FXML private Label lblPorcentaje7;
@FXML private Label lblMarca7;
@FXML private ImageView imgProducto8;
@FXML private Label lblNombre8;
@FXML private Label lblPrecioOriginal8;
@FXML private Label lblPrecioDescuento8;
@FXML private Label lblPorcentaje8;
@FXML private Label lblMarca8;
@FXML private ImageView imgProducto9;
@FXML private Label lblNombre9;
@FXML private Label lblPrecioOriginal9;
@FXML private Label lblPrecioDescuento9;
@FXML private Label lblPorcentaje9;
@FXML private Label lblMarca9;
@FXML private ImageView imgProducto10;
@FXML private Label lblNombre10;
@FXML private Label lblPrecioOriginal10;
@FXML private Label lblPrecioDescuento10;
@FXML private Label lblPorcentaje10;
@FXML private Label lblMarca10;

//variable para filtrar
@FXML private ComboBox<String> comboFiltro;
```

```
//variables pertenecientes a los paneles donde se encuentran distintas funcionalidades
@FXML private Pane panelCarrito;
@FXML private Pane panelFavoritos;
@FXML private Pane panelOpciones;
@FXML private Pane panelListaDeseos;
@FXML private Pane panelHistorialCompra;
@FXML private Pane panelCuenta;
@FXML private Pane opacidad;

//variables de los contenedores de las funcionalidades
@FXML private VBox contenedorCarrito;
@FXML private VBox contenedorFavoritos;
@FXML private VBox contenedorListaDeseos;
@FXML private VBox contenedorHistorialCompra;

//variables pertenecientes a la funcionalidad "carrito de compra"
@FXML private Label lblCantidadCarrito;
@FXML private Label lblSubtotal;
@FXML private Label lblDescuento;
@FXML private Label lblTotal;

//variables pertenecientes a la funcionalidad "mi cuenta"
@FXML private TextField txtNombres;
@FXML private TextField txtApellidos;
@FXML private TextField txtSexo;
@FXML private TextField txtEdad;
@FXML private TextField txtPais;
@FXML private TextField txtDireccion;
@FXML private ImageView btnEditarNombre;
@FXML private ImageView btnEditarApellidos;
@FXML private ImageView btnEditarSexo;
@FXML private ImageView btnEditarEdad;
@FXML private ImageView btnEditarPais;
@FXML private ImageView btnEditarDireccion;
@FXML private Button btnGuardar;
@FXML private Button btnCancelar;
private String nombresOriginal;
private String apellidosOriginal;
private String sexoOriginal;
private String edadOriginal;
private String paisOriginal;
private String direccionOriginal;

//declaramos los productos
private Producto producto1;
private Producto producto2;
private Producto producto3;
private Producto producto4;
private Producto producto5;
private Producto producto6;
private Producto producto7;
private Producto producto8;
private Producto producto9;
private Producto producto10;
```

```

//variable que establece si el usuario registrado está activo (es decir, si un usuario está en linea)
private Usuario usuarioActivo;

//instanciamos la clase listaproducto
private ListaProductos listaCarrito = new ListaProductos();
//es una lista que guarda las compras hechas por el usuario (está asociada con la clase historial
compras)
private List<HistorialCompra> historialCompras = new ArrayList<>();
//un mapa (es un tipo de diccionario) que indica si un producto está marcado como favorito o no.
private Map<String, Boolean> productosEnFavoritos = new HashMap<>();
//un mapa (es un tipo de diccionario) para controlar visualmente qué productos están marcados
como favoritos en la interfaz gráfica.
private Map<String, ImageView> iconosFavoritos = new HashMap<>();
//un mapa (es un tipo de diccionario) que indica si un producto está marcado en lista de deseo o no.
private Map<String, Boolean> productosEnListaDeseos = new HashMap<>();
//un mapa (es un tipo de diccionario) para controlar visualmente qué productos están marcados en
lista de deseo en la interfaz gráfica.
private Map<String, ImageView> iconosListaDeseos = new HashMap<>();

@Override
public void initialize(URL url, ResourceBundle rb) {
    comboFiltro.getItems().addAll("Precio Mayor", "Precio Menor", "Descuento");
    //Configurar el evento de selección para el ComboBox
    comboFiltro.setOnAction(this::filtrarProductos);

    //Inicializamos el arreglo de productos para el filtrado
    inicializarArregloProductos();

    //agregamos los 10 productos
    producto1 = new Producto(
        "Portátil ASUS 16GB RAM, 1TB almacenamiento, tarjeta gráfica NVIDIA 4050RX",
        5200000,
        0.25,
        "ASUS",
        "Producto1.png"
    );
    lblNombre1.setText(producto1.getNombre());
    lblPrecioOriginal1.setText("$" + String.format("%.0f", producto1.getPrecio()));
    lblPrecioDescuento1.setText("$" + String.format("%.0f", producto1.getPrecio() * (1 -
    producto1.getDescuento())));
    lblPorcentaje1.setText((int)(producto1.getDescuento() * 100) + "% OFF");
    lblMarca1.setText(producto1.getMarca());
    Image imagen1 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
    producto1.getImagenUrl()));
    imgProducto1.setImage(imagen1);

    producto2 = new Producto(
        "Portátil HP Victus Gaming, AMD Ryzen 5 con IA, 16GB RAM, 1TB SSD Almacenamiento,
        NVIDIA® GeForce RTX 3050",
        4370000,
        0.25,
        "HP",
        "Producto2.jpg"
    );
}

```

```

);

lblNombre2.setText(producto2.getNombre());
lblPrecioOriginal2.setText("$" + String.format("%.0f", producto2.getPrecio()));
lblPrecioDescuento2.setText("$" + String.format("%.0f", producto2.getPrecio() * (1 -
producto2.getDescuento())));
lblPorcentaje2.setText((int)(producto2.getDescuento() * 100) + "% OFF");
lblMarca2.setText(producto2.getMarca());
Image imagen2 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
producto2.getImagenUrl()));
imgProducto2.setImage(imagen2);

producto3 = new Producto(
"Consola Playstation 5 Slim 1tb Con 2 Controles y Juego Fc 25 Ps5 Fisico",
4360000,
0.2,
"SONY",
"Producto3.jpg"
);
lblNombre3.setText(producto3.getNombre());
lblPrecioOriginal3.setText("$" + String.format("%.0f", producto3.getPrecio()));
lblPrecioDescuento3.setText("$" + String.format("%.0f", producto3.getPrecio() * (1 -
producto3.getDescuento())));
lblPorcentaje3.setText((int)(producto3.getDescuento() * 100) + "% OFF");
lblMarca3.setText(producto3.getMarca());
Image imagen3 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
producto3.getImagenUrl()));
imgProducto3.setImage(imagen3);

producto4 = new Producto(
"Mouse Gamer T-Dagger Imperial RGB",
100000,
0,
"IMPERIAL",
"Producto4.jpg"
);
lblNombre4.setText(producto4.getNombre());
lblPrecioOriginal4.setText("$" + String.format("%.0f", producto4.getPrecio()));
lblPrecioDescuento4.setText("$" + String.format("%.0f", producto4.getPrecio() * (1 -
producto4.getDescuento())));
lblMarca4.setText(producto4.getMarca());
lblPorcentaje4.setText((int)(producto4.getDescuento() * 100) + "% OFF");
Image imagen4 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
producto4.getImagenUrl()));
imgProducto4.setImage(imagen4);

producto5 = new Producto(
"iPhone 16 Pro-Max 1TB, color Desierto",
9240000,
0.4,
"APPLE",
"Producto5.jpg"
);
lblNombre5.setText(producto5.getNombre());
lblPrecioOriginal5.setText("$" + String.format("%.0f", producto5.getPrecio()));

```

```

lblPrecioDescuento5.setText("$" + String.format("%.0f", producto5.getPrecio() * (1 -
producto5.getDescuento())));
lblPorcentaje5.setText((int)(producto5.getDescuento() * 100) + "% OFF");
lblMarca5.setText(producto5.getMarca());
Image imagen5 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
producto5.getImagenUrl()));
imgProducto5.setImage(imagen5);

producto6 = new Producto(
"Samsung Galaxy S25 ultra 256 GB, color Gris titani",
7360000,
0.3,
"SAMSUNG",
"Producto6.png"
);
lblNombre6.setText(producto6.getNombre());
lblPrecioOriginal6.setText("$" + String.format("%.0f", producto6.getPrecio()));
lblPrecioDescuento6.setText("$" + String.format("%.0f", producto6.getPrecio() * (1 -
producto6.getDescuento())));
lblPorcentaje6.setText((int)(producto6.getDescuento() * 100) + "% OFF");
lblMarca6.setText(producto6.getMarca());
Image imagen6 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
producto6.getImagenUrl()));
imgProducto6.setImage(imagen6);

producto7 = new Producto(
"Televisor SAMSUNG 75 pulgadas LED Uhd4K Smart TV",
8470000,
0.1,
"SAMSUNG",
"Producto7.jpg"
);
lblNombre7.setText(producto7.getNombre());
lblPrecioOriginal7.setText("$" + String.format("%.0f", producto7.getPrecio()));
lblPrecioDescuento7.setText("$" + String.format("%.0f", producto7.getPrecio() * (1 -
producto7.getDescuento())));
lblPorcentaje7.setText((int)(producto7.getDescuento() * 100) + "% OFF");
lblMarca7.setText(producto7.getMarca());
Image imagen7 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
producto7.getImagenUrl()));
imgProducto7.setImage(imagen7);

producto8 = new Producto(
"Impresora Multifuncional Epson L5590 Wifi Ecotank",
5230000,
0.2,
"EPSON",
"Producto8.png"
);
lblNombre8.setText(producto8.getNombre());
lblPrecioOriginal8.setText("$" + String.format("%.0f", producto8.getPrecio()));
lblPrecioDescuento8.setText("$" + String.format("%.0f", producto8.getPrecio() * (1 -
producto8.getDescuento())));
lblPorcentaje8.setText((int)(producto8.getDescuento() * 100) + "% OFF");
lblMarca8.setText(producto8.getMarca());

```

```

        Image imagen8 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
producto8.getImagenUrl()));
        imgProducto8.setImage(imagen8);

        producto9 = new Producto(
        "Apple iPad (10th Generation): with A14 Bionic chip, 64GB, 4 RAM",
        6580000,
        0.5,
        "APPLE",
        "Producto9.jpg"
        );
        lblNombre9.setText(producto9.getNombre());
        lblPrecioOriginal9.setText("$" + String.format("%.0f", producto9.getPrecio()));
        lblPrecioDescuento9.setText("$" + String.format("%.0f", producto9.getPrecio() * (1 -
producto9.getDescuento())));
        lblPorcentaje9.setText((int)(producto9.getDescuento() * 100) + "% OFF");
        lblMarca9.setText(producto9.getMarca());
        Image imagen9 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
producto9.getImagenUrl()));
        imgProducto9.setImage(imagen9);

        producto10 = new Producto(
        "AirPods Pro 2",
        4370000,
        0.25,
        "APPLE",
        "Producto10.jpg"
        );
        lblNombre10.setText(producto10.getNombre());
        lblPrecioOriginal10.setText("$" + String.format("%.0f", producto10.getPrecio()));
        lblPrecioDescuento10.setText("$" + String.format("%.0f", producto10.getPrecio() * (1 -
producto10.getDescuento())));
        lblPorcentaje10.setText((int)(producto10.getDescuento() * 100) + "% OFF");
        lblMarca10.setText(producto10.getMarca());
        Image imagen10 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
producto10.getImagenUrl()));
        imgProducto10.setImage(imagen10);
    }

    //metodo para mostrar el carrito de compra
    @FXML
    private void mostrarCarrito(MouseEvent event) {
        panelCarrito.setVisible(true);
        panelCarrito.setManaged(true);
        opacidad.setVisible(true);
        opacidad.setManaged(true);
        panelOpciones.setVisible(false);
        panelOpciones.setManaged(false);
        opacidad.setOpacity(0.5);
    }

    //metodo general para volver de alguna funcionalidad
    @FXML
    private void volver(MouseEvent event){
        volverCarrito();
    }
}

```

```

volverFavoritos();
volverListaDeseos();
volverCuenta();
volverHistorialCompra();
}

//metodo especifico para volver del carrito
@FXML
private void volverCarrito() {
    panelCarrito.setVisible(false);
    panelCarrito.setManaged(false);
    opacidad.setVisible(false);
    opacidad.setManaged(false);
}

//metodo para actualizar los totales del carrito de compra
private void actualizarTotales() {
    double subtotal = 0;
    double descuento = 0;
    double total = 0;

    for (Node node : contenedorCarrito.getChildren()) {
        if (node instanceof HBox) {
            HBox item = (HBox) node;
            VBox info = (VBox) item.getChildren().get(1);
            Label lblNombre = (Label) info.getChildren().get(0);
            String nombreProducto = lblNombre.getText();

            // Busca el producto en la lista lógica
            Producto producto = listaCarrito.buscarPorNombre(nombreProducto);

            //si el producto existe toma su precio y su descuento y lo calcula
            if (producto != null) {
                double precio = producto.getPrecio();
                double porcDesc = producto.getDescuento();

                HBox controles = (HBox) item.getChildren().get(2);
                Label lblCantidad = (Label) controles.getChildren().get(1);
                int cantidad = Integer.parseInt(lblCantidad.getText());

                subtotal += precio * cantidad;
                descuento += precio * porcDesc * cantidad;
                total += (precio * (1 - porcDesc)) * cantidad;
            }
        }
    }

    lblSubtotal.setText("$" + String.format("%.0f", subtotal));
    lblDescuento.setText("$" + String.format("%.0f", descuento));
    lblTotal.setText("$" + String.format("%.0f", total));
}

//metodo para agregar un producto al carrito de compras
private void agregarProductoAlCarrito(Producto producto) {
    // Buscar si el producto ya existe en el carrito visual
}

```

```

for (Node node : contenedorCarrito.getChildren()) {
    if (node instanceof HBox) {
        HBox item = (HBox) node;
        VBox infoBox = (VBox) item.getChildren().get(1);
        Label nameLabel = (Label) infoBox.getChildren().get(0);

        if (nameLabel.getText().equals(producto.getNombre())) {
            // Si el producto ya está en el carrito, solo incrementamos la cantidad
            HBox controles = (HBox) item.getChildren().get(2);
            Label lblCantidad = (Label) controles.getChildren().get(1);
            int cantidad = Integer.parseInt(lblCantidad.getText()) + 1;
            lblCantidad.setText(String.valueOf(cantidad));
            return;
        }
    }
}

// Si llegamos aquí, el producto no existe en el carrito, añadimos uno nuevo
HBox contenedor = new HBox(8);
contenedor.setAlignment(Pos.CENTER_LEFT);
contenedor.setPadding(new Insets(10));
contenedor.setStyle("-fx-background-color: white; -fx-border-color: lightgray;");
contenedor.setPrefWidth(397);

// Imagen del producto
ImageView imagen = new ImageView(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" + producto.getImagenUrl())));
imagen.setFitWidth(50);
imagen.setFitHeight(50);

// Información del producto
VBox infoProducto = new VBox(5);
infoProducto.setPrefWidth(150);
infoProducto.setMinWidth(100);

Label lblNombre = new Label(producto.getNombre());
lblNombre.setStyle("-fx-font-weight: bold;");
lblNombre.setWrapText(true);

Label lblPrecio = new Label("$" + String.format("%.0f", producto.getPrecio() * (1 -
producto.getDescuento())));
infoProducto.getChildren().addAll(lblNombre, lblPrecio);

// Controles de cantidad
HBox controles = new HBox(5);
controles.setAlignment(Pos.CENTER_RIGHT);
HBox.setHgrow(controles, Priority.ALWAYS);

Label lblTextoCantidad = new Label("Cant:");

Label lblCantidad = new Label("1");
lblCantidad.setStyle("-fx-font-size: 12px; -fx-font-weight: bold;");
lblCantidad.setMinWidth(15);

Button btnAumentar = new Button("+");

```

```

btnAumentar.setStyle("-fx-cursor: hand; -fx-background-color: #56D23D");
btnAumentar.setMinWidth(30);
btnAumentar.setPrefWidth(30);

Button btnDisminuir = new Button("-");
btnDisminuir.setStyle("-fx-cursor: hand; -fx-background-color: #FA3C3C");
btnDisminuir.setMinWidth(30);
btnDisminuir.setPrefWidth(30);

ImageView iconoBorrar = new ImageView(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/borrarindividual.png")));
iconoBorrar.setFitWidth(20);
iconoBorrar.setFitHeight(20);
Button btnBorrar = new Button();
btnBorrar.setGraphic(iconoBorrar);
btnBorrar.setStyle("-fx-background-color: transparent;-fx-cursor: hand");

// Eventos
//aumentar cantidad del producto
btnAumentar.setOnAction(e -> {
    int cantidad = Integer.parseInt(lblCantidad.getText()) + 1;
    lblCantidad.setText(String.valueOf(cantidad));
    actualizarCantidadCarrito();
    actualizarTotales();
});

//disminuir cantidad del producto
btnDisminuir.setOnAction(e -> {
    int cantidad = Integer.parseInt(lblCantidad.getText()) - 1;
    if (cantidad <= 0) {
        contenedorCarrito.getChildren().remove(contenedor);
        actualizarCantidadCarrito();
        actualizarTotales();
    }else {
        lblCantidad.setText(String.valueOf(cantidad));
        actualizarCantidadCarrito();
        actualizarTotales();
    }
});

//borrar un producto
btnBorrar.setOnAction(e -> {
    contenedorCarrito.getChildren().remove(contenedor);
    actualizarCantidadCarrito();
    actualizarTotales();
});

//mostrar opciones, contenedor, agregar producto, actualizar los totales
controles.getChildren().addAll(lblTextoCantidad, lblCantidad, btnAumentar, btnDisminuir,
btnBorrar);
HBox.setHgrow(infoProducto, Priority.ALWAYS);
contenedor.getChildren().addAll(imagen, infoProducto, controles);
contenedorCarrito.getChildren().add(contenedor);
listaCarrito.agregarProducto(producto);
actualizarTotales();

```

```
}

//metodos para agregar los productos individualmente al carrito
@FXML
private void agregarAlCarritoProducto1(ActionEvent event) {
    agregarProductoAlCarrito(producto1);
    actualizarCantidadCarrito();
}

@FXML
private void agregarAlCarritoProducto2(ActionEvent event) {
    agregarProductoAlCarrito(producto2);
    actualizarCantidadCarrito();
}

@FXML
private void agregarAlCarritoProducto3(ActionEvent event) {
    agregarProductoAlCarrito(producto3);
    actualizarCantidadCarrito();
}

@FXML
private void agregarAlCarritoProducto4(ActionEvent event) {
    agregarProductoAlCarrito(producto4);
    actualizarCantidadCarrito();
}

@FXML
private void agregarAlCarritoProducto5(ActionEvent event) {
    agregarProductoAlCarrito(producto5);
    actualizarCantidadCarrito();
}

@FXML
private void agregarAlCarritoProducto6(ActionEvent event) {
    agregarProductoAlCarrito(producto6);
    actualizarCantidadCarrito();
}

@FXML
private void agregarAlCarritoProducto7(ActionEvent event) {
    agregarProductoAlCarrito(producto7);
    actualizarCantidadCarrito();
}

@FXML
private void agregarAlCarritoProducto8(ActionEvent event) {
    agregarProductoAlCarrito(producto8);
    actualizarCantidadCarrito();
}

@FXML
private void agregarAlCarritoProducto9(ActionEvent event) {
    agregarProductoAlCarrito(producto9);
    actualizarCantidadCarrito();
}

