Manual Tecnico

Nombre del sistema: Prototipo de tienda
b. Descripción y delimitación del sistema
Este sistema tiene la idea de ser una plantilla para una tienda, ser algo simple que pueda procesar comprar de forma simples (pero sin incluir los pagos ni las formas de pago)
Objetivo general
Ser la base de una tienda online
d. Objetivos específicos
Servir como base para implementar una tienda online de los productos de la UACH

e. Descripción de tipos de usuarios
Unicamente tendra usuarios para comprar los productos que se mostrarán en la tienda
f. Entorno operativo del sistema
Servidor local
Diagrama casos de uso

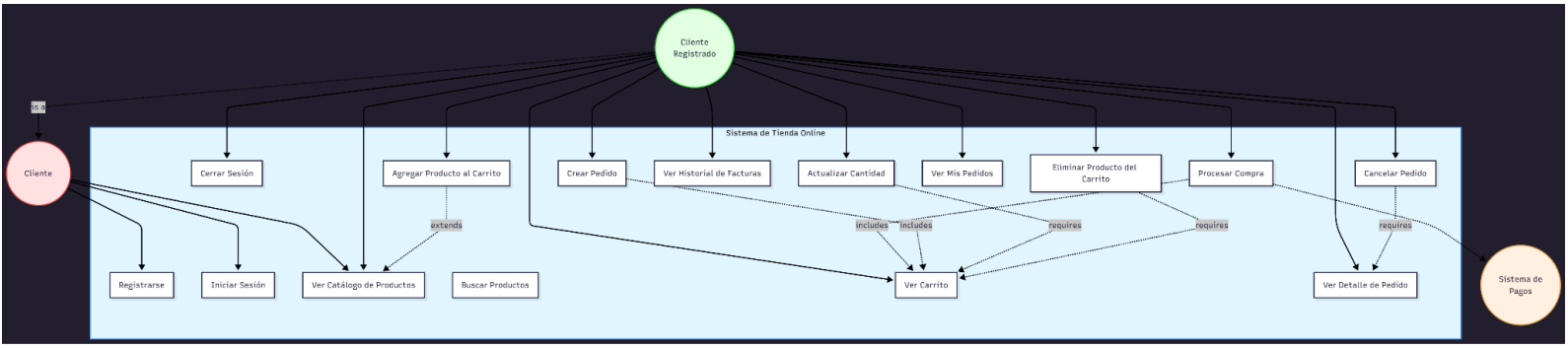


Diagrama de clases

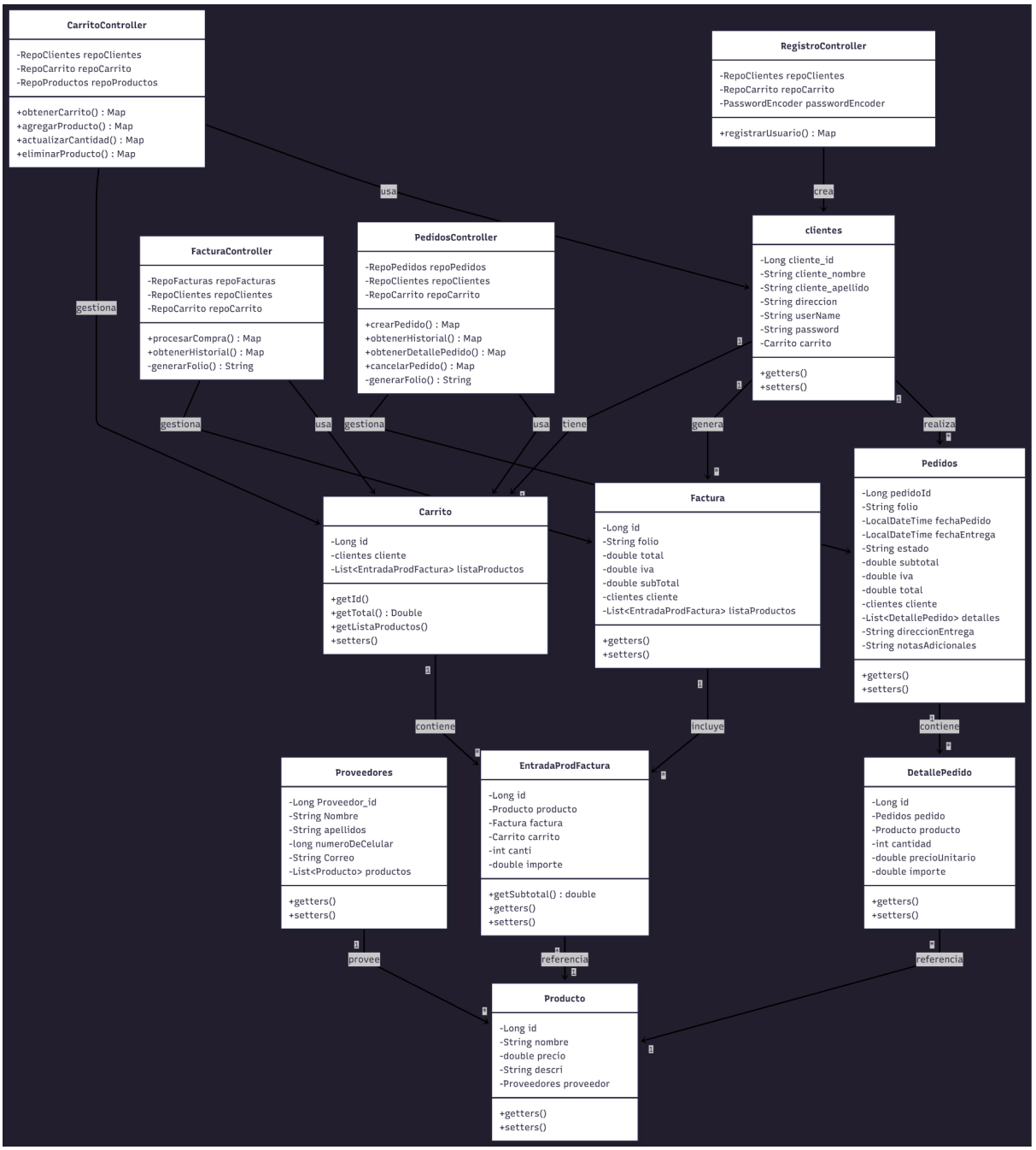
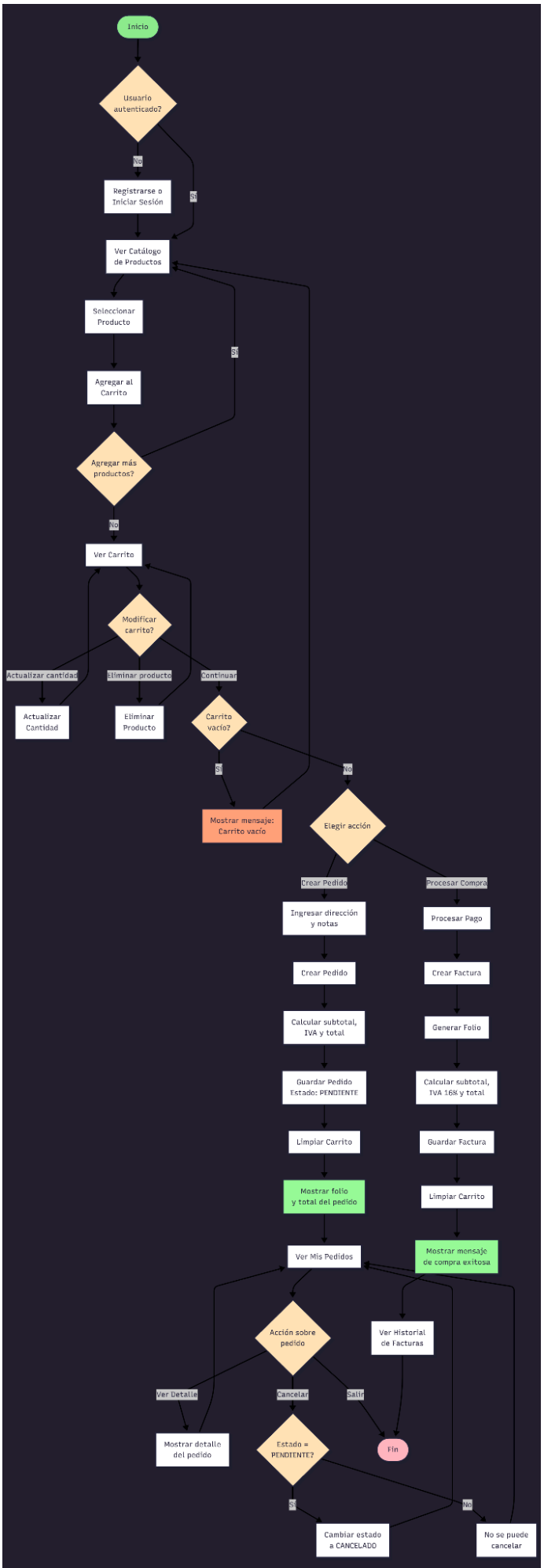


Diagrama de actividad


Inicio
Usuario autenticado?
No
Sí
Registrarse o Iniciar Sesión
Ver Catálogo de Productos
Seleccionar Producto
Sí
Agregar al Carrito
Agregar más productos?
No
Ver Carrito
Modificar carrito?
Actualizar cantidad
