

Universidad Politécnica Internacional

Proyecto: Sistema de Ventas para Ferretería

Entregable #1

Técnicas de programación

Profesor: Luis Felipe Mora Umaña

Oscar Johel Davila Escalante

Ángelo Gabriel Wong Gutiérrez

II Cuatrimestre 2022.

Índice

[**Estructura del Backlog** 3](#_Toc173270817)

[**Gestión de Productos** 3](#_Toc173270818)

[Productos 1.1: Agregar Productos 3](#_Toc173270819)

[Productos 1.2: Editar Productos 3](#_Toc173270820)

[Productos 1.3: Eliminar Productos 3](#_Toc173270821)

[**Gestión de Ventas (Lo realiza el cajero)** 4](#_Toc173270822)

[Usuario 2.1: Procesar Ventas 4](#_Toc173270823)

[Usuario 2.2: Generar Facturas 4](#_Toc173270824)

[**Gestión de Clientes (El registro de clientes lo realiza el cajero)** 4](#_Toc173270825)

[Usuario 3.1: Registrar Clientes 4](#_Toc173270826)

[Usuario 3.2: Ver Historial de Compras 5](#_Toc173270827)

[**Gestión de Cajeros** 5](#_Toc173270828)

[Usuario 4.1: Registrar Cajeros 5](#_Toc173270829)

[Usuario 4.2: Administrar Cajeros 6](#_Toc173270830)

[**Importación de Datos Iniciales** 6](#_Toc173270831)

[Usuario 5.1: Importar Datos desde CSV 6](#_Toc173270832)

[**Manual del código** 6](#_Toc173270833)

[**Aprendizaje** 11](#_Toc173270834)

[Conclusión 12](#_Toc173270835)

# **Estructura del Backlog**

## **Gestión de Productos**

## Productos 1.1: Agregar Productos

**Descripción**: Como administrador, quiero agregar nuevos productos para que se puedan vender en la ferretería.

**Criterios de Aceptación**:

Permitir ingresar el nombre, descripción, precio, y cantidad.

**Tareas**:

1. Crear modelo de producto.
2. Implementar lógica de agregar producto en el controlador.
3. Diseñar formulario para agregar productos.
4. Integrar formulario con el controlador.

## Productos 1.2: Editar Productos

**Descripción**: Como administrador, quiero editar la información de los productos existentes para mantener los datos actualizados.

**Criterios de Aceptación**:

Permitir modificar detalles de productos existentes.

**Tareas**:

1. Implementar lógica de edición en el controlador.
2. Diseñar formulario para editar productos.
3. Integrar formulario con el controlador.

## Productos 1.3: Eliminar Productos

**Descripción**: Como administrador, quiero eliminar productos obsoletos o incorrectos para mantener el inventario limpio.

**Criterios de Aceptación**:

Permitir la eliminación de productos.

**Tareas**:

1. Implementar lógica de eliminación en el controlador.
2. Integrar funcionalidad en la vista de productos.

#### **Gestión de Ventas (Lo realiza el cajero)**

## Usuario 2.1: Procesar Ventas

**Descripción**: Como cajero, quiero procesar ventas para registrar la compra de productos por parte de los clientes.

**Criterios de Aceptación**:

Permitir selección de productos, cantidades y método de pago.

**Tareas**:

1. Crear modelo de venta.
2. Implementar lógica de procesamiento de ventas.
3. Diseñar interfaz para procesar ventas.
4. Integrar interfaz con el controlador.

## Usuario 2.2: Generar Facturas

**Descripción**: Como cajero, quiero generar facturas para entregarlas a los clientes tras realizar una compra.

**Criterios de Aceptación**:

Incluir detalles de la venta y método de pago en la factura.

**Tareas**:

1. Implementar lógica de generación de facturas.
2. Diseñar plantilla para facturas.
3. Integrar generación de facturas en el flujo de ventas.

#### **Gestión de Clientes (El registro de clientes lo realiza el cajero)**

## Usuario 3.1: Registrar Clientes

**Descripción**: Como cajero, quiero registrar nuevos clientes para mantener su información y facilitar futuras ventas.

**Criterios de Aceptación**:

Permitir registrar nombre, contacto y dirección del cliente.

**Tareas**:

1. Crear modelo de cliente.
2. Implementar lógica de registro de clientes.
3. Diseñar formulario para registrar clientes.
4. Integrar formulario con el controlador.

## Usuario 3.2: Ver Historial de Compras

**Descripción**: Como cajero, quiero ver el historial de compras de los clientes para ofrecerles un mejor servicio.

**Criterios de Aceptación**:

Mostrar historial de compras al seleccionar un cliente.

**Tareas**:

* Implementar lógica para obtener historial de compras.
* Diseñar vista para mostrar historial de compras.
* Integrar vista con el controlador.

#### **Gestión de Cajeros**

## Usuario 4.1: Registrar Cajeros

**Descripción**: Como administrador, quiero registrar cajeros para que puedan operar el sistema de ventas.

**Criterios de Aceptación**:

Permitir ingresar detalles de los cajeros como nombre y datos de contacto.

**Tareas**:

1. Crear modelo de cajero.
2. Implementar lógica de registro de cajeros.
3. Diseñar formulario para registrar cajeros.
4. Integrar formulario con el controlador.

## Usuario 4.2: Administrar Cajeros

**Descripción**: Como administrador, quiero actualizar o eliminar la información de los cajeros.

**Criterios de Aceptación**:

Permitir editar y eliminar detalles de cajeros existentes.

**Tareas**:

1. Implementar lógica de edición y eliminación.
2. Diseñar formularios para actualizar y eliminar cajeros.
3. Integrar formularios con el controlador.

## **Importación de Datos Iniciales**

## Usuario 5.1: Importar Datos desde CSV

**Descripción**: Como desarrollador, quiero importar datos iniciales de productos, ventas, clientes, y cajeros desde archivos CSV para iniciar el sistema.

**Criterios de Aceptación**:

Lectura y carga correcta de datos desde CSV.

**Tareas**:

1. Implementar lógica de lectura de CSV.
2. Crear utilidades para cargar datos en el sistema.
3. Verificar la carga de datos.

### **Manual del código**

El código está basado en MVC. Consta de 3 partes para cada tab siendo el controlador el fond y el modelo.

Consta de 4 tabs cliente cajero producto y ventas. Siendo que al cada tab (cliente, cajero, producto y ventas. Mostrando cada uno su apartado correspondiente. Siendo esta primera imagen de la interfaz de cajero. En el se coloca la id del cajero, su nombre y sus apellido dos respectivos, junto con telefono y correo eletronico.

Tabla

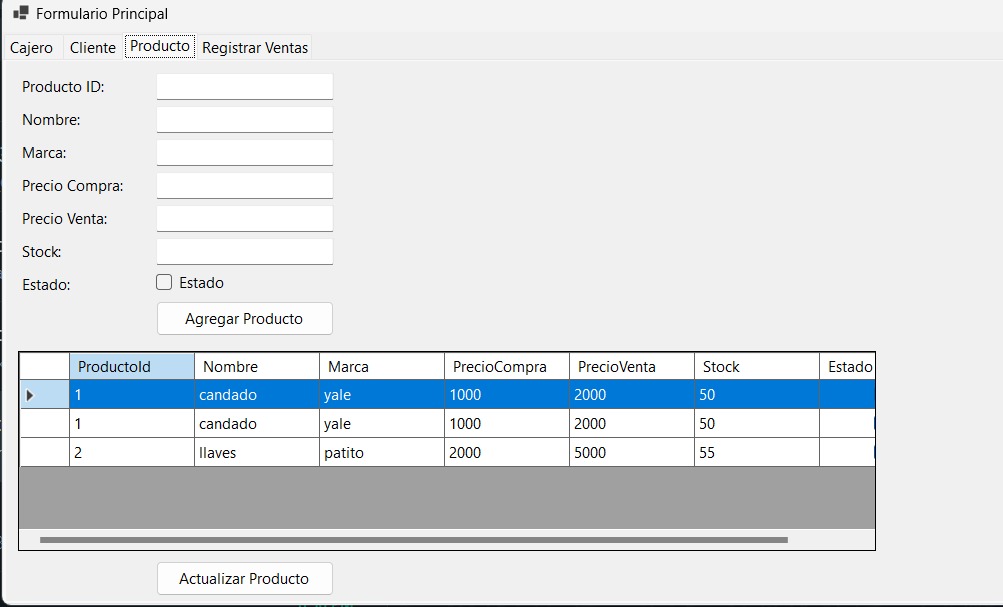
Descripción generada automáticamente

El segundo apartado siendo cliente. En este apartado es para el cliente en la que mantuvimos la misma interfaz de llenado que el cajero.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

En el tercer apartado de Productos podemos agregar los datos de los mismos con su respectivo ID que creamos manualmente, su nombre y marca además del precio de compra y venta y su Stock.



Y por ultimo el apartado el registro de ventas el Id es de la venta y la cantidad y su total. Así mismo con dos botones donde puedes agregar la venta con los datos agregados anteriormente y así mismo actualizar las ventas.

Interfaz de usuario gráfica

Descripción generada automáticamente

Muestra del código:

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente

### **Aprendizaje**

En este apartado se mostrará los errores obtenidos a la hora desarrollar el código en desarrollo y a la hora de desarrollar por los errores y por aprender a usar el framework, además de que nos faltó tiempo para terminar el tab desarrollar mejor la lógica de programación. Aquí evidencias de capturas de los errores que obtuvimos en el desarrollo.

Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamente

### **Conclusión**

En resumen, el desarrollo del Sistema de Ventas para Ferretería nos ha ayudado a aplicar los fundamentos de programación y diseño de sistemas de información utilizando un enfoque basado en modelo-vista-controlador (MVC). A pesar de algunos de los desafíos, como manejar errores de programación y aprender nuevos marcos, el equipo se aseguró de que la plataforma se desarrollara y cumpliera su propósito de facilitar las operaciones en una ferretería. Más que habilidades técnicas, este proyecto dotó a los estudiantes de experiencia en resolución de problemas y colaboración en un entorno de desarrollo de software.