@FXML
private void agregarAlCarritoProducto10(ActionEvent event) {
    agregarProductoAlCarrito(producto10);
    actualizarCantidadCarrito();
}
```

```

//metodo para actualizar la cantidad de elementos que hay en el carrito
private void actualizarCantidadCarrito() {
    int total = 0;
    for (Node node : contenedorCarrito.getChildren()) {
        if (node instanceof HBox) {
            HBox item = (HBox) node;
            HBox controles = (HBox) item.getChildren().get(2); // donde están los botones y la cantidad
            Label lblCantidad = (Label) controles.getChildren().get(1); // el número
            total += Integer.parseInt(lblCantidad.getText());
        }
    }
    lblCantidadCarrito.setText(String.valueOf(total));
}

//metodo para eliminar los productos del carrito
@FXML
private void vaciarCarrito(MouseEvent event) {
    // Eliminar todos los productos de la lista lógica
    listaCarrito.vaciarCarrito();

    // Eliminar todos los elementos visuales del contenedor
    contenedorCarrito.getChildren().clear();

    // Actualizar etiquetas de cantidad y totales
    lblCantidadCarrito.setText("0");
    lblSubtotal.setText("$0");
    lblDescuento.setText("$0");
    lblTotal.setText("$0");
}

//Crear un arreglo de productos para facilitar su ordenamiento
private Producto[] productosArray;

// Método para inicializar el arreglo de productos
private void inicializarArregloProductos() {
    // Verificar que todas los productos existan
    if (producto1 == null || producto2 == null || producto3 == null || producto4 == null || producto5
    == null ||
        producto6 == null || producto7 == null || producto8 == null || producto9 == null || producto10
    == null) {
        System.out.println("ERROR: Uno o más productos no han sido inicializados correctamente");
        return;
    }
    // Si todos los productos existen, crear el arreglo
    productosArray = new Producto[]{
        producto1, producto2, producto3, producto4, producto5,
        producto6, producto7, producto8, producto9, producto10
    };
}

//metodo que se ejecuta cuando el usuario selecciona una opción en el ComboBox de ordenamiento
@FXML
private void filtrarProductos(ActionEvent event) {
    String criterio = comboFiltro.getValue();
    if (criterio == null) return;
}

```

```

try {
    //asegurarnos que el arreglo está inicializado
    if (productosArray == null) {
        inicializarArregloProductos();
    }
    //verificamos si el arreglo se inicializó correctamente
    if (productosArray == null) {
        System.out.println("ERROR: No se pudo inicializar el arreglo de productos");
        return;
    }
    //Ordenamos según el criterio seleccionado
    switch (criterio) {
        case "Precio Mayor":
            ordenarPorPrecioMayor();
            break;
        case "Precio Menor":
            ordenarPorPrecioMenor();
            break;
        case "Descuento":
            ordenarPorDescuento();
            break;
        default:
            System.out.println("Criterio no reconocido: " + criterio);
            return;
    }

    //Actualizamos la visualización con el nuevo orden
    actualizarVisualizacionProductos();

}catch (Exception e) {
    System.out.println("ERROR durante el filtrado: " + e.getMessage());
    e.printStackTrace();
}
}

//metodo que ordena los productos de mayor a menor precio (precio con descuento)
private void ordenarPorPrecioMayor() {
    //verificamos si hay elementos nulos en el arreglo
    boolean hayElementosNulos = false;
    for (int i = 0; i < productosArray.length; i++) {
        if (productosArray[i] == null) {
            hayElementosNulos = true;
            System.out.println("ADVERTENCIA: Elemento nulo encontrado en la posición " + i);
        }
    }
    //Si hay elementos nulos, inicializar o reinicializar el arreglo
    if (hayElementosNulos) {
        inicializarArregloProductos();
        System.out.println("Se ha reinicializado el arreglo de productos debido a elementos nulos");
        return;
    }
    //cocontinuamos con el ordenamiento normal si no hay elementos nulos
    for (int i = 0; i < productosArray.length - 1; i++) {

```

```

for (int j = 0; j < productosArray.length - i - 1; j++) {
    // Comparamos precios con descuento aplicado
    double precioDespuesDescuentoJ = productosArray[j].getPrecio() * (1 -
    productosArray[j].getDescuento());
    double precioDespuesDescuentoJPlus1 = productosArray[j + 1].getPrecio() * (1 -
    productosArray[j + 1].getDescuento());

    if (precioDespuesDescuentoJ < precioDespuesDescuentoJPlus1) {
        // Intercambiar productos
        Producto temp = productosArray[j];
        productosArray[j] = productosArray[j + 1];
        productosArray[j + 1] = temp;
    }
}
}

//metodo que ordena los productos de menor a mayor precio (precio con descuento)
private void ordenarPorPrecioMenor() {
    //Verificamos si hay elementos nulos en el arreglo
    boolean hayElementosNulos = false;
    for (int i = 0; i < productosArray.length; i++) {
        if (productosArray[i] == null) {
            hayElementosNulos = true;
            System.out.println("ADVERTENCIA: Elemento nulo encontrado en la posición " + i);
        }
    }

    //Si hay elementos nulos, inicializar o reinicializar el arreglo
    if (hayElementosNulos) {
        inicializarArregloProductos();
        System.out.println("Se ha reinicializado el arreglo de productos debido a elementos nulos");
        return; //
    }
    //continuamos con el ordenamiento normal si no hay elementos nulos
    for (int i = 0; i < productosArray.length - 1; i++) {
        for (int j = 0; j < productosArray.length - i - 1; j++) {
            //Comparar precios con descuento aplicado
            double precioDespuesDescuentoJ = productosArray[j].getPrecio() * (1 -
            productosArray[j].getDescuento());
            double precioDespuesDescuentoJPlus1 = productosArray[j + 1].getPrecio() * (1 -
            productosArray[j + 1].getDescuento());

            if (precioDespuesDescuentoJ > precioDespuesDescuentoJPlus1) {
                //Intercambiar productos
                Producto temp = productosArray[j];
                productosArray[j] = productosArray[j + 1];
                productosArray[j + 1] = temp;
            }
        }
    }
}

//metodo que ordena los productos por porcentaje de descuento (de mayor a menor)
private void ordenarPorDescuento() {

```

```

//Verificamos si hay elementos nulos en el arreglo
boolean hayElementosNulos = false;
for (int i = 0; i < productosArray.length; i++) {
    if (productosArray[i] == null) {
        hayElementosNulos = true;
        System.out.println("ADVERTENCIA: Elemento nulo encontrado en la posición " + i);
    }
}
// Si hay elementos nulos, inicializar o reinicializar el arreglo
if (hayElementosNulos) {
    inicializarArregloProductos();
    System.out.println("Se ha reinicializado el arreglo de productos debido a elementos nulos");
    return;
}
//Continuamos con el ordenamiento normal si no hay elementos nulos
for (int i = 0; i < productosArray.length - 1; i++) {
    for (int j = 0; j < productosArray.length - i - 1; j++) {
        //Comparar porcentajes de descuento
        if (productosArray[j].getDescuento() < productosArray[j + 1].getDescuento()) {
            // Intercambiar productos
            Producto temp = productosArray[j];
            productosArray[j] = productosArray[j + 1];
            productosArray[j + 1] = temp;
        }
    }
}

//metodo que actualiza la visualización de los productos según el nuevo orden
private void actualizarVisualizacionProductos() {
    // Actualizar producto 1
    lblNombre1.setText(productosArray[0].getNombre());
    lblPrecioOriginal1.setText("$" + String.format("%.0f", productosArray[0].getPrecio()));
    lblPrecioDescuento1.setText("$" + String.format("%.0f", productosArray[0].getPrecio() * (1 - productosArray[0].getDescuento())));
    lblPorcentaje1.setText((int)(productosArray[0].getDescuento() * 100) + "% OFF");
    lblMarca1.setText(productosArray[0].getMarca());
    Image imagen1 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" + productosArray[0].getImagenUrl()));
    imgProducto1.setImage(imagen1);

    // Actualizar producto 2
    lblNombre2.setText(productosArray[1].getNombre());
    lblPrecioOriginal2.setText("$" + String.format("%.0f", productosArray[1].getPrecio()));
    lblPrecioDescuento2.setText("$" + String.format("%.0f", productosArray[1].getPrecio() * (1 - productosArray[1].getDescuento())));
    lblPorcentaje2.setText((int)(productosArray[1].getDescuento() * 100) + "% OFF");
    lblMarca2.setText(productosArray[1].getMarca());
    Image imagen2 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" + productosArray[1].getImagenUrl()));
    imgProducto2.setImage(imagen2);

    // Actualizar producto 3
    lblNombre3.setText(productosArray[2].getNombre());
    lblPrecioOriginal3.setText("$" + String.format("%.0f", productosArray[2].getPrecio()));

```

```

lblPrecioDescuento3.setText("$" + String.format("%.0f", productosArray[2].getPrecio() * (1 -
productosArray[2].getDescuento())));
lblPorcentaje3.setText((int)(productosArray[2].getDescuento() * 100) + "% OFF");
lblMarca3.setText(productosArray[2].getMarca());
Image imagen3 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
productosArray[2].getImagenUrl()));
imgProducto3.setImage(imagen3);

// Actualizar producto 4
lblNombre4.setText(productosArray[3].getNombre());
lblPrecioOriginal4.setText("$" + String.format("%.0f", productosArray[3].getPrecio()));
lblPrecioDescuento4.setText("$" + String.format("%.0f", productosArray[3].getPrecio() * (1 -
productosArray[3].getDescuento())));
lblPorcentaje4.setText((int)(productosArray[3].getDescuento() * 100) + "% OFF");
lblMarca4.setText(productosArray[3].getMarca());
Image imagen4 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
productosArray[3].getImagenUrl()));
imgProducto4.setImage(imagen4);

// Actualizar producto 5
lblNombre5.setText(productosArray[4].getNombre());
lblPrecioOriginal5.setText("$" + String.format("%.0f", productosArray[4].getPrecio()));
lblPrecioDescuento5.setText("$" + String.format("%.0f", productosArray[4].getPrecio() * (1 -
productosArray[4].getDescuento())));
lblPorcentaje5.setText((int)(productosArray[4].getDescuento() * 100) + "% OFF");
lblMarca5.setText(productosArray[4].getMarca());
Image imagen5 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
productosArray[4].getImagenUrl()));
imgProducto5.setImage(imagen5);

// Actualizar producto 6
lblNombre6.setText(productosArray[5].getNombre());
lblPrecioOriginal6.setText("$" + String.format("%.0f", productosArray[5].getPrecio()));
lblPrecioDescuento6.setText("$" + String.format("%.0f", productosArray[5].getPrecio() * (1 -
productosArray[5].getDescuento())));
lblPorcentaje6.setText((int)(productosArray[5].getDescuento() * 100) + "% OFF");
lblMarca6.setText(productosArray[5].getMarca());
Image imagen6 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
productosArray[5].getImagenUrl()));
imgProducto6.setImage(imagen6);

// Actualizar producto 7
lblNombre7.setText(productosArray[6].getNombre());
lblPrecioOriginal7.setText("$" + String.format("%.0f", productosArray[6].getPrecio()));
lblPrecioDescuento7.setText("$" + String.format("%.0f", productosArray[6].getPrecio() * (1 -
productosArray[6].getDescuento())));
lblPorcentaje7.setText((int)(productosArray[6].getDescuento() * 100) + "% OFF");
lblMarca7.setText(productosArray[6].getMarca());
Image imagen7 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
productosArray[6].getImagenUrl()));
imgProducto7.setImage(imagen7);

// Actualizar producto 8
lblNombre8.setText(productosArray[7].getNombre());
lblPrecioOriginal8.setText("$" + String.format("%.0f", productosArray[7].getPrecio()));

```

```

lblPrecioDescuento8.setText("$" + String.format("%.0f", productosArray[7].getPrecio() * (1 -
productosArray[7].getDescuento())));
lblPorcentaje8.setText((int)(productosArray[7].getDescuento() * 100) + "% OFF");
lblMarca8.setText(productosArray[7].getMarca());
Image imagen8 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
productosArray[7].getImagenUrl()));
imgProducto8.setImage(imagen8);

// Actualizar producto 9
lblNombre9.setText(productosArray[8].getNombre());
lblPrecioOriginal9.setText("$" + String.format("%.0f", productosArray[8].getPrecio()));
lblPrecioDescuento9.setText("$" + String.format("%.0f", productosArray[8].getPrecio() * (1 -
productosArray[8].getDescuento())));
lblPorcentaje9.setText((int)(productosArray[8].getDescuento() * 100) + "% OFF");
lblMarca9.setText(productosArray[8].getMarca());
Image imagen9 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
productosArray[8].getImagenUrl()));
imgProducto9.setImage(imagen9);

// Actualizar producto 10
lblNombre10.setText(productosArray[9].getNombre());
lblPrecioOriginal10.setText("$" + String.format("%.0f", productosArray[9].getPrecio()));
lblPrecioDescuento10.setText("$" + String.format("%.0f", productosArray[9].getPrecio() * (1 -
productosArray[9].getDescuento())));
lblPorcentaje10.setText((int)(productosArray[9].getDescuento() * 100) + "% OFF");
lblMarca10.setText(productosArray[9].getMarca());
Image imagen10 = new Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
productosArray[9].getImagenUrl()));
imgProducto10.setImage(imagen10);

// También actualizamos las referencias de productos originales para mantener la coherencia
producto1 = productosArray[0];
producto2 = productosArray[1];
producto3 = productosArray[2];
producto4 = productosArray[3];
producto5 = productosArray[4];
producto6 = productosArray[5];
producto7 = productosArray[6];
producto8 = productosArray[7];
producto9 = productosArray[8];
producto10 = productosArray[9];
}

//metodo que muestra el panel de opciones
@FXML
private void mostrarOpciones(MouseEvent event) {
    panelOpciones.setVisible(true);
    panelOpciones.setManaged(true);
}

//metodo que permite volver del panel de opciones al panel principal
@FXML
private void volverOpciones(MouseEvent event) {
    panelOpciones.setVisible(false);
    panelOpciones.setManaged(false);
}

```

```

}

//metodo que muestra el panel de favoritos
@FXML
private void mostrarFavoritos(MouseEvent event) {
    panelOpciones.setVisible(false);
    panelOpciones.setManaged(false);
    panelFavoritos.setVisible(true);
    panelFavoritos.setManaged(true);
    opacidad.setVisible(true);
    opacidad.setManaged(true);
    opacidad.setOpacity(0.5);
}

//metodo que permite volver del panel de favoritos al panel principal
@FXML
private void volverFavoritos(){
    panelFavoritos.setVisible(false);
    panelFavoritos.setManaged(false);
    opacidad.setVisible(false);
    opacidad.setManaged(false);
}

//metodo que permite agregar un producto a favoritos
private void agregarProductoAFavoritos(Producto producto) {
    // Buscar si el producto ya existe en favoritos
    for (Node node : contenedorFavoritos.getChildren()) {
        if (node instanceof HBox) {
            HBox item = (HBox) node;
            VBox infoBox = (VBox) item.getChildren().get(1);
            Label nameLabel = (Label) infoBox.getChildren().get(0);

            if (nameLabel.getText().equals(producto.getNombre())) {
                return;
            }
        }
    }
    //Si llegamos aquí, el producto no existe en favoritos, así que añadimos uno nuevo
    HBox contenedor = new HBox(8);
    contenedor.setAlignment(Pos.CENTER_LEFT);
    contenedor.setPadding(new Insets(10));
    contenedor.setStyle("-fx-background-color: white; -fx-border-color: lightgray;");
    contenedor.setPrefWidth(397);

    // Imagen del producto
    ImageView imagen = new ImageView(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" + producto.getImagenUrl())));
    imagen.setFitWidth(50);
    imagen.setFitHeight(50);

    // Información del producto
    VBox infoProducto = new VBox(5);
    infoProducto.setPrefWidth(250);
    infoProducto.setMinWidth(150);
}

```

```

// Nombre del producto
Label lblNombre = new Label(producto.getNombre());
lblNombre.setStyle("-fx-font-weight: bold;");
lblNombre.setWrapText(true);

// Mostrar el precio con descuento
Label lblPrecio = new Label("$" + String.format("%.0f", producto.getPrecio() * (1 -
producto.getDescuento())));

//Si tiene descuento, mostrar el precio original tachado
if (producto.getDescuento() > 0) {
    Label lblPrecioOriginal = new Label("$" + String.format("%.0f", producto.getPrecio()));
    lblPrecioOriginal.setStyle("-fx-text-fill: gray; -fx-strikethrough: true;");
    infoProducto.getChildren().addAll(lblNombre, lblPrecioOriginal, lblPrecio);
} else {
    infoProducto.getChildren().addAll(lblNombre, lblPrecio);
}

// Botón para eliminar de favoritos
HBox controles = new HBox(5);
controles.setAlignment(Pos.CENTER_RIGHT);
HBox.setHgrow(controles, Priority.ALWAYS);

// Botón para eliminar de favoritos con ícono
ImageView iconoEliminar = new ImageView(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/borrarindividual.png")));
iconoEliminar.setFitWidth(20);
iconoEliminar.setFitHeight(20);

Button btnEliminar = new Button();
btnEliminar.setGraphic(iconoEliminar);
btnEliminar.setStyle("-fx-background-color: transparent; -fx-cursor: hand;");

// Evento para eliminar de favoritos
btnEliminar.setOnAction(e -> {
    contenedorFavoritos.getChildren().remove(contenedor);
    // Cambiar el estado en el mapa
    productosEnFavoritos.put(producto.getNombre(), false);

    // Cambiar la imagen del producto, si está registrada
    ImageView imgFavorito = iconosFavoritos.get(producto.getNombre());
    if (imgFavorito != null) {
        imgFavorito.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/favoritos.png")));
    }
});

//mostrar controles y contenedor
controles.getChildren().add(btnEliminar);
HBox.setHgrow(infoProducto, Priority.ALWAYS);

contenedor.getChildren().addAll(imagen, infoProducto, controles);
contenedorFavoritos.getChildren().add(contenedor);
}

```

```

// Método que se llama cuando se hace clic en la imagen de favorito
@FXML
private void imagenFavoritos(MouseEvent event) {
    // Obtener la fuente del evento (la ImageView en la que se hizo clic)
    ImageView imgFavorito = (ImageView) event.getSource();

    // Determinar qué producto corresponde a esta ImageView
    Producto productoSeleccionado = obtenerProductoPorImageViewFavorito(imgFavorito);
    iconosFavoritos.put(productoSeleccionado.getNombre(), imgFavorito);

    // Si no se pudo determinar el producto, salir
    if (productoSeleccionado == null) return;

    // Verificar si el producto ya está en favoritos
    boolean estaEnFavoritos =
        productosEnFavoritos.getOrDefault(productoSeleccionado.getNombre(), false);

    if (estaEnFavoritos) {
        // Si ya está en favoritos, eliminarlo
        eliminarProductoDeFavoritos(productoSeleccionado);
        //Cambiamos la imagen a la original
        imgFavorito.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/favoritos.png")));
        // Actualizamos estado
        productosEnFavoritos.put(productoSeleccionado.getNombre(), false);
    } else {
        // Si no está en favoritos, lo agregamos
        agregarProductoAFavoritos(productoSeleccionado);
        // Cambiar la imagen a la imagen cuando está en favoritos
        imgFavorito.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/favoritoAgregado.png")));
        // Actualizamos estado
        productosEnFavoritos.put(productoSeleccionado.getNombre(), true);
    }
}

