Gestor Tienda

Sistema de Gestión de Información con JDBC y Arquitectura MVC

Proyecto Unidad 2 • Acceso a Datos

Definición del Problema y Objetivos

⚠El Problema

Un negocio de venta de productos quiere digitalizarse y para ello necesita dejar atras todo lo manual para ahorrar tiempo y trabajo. Es decir buscamos crear una aplicación que realize este trabajo y gestione los productos, proveedores, clientes y ventas

🎯La Solución

Una aplicación de escritorio robusta que permite operaciones CRUD completas, garantizando la persistencia de datos en base de datos relacional y resolviendo la impedancia objeto-relacional.

Tecnológias

Java

Lógica de negocio robusta y orientada a objetos.

JavaFX

Interfaz gráfica moderna con FXML y CSS.

JDBC & MySQL

Conectividad y persistencia de datos relacionales.

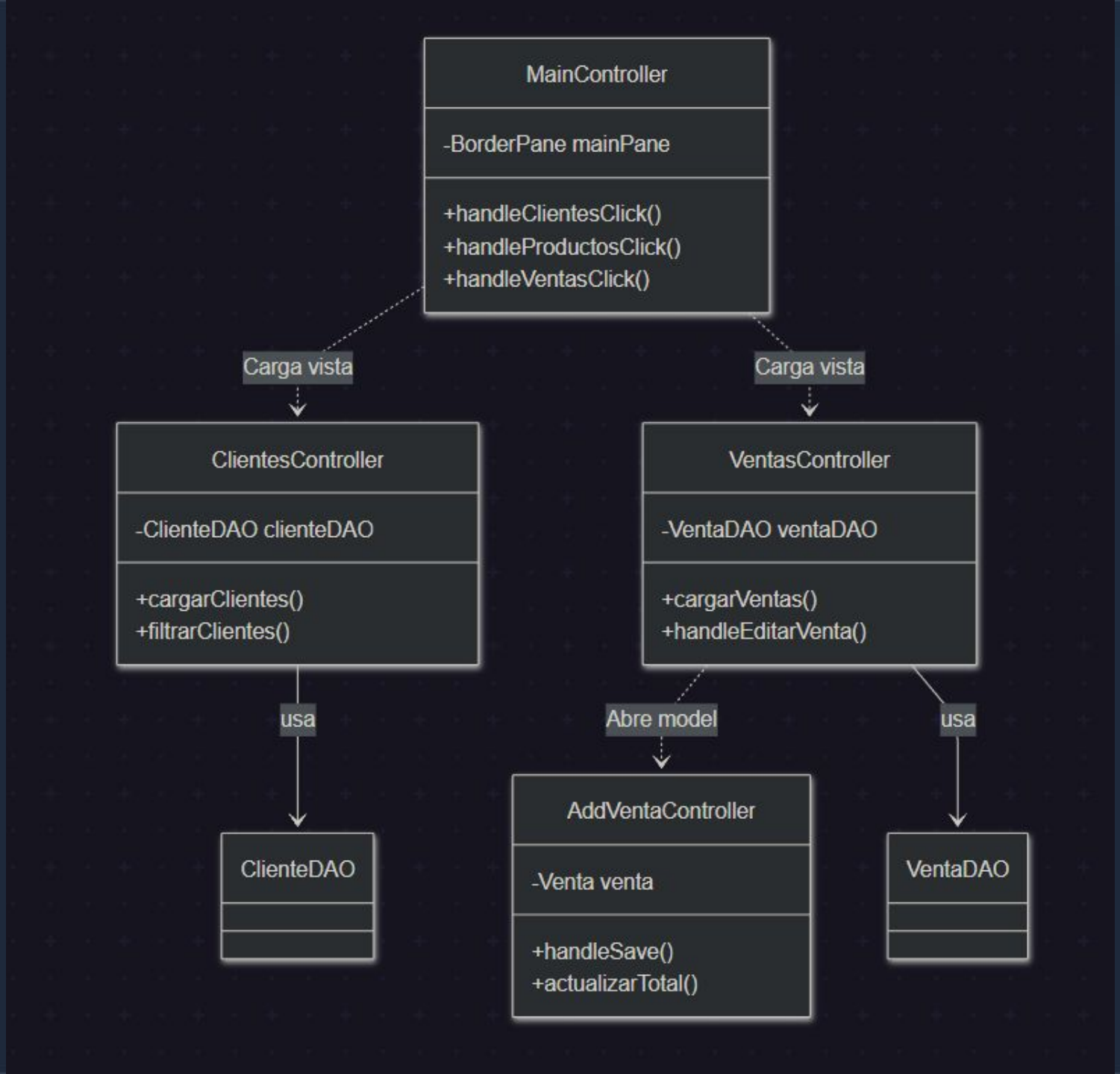
Maven

Gestión de dependencias y construcción del proyecto.

