



ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
TECNOLOGIA SUPERIOR EN DESARROLLO DE SOFTWARE



**PROYECTO: SISTEMA INTEGRAL PARA REGISTRAR
COMPRAS EN UNA TIENDA DE GORRAS**

PROGRAMACION ORIENTADA A OBJETOS

AUTOR

ISAAC VALENZUELA

QUITO – ECUADOR

AGOSTO - 2024

Tabla de Contenidos:

- 1. Resumen del Proyecto**
- 2. Objetivos**
 - 2.1. Objetivo General**
 - 2.2. Objetivos Específicos**
- 3. Alcance del Proyecto**
- 4. Desarrollo del Proyecto**
 - 4.1Requerimientos del Sistema**
 - 4.2Arquitectura del Sistema**
 - 4.3Diseño del Sistema**
 - 4.4 Implementación**
- 5. Recomendaciones**
- 6. Conclusiones**
- 7. Anexos**

1. RESUMEN

Este proyecto se orienta a la creación de una solución tecnológica práctica y funcional que responda a las necesidades específicas de una tienda de gorras física, aplicando los conocimientos teóricos y prácticos adquiridos durante el semestre. Con la implementación del sistema propuesto, se espera mejorar la eficiencia operativa y la satisfacción del cliente, al mismo tiempo que se brinda una herramienta eficaz para la administración y el control de ventas e inventarios.

2. OBJETIVOS

2.1. Objetivo General

Desarrollar un sistema integral que facilite las operaciones diarias de una tienda de gorras física, mediante la automatización de procesos clave como la gestión de inventarios, la realización de transacciones de compra y la administración de usuarios.

2.2. Objetivos Específicos

- Implementar un sistema de login que permita el acceso diferenciado para administradores y cajeros.
- Desarrollar un módulo de transacciones de compra que registre cada operación realizada.
- Generar notas de venta en formato PDF al finalizar cada transacción.
- Generar reporte de ventas filtrando por fecha
- Implementar la reducción automática del stock de productos tras cada compra.
- Ingresar imágenes de los productos en la base de datos.
- Desarrollar funcionalidades de administración tipo CRUD

3. ALCANCE DEL PROYECTO

El proyecto abarca el desarrollo de un sistema software que se instalará en las instalaciones de la tienda de gorras, incluyendo los siguientes componentes:

- Login de Usuarios: Diferenciación entre accesos de administradores y cajeros.
- Gestión de Transacciones: Registro y almacenamiento de cada compra realizada.
- Generación de Notas de Venta: Emisión de un comprobante en PDF para cada transacción.
- Generación de reportes de venta: Emisión de un PDF con las compras filtradas por fechas

- Gestión de Inventarios: Actualización automática del stock tras cada venta y visualización de imágenes de productos.
- Administración de Usuarios y Productos tipo CRUD

4. DESARROLLO DEL PROYECTO

4.1. Requisitos del sistema

Funcionalidades del Cajero:

- Realizar transacciones de compra y registrar cada operación.
- Generar y guardar notas de venta en formato PDF.
- Generar reportes de ventas en formato PDF

Funcionalidades del Administrador:

- Gestionar productos con el formato CRUD
- Gestionar cajeros con el formato CRUD

Interfaz de Usuario:

- **Login:** Pantalla de inicio de sesión con opciones para administrador y cajero.
- **Transacciones de Compra:** Interfaz para la realización de compras, visualización de productos seleccionados e impresión de notas de venta.
- **Generación de Reportes:** Interfaz para filtrar por fecha los productos comprados para después realizar un informe de ventas.
- **Gestión de productos:** Interfaz para el ingreso, visualización, eliminación y actualización de productos en inventario.
- **Administración de Usuarios:** Pantalla para la gestión de usuarios cajeros.

4.2. Arquitectura del sistema

Frontend:

- Lenguaje de programación: Java con Swing para la interfaz gráfica.
- Herramienta de desarrollo: IntelliJ IDEA.
- Bibliotecas: PDF para la generación de PDF.
- JAR: para crear un archivo formato jar
- Launch4j: para crear un archivo formato exe

Backend:

- Lenguaje de programación: MySQL para el almacenamiento de datos de cajeros y administradores, productos, transacciones de venta, detalle de venta y stock.
- Base de datos: Clever Cloud para el almacenamiento de datos en la nube.
- Frameworks: JDBC para la conexión con la base de datos.