//método que determina qué producto corresponde a la ImageView en la que se hizo clic
private Producto obtenerProductoPorImageViewFavorito(ImageView imgFavorito) {
    // Obtener el ID de la ImageView para identificar qué producto es
    String id = imgFavorito.getId();

    // Según el id de la ImageView, devolvemos el producto correspondiente
    if (id.equals("imgFavorito1")) return producto1;
    if (id.equals("imgFavorito2")) return producto2;
    if (id.equals("imgFavorito3")) return producto3;
    if (id.equals("imgFavorito4")) return producto4;
    if (id.equals("imgFavorito5")) return producto5;
    if (id.equals("imgFavorito6")) return producto6;
    if (id.equals("imgFavorito7")) return producto7;
    if (id.equals("imgFavorito8")) return producto8;
    if (id.equals("imgFavorito9")) return producto9;
    if (id.equals("imgFavorito10")) return producto10;

    // Si todo falla, devolver null
}

```

```

        System.out.println("No se pudo determinar el producto para la ImageView: " + id);
        return null;
    }

    //metodo para eliminar un producto de la lista de favoritos
    private void eliminarProductoDeFavoritos(Producto producto) {
        // Buscar y eliminar el producto de la vista de favoritos
        for (Node node : contenedorFavoritos.getChildren()) {
            if (node instanceof HBox) {
                HBox item = (HBox) node;
                // Verificar que hay al menos 2 elementos en el HBox
                if (item.getChildren().size() >= 2) {
                    VBox infoBox = (VBox) item.getChildren().get(1);
                    // Verificar que hay elementos en el VBox
                    if (!infoBox.getChildren().isEmpty()) {
                        Label nameLabel = (Label) infoBox.getChildren().get(0);

                        if (nameLabel.getText().equals(producto.getNombre())) {
                            // Encontramos el producto, lo eliminamos
                            contenedorFavoritos.getChildren().remove(node);
                            return;
                        }
                    }
                }
            }
        }
    }

    //metodo para varias todos los favoritos
    @FXML
    private void vaciarFavoritos(MouseEvent event) {
        //Vaciamos visualmente
        contenedorFavoritos.getChildren().clear();
        //Cambiamos el estado en el Map
        for (String nombreProducto : productosEnFavoritos.keySet()) {
            productosEnFavoritos.put(nombreProducto, false);
            //Cambiemos la imagen
            ImageView img = iconosFavoritos.get(nombreProducto);
            if (img != null) {
                img.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/favoritos.png")));
            }
        }
    }

    //metodo para agregar un producto a la lista de deseos
    private void agregarProductoAListaDeseos(Producto producto) {
        //creamos el contenedor del producto
        HBox contenedor = new HBox(10);
        contenedor.setAlignment(Pos.CENTER_LEFT);
        contenedor.setPadding(new Insets(10));
        contenedor.setStyle("-fx-background-color: white; -fx-border-color: lightgray;");
        contenedor.setPrefWidth(400);

        //Imagen del producto

```

```

ImageView imagen = new ImageView(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" + producto.getImagenUrl())));
imagen.setFitWidth(50);
imagen.setFitHeight(50);

//Información del producto
VBox infoProducto = new VBox(5);
infoProducto.setPrefWidth(220);

//Nombre del producto
Label lblNombre = new Label(producto.getNombre());
lblNombre.setStyle("-fx-font-weight: bold;");
lblNombre.setWrapText(true);

Label lblPrecioConDescuento = new Label("$" + String.format("%.0f", producto.getPrecio() *
(1 - producto.getDescuento())));

//precio con descuento del producto
if (producto.getDescuento() > 0) {
    Label lblPrecioOriginal = new Label("$" + String.format("%.0f", producto.getPrecio()));
    lblPrecioOriginal.setStyle("-fx-text-fill: gray; -fx-strikethrough: true;");
    infoProducto.getChildren().addAll(lblNombre, lblPrecioOriginal, lblPrecioConDescuento);
} else {
    infoProducto.getChildren().addAll(lblNombre, lblPrecioConDescuento);
}

//Botón para añadir al carrito
Button btnAregarCarrito = new Button("Añadir al carrito");
btnAregarCarrito.setStyle("-fx-background-color: #56D23D; -fx-text-fill: white; -fx-cursor:
hand;");

//evento si la persona le da al boton de añadir al carrito
btnAregarCarrito.setOnAction(e -> {
    agregarProductoAlCarrito(producto);
    actualizarCantidadCarrito();
    actualizarTotales();
    contenedorListaDeseos.getChildren().remove(contenedor);
    //Cambiar el estado en el mapa
    productosEnListaDeseos.put(producto.getNombre(), false);
    //Cambiar la imagen del producto
    ImageView imgListaDeseos = iconosListaDeseos.get(producto.getNombre());
    if (imgListaDeseos != null) {
        imgListaDeseos.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/Listadeseos.png")));
    }
});

//imagen para eliminar de lista de deseos
ImageView iconoEliminar = new ImageView(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/borrarindividual.png")));
iconoEliminar.setFitWidth(20);
iconoEliminar.setFitHeight(20);
Button btnEliminar = new Button();
btnEliminar.setGraphic(iconoEliminar);
btnEliminar.setStyle("-fx-background-color: transparent; -fx-cursor: hand;");
```

```

//evento si la persona le da en eliminar de favoritos
btnEliminar.setOnAction(e -> {
    contenedorListaDeseos.getChildren().remove(contenedor);
    //Cambiemos el estado en el mapa
    productosEnListaDeseos.put(producto.getNombre(), false);

    //Cambiemos la imagen del producto
    ImageView imgListaDeseos = iconosListaDeseos.get(producto.getNombre());
    if (imgListaDeseos != null) {
        imgListaDeseos.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/Listadeseos.png")));
    }
});

//mostramos botones y contenedor
VBox botones = new VBox(5, btnAgregarCarrito, btnEliminar);
botones.setAlignment(Pos.CENTER_RIGHT);
botones.setPrefWidth(120);
contenedor.getChildren().addAll(imagen, infoProducto, botones);
contenedorListaDeseos.getChildren().add(contenedor);
}

//metodo que se llama para cambiar la imagen que se muestra de lista de deseos en el producto
@FXML
private void imagenListaDeseos(MouseEvent event) {
    // Obtener la fuente del evento (la ImageView en la que se hizo clic)
    ImageView imgListaDeseos = (ImageView) event.getSource();

    // Determinar qué producto corresponde a esta ImageView
    Producto productoSeleccionado = obtenerProductoPorImageViewDeseo(imgListaDeseos);
    iconosListaDeseos.put(productoSeleccionado.getNombre(), imgListaDeseos);

    // Si no se pudo determinar el producto, salir
    if (productoSeleccionado == null) return;

    // Verificar si el producto ya está en favoritos
    boolean estaEnListaDeseos =
    productosEnListaDeseos.getOrDefault(productoSeleccionado.getNombre(), false);

    if (estaEnListaDeseos) {
        // Si ya está en favoritos, eliminarlo
        eliminarProductoDeListaDeseos(productoSeleccionado);
        // Cambiar la imagen a la original
        imgListaDeseos.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/Listadeseos.png")));
        // Actualizar estado
        productosEnListaDeseos.put(productoSeleccionado.getNombre(), false);
    } else {
        // Si no está en favoritos, agregarlo
        agregarProductoAListaDeseos(productoSeleccionado);
        // Cambiar la imagen a la imagen cuando está en lista de deseos
        imgListaDeseos.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/ListadeseosAgregado.png")));
    }
}

```

```

        // Actualizar estado
        productosEnListaDeseos.put(productoSeleccionado.getNombre(), true);
    }

}

//metodo que permite determinar qué producto corresponde a la ImageView en la que se hizo clic
private Producto obtenerProductoPorImageViewDeseo(ImageView imgListaDeseos) {
    // Obtener el ID de la ImageView para identificar qué producto es
    String id = imgListaDeseos.getId();

    // Según el id de la ImageView, devolvemos el producto correspondiente
    if (id.equals("imgDeseos1")) return producto1;
    if (id.equals("imgDeseos2")) return producto2;
    if (id.equals("imgDeseos3")) return producto3;
    if (id.equals("imgDeseos4")) return producto4;
    if (id.equals("imgDeseos5")) return producto5;
    if (id.equals("imgDeseos6")) return producto6;
    if (id.equals("imgDeseos7")) return producto7;
    if (id.equals("imgDeseos8")) return producto8;
    if (id.equals("imgDeseos9")) return producto9;
    if (id.equals("imgDeseos10")) return producto10;

    // Si todo falla, devolver null
    System.out.println("No se pudo determinar el producto para la ImageView: " + id);
    return null;
}

//metodo para eliminar un producto de la lista de favoritos
private void eliminarProductoDeListaDeseos(Producto producto) {
    // Buscar y eliminar el producto de la vista de favoritos
    for (Node node : contenedorListaDeseos.getChildren()) {
        if (node instanceof HBox) {
            HBox item = (HBox) node;
            // Verificar que hay al menos 2 elementos en el HBox
            if (item.getChildren().size() >= 2) {
                VBox infoBox = (VBox) item.getChildren().get(1);
                // Verificar que hay elementos en el VBox
                if (!infoBox.getChildren().isEmpty()) {
                    Label nameLabel = (Label) infoBox.getChildren().get(0);

                    if (nameLabel.getText().equals(producto.getNombre())) {
                        // Encontramos el producto, lo eliminamos
                        contenedorListaDeseos.getChildren().remove(node);
                        return;
                    }
                }
            }
        }
    }
}

//metodo para vaciar toda la lista de deseos
@FXML
private void vaciarListaDeseos(MouseEvent event) {
    //Vaciamos visualmente
}

```

```

contenedorListaDeseos.getChildren().clear();

//Cambiamos el estado en el Map
for (String nombreProducto : productosEnListaDeseos.keySet()) {
    productosEnListaDeseos.put(nombreProducto, false);

    //Cambiar la imagen
    ImageView img = iconosListaDeseos.get(nombreProducto);
    if (img != null) {
        img.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/Listadeseos.png")));
    }
}

//metodo para mostrar la lista de deseos
@FXML
private void mostrarListaDeseos(MouseEvent event) {
    panelOpciones.setVisible(false);
    panelOpciones.setManaged(false);
    panelListaDeseos.setVisible(true);
    panelListaDeseos.setManaged(true);
    opacidad.setVisible(true);
    opacidad.setManaged(true);
    opacidad.setOpacity(0.5);
}

//metodo que permite volver del pane de la lista de deseos al panel principal
@FXML
private void volverListaDeseos(){
    panelListaDeseos.setVisible(false);
    panelListaDeseos.setManaged(false);
    opacidad.setVisible(false);
    opacidad.setManaged(false);
}

//metodo para agregar todos los productos de la lista de deseo al carrito
@FXML
private void agregarTodosAlCarrito(ActionEvent event) {
    // Si no hay productos en la lista de deseos, salimos del método
    if (contenedorListaDeseos.getChildren().isEmpty()) {
        return;
    }

    // Creamos una copia de los elementos para evitar problemas de concurrencia
    List<Node> elementosAEliminar = new ArrayList<>(contenedorListaDeseos.getChildren());

    // iteramos por cada elemento en la lista de deseos
    for (Node node : elementosAEliminar) {
        if (node instanceof HBox) {
            HBox item = (HBox) node;
            // Verificar que hay al menos 2 elementos en el HBox
            if (item.getChildren().size() >= 2) {
                VBox infoBox = (VBox) item.getChildren().get(1);
                // Verificar que hay elementos en el VBox
            }
        }
    }
}

```

```

if (!infoBox.getChildren().isEmpty()) {
    Label nameLabel = (Label) infoBox.getChildren().get(0);
    String nombreProducto = nameLabel.getText();

    // Buscar el producto correspondiente usando nuestro método auxiliar
    Producto producto = buscarProductoPorNombre(nombreProducto);

    if (producto != null) {
        // Agregar al carrito
        agregarProductoAlCarrito(producto);

        // Actualizar el estado en el mapa
        productosEnListaDeseos.put(nombreProducto, false);

        // Cambiar la imagen del producto si está registrada
        ImageView imgListaDeseos = iconosListaDeseos.get(nombreProducto);
        if (imgListaDeseos != null) {
            imgListaDeseos.setImage(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/Listadeseos.png")));
        }
    }
}

// Vaciar la lista de deseos
contenedorListaDeseos.getChildren().clear();
// Actualizar los totales y la cantidad en el carrito
actualizarCantidadCarrito();
actualizarTotales();
}

//metodo auxiliar para buscar un producto por su nombre (ayudado por el de listaproductos)
private Producto buscarProductoPorNombre(String nombre) {
    // Verificar cada producto
    if (producto1 != null && producto1.getNombre().equals(nombre)) return producto1;
    if (producto2 != null && producto2.getNombre().equals(nombre)) return producto2;
    if (producto3 != null && producto3.getNombre().equals(nombre)) return producto3;
    if (producto4 != null && producto4.getNombre().equals(nombre)) return producto4;
    if (producto5 != null && producto5.getNombre().equals(nombre)) return producto5;
    if (producto6 != null && producto6.getNombre().equals(nombre)) return producto6;
    if (producto7 != null && producto7.getNombre().equals(nombre)) return producto7;
    if (producto8 != null && producto8.getNombre().equals(nombre)) return producto8;
    if (producto9 != null && producto9.getNombre().equals(nombre)) return producto9;
    if (producto10 != null && producto10.getNombre().equals(nombre)) return producto10;

    System.out.println("No se encontró un producto con el nombre: " + nombre);
    return null;
}

//metodo que se ejecuta cuando el usuario le da en comprar producto (individual, solo para 1
producto)
@FXML
private void comprarProducto(Producto producto) {

```

```

try {
    // Calcular precio con descuento si aplica
    double precioFinal = producto.getPrecio() * (1 - producto.getDescuento());

    //cargamos el fxml de pagos
    FXMLLoader loader = new
    FXMLLoader(getClass().getResource("/Vista/PaginaPagos.fxml"));
    Parent root = loader.load();

    Stage stage = new Stage();
    stage.setScene(new Scene(root));
    stage.setTitle("Pagos");
    stage.setResizable(false);
    stage.centerOnScreen();
    stage.show();

    // Obtenemos el controlador y pasar el total
    PaginaPagosControlador pagosControlador = loader.getController();
    pagosControlador.setTotalPagar(precioFinal);

    //Configuramos el producto y callback para historial
    pagosControlador.setProductoComprado(producto);
    pagosControlador.setCallbackAgregarHistorial(() -> {
        String fechaActual =
LocalDate.now().format(DateTimeFormatter.ofPattern("dd/MM/yyyy"));
        HistorialCompra compra = new HistorialCompra(producto, fechaActual, precioFinal);
        historialCompras.add(compra);
    });
} catch (IOException e) {
    e.printStackTrace();
}
}

//metodos para comprar el producto seleccionado
@FXML
private void comprarProducto1(ActionEvent event) {
    comprarProducto(producto1);
}
@FXML
private void comprarProducto2(ActionEvent event) {
    comprarProducto(producto2);
}
@FXML
private void comprarProducto3(ActionEvent event) {
    comprarProducto(producto3);
}
@FXML
private void comprarProducto4(ActionEvent event) {
    comprarProducto(producto4);
}
@FXML
private void comprarProducto5(ActionEvent event) {
    comprarProducto(producto5);
}

```

```

private void comprarProducto6(ActionEvent event) {
    comprarProducto(producto6);
}
@FXML
private void comprarProducto7(ActionEvent event) {
    comprarProducto(producto7);
}
@FXML
private void comprarProducto8(ActionEvent event) {
    comprarProducto(producto8);
}
@FXML
private void comprarProducto9(ActionEvent event) {
    comprarProducto(producto9);
}
@FXML
private void comprarProducto10(ActionEvent event) {
    comprarProducto(producto10);
}

//metodo para comprar los productos(los que están dentro del carrito de compras)
@FXML
private void comprarCarritoCompleto(ActionEvent event) {
    try {
        double total = 0;
        //inicializamos una nueva lista con los productos dentro del carrito
        List<Producto> productosEnCarrito = new ArrayList<>();

        //recorremos el nodo para extraer los productos en el
        for (Node node : contenedorCarrito.getChildren()) {
            if (node instanceof HBox) {
                HBox item = (HBox) node;
                VBox infoBox = (VBox) item.getChildren().get(1);
                Label lblNombre = (Label) infoBox.getChildren().get(0);
                Label lblPrecio = (Label) infoBox.getChildren().get(infoBox.getChildren().size() - 1);

                HBox controles = (HBox) item.getChildren().get(2);
                Label lblCantidad = (Label) controles.getChildren().get(1);

                String nombre = lblNombre.getText();
                int cantidad = Integer.parseInt(lblCantidad.getText());

                Producto producto = buscarProductoPorNombre(nombre);
                if (producto != null) {
                    double precioConDescuento = producto.getPrecio() * (1 - producto.getDescuento());
                    total += precioConDescuento * cantidad;
                }
                for (int i = 0; i < cantidad; i++) {
                    productosEnCarrito.add(producto);
                }
            }
        }
        // Abrimos la ventana de pago
    }
}

```

```

FXMLLoader loader = new
FXMLLoader(getClass().getResource("/Vista/PaginaPagos.fxml"));
Parent root = loader.load();

PaginaPagosControlador pagosControlador = loader.getController();
pagosControlador.setTotalPagar(total);

Scene scene = new Scene(root);
Stage stage = new Stage();
stage.setTitle("Información de Pago");
stage.setScene(scene);
stage.show();

//llamos los controlador requeridos
pagosControlador.setTotalPagar(total);
pagosControlador.setEsCompraDesdeCarrito(true);
pagosControlador.setProductosComprados(productosEnCarrito);
//permite agregar al archivo txt los datos
pagosControlador.setCallbackAgregarHistorial(() -> {
    String fechaActual =
LocalDate.now().format(DateTimeFormatter.ofPattern("dd/MM/yyyy"));
    for (Producto producto : productosEnCarrito) {
        double precioConDescuento = producto.getPrecio() * (1 - producto.getDescuento());
        HistorialCompra compra = new HistorialCompra(producto, fechaActual,
precioConDescuento);
        historialCompras.add(compra);
        agregarProductoAHistorial(compra);
    }
    guardarHistorialEnArchivo(usuarioActivo.getUsuario());
});
}

// pasamos el metodo para vaciar carrito
pagosControlador.setCallbackVaciarCarrito(() -> {
    listaCarrito.vaciarCarrito();
    contenedorCarrito.getChildren().clear();
    actualizarCantidadCarrito();
    actualizarTotales();
});
} catch (IOException e) {
    e.printStackTrace();
}
}

//metodo que permite ver que usuario registrado está activo
public void setUsuarioActivo(Usuario usuario) {
    this.usuarioActivo = usuario;
    mostrarDatosUsuario();
}

//metodo que permite mostrar los datos del usuario activo
private void mostrarDatosUsuario() {
    if (usuarioActivo == null) {
        System.out.println("usuarioActivo es null!");
        return;
    }
}

```

```

System.out.println("Mostrando datos de: " + usuarioActivo.getNombres());
txtNombres.setText(usuarioActivo.getNombres());
txtApellidos.setText(usuarioActivo.getApellidos());
txtSexo.setText(usuarioActivo.getSexo());
txtEdad.setText(String.valueOf(usuarioActivo.getEdad()));
txtPais.setText(usuarioActivo.getPais());
txtDireccion.setText(usuarioActivo.getDireccion());
}

//metodo que permite mostrar el panel de cuenta
@FXML
private void mostrarCuenta(MouseEvent event) {
    panelOpciones.setVisible(false);
    panelOpciones.setManaged(false);
    panelCuenta.setVisible(true);
    panelCuenta.setManaged(true);
    opacidad.setVisible(true);
    opacidad.setManaged(true);
    opacidad.setOpacity(0.5);
    mostrarDatosUsuario();
}