Arquitectura MVC

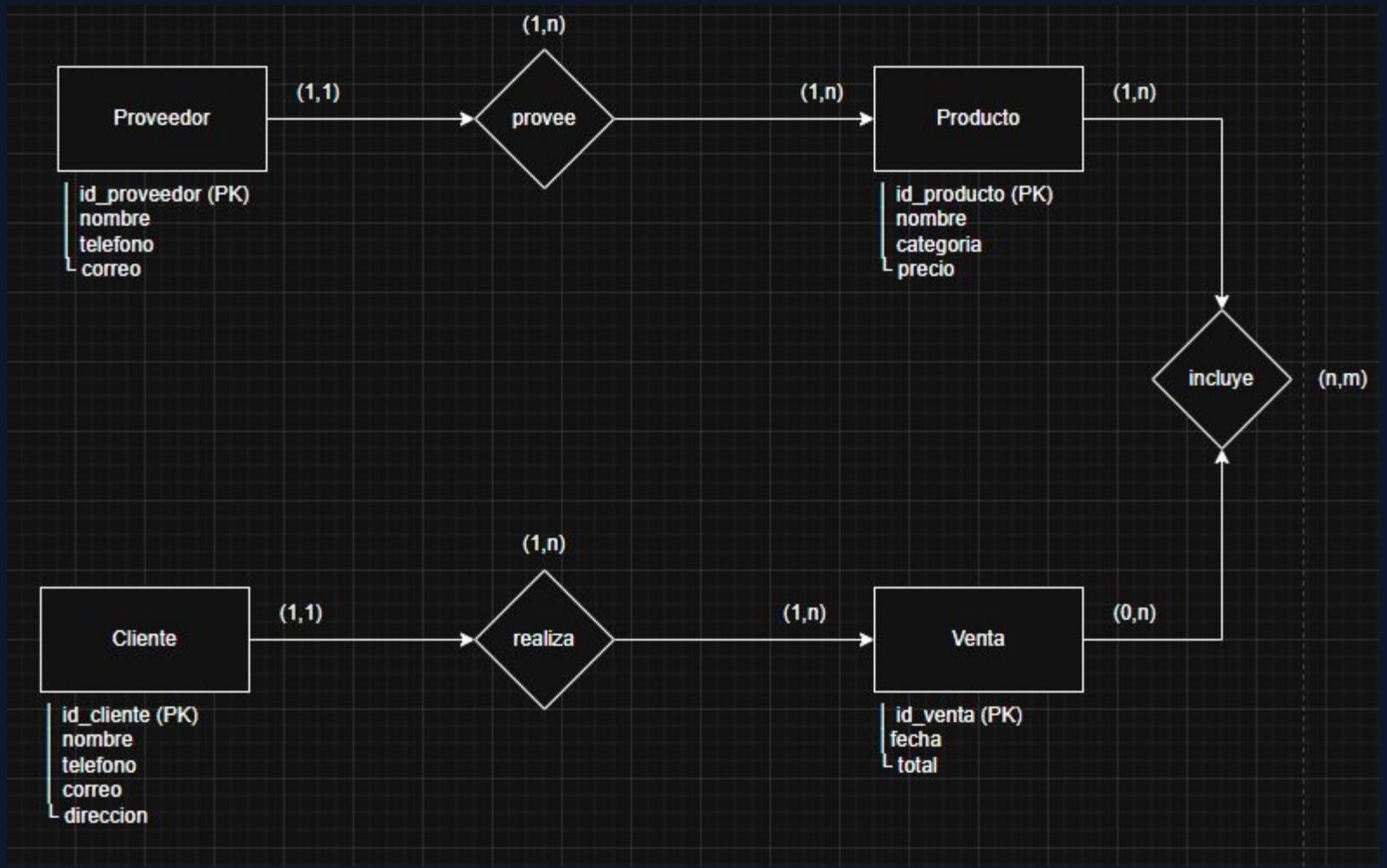
Implementación del patrón **Modelo-Vista-Controlador** para desacoplar responsabilidades.

- **MainController:** Orquesta la navegación principal.
- **Controladores:** Gestionan lógica específica y usan DAOs.
- **Vistas:** FXML cargados dinámicamente.



Modelo de Datos

- **Entidades:** Cliente, Producto, Proveedor y venta
- **Transaccional:** Tabla en medio entre producto y venta, llamada detalle_venta
- **Relaciones:** 1:N y N:M resueltas.



Gestión de Conexiones

Patrón Singleton

La clase ConnectionFactory asegura que solo exista una instancia del manejador de conexiones, optimizando recursos y evitando múltiples hilos de conexión innecesarios.