4.3. Diseño del sistema

Diagrama de Clases:

- **Clases principales:** Login, Administrador, Cajero, Facturación, Reportes, Gestión de cajero, Gestión de productos.
- **Relación entre clases:** Se instancian las clases para poder navegar entre ellas, estas clases están diseñadas con formularios lo que hace que se cree una arquitectura de navegación.

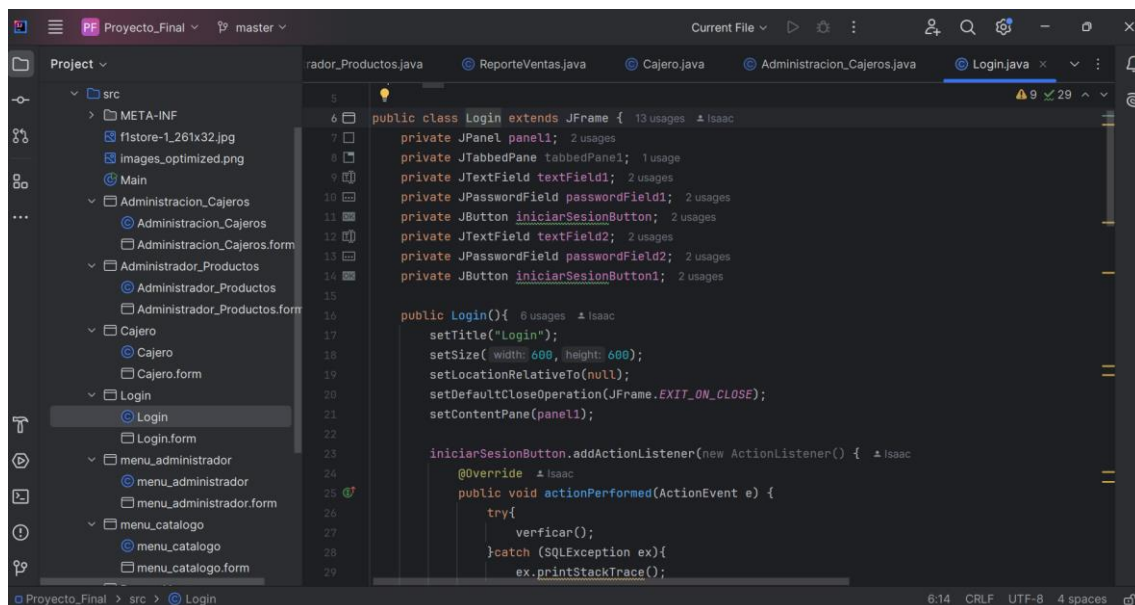
Diagrama de Casos de Uso:

- **Login:** El usuario (administrador o cajero) ingresa sus credenciales para acceder al sistema.
- **Transacciones de Compra:** El cajero selecciona productos, registra la compra y genera una nota de venta formato PDF.
- **Generación de Reportes:** El cajero genera reportes de ventas dependiendo de la fecha de digite.
- **Gestión de Stock:** El administrador gestiona los productos en el inventario.
- **Administración de Usuarios:** El administrador gestiona usuarios cajeros.

4.4. Implementación

Desarrollo del Login:

- Creación de la interfaz de inicio de sesión.
- Codificación de clases, instancias, botones, conexiones
- Implementación de la validación de credenciales contra la base de datos.
- Selección de las interfaces de administrador o cajero según el tipo de usuario.



Iniciar Sesión Empresa X.Y.Z



Cajero Administrador


Correo:

Contraseña:

Iniciar Sesión

Login

Iniciar Sesion Empresa X.Y.Z



Cajero

Administrador

Correo:

Contraseña:

Iniciar Sesion

Módulo de Transacciones de Compra:

- Desarrollo de la interfaz de cajero.
- Implementación de la lógica para registrar transacciones y actualizar el stock.
- Generación de notas de venta en formato PDF utilizando PDFBOX.

Project

src

META-INF

f1store-1_261x32.jpg

images_optimized.png

Main

Administracion_Cajeros

Administracion_Cajeros

Administracion_Cajeros.form

Administrador_Productos

Administrador_Productos

Administrador_Productos.form

Cajero

Cajero

Cajero.form

21

public class Cajero extends JFrame {

6 usages

Isaac

285

public void registrarVenta() throws SQLException {

1 usage

Isaac

236

ResultSet generatedKeys = pstmt.getGeneratedKeys();

237

if (generatedKeys.next()) {

238

int ventaId = generatedKeys.getInt(columnIndex: 1);

239

240

for (Product product : productList) {

241

String detalleSql = "INSERT INTO DetalleVentas (VentaID, ProductoID, ProductoNombre