//metodo que permite volver del panel de cuenta al panel principal
@FXML
private void volverCuenta() {
    panelCuenta.setVisible(false);
    panelCuenta.setManaged(false);
    opacidad.setVisible(false);
    opacidad.setManaged(false);
}

//metodo que habilita el campo a editar en mi cuenta
private void habilitarCampo(TextField campo) {
    campo.setEditable(true);
    campo.requestFocus();
}

//metodos para editar los campos en mi cuenta
@FXML
private void editarNombre(MouseEvent event) {
    habilitarCampo(txtNombres);
}
@FXML
private void editarApellidos(MouseEvent event) {
    habilitarCampo(txtApellidos);
}
@FXML
private void editarSexo(MouseEvent event) {
    habilitarCampo(txtSexo);
}
@FXML
private void editarEdad(MouseEvent event) {
    habilitarCampo(txtEdad);
}

```

```

private void editarPais(MouseEvent event) {
    habilitarCampo(txtPais);
}
@FXML
private void editarDireccion(MouseEvent event) {
    habilitarCampo(txtDireccion);
}

//metodo para guardar los cambios de los campos editados en mi cuenta
@FXML
private void guardarCambios(ActionEvent event) {
    // Deshabilitar edición
    txtNombres.setEditable(false);
    txtApellidos.setEditable(false);
    txtSexo.setEditable(false);
    txtEdad.setEditable(false);
    txtPais.setEditable(false);
    txtDireccion.setEditable(false);

    // Actualizar los datos en el objeto Usuario
    if (usuarioActivo != null) {
        usuarioActivo.setNombre(txtNombres.getText());
        usuarioActivo.setApellido(txtApellidos.getText());
        usuarioActivo.setSexo(txtSexo.getText());
        usuarioActivo.setEdad(txtEdad.getText());
        usuarioActivo.setPais(txtPais.getText());
        usuarioActivo.setDireccion(txtDireccion.getText());

        //Buscar y actualizar el usuario en la lista enlazada
        nodo<Usuario> actual = Controlador.RegisterControlador.listaUsuarios.cabeza;
        while (actual != null) {
            if (actual.dato.getUsuario().equals(usuarioActivo.getUsuario())) {
                // Actualizar el usuario en el nodo
                actual.dato = usuarioActivo;
                break;
            }
            actual = actual.sig;
        }
        guardarTodosLosUsuarios();
    }
    // Mostrar mensaje de confirmación
    Alert alerta = new Alert(Alert.AlertType.INFORMATION);
    alerta.setTitle("Información actualizada");
    alerta.setHeaderText(null);
    alerta.setContentText("Tu información fue actualizada correctamente.");
    alerta.showAndWait();
}

//metodo para cancelar la edición de los cambios y volver a los que estaban originalmente
@FXML
private void cancelarCambios(ActionEvent event) {
    // Restaurar los valores originales
    if (usuarioActivo != null) {
        nombresOriginal = usuarioActivo.getNombres();
        apellidosOriginal = usuarioActivo.getApellidos();
    }
}

```

```

sexoOriginal = usuarioActivo.getSexo();
edadOriginal = usuarioActivo.getEdad();
paisOriginal = usuarioActivo.getpais();
direccionOriginal = usuarioActivo.getDireccion();

txtNombres.setText(nombresOriginal);
txtApellidos.setText(apellidosOriginal);
txtSexo.setText(sexoOriginal);
txtEdad.setText(edadOriginal);
txtPais.setText(paisOriginal);
txtDireccion.setText(direccionOriginal);
}

// Deshabilitar edición
txtNombres.setEditable(false);
txtApellidos.setEditable(false);
txtSexo.setEditable(false);
txtEdad.setEditable(false);
txtPais.setEditable(false);
txtDireccion.setEditable(false);

// Mostrar mensaje de cancelación
Alert alerta = new Alert(Alert.AlertType.INFORMATION);
alerta.setTitle("Cambios cancelados");
alerta.setHeaderText(null);
alerta.setContentText("Los cambios han sido cancelados.");
alerta.show();
}

//metodo para guardar los usuarios en un archivo txt
private void guardarTodosLosUsuarios() {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter("usuarios.txt", false))) {
        nodo<Usuario> actual = Controlador.RegisterControlador.listaUsuarios.cabeza;
        while (actual != null) {
            Usuario u = actual.datos;
            writer.write(u.getUsuario() + ";" +
                        u.getContrasena() + ";" +
                        u.getNombres() + ";" +
                        u.getApellidos() + ";" +
                        u.getSexo() + ";" +
                        u.getEdad() + ";" +
                        u.getPais() + ";" +
                        (u.getDireccion() != null ? u.getDireccion() : ""));
            writer.newLine();
            actual = actual.sig;
        }
    } catch (IOException e) {
        e.printStackTrace();
        System.out.println("Error al guardar los usuarios.");
    }
}

//metodo para cerrar sesion
@FXML
private void cerrarSesion(MouseEvent event){
}

```

```

Alert alerta = new Alert(Alert.AlertType.INFORMATION);
alerta.setTitle("Cierre de sesión exitoso");
alerta.setHeaderText(null);
alerta.setContentText("¡Gracias por comprar con nosotros, te esperamos nuevamente!");
alerta.show();
// Pausa de 2 segundos antes de abrir página principal
PauseTransition pausa = new PauseTransition(Duration.seconds(2));
pausa.setOnFinished(e -> {
    try {
        alerta.close();

        //cargamos fxml del login
        FXMLLoader loader = new FXMLLoader(getClass().getResource("/Vista/Login.fxml"));
        Parent root = loader.load();
        Stage stage = new Stage();
        stage.setScene(new Scene(root));
        stage.setTitle("SantiShop - Inicio de sesión");
        stage.getIcons().add(new Image(getClass().getResourceAsStream("/Vista/Imagenes/LOGO
FONDO BLANCO SANTISHOP.png")));
        stage.setResizable(false);
        stage.centerOnScreen();
        stage.show();

        ((Stage)((Label)event.getSource()).getScene().getWindow())).close();
    } catch (IOException ex) {
        Logger.getLogger(PaginaPrincipalControlador.class.getName()).log(Level.SEVERE, null,
ex);
    }
});

pausa.play();
}

//metodo para abrir el panel de historial de compra
@FXML
private void mostrarHistorialCompra(MouseEvent event) {
    panelOpciones.setVisible(false);
    panelOpciones.setManaged(false);
    panelHistorialCompra.setVisible(true);
    panelHistorialCompra.setManaged(true);
    opacidad.setVisible(true);
    opacidad.setManaged(true);
    opacidad.setOpacity(0.5);
    mostrarHistorialCompleto();
}

//metodo para cerrar el panel de historial de compra y volver al panel principal
@FXML
private void volverHistorialCompra() {
    panelHistorialCompra.setVisible(false);
    panelHistorialCompra.setManaged(false);
    opacidad.setVisible(false);
    opacidad.setManaged(false);
}

```

```

//metodo para agregar un producto comprado al historial de compra
private void agregarProductoAHistorial(HistorialCompra historialCompra) {
    // Crear el contenedor del producto
    HBox contenedor = new HBox(10);
    contenedor.setAlignment(Pos.CENTER_LEFT);
    contenedor.setPadding(new Insets(10));
    contenedor.setStyle("-fx-background-color: white; -fx-border-color: lightgray;");
    contenedor.setPrefWidth(400);

    // Imagen del producto
    ImageView imagen = new ImageView(new
    Image(getClass().getResourceAsStream("/Vista/Imagenes/Productos/" +
    historialCompra.getProducto().getImagenUrl())));
    imagen.setFitWidth(50);
    imagen.setFitHeight(50);

    // Información del producto
    VBox infoProducto = new VBox(5);
    infoProducto.setPrefWidth(300);

    //Nombre del producto
    Label lblNombre = new Label(historialCompra.getProducto().getNombre());
    lblNombre.setStyle("-fx-font-weight: bold;");
    lblNombre.setWrapText(true);

    Label lblPrecioCompra = new Label("$" + String.format("%,.0f",
    historialCompra.getPrecioCompra()));
    lblPrecioCompra.setStyle("-fx-font-weight: bold; -fx-text-fill: #2E8B57;");

    Label lblFecha = new Label(historialCompra.getFechaCompra());
    lblFecha.setStyle("-fx-font-size: 12px; -fx-text-fill: #666666;");

    infoProducto.getChildren().addAll(lblNombre, lblPrecioCompra, lblFecha);

    // Agregar elementos al contenedor
    contenedor.getChildren().addAll(imagen, infoProducto);

    // Agregar al contenedor del historial
    contenedorHistorialCompra.getChildren().add(contenedor);
}

//metodo para mostrar el historial de compra con todos los productos comprados
private void mostrarHistorialCompleto() {
    contenedorHistorialCompra.getChildren().clear();
    for (HistorialCompra compra : historialCompras) {
        agregarProductoAHistorial(compra);
    }
}

//metodo para guardar el historial de compra en un archivo txt
private void guardarHistorialEnArchivo(String usuario) {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter("historial_" + usuario + ".txt",
    true))) {
        for (HistorialCompra compra : historialCompras) {
            String linea = compra.getProducto().getNombre() + "|" +

```

```
compra.getPrecioCompra() + "|" +
compra.getProducto().getImagenUrl() + "|" +
compra.getFechaCompra());
writer.write(linea);
writer.newLine();
}
} catch (IOException e) {
e.printStackTrace();
}
}

//metodo para cargar el historial de compra de cada usuario
void cargarHistorialDesdeArchivo(String usuario) {
//limpiamos el historial de compra
historialCompras.clear();
File archivo = new File("historial_" + usuario + ".txt");

if (!archivo.exists()) return;

try (BufferedReader reader = new BufferedReader(new FileReader(archivo))) {
String linea;
while ((linea = reader.readLine()) != null) {
String[] partes = linea.split("\\|");
if (partes.length == 4) {
String nombre = partes[0];
double precio = Double.parseDouble(partes[1]);
String imagenUrl = partes[2];
String fecha = partes[3];

Producto producto = buscarProductoPorNombre(nombre);
if (producto == null) {
producto = new Producto(nombre, precio, 0, "", imagenUrl);
}

HistorialCompra compra = new HistorialCompra(producto, fecha, precio);
historialCompras.add(compra);
}
}
} catch (IOException e) {
e.printStackTrace();
}
}
}
```

Ahora agregaremos el código de la clase PaginaPagosControlador, este controlador es el que se encarga de procesar el pago de los productos y añadir el historial visualmente y guardarlo en el archivo.txt

```
package Controlador;  
  
import Modelo.HistorialCompra;  
import Modelo.Producto;  
import java.net.URL;  
import java.util.ArrayList;
```

```

import java.util.List;
import java.util.ResourceBundle;
import javafx.animation.PauseTransition;
import javafx.fxml.FXML;
import javafx.fxml.Initializable;
import javafx.scene.control.Alert;
import javafx.scene.control.Button;
import javafx.scene.control.ComboBox;
import javafx.scene.control.Label;
import javafx.scene.control.PasswordField;
import javafx.scene.control.TextField;
import javafx.stage.Stage;
import javafx.util.Duration;

/**
 * FXML Controller class
 *
 * @author santi
 */
public class PaginaPagosControlador implements Initializable {

    @FXML
    private ComboBox<String> comboBanco;
    @FXML
    private TextField txtNumeroCuenta;
    @FXML
    private ComboBox<String> comboTipoCuenta;
    @FXML
    private PasswordField txtClave;
    @FXML
    private Label lblTotal;
    @FXML
    private Button btnPagar;
    @FXML
    private Button btnCancelar;

    //declaramos e inicializamos una lista que almacenará objetos de tipo
    private List<Producto> productosComprados = new ArrayList<>();
    //Declara una variable de tipo Runnable.
    private Runnable callbackAgregarHistorial;

    private boolean esCompraDesdeCarrito = false;

    private Runnable callbackVaciarCarrito = null;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        comboBanco.getItems().addAll("Bancolombia", "Davivienda", "Nequi");
        comboTipoCuenta.getItems().addAll("Ahorros", "Corriente", "Cuenta Digital");
    }

    //método que marca de dónde proviene la compra
    public void setEsCompraDesdeCarrito(boolean valor) {
        this.esCompraDesdeCarrito = valor;
    }
}

```

```

//método que asigna un bloque de código (callback) que permite que otra clase defina qué debe
ocurrir automáticamente cuando se vacíe el carrito
public void setCallbackVaciarCarrito(Runnable callback) {
    this.callbackVaciarCarrito = callback;
}

//metodo que se ejecuta cuando el usuario le da en el boton pagar
@FXML
private void pagar() {
    String banco = comboBanco.getValue();
    String numeroCuenta = txtNumeroCuenta.getText();
    String tipoCuenta = comboTipoCuenta.getValue();
    String clave = txtClave.getText();

    // Verificar si alguno está vacío o no seleccionado
    if (banco == null || banco.isEmpty() ||
        numeroCuenta == null || numeroCuenta.isEmpty() ||
        tipoCuenta == null || tipoCuenta.isEmpty() ||
        clave == null || clave.isEmpty()) {

        // Mostrar alerta de advertencia
        Alert alerta = new Alert(Alert.AlertType.WARNING);
        alerta.setTitle("Campos incompletos");
        alerta.setHeaderText(null);
        alerta.setContentText("Por favor, complete todos los campos para continuar con el pago.");
        alerta.showAndWait();
        return;
    }

    // Mostrar alerta de "Procesando pago..."
    Alert procesando = new Alert(Alert.AlertType.INFORMATION);
    procesando.setTitle("Pago");
    procesando.setHeaderText(null);
    procesando.setContentText("Procesando pago... ");
    procesando.show();
    // Esperar 3 segundos y luego mostrar mensaje de éxito
    PauseTransition espera = new PauseTransition(Duration.seconds(3));
    espera.setOnFinished(e -> {
        procesando.close();
        //Aregar al historial antes de mostrar exito
        if (callbackAgregarHistorial != null) {
            callbackAgregarHistorial.run();
        }
    });

    Alert exito = new Alert(Alert.AlertType.INFORMATION);
    exito.setTitle("Compra exitosa");
    exito.setHeaderText(null);
    exito.setContentText("Compra completada. Los productos serán enviados a tu dirección
registrada. ¡Gracias por comprar con nosotros!");
    exito.show();
    // Esperar otros 3 segundos y cerrar ventana actual, volver al principal
    PauseTransition volver = new PauseTransition(Duration.seconds(3));
    volver.setOnFinished(ev -> {
        // Cerrar esta ventana
    });
}

```

```

Stage stage = (Stage) comboBanco.getScene().getWindow();
stage.close();

// Volver al principal (asegúrate de tener acceso)
// Y vaciar el carrito si fue una compra desde ahí
if(esCompraDesdeCarrito && callbackVaciarCarrito != null) {
    callbackVaciarCarrito.run();
}

});

volver.play();
};

espera.play();
}

//metodo que se ejecuta cuando la persona cancela la compra de un producto
@FXML
private void cancelar() {
    // Aquí va la lógica para cancelar el pago
    // Por ejemplo, limpiar los campos o cerrar la ventana
    txtNumeroCuenta.clear();
    txtClave.clear();
    // Para cerrar la ventana:
    Stage stage = (Stage) btnCancelar.getScene().getWindow();
    stage.close();
}

//metodo que muestra el total a pagar
public void setTotalPagar(double precioFinal) {
    // Formatear el número con puntos de miles y sin decimales
    String precioFormateado = String.format("%,.0f", precioFinal);
    lblTotal.setText("Total a pagar: $" + precioFormateado);
}

//metodo para ver los productos comprados
public void setProductoComprado(Producto producto) {
    this.productosComprados.clear();
    this.productosComprados.add(producto);
}

// Método para recibir lista de productos del carrito
public void setProductosComprados(List<Producto> productos) {
    this.productosComprados = new ArrayList<>(productos);
}

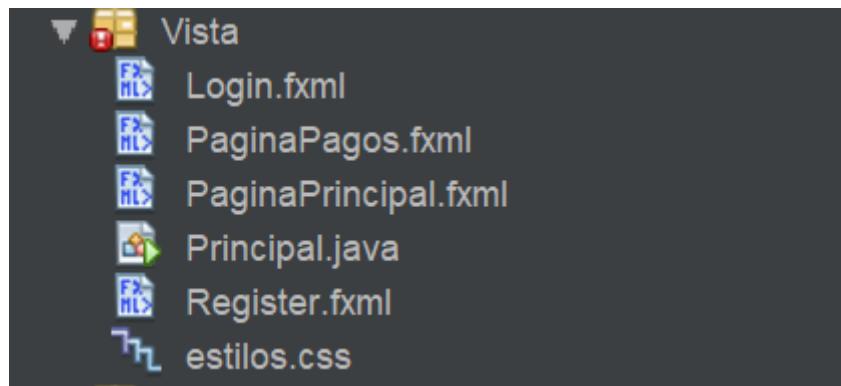
// Método para recibir el callback del historial
public void setCallbackAregarHistorial(Runnable callback) {
    this.callbackAregarHistorial = callback;
}
}

```

Como ya hemos terminado la parte del empaquetado de Controlador, finalizamos agregando los archivos FXML en el empaquetado Vista, aquí debemos añadir 1 archivo MainClass, el cual se

encarga de mostrar o ejecutar el programa inicialmente, 4 archivos FXML y 1 archivo de tipo css donde se aplican cambios de propiedades de las clases en los FXML.

Nos debe quedar algo asi:



iniciaremos codificando el archivo Principal, este archivo es el que se ejecuta cuando se corre el programa y es el que muestra el archivo fxml que decidamos, en este caso, el Login inicialmente:

```
package Vista;

import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.image.Image;
import javafx.scene.input.MouseEvent;
import javafx.stage.Stage;
import javafx.stage.StageStyle;

/**
 *
 * @author santi
 */
public class Principal extends Application {

    @Override
    public void start(Stage primaryStage){
        Parent root = null;

        //Mostramos el archivo Login.FXML
        try {
            root = FXMLLoader.load(getClass().getResource("Login.fxml"));
        } catch (IOException ex) {
            Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
        }

        //le aplicamos atributos
        Scene escena = new Scene(root);
        primaryStage.setScene(escena);
    }
}
```