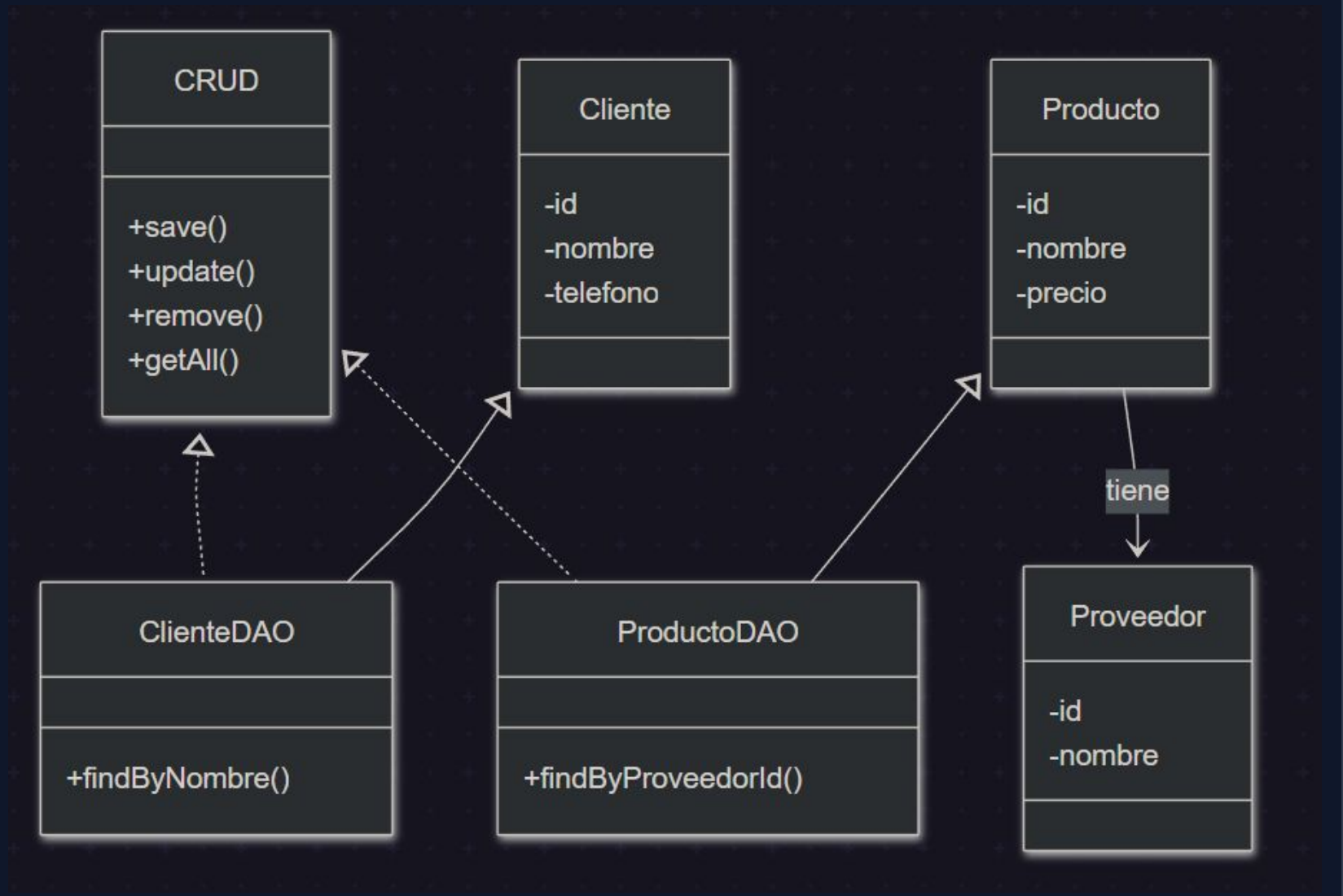
Configuración Externa

Uso de config.properties para definir credenciales. Permite cambiar entre **MySQL** y **H2** (Embebida) sin recompilar el código.

Patrón DAO

Abstracción de la capa de persistencia mediante herencia e interfaces.

- Interfaz Genérica **CRUD**.
- Los DAOs heredan de la Entidad y extienden la funcionalidad.
- Desacople total de la lógica SQL.



Lógica Transaccional: Ventas

1

Crear Venta

Se genera la cabecera con el Cliente y Fecha actual.

2

Detalles

Se guardan los productos línea a línea.

3

Stock

Se descuenta el stock automáticamente en la BD.

Conclusiones

Retos Superados

Gestión de la impedancia objeto-relacional mapeando ResultSets manualmente y mantenimiento de la integridad referencial al borrar datos vinculados.

Futuras Mejoras

Implementación de Hibernate/JPA para automatizar el mapeo, sistema de Login con roles y generación de facturas en PDF.

DEMO TÉCNICA

Gracias por su atención.

Gestor Tienda v1.0