242

PreparedStatement detallePstmt = conectamos.prepareStatement(detalleSql);

243

detallePstmt.setInt(parameterIndex: 1, ventaId);

244

detallePstmt.setInt(parameterIndex: 2, product.getId());

245

detallePstmt.setString(parameterIndex: 3, product.getNombre());

246

detallePstmt.setDouble(parameterIndex: 4, product.getPrecio());

247

detallePstmt.setInt(parameterIndex: 5, product.getCantidad());

248

detallePstmt.executeUpdate();

249

detallePstmt.close();

Run

Main

C:\Users\Isaac\.jdk\openjdk-22.0.1\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.1.1\lib\idea_rt.jar

Nombre del Cliente:	<input type="text"/>
Cédula del Cliente:	<input type="text"/>
Dirección del Cliente:	<input type="text"/>
ID del Producto:	<input type="text"/>
Nombre del Producto:	<input type="text"/>
Precio del Producto:	<input type="text"/>
Cantidad:	<input type="text"/>
<input type="button" value="Agregar Producto"/>	
<input type="button" value="Registrar Venta"/>	
<div></div>	
<input type="button" value="Volver"/>	

Módulo de Generar Reporte de Ventas:

- Desarrollo de la interfaz de reportes.
- Implementación de la lógica para buscar compras por fecha.
- Generación de reportes de venta en formato PDF utilizando PDFBOX.


```

public Administrador_Productos(){ 1 usage  Isaac
    setTitle("Administrador");
    setSize( width: 600, height: 665);
    setLocationRelativeTo(null);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setContentPane(panel1);

    ingresarButton.addActionListener(new ActionListener() {  Isaac
        @Override  Isaac
        public void actionPerformed(ActionEvent e) {
            try {
                ingresa();
            } catch (SQLException | IOException ex) {
                ex.printStackTrace();
            }
        }
    });
}

```

```

public void crear()throws SQLException { 1 usage  Isaac
    String cedula = textField1.getText();
    String correo = textField2.getText();
    String nombre = textField3.getText();
    String contrasenia = textField4.getText();
    String edad = textField5.getText();
    String direccion = textField6.getText();

    Connection conectamos = connection();
    String sql = "INSERT INTO cajero(cedula, correo, nombre, contrasenia, edad, direccion)"
        "VALUES(?,?,?,?,?,?)";
    PreparedStatement pstmt = conectamos.prepareStatement(sql);
    pstmt.setString( parameterIndex: 1, cedula);
    pstmt.setString( parameterIndex: 2, correo);
    pstmt.setString( parameterIndex: 3, nombre);
    pstmt.setString( parameterIndex: 4, contrasenia);
    pstmt.setInt( parameterIndex: 5, Integer.parseInt(edad));
    pstmt.setString( parameterIndex: 6, direccion);
}

```

Administracion de cajeros

[Crear](#)[Ver](#)[Actualizar](#)[Eliminar](#)

Inserte datos:

Cedula:

Correo:

Nombre:

Contraseña:

Edad:

Direccion:

Crear

Administracion de Productos

Agregar

Ver

Actualizar

Eliminar

Busqueda de producto por su ID

ngrese ID:

D: 1

Nombre: Gorra Mercedes - Lewis Hamilton 44


Descripción: Mercedes

Precio: 62.99

Cantidad: 9

Fecha de entrada: 2024-07-01

Fecha cuando se actualizó: 2024-08-07 02:56:21



Buscar

Cerrar Sesion

bmfp6c3mefmlhvjslupe

Nueva

administrador

cajero

DetalleVentas

InventarioProductos

Ventas

information_schema

Filtros

Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento
<input type="checkbox"/> administrador	<div><div>★</div><div>Examinar</div><div>Estructura</div><div>Buscar</div><div>Insertar</div><div>Vaciar</div><div>Eliminar</div></div>	2	InnoDB	utf8_general_c
<input type="checkbox"/> cajero	<div><div>★</div><div>Examinar</div><div>Estructura</div><div>Buscar</div><div>Insertar</div><div>Vaciar</div><div>Eliminar</div></div>	2	InnoDB	utf8_general_c
<input type="checkbox"/> DetalleVentas	<div><div>★</div><div>Examinar</div><div>Estructura</div><div>Buscar</div><div>Insertar</div><div>Vaciar</div><div>Eliminar</div></div>	14	InnoDB	utf8_general_c
<input type="checkbox"/> InventarioProductos	<div><div>★</div><div>Examinar</div><div>Estructura</div><div>Buscar</div><div>Insertar</div><div>Vaciar</div><div>Eliminar</div></div>	20	InnoDB	utf8_general_c
<input type="checkbox"/> Ventas	<div><div>★</div><div>Examinar</div><div>Estructura</div><div>Buscar</div><div>Insertar</div><div>Vaciar</div><div>Eliminar</div></div>	6	InnoDB	utf8_general_c
5 tablas	Número de filas	44	InnoDB	utf8_general_c