```
primaryStage.setTitle("SantiShop - Inicio de sesión");
primaryStage.getIcons().add(new
Image(getClass().getResourceAsStream("/Vista/Imagenes/LOGO FONDO BLANCO
SANTISHOP.png")));
primaryStage.setResizable(false);
primaryStage.show();
}
public static void main(String[] args) {
    launch(args);
}
}
```

Una vez tengamos el código agregado, tenemos dos maneras de editar/hacer la parte visual de nuestro aplicativo.

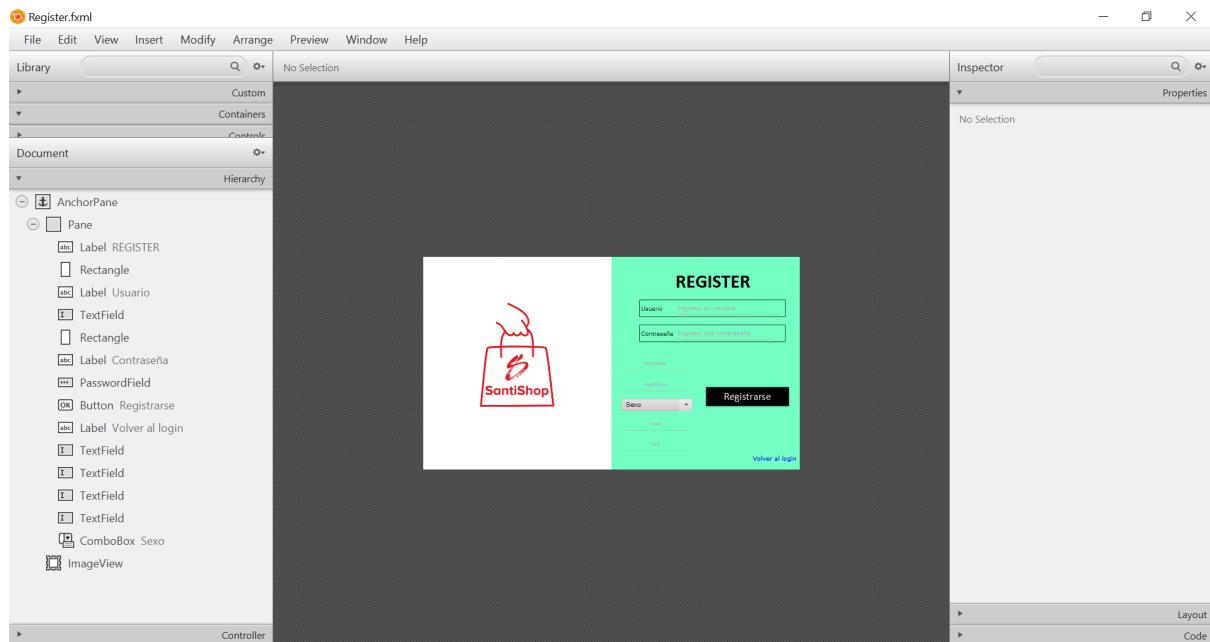
1. SceneBuilder
 2. Código

Aquí les mostraré ambas maneras y ya depende de tu elección cual usar.

Iniciaremos con el archivo Register, este archivo podemos modificarlo:

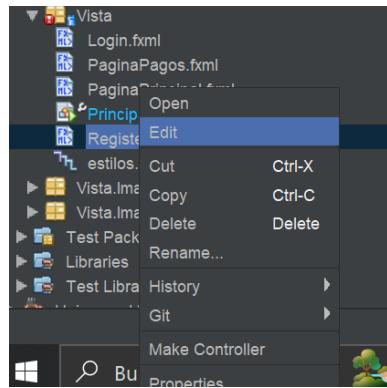
1. SCENE BUILDER

Daremos doble click sobre el archivo Register.fxml y nos abrirá la ventana de scenebuilder, una vez aquí debemos organizar todos los elementos manualmente.



2. CÓDIGO

Daremos click derecho sobre el archivo Register.fxml y seleccionaremos la opción “edite” una vez aquí podremos generar todo mediante código.



Una vez nos abra el editor, agregaremos lo siguiente:

```

<?import java.lang.String?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.shape.Rectangle?>
<?import javafx.scene.text.Font?>

<AnchorPane id="AnchorPane" prefHeight="450.0" prefWidth="800.0" style="-fx-background-color: white;" stylesheets="@estilos.css" xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Controlador.RegisterControlador">
    <children>
        <Pane layoutX="400.0" prefHeight="452.0" prefWidth="400.0" style="-fx-background-color: #76FFC6;">
            <children>
                <Label layoutX="136.0" layoutY="29.0" prefHeight="46.0" prefWidth="163.0" style="-fx-text-fill: black;" text="REGISTER" textAlignment="CENTER">
                    <font>
                        <Font name="Calibri Bold" size="40.0" />
                    </font>
                </Label>
                <Rectangle arcHeight="5.0" arcWidth="5.0" fill="#fffff00" height="37.0" layoutX="59.0" layoutY="91.0" stroke="BLACK" strokeType="INSIDE" width="311.0" />
                <Label layoutX="63.0" layoutY="101.0" text="Usuario" />
                <TextField fx:id="txtUsuario" layoutX="136.0" layoutY="96.0" prefHeight="25.0" prefWidth="225.0" promptText="Ingresa un usuario" styleClass="tipoLetra" />
                <Rectangle arcHeight="5.0" arcWidth="5.0" fill="#fffff00" height="37.0" layoutX="59.0" layoutY="144.0" stroke="BLACK" strokeType="INSIDE" width="311.0" />
            </children>
        </Pane>
    </children>
</AnchorPane>
```

```

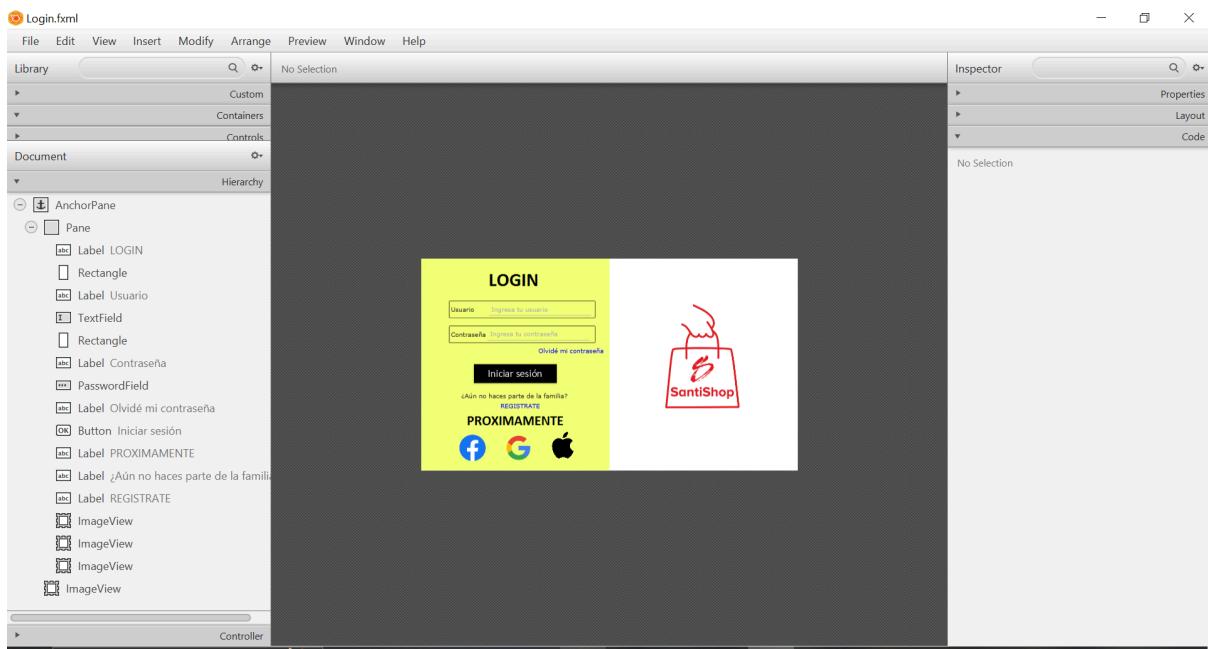
<Label layoutX="63.0" layoutY="154.0" text="Contraseña" />
<PasswordField fx:id="txtContrasena" layoutX="136.0" layoutY="149.0" prefHeight="26.0"
prefWidth="206.0" promptText="Ingresa una contraseña" styleClass="tipoLetra" />
<Button layoutX="201.0" layoutY="277.0" mnemonicParsing="false"
onAction="#registrarUsuario" prefHeight="39.0" prefWidth="176.0"
styleClass="botonRegistrarIniciar" text="Registrarse" textFill="WHITE">
<font>
<Font name="Calibri" size="20.0" />
</font>
</Button>
<Label layoutX="300.0" layoutY="420.0" onMouseClicked="#volverLogin" text="Volver al
login">
<styleClass>
<String fx:value="vinculos" />
<String fx:value="tipoLetra" />
</styleClass></Label>
<TextField fx:id="txtNombres" alignment="CENTER" layoutX="3.0" layoutY="208.0"
prefHeight="35.0" prefWidth="180.0" promptText="Nombres" scaleX="0.8" scaleY="0.8"
scaleZ="0.8" styleClass="tipoLetra" />
<TextField fx:id="txtApellidos" alignment="CENTER" layoutX="3.0" layoutY="253.0"
prefHeight="35.0" prefWidth="180.0" promptText="Apellidos" scaleX="0.8" scaleY="0.8"
scaleZ="0.8" styleClass="tipoLetra" />
<TextField fx:id="txtEdad" alignment="CENTER" layoutX="3.0" layoutY="337.0"
prefHeight="35.0" prefWidth="180.0" promptText="Edad" scaleX="0.8" scaleY="0.8" scaleZ="0.8"
styleClass="tipoLetra" />
<TextField fx:id="txtPais" alignment="CENTER" layoutX="3.0" layoutY="380.0"
prefHeight="35.0" prefWidth="180.0" promptText="País" scaleX="0.8" scaleY="0.8" scaleZ="0.8"
styleClass="tipoLetra" />
<ComboBox fx:id="comboSexo" layoutX="22.0" layoutY="302.0" prefWidth="150.0"
promptText="Sexo" styleClass="tipoLetra" />
</children>
</Pane>
<ImageView fitHeight="301.0" fitWidth="331.0" layoutX="36.0" layoutY="86.0"
pickOnBounds="true" preserveRatio="true">
<image>
<Image url="@Imagenes/LOGO%20FONDO%20BLANCO%20SANTISHOP.png" />
</image>
</ImageView>
</children>
</AnchorPane>

```

Sin importar qué método usamos, nos dará el mismo resultado.

Ahora, organizaremos el archivo Login.fxml, usando cualquiera de los dos métodos.

1. SCENE BUILDER



2. CÓDIGO

```

<?import java.lang.String?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.shape.Rectangle?>
<?import javafx.scene.text.Font?>

<AnchorPane id="AnchorPane" prefHeight="450.0" prefWidth="800.0" style="-fx-background-color: white;" stylesheets="@estilos.css" xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="Controlador.LoginControlador">
    <children>
        <Pane layoutY="-1.0" prefHeight="452.0" prefWidth="400.0" style="-fx-background-color: F4FF76;">
            <children>
                <Label layoutX="143.0" layoutY="24.0" prefHeight="40.0" prefWidth="115.0" style="-fx-text-fill: black;" text="LOGIN" textAlignment="CENTER">
                    <font>
                        <Font name="Calibri Bold" size="40.0" />
                    </font>
                </Label>
                <Rectangle arcHeight="5.0" arcWidth="5.0" fill="#fffff00" height="37.0" layoutX="59.0" layoutY="91.0" stroke="BLACK" strokeType="INSIDE" width="311.0" />
                <Label layoutX="63.0" layoutY="101.0" styleClass="tipoLetra" text="Usuario" />
            </children>
        </Pane>
    </children>
</AnchorPane>

```

```

<TextField fx:id="txtUsuario" layoutX="144.0" layoutY="96.0" prefHeight="25.0"
prefWidth="217.0" promptText="Ingresa tu usuario" styleClass="tipoLetra"
styleSheets="@estilos.css" />
    <Rectangle arcHeight="5.0" arcWidth="5.0" fill="#fffff00" height="37.0" layoutX="59.0"
layoutY="144.0" stroke="BLACK" strokeType="INSIDE" width="311.0" />
        <Label layoutX="63.0" layoutY="154.0" prefHeight="16.0" prefWidth="77.0"
styleClass="tipoLetra" text="Contraseña" />
            <PasswordField fx:id="txtContrasena" layoutX="141.0" layoutY="149.0" prefHeight="25.0"
prefWidth="217.0" promptText="Ingresa tu contraseña" styleClass="tipoLetra" />
                <Label fx:id="lblOlvidarContrasena" layoutX="249.0" layoutY="189.0"
onMouseClicked="#olvidarContrasena" text="Olvidé mi contraseña">
                    <styleClass>
                        <String fx:value="tipoLetra" />
                        <String fx:value="vinculos" />
                    </styleClass></Label>
                    <Button layoutX="112.0" layoutY="226.0" mnemonicParsing="false"
onAction="#iniciarSesion" prefHeight="39.0" prefWidth="176.0" styleClass="botonRegistrarIniciar"
text="Iniciar sesión" textFill="WHITE">
                        <font>
                            <Font name="Calibri" size="20.0" />
                        </font>
                    </Button>
                    <Label layoutX="97.0" layoutY="323.0" prefHeight="46.0" prefWidth="206.0"
style="-fx-text-fill: black;" text="PROXIMAMENTE" textAlignment="CENTER">
                        <font>
                            <Font name="Calibri Bold" size="29.0" />
                        </font>
                    </Label>
                    <Label layoutX="85.0" layoutY="285.0" styleClass="tipoLetra" text="¿Aún no haces parte de
la familia? " />
                    <Label layoutX="168.0" layoutY="307.0" onMouseClicked="#openRegister"
prefHeight="16.0" prefWidth="93.0" text="REGISTRATE" textAlignment="JUSTIFY">
                        <styleClass>
                            <String fx:value="tipoLetra" />
                            <String fx:value="vinculos" />
                        </styleClass></Label>
                        <ImageView fitHeight="56.0" fitWidth="60.0" layoutX="81.0" layoutY="376.0"
pickOnBounds="true" preserveRatio="true">
                            <image>
                                <Image url="@Imagenes/facebook%20logo.png" />
                            </image>
                        </ImageView>
                        <ImageView fitHeight="56.0" fitWidth="60.0" layoutX="179.0" layoutY="376.0"
pickOnBounds="true" preserveRatio="true">
                            <image>
                                <Image url="@Imagenes/google%20logo.png" />
                            </image>
                        </ImageView>
                        <ImageView fitHeight="56.0" fitWidth="60.0" layoutX="278.0" layoutY="370.0"
pickOnBounds="true" preserveRatio="true">
                            <image>
                                <Image url="@Imagenes/apple%20logo.png" />
                            </image>
                        </ImageView>
                    </children>

```

```

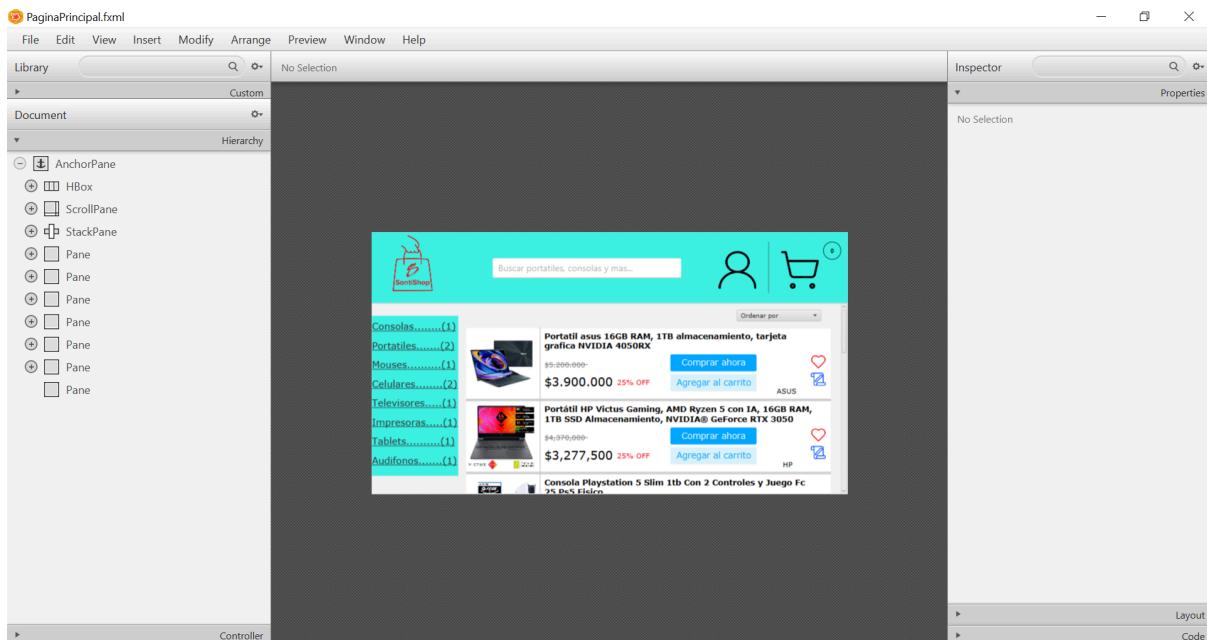
</Pane>
<ImageView fitHeight="301.0" fitWidth="331.0" layoutX="434.0" layoutY="86.0"
pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@Imagenes/LOGO%20FONDO%20BLANCO%20SANTISHOP.png" />
    </image>
</ImageView>
</children>
</AnchorPane>

```

Luego de editar estos archivos, continuaremos con el más importante: PaginaPrincipal.fxml, este archivo es donde se concentra toda nuestra tienda.

En este caso, recomiendo personalmente usar el método de código, puesto que hacerlo manualmente en scenebuilder es más tedioso de organizar, sin embargo, dejo ambos métodos.

1. SCENE BUILDER



NOTA: Se hizo uso de muchos elementos, al ser tantos, en la parte izquierda no se alcanzan a ver todos.

2. CÓDIGO

```

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.ScrollPane?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.image.Image?>
<?import javafx.scene.image.ImageView?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.Pane?>

```

```

<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.shape.Circle?>
<?import javafx.scene.shape.Line?>
<?import javafx.scene.text.Font?>

<AnchorPane id="AnchorPane" prefHeight="555.0" prefWidth="1012.0"
style="-fx-background-color: #E7E7E7;" stylesheets="@estilos.css"
xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="Controlador.PaginaPrincipalControlador">
    <children>
        <HBox prefHeight="150.0" prefWidth="1012.0" style="-fx-background-color: #3df0e5;">
            <children>
                <ImageView fitHeight="150.0" fitWidth="200.0" pickOnBounds="true"
preserveRatio="true">
                    <image>
                        <Image url="@Imagenes/LOGO%20SIN%20FONDO%20SANTISHOP.png" />
                    </image>
                </ImageView>
                <Pane maxHeight="-Infinity" prefHeight="42.0" prefWidth="400.0"
style="-fx-background-color: D9D9D9;">
                    <children>
                        <TextField focusTraversable="false" prefHeight="40.0" prefWidth="400.0"
promptText="Buscar portatiles, consolas y mas..." style="-fx-background-color: white;">
                            <font>
                                <Font size="19.0" />
                            </font>
                        </TextField>
                    </children>
                    <HBox.margin>
                        <Insets left="80.0" top="55.0" />
                    </HBox.margin>
                    <opaqueInsets>
                        <Insets />
                    </opaqueInsets>
                </Pane>
                <ImageView fitHeight="80.0" fitWidth="100.0" onMouseClicked="#mostrarOpciones"
pickOnBounds="true" preserveRatio="true" styleClass="manoImagen">
                    <image>
                        <Image url="@Imagenes/Icono%20usuario.png" />
                    </image>
                    <HBox.margin>
                        <Insets left="80.0" top="40.0" />
                    </HBox.margin>
                </ImageView>
                <Line endX="-97.85885620117188" endY="40.93291473388672"
startX="-97.85879516601562" startY="-66.8044204711914" stroke="#515151">
                    <HBox.margin>
                        <Insets left="25.0" top="20.0" />
                    </HBox.margin>
                </Line>
                <ImageView fitHeight="80.0" fitWidth="92.0" onMouseClicked="#mostrarCarrito"
pickOnBounds="true" preserveRatio="true" styleClass="manoImagen">
                    <image>
                        <Image url="@Imagenes/carro%20compras.png" />

```

