Imagen que contiene Logotipo

El contenido generado por IA puede ser incorrecto.

Implementación de Software para Tienda de Calzado *FLY RP*

**Integrantes:**

Damián Rodriguez

Juan Carlos Peña

Carrera: Tecnicatura en Desarrollo de Software  
Materia: Programación Orientada a Objeto  
Docente: Emir García  
Año: 2025

Sistema de Gestión - ***Fly RP***

# 1. Introducción

Como parte de un trabajo integrador en la materia de Programación Orientada a Objetos, como equipo hemos pensado en el Desarrollo, diseñado e implementado un sistema de gestión integral para una tienda de calzados que Denominamos “Fly RP”.

El sistema se desarrolla en C# utilizando Windows Forms con arquitectura MDI (Interfaz de Documentos Múltiples) y MySQL como sistema gestor de base de datos. El objetivo es optimizar la gestión diaria de los procesos clave del negocio, tales como:

- Ventas  
- Gestión de productos  
- Clientes  
- Usuarios y empleados  
- Stock  
- Administración (Accesos)

El diseño se orienta hacia una arquitectura escalable, que sea segura y mantenible, aplicando buenas prácticas de POO.

# 2. Historia de Usuario General

Haciendo una evaluación y analizando que requerimientos necesita la tienda Fly RP, se busca tener un sistema que permita gestionar productos, ventas, caja, clientes, usuarios y configuraciones, para poder administrar eficientemente todas las operaciones diarias de la tienda y tomar decisiones basadas en reportes precisos, con visión a crecimiento y nuevas incorporaciones de módulos funcionales.

# 3. Requerimientos Funcionales

El sistema debe cumplir con las siguientes funcionalidades:

- Login de usuario con validación de credenciales  
- Control de acceso por rol: Administrador, Vendedor, Cajero  
- Registro, edición y eliminación de productos (CRUD)  
- Registro de ventas con detalle de productos y cliente  
- Descuento automático de stock al concretar una venta  
- Gestión de clientes frecuentes  
- Control de caja diaria: apertura, cierre, ingresos y egresos  
- Generación de reportes de ventas, stock y caja  
- Configuración de parámetros generales del sistema  
- Registro de logs para auditoría de acciones críticas

# 4. Requerimientos No Funcionales

- Seguridad: contraseñas encriptadas, control de accesos por roles  
- Usabilidad: interfaz amigable para usuarios no técnicos  
- Rendimiento: tiempo de respuesta menor a 3 segundos  
- Portabilidad: ejecutable en diferentes equipos con Windows  
- Mantenibilidad: modularidad del código para futuras mejoras  
- Escalabilidad: manejo eficiente de volúmenes crecientes de datos  
- Disponibilidad: sistema disponible al menos el 95% del tiempo  
- Tolerancia a errores: gestión de errores clara sin cierre inesperado  
- Integridad: sin duplicación de datos claves como productos o usuarios  
- Respaldo: posibilidad de backup periódico de la base de datos

# 5. Modelo de Base de Datos

El sistema utiliza una base de datos relacional en MySQL. Incluye tablas como usuarios, roles, productos, clientes, ventas, detalle de ventas, caja, movimientos de caja y configuración general.

Se contempla la opción de migrar a SQL Server en caso de ser necesario.

# 6. Roles del Sistema

- Administrador: acceso total al sistema, configuración, reportes, caja y gestión de usuarios  
- Vendedor: carga de ventas, consulta de productos, clientes y stock  
- Cajero: registro de movimientos de caja, cierre diario  
- Almacenero: gestión de inventario, actualización de stock

# 7. Implementación del MDI en C#

La estructura del sistema se basa en un formulario principal MDI (`frmPrincipal`) desde el cual se accede a los distintos módulos del sistema. Al iniciar sesión correctamente desde `frmLogin`, se habilitan accesos según el rol.

