Desarrollo de un Sistema de Registro de Venta de Productos

*Desarrollo de Proyectos Empresariales*

Autor: Farid Yasser Romero Grabiel

Año: 2025

# 1. GENERALIDADES DEL PROYECTO

## 1.1. Descripción del caso

Muchas PYMEs gestionan sus ventas manualmente o con hojas de cálculo, lo que provoca errores, pérdida de datos y poca trazabilidad. Este proyecto propone un sistema de registro de venta de productos que gestione inventario, clientes, vendedores y transacciones, con énfasis en integridad de datos y usabilidad.

## 1.2. Objetivo

### 1.2.1. Objetivo general

* Desarrollar un sistema que permita administrar productos, clientes, vendedores y ventas, optimizando procesos y garantizando integridad de la información.

### 1.2.2. Objetivos específicos

* Registrar, editar y eliminar productos, categorías y clientes.
* Controlar el stock en base a las ventas.
* Registrar vendedores y su asistencia vía login.
* Ejecutar ventas con cálculo automático de totales.
* Reportes exportables.
* Ofrecer una interfaz clara e intuitiva.

## 1.3. Requerimientos

### 1.3.1. Requerimientos Funcionales (RF)

* RF1: El sistema debe permitir el registro de productos.
* RF2: Los vendedores registran su asistencia mediante login o código único.
* RF3: Registrar y actualizar datos de clientes.
* RF4: Generar boletas y facturas
* RF5: Revisión de productos más vendidos.
* RF6: Exportar reportes a PDF y Excel.

### 1.3.2. Requerimientos Específicos (RE)

* RE1: BD seguro con respaldo automático.
* RE2: Registro de productos en oferta.
* RE3: Interfaz amigable y accesible desde móviles (línea evolutiva).
* RE4: Alertas de bajo stock.

# 2. ANÁLISIS DEL SISTEMA

## 2.1. Actores y Casos de Uso

### 2.1.1. Actores

* Administrador: gestiona productos, categorías, clientes, vendedores y usuarios.
* Vendedor: registra ventas, consulta historial y marca asistencia.
* Cliente: entidad pasiva asociada a ventas.

### 2.1.2. Casos de uso principales

* Gestionar productos y categorías (CRUD).
* Gestionar clientes (CRUD).
* Gestionar vendedores y usuarios del sistema.
* Iniciar sesión y registrar asistencia.
* Registrar venta (carrito, totales, descuento de stock).
* Consultar historial por fecha, cliente o vendedor.
* Generar reportes y exportaciones.

## 2.2. Arquitectura de Desarrollo Seleccionada

Se adopta una arquitectura Cliente–Servidor, una aplicación cliente (escritorio/web) que interactúa con un servidor de base de datos relacional. Para el MVP se considera una variante monolítica de escritorio con capa de datos local para producción, despliegue en red con servidor de BD central.

## 2.3. Arquitectura Lógica del Sistema

* Capa de Presentación: interfaces de usuario (pantallas) para productos, clientes, ventas, reportes y autenticación.
* Capa de Lógica de Negocio: reglas de stock, cálculo de totales, validaciones, emisión de documentos y control de asistencia.
* Capa de Acceso a Datos: repositorios/DAO para entidades y consultas de reportes, con integridad referencial.

## 2.4. Patrón Arquitectónico

Se propone utilizar el patrón **MVC (Modelo–Vista–Controlador)** como base de la arquitectura.

## 2.5. Diagramas y Modelos

* Diagrama de Casos de Uso: Administrador (gestión total)
* Vendedor (ventas/historial/asistencia), Cliente (consultado).
* Diagrama de Clases: Persona (abstracta) → Cliente/Vendedor; Producto; Categoría; Venta; Detalle Venta; Usuario.
* Diagrama de Arquitectura General: Presentación → Lógica → Datos (BD relacional).
* Diagrama de Secuencia (Registrar Venta): Vendedor → UI → Controlador → Repositorios → BD: retorno de totales y confirmación.

# 3. DISEÑO DEL SISTEMA

## 3.1. Modelo Lógico de la Base de Datos

* Cliente (DNI, nombre, dirección, fecha nacimiento, teléfono, correo).
* Vendedor (id Vendedor, nombre, usuario, contraseña, datos personales).
* Categoría (id categoría, nombre, descripción).
* Producto (id producto, nombre, descripción, precio, stock, id categoría).
* Venta (id venta, fecha, DNI Cliente, id Vendedor, total).
* Detalle Venta (id detalle, id venta, id producto, cantidad, precio unitario, subtotal).
* Usuario/Vendedor: unificado para control de acceso.

## 3.2. Reglas de Integridad

* Stock no negativo, validación de existencia de cliente/producto al vender.
* Usuarios con contraseña almacenada: sesiones registran asistencia.

## 3.3. Diseño de Interfaces

* Inicio de sesión (registro de asistencia).
* Dashboard por rol (Administrador/Vendedor).
* Gestión de productos, categorías y clientes.
* Registro de ventas con carrito y totales.
* Historial y reportes (filtro por fecha/usuario).
* Usuarios/Vendedores (alta/baja/modificación).

## 3.4. Servicios por Componente

* LoginService: login, hash de contraseñas, registro de asistencia.
* ProductoService: control de stock, ofertas y umbrales de alerta.
* ClienteService: validaciones de DNI/correo.
* VentaService: carrito, totales, persistencia de venta y detalle, emisión de comprobante.
* ReporteService: ventas por día/mes/vendedor, top productos, exportación CSV/PDF/Excel.
* NotificaciónService: alertas de bajo stock.

# 4. PRUEBAS

## 4.1. Unitarias

* Auth: login válido/ inválido:correcto.
* Producto: creación/edición: stock no negativo.
* Venta: cálculo de totales: validación de stock - detalle consistente.

## 4.2. Integración

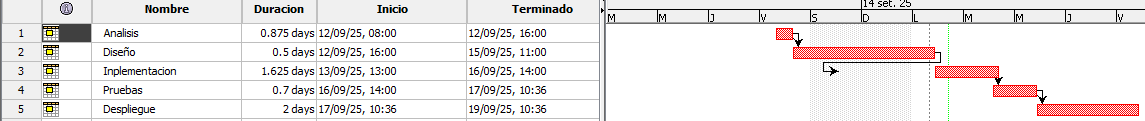
* Flujo completo de venta: login → seleccionar cliente → añadir productos → confirmar → descontar stock → registrar detalle.
* Reportes: ranking de productos y ventas por periodo.
* Roles: permisos de Administrador vs Vendedor.

# 5. RIESGOS Y MITIGACIONES

* Pérdida de datos: backup automático y restauración.
* Fallas de seguridad: control de roles.
* Resistencia al cambio: capacitación breve y manual de usuario.
* Conectividad (si BD remoto): modo offline con cola de sincronización.
* Escalabilidad: plan de migración a arquitectura web (API REST + frontend).

# 7. CRONOGRAMA

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Duración | Inicio | Terminado |
| Análisis | 0.875 days | 12/09/25, 08:00 | 12/09/25, 16:00 |
| Diseño | 0.5 days | 12/09/25, 16:00 | 15/09/25, 11:00 |
| Implementación | 1.625 days | 13/09/25, 13:00 | 16/09/25, 14:00 |
| Pruebas | 0.7 days | 16/09/25, 14:00 | 17/09/25, 10:36 |
| Despliegue | 2 days | 17/09/25, 10:36 | 19/09/25, 10:36 |



# 8. CONCLUSIONES

* El sistema propuesto aborda trazabilidad y control de inventario con una arquitectura simple y escalable.
* El patrón MVC separa responsabilidades y facilita.
* Los requerimientos críticos (ventas, stock, reportes) quedan cubiertos en el diseño.

# 9. RECOMENDACIONES

* Implementar backups programados y auditoría de cambios.
* Migrar a BD centralizada (MySQL/PostgreSQL) para multiusuario concurrente.
* Agregar exportación a PDF/Excel y alertas configurables de stock.
* Plan de formación breve para vendedores y administradores.