```

</image>
<HBox.margin>
  <Insets left="25.0" top="40.0" />
</HBox.margin>
</Image>
</children>
</HBox>
<ScrollPane hbarPolicy="NEVER" layoutY="151.0" prefHeight="405.0" prefWidth="1012.0"
vbarPolicy="ALWAYS">
  <content>
    <AnchorPane minHeight="0.0" minWidth="0.0" onMouseClicked="#volverOpciones"
prefWidth="1000.0">
      <children>
        <VBox layoutY="25.0" prefHeight="340.0" prefWidth="183.0"
style="-fx-background-color: #3df0e5;">
          <children>
            <Label prefHeight="24.0" prefWidth="183.0"
styleClass="tipoLetraVinculosPagPrincipal" text="Consolas.....(1)" />
            <Label prefHeight="24.0" prefWidth="183.0"
styleClass="tipoLetraVinculosPagPrincipal" text="Portatiles.....(2)">
              <VBox.margin>
                <Insets top="15.0" />
              </VBox.margin>
            </Label>
            <Label prefHeight="24.0" prefWidth="183.0"
styleClass="tipoLetraVinculosPagPrincipal" text="Mouses.....(1)">
              <VBox.margin>
                <Insets top="15.0" />
              </VBox.margin>
            </Label>
            <Label prefHeight="24.0" prefWidth="183.0"
styleClass="tipoLetraVinculosPagPrincipal" text="Celulares.....(2)">
              <VBox.margin>
                <Insets top="15.0" />
              </VBox.margin>
            </Label>
            <Label prefHeight="24.0" prefWidth="183.0"
styleClass="tipoLetraVinculosPagPrincipal" text="Televisores.....(1)">
              <VBox.margin>
                <Insets top="15.0" />
              </VBox.margin>
            </Label>
            <Label prefHeight="24.0" prefWidth="183.0"
styleClass="tipoLetraVinculosPagPrincipal" text="Impresoras.....(1)">
              <VBox.margin>
                <Insets top="15.0" />
              </VBox.margin>
            </Label>
            <Label prefHeight="24.0" prefWidth="183.0"
styleClass="tipoLetraVinculosPagPrincipal" text="Tablets.....(1)">
              <VBox.margin>
                <Insets top="15.0" />
              </VBox.margin>
            </Label>

```

```

<Label prefHeight="24.0" prefWidth="183.0"
styleClass="tipoLetraVinculosPagPrincipal" text="Audifonos.....(1)">
    <VBox.margin>
        <Insets top="15.0" />
    </VBox.margin>
</Label>
</children>
<padding>
    <Insets top="10.0" />
</padding>
</VBox>
<ComboBox fx:id="comboFiltro" layoutX="775.0" layoutY="12.0" prefHeight="25.0"
prefWidth="179.0" promptText="Ordenar por" styleClass="tipoLetra" />
<VBox layoutX="200.0" layoutY="51.0" prefHeight="1551.0" prefWidth="775.0"
style="-fx-background-color: #d1d1d1;">
    <children>
        <Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
            <children>
                <ImageView fx:id="imgProducto1" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
                    <image>
                        <Image url="@Imagenes/Productos/Producto1.png" />
                    </image>
                </ImageView>
                <Label fx:id="lblNombre1" layoutX="166.0" prefHeight="53.0"
prefWidth="569.0" styleClass="LetraNombreProducto" text="Portatil asus 16GB RAM, 1TB
almacenamiento, tarjeta grafica NVIDIA 4050RX" wrapText="true" />
                <Label fx:id="lblPrecioOriginal1" layoutX="166.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="$5.200.000" />
                    <Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
                    <Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="197.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />
                <Label fx:id="lblPrecioDescuento1" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="$3.900.000" />
                <Label fx:id="lblPorcentaje1" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="25% OFF" />
                <Label fx:id="lblMarca1" layoutX="660.0" layoutY="125.0"
styleClass="LetraMarca" text="ASUS" />
                <Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto1" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
                    <Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto1" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
                    <ImageView fx:id="imgDeseos1" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
                        <image>
                            <Image url="@Imagenes/Listadeseos.png" />
                        </image>
                    </ImageView>
    </children>

```

```

<ImageView fx:id="imgFavorito1" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
    <image>
        <Image url="@Imagenes/favoritos.png" />
    </image>
</ImageView>
</children>
</Pane>
<Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
    <children>
        <ImageView fx:id="imgProducto2" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@Imagenes/Productos/Producto2.jpg" />
            </image>
</ImageView>
<Label fx:id="lblNombre2" layoutX="166.0" prefHeight="53.0"
prefWidth="571.0" styleClass="LetraNombreProducto" text="Portátil HP Victus Gaming, AMD
Ryzen 5 con IA, 16GB RAM, 1TB SSD Almacenamiento, NVIDIA® GeForce RTX 3050 "
wrapText="true" />
<Label fx:id="lblPrecioOriginal2" layoutX="166.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="$4,370,000" />
<Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
<Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="197.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />
<Label fx:id="lblPrecioDescuento2" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="$3,277,500" />
<Label fx:id="lblPorcentaje2" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="25% OFF" />
<Label fx:id="lblMarca2" layoutX="673.0" layoutY="125.0"
styleClass="LetraMarca" text="HP" />
<Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto2" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
<Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto2" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
<ImageView fx:id="imgDeseos2" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
    <image>
        <Image url="@Imagenes/Listadeseos.png" />
    </image>
</ImageView>
<ImageView fx:id="imgFavorito2" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
    <image>
        <Image url="@Imagenes/favoritos.png" />
    </image>
</ImageView>
</children>

```

```

<VBox.margin>
    <Insets top="5.0" />
</VBox.margin>
</Pane>
<Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
    <children>
        <ImageView fx:id="imgProducto3" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@Imagenes/Productos/Producto3.jpg" />
            </image>
        </ImageView>
        <Label fx:id="lblNombre3" layoutX="166.0" prefHeight="53.0"
prefWidth="571.0" styleClass="LetraNombreProducto" text="Consola Playstation 5 Slim 1tb Con 2
Controles y Juego Fc 25 Ps5 Fisico" wrapText="true" />
        <Label fx:id="lblPrecioOriginal3" layoutX="166.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="$4.360.000" />
        <Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
        <Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="197.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />
        <Label fx:id="lblPrecioDescuento3" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="$3.488.000" />
        <Label fx:id="lblPorcentaje3" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="20% OFF" />
        <Label fx:id="lblMarca3" layoutX="660.0" layoutY="125.0"
styleClass="LetraMarca" text="SONY" />
        <Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto3" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
        <Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto3" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
        <ImageView fx:id="imgDeseos3" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
            <image>
                <Image url="@Imagenes/Listadeseos.png" />
            </image>
        </ImageView>
        <ImageView fx:id="imgFavorito3" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
            <image>
                <Image url="@Imagenes/favoritos.png" />
            </image>
        </ImageView>
    </children>
<VBox.margin>
    <Insets top="5.0" />
</VBox.margin>
</Pane>
<Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
    <children>

```

```

<ImageView fx:id="imgProducto4" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@Imagenes/Productos/Producto4.jpg" />
    </image>
</ImageView>
<Label fx:id="lblNombre4" layoutX="166.0" prefHeight="53.0"
prefWidth="571.0" styleClass="LetraNombreProducto" text="Mouse Gamer T-Dagger Imperial
RGB" wrapText="true" />
    <Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
        <Label fx:id="lblPrecioDescuento4" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="$100.000" />
            <Label fx:id="lblMarca4" layoutX="651.0" layoutY="125.0"
styleClass="LetraMarca" text="IMPERIAL" />
                <Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto4" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
                    <Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto4" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
                        <ImageView fx:id="imgDeseos4" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
                            <image>
                                <Image url="@Imagenes>Listadeseos.png" />
                            </image>
</ImageView>
<ImageView fx:id="imgFavorito4" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
                            <image>
                                <Image url="@Imagenes/favoritos.png" />
                            </image>
</ImageView>
<Label fx:id="lblPrecioOriginal4" layoutX="177.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="$100.000" />
    <Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="200.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />
        <Label fx:id="lblPorcentaje4" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="0%" />
    </children>
    <VBox.margin>
        <Insets top="5.0" />
    </VBox.margin>
</Pane>
<Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
    <children>
        <ImageView fx:id="imgProducto5" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@Imagenes/Productos/Producto5.jpg" />
            </image>
</ImageView>

```

```

        <Label fx:id="lblNombre5" layoutX="166.0" prefHeight="53.0"
prefWidth="571.0" styleClass="LetraNombreProducto" text="iPhone 16 Pro-Max 1TB, color
Desierto" wrapText="true" />
        <Label fx:id="lblPrecioOriginal5" layoutX="166.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="$9,240,000" />
        <Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
        <Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="197.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />
        <Label fx:id="lblPrecioDescuento5" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="\$5,544,000" />
        <Label fx:id="lblPorcentaje5" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="40% OFF" />
        <Label fx:id="lblMarca5" layoutX="660.0" layoutY="125.0"
styleClass="LetraMarca" text="APPLE" />
        <Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto5" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
        <Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto5" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
        <Image fx:id="imgDeseos5" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
        <image>
            <Image url="@Imagenes>Listadeseos.png" />
        </image>
    </Image>
        <Image fx:id="imgFavorito5" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
        <image>
            <Image url="@Imagenes/favoritos.png" />
        </image>
    </Image>
</children>
<VBox.margin>
    <Insets top="5.0" />
</VBox.margin>
</Pane>
<Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
    <children>
        <Image fx:id="imgProducto6" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@Imagenes/Productos/Producto6.png" />
            </image>
        </Image>
        <Label fx:id="lblNombre6" layoutX="166.0" prefHeight="53.0"
prefWidth="571.0" styleClass="LetraNombreProducto" text="Samsung Galaxy S25 ultra 256 GB,
color Gris titani" wrapText="true" />
        <Label fx:id="lblPrecioOriginal6" layoutX="166.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="\$7,360,000" />
    </children>
</Pane>

```

```

        <Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
        <Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="197.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />
        <Label fx:id="lblPrecioDescuento6" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="\$5,152,000 " />
        <Label fx:id="lblPorcentaje6" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="30% OFF" />
        <Label fx:id="lblMarca6" layoutX="648.0" layoutY="125.0"
styleClass="LetraMarca" text="SAMSUNG" />
        <Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto6" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
        <Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto6" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
        <ImageView fx:id="imgDeseos6" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
        <image>
            <Image url="@Imagenes>Listadeseos.png" />
        </image>
    </ImageView>
        <ImageView fx:id="imgFavorito6" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
        <image>
            <Image url="@Imagenes/favoritos.png" />
        </image>
    </ImageView>
</children>
<VBox.margin>
    <Insets top="5.0" />
</VBox.margin>
</Pane>
<Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
<children>
    <ImageView fx:id="imgProducto7" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
        <image>
            <Image url="@Imagenes/Productos/Producto7.jpg" />
        </image>
    </ImageView>
    <Label fx:id="lblNombre7" layoutX="166.0" prefHeight="53.0"
prefWidth="571.0" styleClass="LetraNombreProducto" text="Televisor SAMSUNG 75 pulgadas
LED Uhd4K Smart TV" wrapText="true" />
    <Label fx:id="lblPrecioOriginal7" layoutX="166.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="\$8,470,000" />
        <Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
        <Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="197.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />

```

```

        <Label fx:id="lblPrecioDescuento7" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="\$7,623,000 " />
        <Label fx:id="lblPorcentaje7" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="10% OFF" />
        <Label fx:id="lblMarca7" layoutX="648.0" layoutY="125.0"
styleClass="LetraMarca" text="SAMSUNG" />
        <Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto7" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
        <Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto7" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
        <ImageView fx:id="imgDeseos7" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
            <image>
                <Image url="@Imagenes>Listadeseos.png" />
            </image>
        </ImageView>
        <ImageView fx:id="imgFavorito7" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
            <image>
                <Image url="@Imagenes/favoritos.png" />
            </image>
        </ImageView>
    </children>
    <VBox.margin>
        <Insets top="5.0" />
    </VBox.margin>
</Pane>
<Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
    <children>
        <ImageView fx:id="imgProducto8" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@Imagenes/Productos/Producto8.png" />
            </image>
        </ImageView>
        <Label fx:id="lblNombre8" layoutX="166.0" prefHeight="53.0"
prefWidth="571.0" styleClass="LetraNombreProducto" text="Impresora Multifuncional Epson L5590
Wifi Ecotank" wrapText="true" />
        <Label fx:id="lblPrecioOriginal8" layoutX="166.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="\$5,230,000 " />
        <Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
        <Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="197.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />
        <Label fx:id="lblPrecioDescuento8" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="\$4,184,000 " />
        <Label fx:id="lblPorcentaje8" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="20% OFF" />
        <Label fx:id="lblMarca8" layoutX="660.0" layoutY="125.0"
styleClass="LetraMarca" text="EPSON" />
    </children>
</Pane>

```

```

<Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto8" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
    <Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto8" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
        <ImageView fx:id="imgDeseos8" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
            <image>
                <Image url="@Imagenes/Listadeseos.png" />
            </image>
        </ImageView>
        <ImageView fx:id="imgFavorito8" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
            <image>
                <Image url="@Imagenes/favoritos.png" />
            </image>
        </ImageView>
    </children>
<VBox.margin>
    <Insets top="5.0" />
</VBox.margin>
</Pane>
<Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
    <children>
        <ImageView fx:id="imgProducto9" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@Imagenes/Productos/Producto9.jpg" />
            </image>
        </ImageView>
        <Label fx:id="lblNombre9" layoutX="166.0" prefHeight="53.0"
prefWidth="571.0" styleClass="LetraNombreProducto" text="Apple iPad (10th Generation): with
A14 Bionic chip, 64GB, 4 RAM" wrapText="true" />
            <Label fx:id="lblPrecioOriginal9" layoutX="166.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="$6,580,000" />
            <Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
            <Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="197.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />
            <Label fx:id="lblPrecioDescuento9" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="$3,290,000 " />
            <Label fx:id="lblPorcentaje9" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="50% OFF" />
            <Label fx:id="lblMarca9" layoutX="660.0" layoutY="125.0"
styleClass="LetraMarca" text="APPLE" />
            <Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto9" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
            <Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto9" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
    </children>

```

```

        <ImageView fx:id="imgDeseos9" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
<image>
<Image url="@Imagenes/Listadeseos.png" />
</image>
</ImageView>
<ImageView fx:id="imgFavorito9" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
<image>
<Image url="@Imagenes/favoritos.png" />
</image>
</ImageView>
</children>
<VBox.margin>
<Insets top="5.0" />
</VBox.margin>
</Pane>
<Pane prefHeight="150.0" prefWidth="775.0" style="-fx-background-color: white;">
<children>
<ImageView fx:id="imgProducto10" fitHeight="150.0" fitWidth="200.0"
pickOnBounds="true" preserveRatio="true">
<image>
<Image url="@Imagenes/Productos/Producto10.jpg" />
</image>
</ImageView>
<Label fx:id="lblNombre10" layoutX="166.0" prefHeight="53.0"
prefWidth="571.0" styleClass="LetraNombreProducto" text="AirPods Pro 2" wrapText="true" />
<Label fx:id="lblPrecioOriginal10" layoutX="166.0" layoutY="68.0"
styleClass="LetraPrecioOriginal" text="$4,370,000" />
<Line endX="-96.77783203125" endY="53.00001525878906" layoutX="254.0"
layoutY="97.0" startX="-96.77780151367188" startY="-97.99996948242188" stroke="#acacac" />
<Line endX="61.202178955078125" endY="-0.4999847412109375"
layoutX="197.0" layoutY="78.0" startX="-25.999969482421875" startY="-0.4999847412109375"
stroke="#575757" />
<Label fx:id="lblPrecioDescuento10" layoutX="166.0" layoutY="99.0"
styleClass="LetraPrecioDescuento" text="$3,277,500 " />
<Label fx:id="lblPorcentaje10" layoutX="321.0" layoutY="106.0"
styleClass="LetraPorcentaje" text="25% OFF" />
<Label fx:id="lblMarca10" layoutX="660.0" layoutY="125.0"
styleClass="LetraMarca" text="APPLE" />
<Button layoutX="434.0" layoutY="56.0" mnemonicParsing="false"
onAction="#comprarProducto10" prefHeight="35.0" prefWidth="183.0" styleClass="botonComprar"
text="Comprar ahora" />
<Button layoutX="434.0" layoutY="98.0" mnemonicParsing="false"
onAction="#agregarAlCarritoProducto10" prefHeight="35.0" prefWidth="183.0"
styleClass="botonAddCarrito" text="Aregar al carrito" />
<ImageView fx:id="imgDeseos10" fitHeight="28.0" fitWidth="33.0"
layoutX="732.0" layoutY="95.0" onMouseClicked="#imagenListaDeseos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
<image>
<Image url="@Imagenes/Listadeseos.png" />
</image>
</ImageView>

```

```

        <ImageView fx:id="imgFavorito10" fitHeight="28.0" fitWidth="33.0"
layoutX="733.0" layoutY="58.0" onMouseClicked="#imagenFavoritos" pickOnBounds="true"
preserveRatio="true" styleClass="manoImagen">
<image>
<Image url="@Imagenes/favoritos.png" />
</image>
</ImageView>
</children>
<VBox.margin>
<Insets top="5.0" />
</VBox.margin>
</Pane>
</children>
</VBox>
</children>
</AnchorPane>
</content>
</ScrollPane>
<StackPane layoutX="960.0" layoutY="20.0">
<Circle fx:id="cantidadCarrito" fill="#3df0e5" radius="19.0" stroke="black"
strokeType="INSIDE" />
<Label fx:id="lblCantidadCarrito" style="-fx-font-size: 12px; -fx-font-weight: bold;" text="0"
textFill="black" />
</StackPane>

<Pane fx:id="panelCarrito" layoutX="600.0" prefHeight="554.0" prefWidth="411.0"
style="-fx-background-color: white;" visible="false">
<children>
<Line endY="101.3332748413086" layoutY="38.0" startX="412.00006103515625"
 startY="102.0" />
<Label layoutX="156.0" layoutY="14.0" prefHeight="102.0" prefWidth="214.0"
styleClass="LetraTitulos" text="Carrito de compras" wrapText="true" />
<Circle fill="WHITE" layoutX="73.0" layoutY="69.0" radius="54.0" stroke="BLACK"
strokeType="INSIDE" />
<ImageView fitHeight="73.0" fitWidth="80.0" layoutX="33.0" layoutY="35.0"
onMouseClicked="#mostrarCarrito" pickOnBounds="true" preserveRatio="true">
<image>
<Image url="@Imagenes/carro%20compras.png" />
</image>
</ImageView>
<ScrollPane fitToWidth="true" hbarPolicy="NEVER" layoutY="140.0" prefHeight="250.0"
prefWidth="412.0">
<content>
<AnchorPane minHeight="0.0" minWidth="0.0">
<children>
<VBox fx:id="contenedorCarrito" layoutX="-1.0" layoutY="-8.0" prefHeight="100.0"
prefWidth="397.0" AnchorPane.leftAnchor="-0.0" AnchorPane.rightAnchor="0.0"
AnchorPane.topAnchor="0.0" />
</children>
</AnchorPane>
</content>
</ScrollPane>
<Label layoutX="8.0" layoutY="406.0" styleClass="LetraNombreProducto" text="Subtotal:"/>
</>

```