PLAN	CLUSTER	VERSION	REGION	STATUS	CREATION DATE	ID
Dev	par-mysql-c6	8.0	par	ACTIVE	2024-07-23	mysql_578ed4d8-1b18-4f8a-b56c-46ea29da2853

Examinar	Estructura	SQL	Buscar	Insertar	Exportar	Importar	Operaciones
----------	------------	-----	--------	----------	----------	----------	-------------

ID	Nombre	Descripción	Precio	Cantidad	Fecha_entrada	Fecha_actualización	Imagen
1	Gorra Mercedes - Lewis Hamilton 44	Mercedes	62.99	9	2024-07-01	2024-08-07 02:56:21	[BLO 22.3
2	Gorra Mercedes - George Russell 63	Mercedes	62.99	15	2024-07-01	2024-08-07 02:56:21	[BLO 59.4
3	Gorra Ferrari - Charles Leclerc 16	Ferrari	51.99	17	2024-07-01	2024-08-07 02:54:20	[BLO 38.9
4	Gorra Ferrari - Carlos Sainz 55	Ferrari	51.99	16	2024-07-01	2024-08-07 15:42:45	[BLO 58.7

5. RECOMENDACIONES

Capacitación del Personal: Realizar sesiones de capacitación para que tanto administradores como cajeros se familiaricen con el uso del sistema.

Mantenimiento Regular: Establecer un plan de mantenimiento regular para asegurar la correcta operación del sistema y la actualización de software.

Seguridad: Implementar medidas de seguridad robustas para proteger la información de usuarios y transacciones, incluyendo el cifrado de datos sensibles.

Pruebas Exhaustivas: Realizar pruebas exhaustivas antes del lanzamiento para identificar y corregir posibles errores.

6. CONCLUSIONES

El desarrollo de este sistema integral permitirá a la tienda de gorras optimizar sus operaciones diarias, mejorar la gestión de inventarios, y proporcionar un servicio más eficiente a sus clientes. La implementación de un sistema automatizado reducirá los errores manuales y facilitará el trabajo tanto del personal administrativo como de los cajeros, contribuyendo así a un mejor desempeño del negocio.

7. ANEXOS

Link de GitHub con todos el repositorio del proyecto: <https://github.com/Isaac-Valenzuela/Proyecto-Final.git>

Link del Video de Youtube explicando el sistema modulo por modulo: https://youtu.be/CFN7G34Gkqg?si=_ty4SRSzlyKdi6xk

Imagen del programa en formato exe:

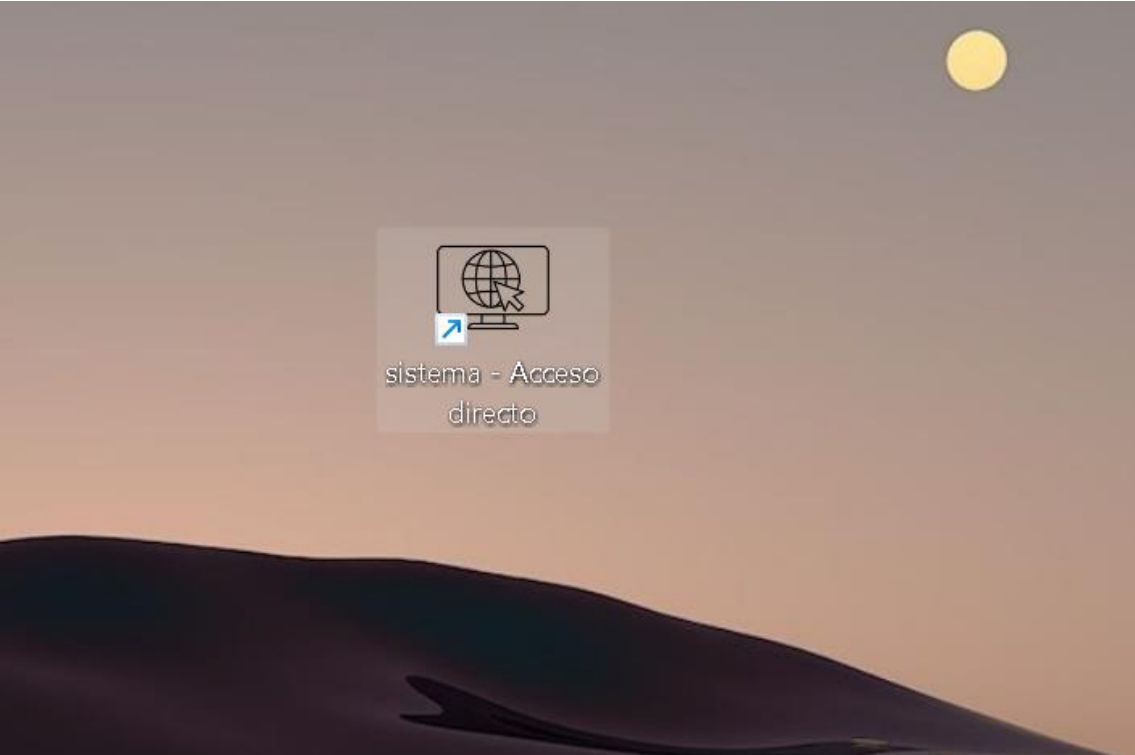


Imagen de la base de datos en la nube, mediante clever cloud:

MySQL by Clever Cloud

Admin PHPMyAdmin Documentation

TYPE	PLAN	CLUSTER	VERSION	REGION	STATUS	CREATION DATE	ID
MySQL	Dev	par-mysql-c6	8.0	par	ACTIVE	2024-07-23	mysql_578ed4d8-1b18-4f8a-b56c-46ea29da2853

phpMyAdmin

Reciente Favoritas

bmfp6c3mefmlhvjslupe
information_schema

Servidor: bmfp6c3mefmlhvjslupe-mysql.services.clever-cloud.com:3306 > Base de datos: bmfp6c3mefmlhvjslupe

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Más

Filtros

Que contengan la palabra:

Tabla	Acción	Filas	Tipo	Cotejamiento
<input type="checkbox"/> administrador	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_c
<input type="checkbox"/> cajero	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	2	InnoDB	utf8_general_c
<input type="checkbox"/> DetalleVentas	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	14	InnoDB	utf8_general_c
<input type="checkbox"/> InventarioProductos	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	20	InnoDB	utf8_general_c
<input type="checkbox"/> Ventas	★ Examinar Estructura Buscar Insertar Vaciar Eliminar	6	InnoDB	utf8_general_c
5 tablas	Número de filas	44	InnoDB	utf8_general_c

Imagen de los commits que se hicieron durante la elaboración del proyecto:



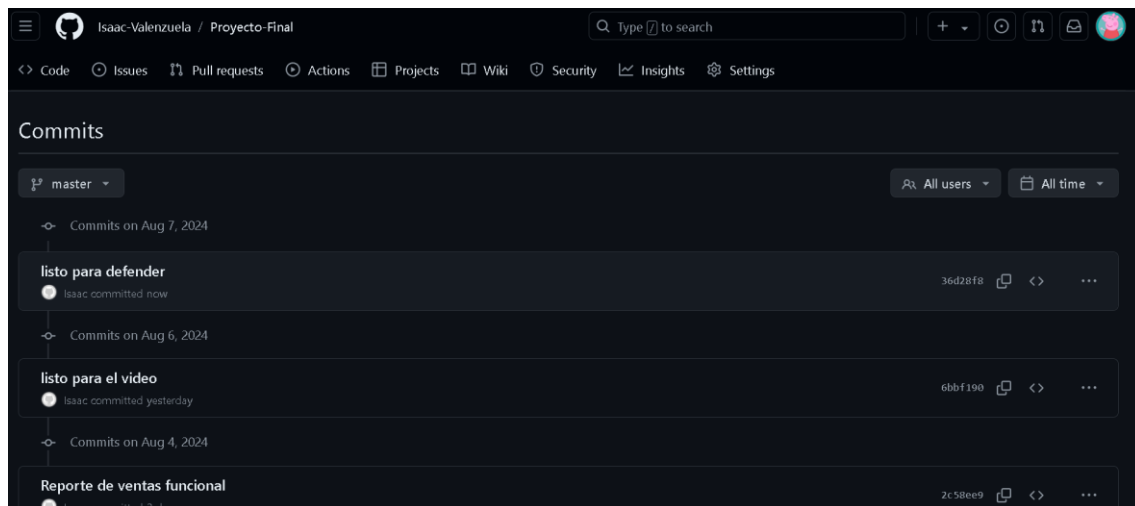


Imagen del repositorio en GitHub:

