

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](#_Toc171259252)

[**Gestión de Productos** 3](#_Toc171259253)

[Productos 1.1: Agregar Productos 3](#_Toc171259254)

[Productos 1.2: Editar Productos 3](#_Toc171259255)

[Productos 1.3: Eliminar Productos 3](#_Toc171259256)

[**Gestión de Ventas (Lo realiza el cajero)** 4](#_Toc171259257)

[Usuario 2.1: Procesar Ventas 4](#_Toc171259258)

[Usuario 2.2: Generar Facturas 4](#_Toc171259259)

[**Gestión de Clientes (El registro de clientes lo realiza el cajero)** 4](#_Toc171259260)

[Usuario 3.1: Registrar Clientes 4](#_Toc171259261)

[Usuario 3.2: Ver Historial de Compras 5](#_Toc171259262)

[**Gestión de Cajeros** 5](#_Toc171259263)

[Usuario 4.1: Registrar Cajeros 5](#_Toc171259264)

[Usuario 4.2: Administrar Cajeros 5](#_Toc171259265)

[**Importación de Datos Iniciales** 6](#_Toc171259266)

[Usuario 5.1: Importar Datos desde CSV 6](#_Toc171259267)

[**Ejemplo de Backlog en Jira** 6](#_Toc171259268)

# **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.

(LA SIGUIENTE PARTE ES DE OSCAR)

### **Ejemplo de Backlog en Jira**

A continuación, se muestra cómo estructurar las historias y tareas en Jira:

#### **Épicas**

* **[PRO] Gestión de Productos**
* **[VEN] Gestión de Ventas**
* **[CLI] Gestión de Clientes**
* **[CAJ] Gestión de Cajeros**
* **[CFG] Configuración Inicial del Sistema**

#### **Historias de Usuario y Tareas**

* **[PRO-1] Agregar Productos**
  + **Subtarea**: Crear modelo de producto (4h)
  + **Subtarea**: Implementar lógica de agregar producto (6h)
  + **Subtarea**: Diseñar formulario para agregar productos (4h)
  + **Subtarea**: Integrar formulario con el controlador (4h)
* **[PRO-2] Editar Productos**
  + **Subtarea**: Implementar lógica de edición (4h)
  + **Subtarea**: Diseñar formulario para editar productos (4h)
  + **Subtarea**: Integrar formulario con el controlador (4h)
* **[VEN-1] Procesar Ventas**
  + **Subtarea**: Crear modelo de venta (4h)
  + **Subtarea**: Implementar lógica de procesamiento de ventas (6h)
  + **Subtarea**: Diseñar interfaz para procesar ventas (4h)
  + **Subtarea**: Integrar interfaz con el controlador (4h)
* **[CLI-1] Registrar Clientes**
  + **Subtarea**: Crear modelo de cliente (4h)
  + **Subtarea**: Implementar lógica de registro de clientes (6h)
  + **Subtarea**: Diseñar formulario para registrar clientes (4h)
  + **Subtarea**: Integrar formulario con el controlador (4h)
* **[CAJ-1] Registrar Cajeros**
  + **Subtarea**: Crear modelo de cajero (4h)
  + **Subtarea**: Implementar lógica de registro de cajeros (4h)
  + **Subtarea**: Diseñar formulario para registrar cajeros (4h)
  + **Subtarea**: Integrar formulario con el controlador (4h)
* **[CFG-1] Importar Datos desde CSV**
  + **Subtarea**: Implementar lógica de lectura de CSV (4h)
  + **Subtarea**: Crear utilidades para cargar datos en el sistema (4h)
  + **Subtarea**: Verificar la carga de datos (4h)