```

<Label layoutX="8.0" layoutY="442.0" styleClass="LetraNombreProducto"
text="Descuento:" />
    <Label layoutX="8.0" layoutY="477.0" styleClass="LetraNombreProducto" text="Total:" />
        <Button fx:id="btnComprarTodo" layoutX="164.0" layoutY="511.0"
mnemonicParsing="false" onAction="#comprarCarritoCompleto" prefHeight="35.0"
prefWidth="133.0" styleClass="botonComprar" text="Comprar" />
            <ImageView fitHeight="43.0" fitWidth="41.0" layoutX="337.0" layoutY="505.0"
onMouseClicked="#vaciarCarrito" pickOnBounds="true" preserveRatio="true"
styleClass="manoImagen">
                <image>
                    <Image url="@Imagenes/borrargrupo.png" />
                </image>
            </ImageView>
        <Label fx:id="lblSubtotal" layoutX="120.0" layoutY="408.0"
styleClass="LetraPrecioOriginal" />
        <Label fx:id="lblDescuento" layoutX="134.0" layoutY="444.0" styleClass="LetraPorcentaje"
/>
        <Label fx:id="lblTotal" layoutX="79.0" layoutY="472.0" styleClass="LetraPrecioDescuento"
/>
    </children>
</Pane>
<Pane fx:id="panelFavoritos" layoutX="600.0" prefHeight="554.0" prefWidth="411.0"
style="-fx-background-color: white;" visible="false">
    <children>
        <Line endY="101.3332748413086" layoutY="38.0" startX="412.00006103515625"
 startY="102.0" />
        <Label layoutX="156.0" layoutY="14.0" prefHeight="102.0" prefWidth="214.0"
styleClass="LetraTitulos" text="Favoritos" wrapText="true" />
        <Circle fill="WHITE" layoutX="73.0" layoutY="69.0" radius="54.0" stroke="BLACK"
strokeType="INSIDE" />
        <ImageView fitHeight="73.0" fitWidth="80.0" layoutX="33.0" layoutY="35.0"
onMouseClicked="#mostrarCarrito" pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@Imagenes/favoritoAgregado.png" />
            </image>
        </ImageView>
        <ScrollPane fitToWidth="true" hbarPolicy="NEVER" layoutY="140.0" prefHeight="324.0"
prefWidth="412.0">
            <content>
                <AnchorPane minHeight="0.0" minWidth="0.0">
                    <children>
                        <VBox fx:id="contenedorFavoritos" layoutX="-1.0" layoutY="-8.0"
prefHeight="100.0" prefWidth="397.0" AnchorPane.leftAnchor="-0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0" />
                    </children>
                </AnchorPane>
            </content>
        </ScrollPane>
        <ImageView fitHeight="54.0" fitWidth="55.0" layoutX="185.0" layoutY="476.0"
onMouseClicked="#vaciarFavoritos" pickOnBounds="true" preserveRatio="true"
styleClass="manoImagen">
            <image>
                <Image url="@Imagenes/borrargrupo.png" />
            </image>
        </ImageView>
    </children>

```

```

<Line endY="101.3332748413086" layoutY="362.0" startX="412.00006103515625"
startY="102.0" />
</children>
</Pane>
<Pane fx:id="panelListaDeseos" layoutX="600.0" prefHeight="554.0" prefWidth="411.0"
style="-fx-background-color: white;" visible="false">
<children>
<Line endY="101.3332748413086" layoutY="38.0" startX="412.00006103515625"
startY="102.0" />
<Label layoutX="156.0" layoutY="14.0" prefHeight="102.0" prefWidth="214.0"
styleClass="LetraTitulos" text="Lista de deseos" wrapText="true" />
<Circle fill="WHITE" layoutX="73.0" layoutY="69.0" radius="54.0" stroke="BLACK"
strokeType="INSIDE" />
<ImageView fitHeight="73.0" fitWidth="80.0" layoutX="33.0" layoutY="35.0"
onMouseClicked="#mostrarCarrito" pickOnBounds="true" preserveRatio="true">
<image>
<Image url="@Imagenes>ListadeseosAgregado.png" />
</image>
</ImageView>
<ScrollPane fitToWidth="true" hbarPolicy="NEVER" layoutY="140.0" prefHeight="324.0"
prefWidth="412.0">
<content>
<AnchorPane minHeight="0.0" minWidth="0.0">
<children>
<VBox fx:id="contenedorListaDeseos" layoutX="-1.0" layoutY="-8.0"
prefHeight="100.0" prefWidth="397.0" AnchorPane.leftAnchor="-0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0" />
</children>
</AnchorPane>
</content>
</ScrollPane>
<ImageView fitHeight="54.0" fitWidth="55.0" layoutX="293.0" layoutY="488.0"
onMouseClicked="#vaciarListaDeseos" pickOnBounds="true" preserveRatio="true"
styleClass="manoImagen">
<image>
<Image url="@Imagenes/borrargrupo.png" />
</image>
</ImageView>
<Line endY="101.3332748413086" layoutY="362.0" startX="412.00006103515625"
startY="102.0" />
<Button layoutX="58.0" layoutY="491.0" mnemonicParsing="false"
onAction="#agregarTodosAlCarrito" prefHeight="48.0" prefWidth="196.0"
styleClass="botonAddCarrito" text="Agregar al carrito" />
</children>
</Pane>
<Pane fx:id="panelHistorialCompra" layoutX="600.0" prefHeight="554.0" prefWidth="411.0"
style="-fx-background-color: white;" visible="false">
<children>
<Line endY="101.3332748413086" layoutY="38.0" startX="412.00006103515625"
startY="102.0" />
<Label layoutX="156.0" layoutY="14.0" prefHeight="102.0" prefWidth="231.0"
styleClass="LetraTitulos" text="Historial de compras" wrapText="true" />
<Circle fill="WHITE" layoutX="73.0" layoutY="69.0" radius="54.0" stroke="BLACK"
strokeType="INSIDE" />

```

```

<ImageView fitHeight="73.0" fitWidth="80.0" layoutX="38.0" layoutY="35.0"
onMouseClicked="#mostrarCarrito" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@Imagenes/historialcompra.png" />
    </image>
</ImageView>
<ScrollPane fitToWidth="true" hbarPolicy="NEVER" layoutY="140.0" prefHeight="417.0"
prefWidth="412.0">
    <content>
        <AnchorPane minHeight="0.0" minWidth="0.0">
            <children>
                <VBox fx:id="contenedorHistorialCompra" layoutX="-1.0" layoutY="-8.0"
prefHeight="100.0" prefWidth="397.0" AnchorPane.leftAnchor="-0.0"
AnchorPane.rightAnchor="0.0" AnchorPane.topAnchor="0.0" />
            </children>
        </AnchorPane>
    </content>
</ScrollPane>
</children>
</Pane>
<Pane fx:id="panelCuenta" layoutX="600.0" prefHeight="554.0" prefWidth="411.0"
style="-fx-background-color: white;" visible="false">
    <children>
        <Line endY="101.3332748413086" layoutY="38.0" startX="412.00006103515625"
 startY="102.0" />
        <Label layoutX="156.0" layoutY="14.0" prefHeight="102.0" prefWidth="214.0"
styleClass="LetraTitulos" text="Mi cuenta" wrapText="true" />
        <Circle fill="WHITE" layoutX="73.0" layoutY="69.0" radius="54.0" stroke="BLACK"
strokeType="INSIDE" />
        <ImageView fitHeight="73.0" fitWidth="80.0" layoutX="37.0" layoutY="32.0"
onMouseClicked="#mostrarCarrito" pickOnBounds="true" preserveRatio="true">
            <image>
                <Image url="@Imagenes/Icono%20usuario.png" />
            </image>
        </ImageView>
        <Line endY="101.3332748413086" layoutY="362.0" startX="412.00006103515625"
startY="102.0" />
        <Button fx:id="btnGuardar" layoutX="58.0" layoutY="491.0" mnemonicParsing="false"
onAction="#guardarCambios" prefHeight="48.0" prefWidth="133.0" styleClass="botonComprar"
text="Guardar" />
        <Button fx:id="btnCancelar" layoutX="217.0" layoutY="492.0" mnemonicParsing="false"
onAction="#cancelarCambios" prefHeight="48.0" prefWidth="133.0" styleClass="botonCancelar"
text="Cancelar" />
        <Label layoutX="15.0" layoutY="160.0" styleClass="LetraPrecioDescuento" text="Nombres"
/>
        <Label layoutX="15.0" layoutY="210.0" styleClass="LetraPrecioDescuento"
text="Apellidos" />
        <Label layoutX="15.0" layoutY="260.0" styleClass="LetraPrecioDescuento" text="Sexo" />
        <Label layoutX="15.0" layoutY="310.0" styleClass="LetraPrecioDescuento" text="Edad" />
        <Label layoutX="15.0" layoutY="360.0" styleClass="LetraPrecioDescuento" text="Pais" />
        <Label layoutX="15.0" layoutY="410.0" styleClass="LetraPrecioDescuento"
text="Dirección" />
        <TextField fx:id="txtNombres" editable="false" layoutX="170.0" layoutY="165.0" />
        <TextField fx:id="txtApellidos" editable="false" layoutX="170.0" layoutY="212.0" />
        <TextField fx:id="txtSexo" editable="false" layoutX="170.0" layoutY="261.0" />
    </children>

```

```

<TextField fx:id="txtEdad" editable="false" layoutX="170.0" layoutY="311.0" />
<TextField fx:id="txtPais" editable="false" layoutX="170.0" layoutY="367.0" />
<TextField fx:id="txtDireccion" editable="false" layoutX="170.0" layoutY="412.0" />
<ImageView fx:id="btnEditarNombre" fitHeight="26.0" fitWidth="29.0" layoutX="355.0"
layoutY="164.0" onMouseClicked="#editarNombre" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@Imagenes/editar.png" />
    </image>
</ImageView>
<ImageView fx:id="btnEditarApellidos" fitHeight="26.0" fitWidth="29.0" layoutX="355.0"
layoutY="212.0" onMouseClicked="#editarApellidos" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@Imagenes/editar.png" />
    </image>
</ImageView>
<ImageView fx:id="btnEditarSexo" fitHeight="26.0" fitWidth="29.0" layoutX="355.0"
layoutY="260.0" onMouseClicked="#editarSexo" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@Imagenes/editar.png" />
    </image>
</ImageView>
<ImageView fx:id="btnEditarEdad" fitHeight="26.0" fitWidth="29.0" layoutX="355.0"
layoutY="311.0" onMouseClicked="#editarEdad" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@Imagenes/editar.png" />
    </image>
</ImageView>
<ImageView fx:id="btnEditarPais" fitHeight="26.0" fitWidth="29.0" layoutX="355.0"
layoutY="365.0" onMouseClicked="#editarPais" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@Imagenes/editar.png" />
    </image>
</ImageView>
<ImageView fx:id="btnEditarDireccion" fitHeight="26.0" fitWidth="29.0" layoutX="355.0"
layoutY="412.0" onMouseClicked="#editarDireccion" pickOnBounds="true" preserveRatio="true">
    <image>
        <Image url="@Imagenes/editar.png" />
    </image>
</ImageView>
</children>
</Pane>
<Pane fx:id="panelOpciones" layoutX="668.0" layoutY="150.0" prefHeight="200.0"
prefWidth="231.0" style="-fx-background-color: white;" visible="false">
    <children>
        <Label layoutX="70.0" layoutY="10.0" onMouseClicked="#mostrarFavoritos"
styleClass="tipoLetraVinculosPagPrincipal" text="Favoritos" />
        <Label layoutX="38.0" layoutY="50.0" onMouseClicked="#mostrarListaDeseos"
styleClass="tipoLetraVinculosPagPrincipal" text="Lista de deseos" />
        <Label layoutX="10.0" layoutY="90.0" onMouseClicked="#mostrarHistorialCompra"
styleClass="tipoLetraVinculosPagPrincipal" text="Historial de compras" />
        <Label layoutX="64.0" layoutY="130.0" onMouseClicked="#mostrarCuenta"
styleClass="tipoLetraVinculosPagPrincipal" text="Mi cuenta" />
        <Label layoutX="45.0" layoutY="170.0" onMouseClicked="#cerrarSesion"
styleClass="tipoLetraVinculosPagPrincipal" text="Cerrar sesión" />
    </children>

```

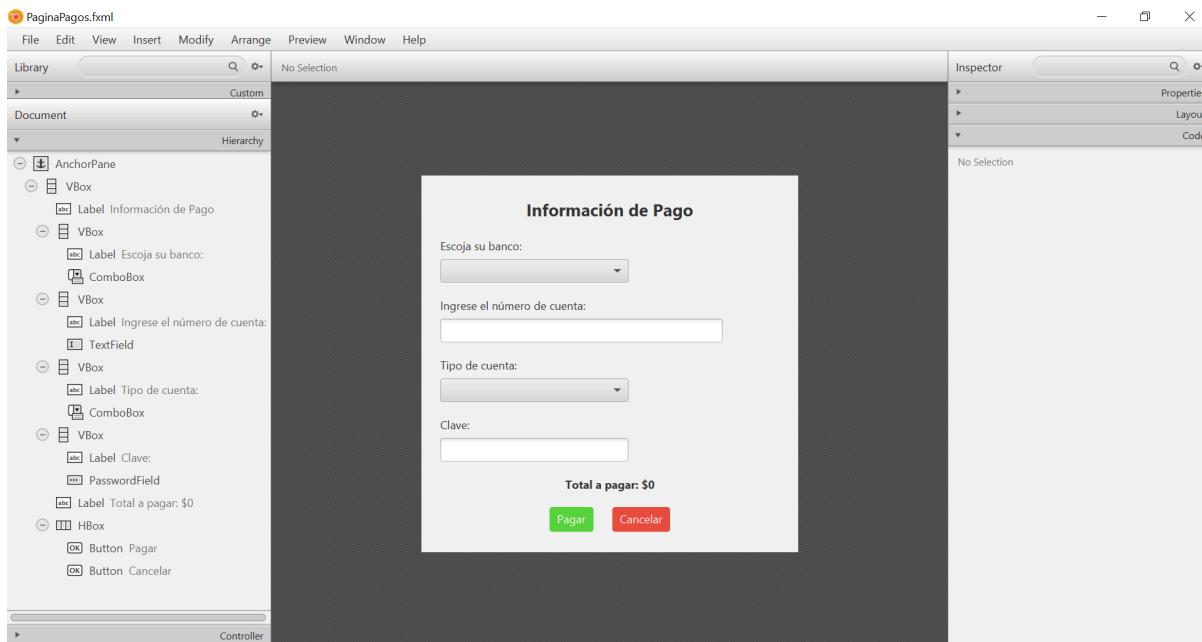
```

</Pane>
<Pane fx:id="opacidad" onMouseClicked="#volver" prefHeight="554.0" prefWidth="601.0"
style="-fx-background-color: white;" visible="false" />
</children>
</AnchorPane>

```

Para finalizar con los archivos FXML, editaremos PaginaPagos.fxml.

1. SCENE BUILDER



2. CÓDIGO

```

<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ComboBox?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.PasswordField?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>

<AnchorPane prefHeight="400.0" prefWidth="400.0" stylesheets="@estilos.css"
xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1"
fx:controller="Controlador.PaginaPagosControlador">
    <VBox alignment="CENTER" spacing="15" AnchorPane.bottomAnchor="20"
AnchorPane.leftAnchor="20" AnchorPane.rightAnchor="20" AnchorPane.topAnchor="20">
        <Label style="-fx-font-size: 18; -fx-font-weight: bold;" text="Información de Pago" />
        <VBox spacing="5">
            <Label text="Escoja su banco:" />
            <ComboBox fx:id="comboBanco" prefWidth="200" />
        </VBox>
        <VBox spacing="5">

```

```

<Label text="Ingrese el número de cuenta:" />
<TextField fx:id="txtNumeroCuenta" maxWidth="300" />
</VBox>

<VBox spacing="5">
    <Label text="Tipo de cuenta:" />
    <ComboBox fx:id="comboTipoCuenta" prefWidth="200" />
</VBox>

<VBox spacing="5">
    <Label text="Clave:" />
    <PasswordField fx:id="txtClave" maxWidth="200" />
</VBox>

<Label fx:id="lblTotal" style="-fx-font-weight: bold;" text="Total a pagar: $0" />