Módulos incluidos:  
- frmProductos  
- frmClientes  
- frmVentas  
- frmUsuarios  
- frmCaja (Para incorporar mas Adelante)   
- frmConfiguracion (Para incorporar mas Adelante)

# 8. Ejemplo CRUD - Productos

Se implementaron las funciones básicas:

- INSERT: Agregar nuevos productos  
- SELECT: Consultar productos  
- UPDATE: Modificar productos existentes  
- DELETE: Eliminar productos

Todas estas funciones se realizan mediante sentencias SQL

# 9. Casos de Prueba Principales

- CT001: Login correcto  
- CT002: Login incorrecto  
- CT003: Agregar producto  
- CT004: Buscar producto  
- CT005: Armado de pedido  
- CT006: Registro de cliente  
- CT007: Confirmación y guardado de venta

# 10. Diagrama Entidad-Relación (Resumen MER)

- USUARIOS (id\_usuario, nombre, email, contraseña, id\_rol)  
 └──▶ ROLES (id\_rol, nombre)  
- VENTAS (id\_venta, fecha, id\_cliente, id\_usuario, total)  
 ├──▶ CLIENTES (id\_cliente, nombre, teléfono, email)  
 ├──▶ USUARIOS (id\_usuario)  
- DETALLE\_VENTA (id\_detalle, id\_venta, id\_producto, cantidad, precio\_unit)  
 └──▶ PRODUCTOS (id\_producto, nombre, talla, color, marca, stock, precio\_venta)  
- CAJA (id\_caja, fecha\_apertura, fecha\_cierre, monto\_inicial, monto\_final, id\_usuario)  
- MOVIMIENTOS\_CAJA (id\_movimiento, id\_caja, tipo, descripción, monto, fecha)  
- CONFIGURACION (id\_config, nombre\_tienda, moneda, impuesto, logo\_url)

# 11. Casos de Uso Destacados

Caso de Uso: Registrar Venta  
Actor: Vendedor  
Precondición: El vendedor ha iniciado sesión

Flujo principal:  
1. El vendedor selecciona al cliente  
2. Busca los productos y los añade al carrito  
3. El sistema calcula el total  
4. El vendedor confirma la venta  
5. El sistema guarda la venta, descuenta el stock y actualiza la caja

# 12. Diseño de Interfaz (Pantallas)

Pantallas desarrolladas:  
- Login  
- Menú principal por rol  
- Gestión de productos  
- Gestión de clientes  
- Ventas (buscador y armado de pedidos)  
- Gestión de caja  
- Configuración general  
- Reportes

# 13. Anexos

Ingreso de Usuarios a la BD En My Sql

INSERT INTO usuarios (nombre, usuario, contraseña, rol) VALUES

('Peña Juan Carlos', 'pjuan', 'admin123', 'admin'),

('Rodriguez Damian', 'rdamian', 'cajero123', 'cajero'),

('Cosme Fulano', 'cfulano', 'vendedor1', 'vendedor'),

('Armando Paredes', 'aparedes', 'vendedor2', 'vendedor');  
 **A screenshot of a computer

AI-generated content may be incorrect.**

Ya tenemos los usuarios y los artículos cargados a la base de datos tbm 1 cliente de prueba

**A screenshot of a computer

AI-generated content may be incorrect.**

El inicio de sesión ya está programado valida el ingreso de los usuarios ya existentes,

**A computer screen shot of a login form

AI-generated content may be incorrect.**Solo falta unir con lo creado para la visualización del vendedor.

Ya en el diseño de vista del vendedor tenemos el grid de los artículos que carga estos desde la base de datos con el botón agregar o doble click en la fila se pasan al grid de pedido también está el cuadro de búsqueda funcional mostrando los resultados de esta en el grid en el cuadro del total se va mostrando la suma de los artículos dentro del grid de pedido, se modificó del diseño original los botones y el orden para ser más entendible.

**A screenshot of a computer

AI-generated content may be incorrect.**

A computer screen shot of a computer screen

AI-generated content may be incorrect.La siguiente pantalla una vez finalizado el armado del pedido, con el botón siguiente se trasladará a la sección de carga de datos del comprador, esta solución carga los datos correctamente, pero por falta de acoplamiento falta el enlace de botones con la pantalla anterior y la visual del vendedor asignado al puesto, esta parte se emulo con datos de la base.

Estos son algunos modelos analizados como alternativos en Base de datos y portal de la tienda

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.