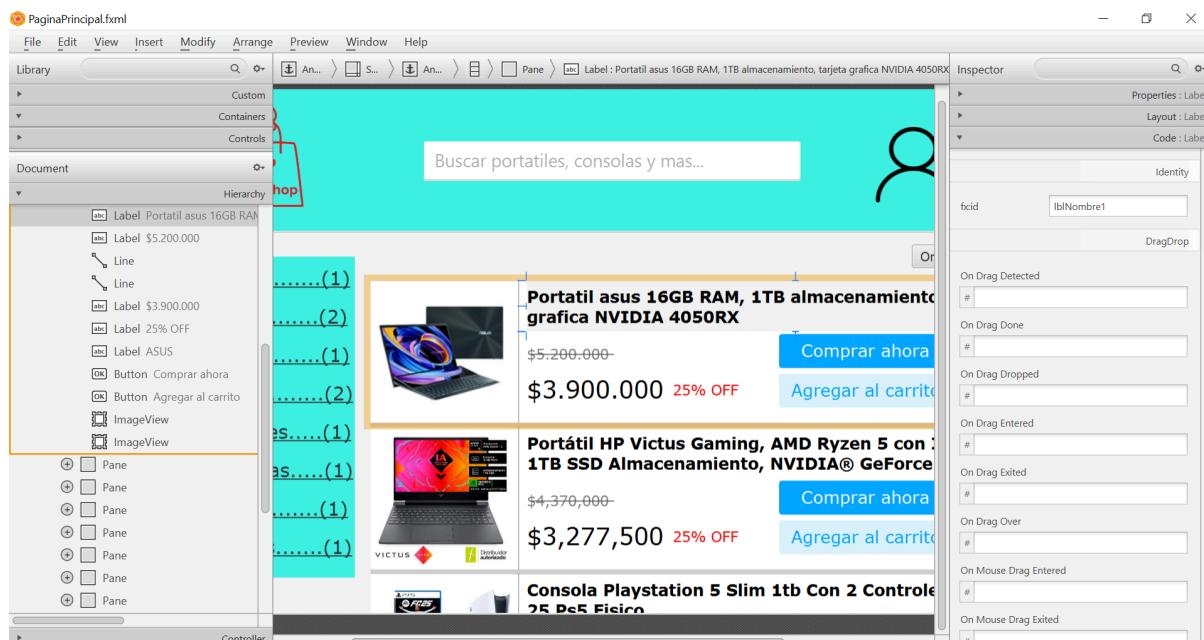
<HBox alignment="CENTER" spacing="20">
    <Button fx:id="btnPagar" onAction="#pagar" style="-fx-background-color: #56D23D;
-fx-text-fill: white;" styleClass="manoImagen" text="Pagar" />
    <Button fx:id="btnCancelar" onAction="#cancelar" style="-fx-background-color: #E74C3C;
-fx-text-fill: white;" styleClass="manoImagen" text="Cancelar" />
</HBox>
</VBox>
</AnchorPane>

```

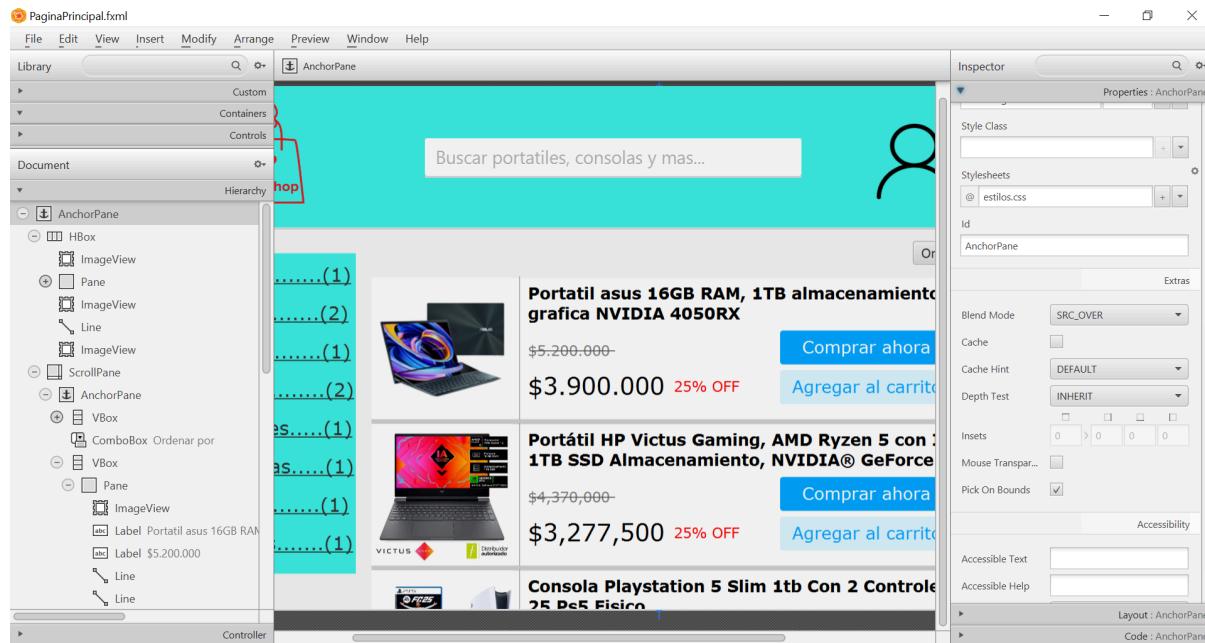
Una vez agregado y editados los archivos FXML, editaremos el último archivo de nuestro proyecto: Estilos.css, este archivo se encarga de agregar estilos a los componentes de los archivos fxml tales como cambiar color, letra, tamaños, etc. (esto se puede hacer directamente en el apartado Properties=Style, pero es menos dinámico)

NOTA: Si usaste el método de SCENE BUILDER, debes cerciorarte de agregar los respectivos FX:ID, a los elementos y enlazar la hoja de estilos para que se puedan aplicar.

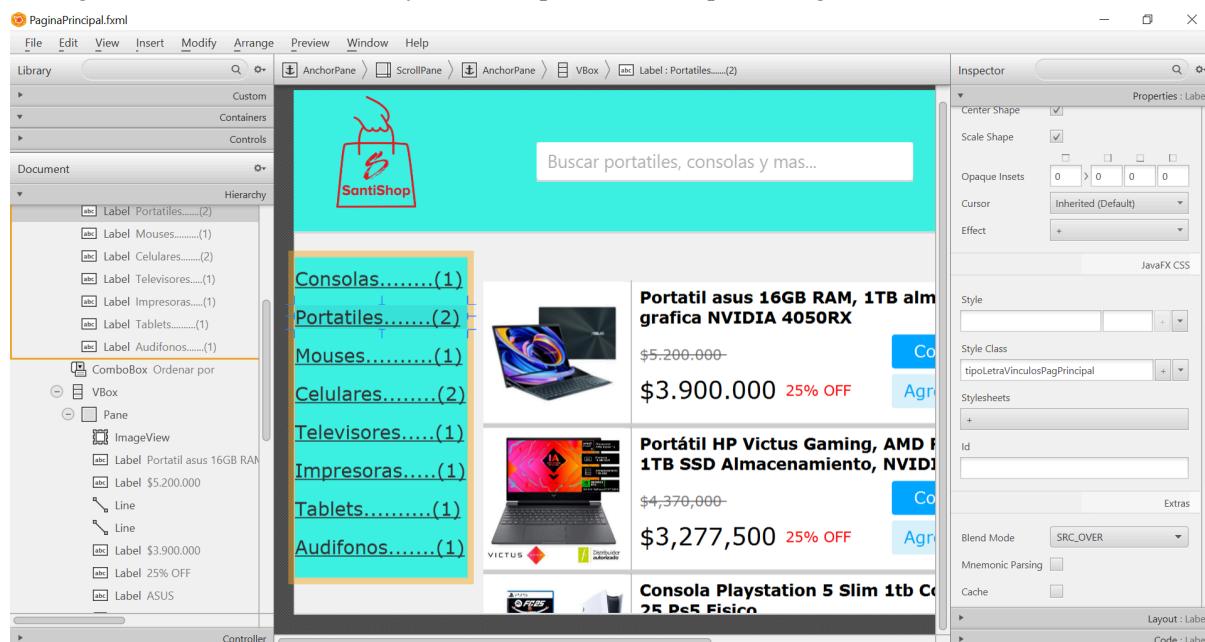
FX:ID: se agregan seleccionando un elemento y yendo a la pestaña Code en la derecha:



HOJA DE ESTILOS: esto se divide en dos pasos: 1. enlazar la hoja de estilos escogiendo el AnchorPane principal y en Properties bajar hasta Stylesheets y escoger el archivo de estilos



2. Asignarle a cada elemento el StyleClass dependiendo lo que se asigne en el archivo Estilos



Si escogiste el método por código, solo copia y pega este código y ya:

```
.vinculos {
    -fx-text-fill: blue;
    -fx-cursor: hand;
}
.vinculos:hover{
    -fx-text-fill: #070287;
}
```

```
.tipoLetra{  
    -fx-font-family: verdana;  
    -fx-font-size: 13px;  
}  
.botonRegistrarIniciar{  
    -fx-background-color: black;  
    -fx-font-size: 22px;  
    -fx-font-family: calibri;  
    -fx-cursor: hand;  
}  
  
.botonRegistrarIniciar:hover{  
    -fx-background-color: #2a2a2a;  
}  
  
.botonRegistrarIniciar:preset{  
    -fx-background-color: #aca7a7;  
}  
  
.botonCerrar{  
    -fx-background-color: red;  
    -fx-cursor: hand;  
}  
.botonCerrar:hover{  
    -fx-background-color: #f84d4d;  
}  
  
.tipoLetraVinculosPagPrincipal{  
    -fx-font-family: verdana;  
    -fx-font-size: 20px;  
    -fx-underline: true;  
    -fx-cursor: hand;  
}  
  
.LetraNombreProducto{  
    -fx-font-family: verdana;  
    -fx-font-size: 17px;  
    -fx-text-fill: black;  
    -fx-font-weight: bold;  
}  
  
.LetraPrecioOriginal{  
    -fx-font-family: verdana;  
    -fx-font-size: 15px;  
    -fx-text-fill: #817e7e;  
}  
  
.LetraPrecioDescuento{  
    -fx-font-family: verdana;  
    -fx-font-size: 25px;  
    -fx-text-fill: black;  
}  
  
.LetraPorcentaje{  
    -fx-font-family: verdana;
```

```
-fx-font-size: 15px;
-fx-text-fill: red;
}

.LetraMarca{
    -fx-font-family: verdana;
    -fx-font-size: 15px;
    -fx-text-fill: black;
}

.botonComprar{
    -fx-background-color: #00a7ff;
    -fx-font-size: 18px;
    -fx-font-family: verdana;
    -fx-text-fill: white;
    -fx-cursor: hand;
}

.botonComprar:hover{
    -fx-background-color: #036497;
}

.botonComprar:preset{
    -fx-background-color: #033e5c;
}

.botonAddCarrito{
    -fx-background-color: #daf2fe;
    -fx-font-size: 18px;
    -fx-font-family: verdana;
    -fx-text-fill: #00a7ff;
    -fx-cursor: hand;
}

.botonAddCarrito:hover{
    -fx-background-color: #c5dce8;
}

.botonAddCarrito:preset{
    -fx-background-color: #849299;
}

.LetraTitulos{
    -fx-font-family: verdana;
    -fx-font-size: 40px;
    -fx-text-fill: black;
}

.opacidad{
    -fx-opacity: 0.5;
}

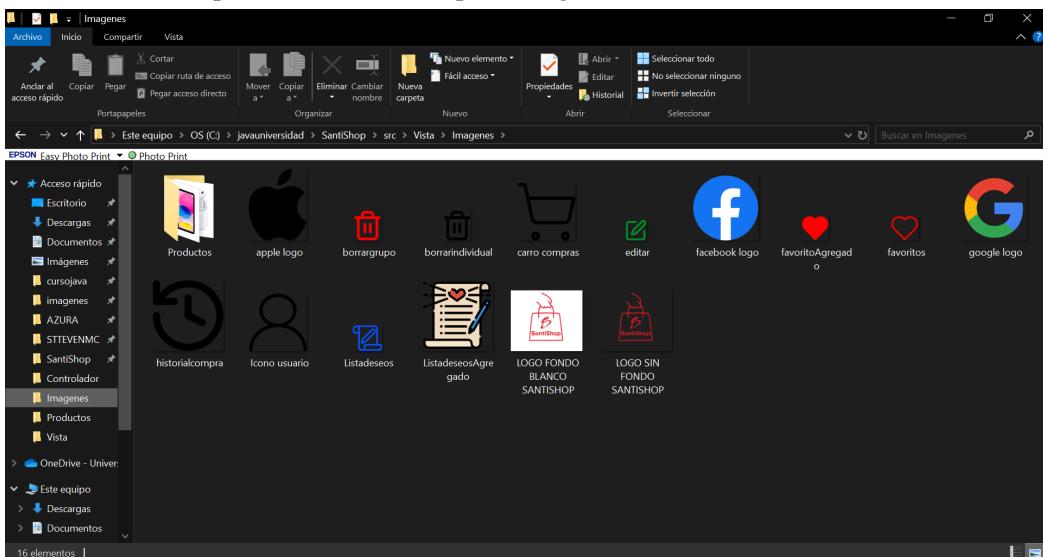
.manoImagen{
    -fx-cursor: hand;
}
```

```
.botonCancelar{
    -fx-background-color: red;
    -fx-font-size: 18px;
    -fx-font-family: verdana;
    -fx-text-fill: white;
    -fx-cursor: hand;
}

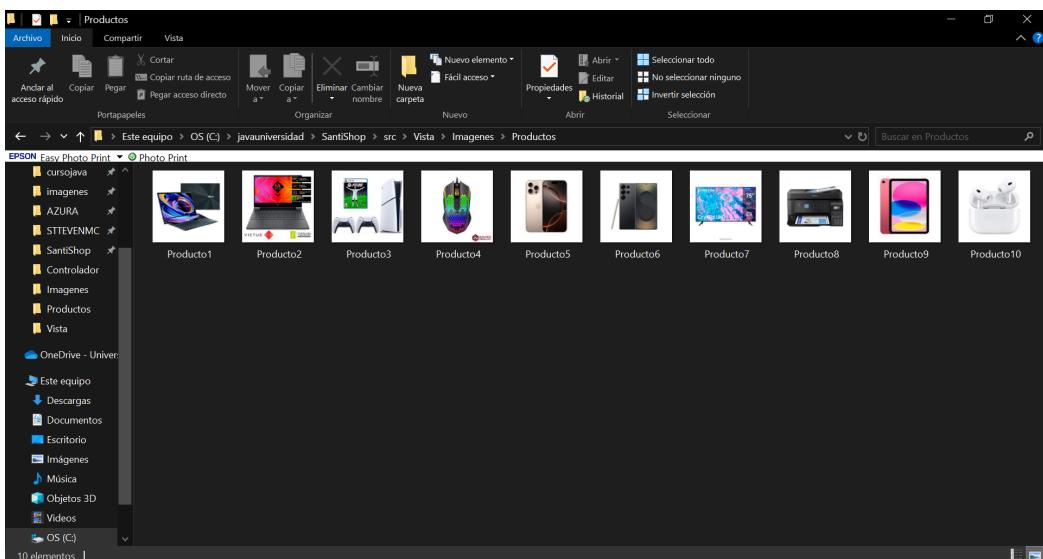
.botonCancelar:hover{
    -fx-background-color: #c30202;
}

.botonCancelar:preset{
    -fx-background-color: #940202;
}
```

estos son los elementos que debe llevar la carpeta imágenes:



y la subcarpeta Producto:



RESULTADOS

Para verificar que la tienda funcione correctamente se pone a prueba:

1. Registro

SantiShop - Registro

REGISTER

Usuario

Contraseña Ingresar una contraseña

Nombres

Apellidos

Sexo

Edad

País

Registrarse

[Volver al login](#)

2. Login

SantiShop - Login

LOGIN

Usuario

Contraseña Ingresar tu contraseña

[Olvidé mi contraseña](#)

Iniciar sesión

¿Aún no haces parte de la familia?
[REGISTRATE](#)

PROXIMAMENTE

3. Página Principal

Screenshot of the SantiShop - Página Principal website. The page features a teal header with a logo, a search bar, and user icons. A sidebar on the left lists categories with counts: Consolas (1), Portatiles (2), Mouses (1), Celulares (2), Televisores (1), Impresoras (1), Tablets (1), and Audifonos (1). The main content area displays a grid of products:

- Portátil ASUS 16GB RAM, 1TB almacenamiento, tarjeta gráfica NVIDIA 4050RX**
\$5.200.000 → \$3.900.000 25% OFF
ASUS
- Portátil HP Victus Gaming, AMD Ryzen 5 con IA, 16GB RAM, 1TB SSD Almacenamiento, NVIDIA® GeForce RTX 3050**
\$4.370.000 → \$3.277.500 25% OFF
HP
- Consola Playstation 5 Slim 1tb Con 2 Controles y Juego Fc 25 Días Fisico**

4. Carrito de compras

Screenshot of the SantiShop - Página Principal website showing the shopping cart interface. The cart summary on the right side includes:

- Carrito de compras**
- Subtotal:**
- Descuento:**
- Total:**

The main content area shows the same product grid as the previous screenshot.

5. Mi cuenta

SantiShop - Página Principal

Buscar portatiles, consolas y mas...

Favoritos

Lista de deseos

Historial de compras

Mi cuenta

Cerrar sesión

Portátil ASUS 16GB RAM, 1TB a gráfica NVIDIA 4050RX
\$5.200.000-\$3.900.000 25% OFF

Portátil HP Victus Gaming, AMD Ryzen 5 con IA, 16GB RAM, 1TB SSD Almacenamiento, NVIDIA® GeForce RTX 3050
\$4.370.000-\$3.277.500 25% OFF

Consola Playstation 5 Slim 1tb Con 2 Controles y Juego Fc
25 Días Fisico

Comprar ahora

Agregar al carrito

HP

Consolas.....(1)

Portatiles.....(2)

Mouses.....(1)

Celulares.....(2)

Televisores.....(1)

Impresoras.....(1)

Tablets.....(1)

Audifonos.....(1)

6. Favoritos

SantiShop - Página Principal

Buscar portatiles, consolas y mas...

Favoritos

Portátil ASUS 16GB RAM gráfica NVIDIA 4050RX
\$5.200.000-\$3.900.000 25% OFF

Portátil HP Victus Gamin 1TB SSD Almacenamient
\$4.370.000-\$3.277.500 25% OFF

Consola Playstation 5 Sli
25 Días Fisico

Consolas.....(1)

Portatiles.....(2)

Mouses.....(1)

Celulares.....(2)

Televisores.....(1)

Impresoras.....(1)

Tablets.....(1)

Audifonos.....(1)

7. Lista de deseos

SantiShop - Página Principal

Buscar portatiles, consolas y mas...

Lista de deseos

Consolas.....(1)		Portátil ASUS 16GB RAM gráfica NVIDIA 4050RX
Portatiles.....(2)		\$5.200.000- \$3.900.000 25% OFF
Mouses.....(1)		Portátil HP Victus Gamin 1TB SSD Almacenamiento
Celulares.....(2)		\$4.370.000- \$3.277.500 25% OFF
Televisores.....(1)		Consola Playstation 5 Slim 25 De Fisico
Impresoras.....(1)		
Tablets.....(1)		
Audifonos.....(1)		

[Agregar al carrito](#)

8. Historial de Compras

SantiShop - Página Principal

Buscar portatiles, consolas y mas...

Historial de compras

Consolas.....(1)		Portátil ASUS 16GB RAM gráfica NVIDIA 4050RX
Portatiles.....(2)		\$5.200.000- \$3.900.000 25% OFF
Mouses.....(1)		Portátil HP Victus Gamin 1TB SSD Almacenamiento
Celulares.....(2)		\$4.370.000- \$3.277.500 25% OFF
Televisores.....(1)		Consola Playstation 5 Slim 25 De Fisico
Impresoras.....(1)		
Tablets.....(1)		
Audifonos.....(1)		

9. Mi Cuenta

The screenshot shows the SantiShop website's main page. On the left, there is a sidebar with categories: Consolas (1), Portatiles (2), Mouses (1), Celulares (2), Televisores (1), Impresoras (1), Tablets (1), and Audifonos (1). The main area features a search bar with placeholder text 'Buscar portatiles, consolas y mas...'. To the right, there is a user profile icon labeled 'Mi cuenta'. Below it, there are input fields for personal information: Nombres (with value 'jjj'), Apellidos (with value 'ggg'), Sexo (with value 'Masculino'), Edad (with value '78'), País (with value 'sel'), and Dirección (with value 'col'). There are also two blue edit icons next to each input field. At the bottom right are 'Guardar' and 'Cancelar' buttons.

10. Comprar

The screenshot shows a 'Información de Pago' (Payment Information) dialog box. It contains fields for selecting a bank, entering a bank account number, choosing an account type, and entering a key. At the bottom, it displays the total amount to be paid: 'Total a pagar: \$3.900.000'. There are 'Pagar' and 'Cancelar' buttons at the bottom.

CONCLUSIONES

El desarrollo de este proyecto permitió aplicar de forma práctica los conocimientos adquiridos en el curso de estructuras de datos, mediante el diseño e implementación de un sistema funcional que simula un entorno de comercio electrónico. Se logró integrar correctamente estructuras como listas enlazadas, pilas y colas dentro de un flujo de trabajo realista, fortaleciendo así la comprensión sobre su uso y eficiencia.

Además, la implementación de JavaFX nos propuso como grupo un reto adicional que aportó significativamente al desarrollo de interfaces gráficas interactivas. Aprendí a conectar la lógica de las estructuras de datos con elementos visuales y me ofreció una experiencia más dinámica. Esta integración entre la lógica y la presentación visual fue fundamental para ofrecer una aplicación más atractiva, intuitiva y cercana a una solución profesional.

En conjunto, este proyecto me permitió aplicar de forma práctica los conceptos de estructuras de datos, mejorando nuestra lógica, capacidad de resolución de problemas y habilidades para desarrollar software funcional y organizado. También fomenta el trabajo en equipo, la planificación modular y la adaptación a nuevas herramientas de desarrollo como JavaFX. El resultado final es una plataforma sólida que cumple con los requisitos propuestos y refleja el avance logrado durante el curso.

BIBLIOGRAFÍAS

- Deitel, P. J., & Deitel, H. M. (2019). *Java: How to Program*. Pearson Education.
- Liang, Y. D. (2021). *Introduction to Java Programming and Data Structures*. Pearson.
- Oracle. (s.f.). *JavaFX Documentation*. Recuperado de: <https://openjfx.io/>
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. MIT Press.
- GeeksforGeeks. (s.f.). *Data Structures*. Recuperado de:
<https://www.geeksforgeeks.org/data-structures/>
- Weiss, M. A. (2013). *Data Structures and Algorithm Analysis in Java* (3rd ed.). Pearson Education.
- Lafore, R. (2002). *Data Structures and Algorithms in Java* (2nd ed.). Sams Publishing.
- Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2014). *Data Structures and Algorithms in Java* (6th ed.). Wiley.