Eliminar producto
Continuar
Actualizar Cantidad
Eliminar Producto
Carrito vacío?
Sí
No
Mostrar mensaje: Carrito vacío
Elegir acción
Crear Pedido
Procesar Compra
Ingresar dirección y notas
Procesar Pago
Crear Pedido
Crear Factura
Calcular subtotal, IVA y total
Generar Folio
Guardar Pedido Estado: PENDIENTE
Calcular subtotal, IVA 16% y total
Limpiar Carrito
Guardar Factura
Mostrar folio y total del pedido
Limpiar Carrito
Ver Mis Pedidos
Mostrar mensaje de compra exitosa
Acción sobre pedido
Ver Historial de Facturas
Ver Detalle
Cancelar
Salir
Mostrar detalle del pedido
Estado = PENDIENTE?
Fin
Sí
No
Cambiar estado a CANCELADO
No se puede cancelar

Requerimientos del Sistema de Tienda Local

1. REQUERIMIENTOS FUNCIONALES

1.1 Gestión de Usuarios

RF-01: Registro de Clientes

- El sistema debe permitir que nuevos clientes se registren proporcionando:
    - Nombre y apellido
    - Dirección de entrega
    - Nombre de usuario
    - Contraseña
- El sistema debe validar que el nombre de usuario sea único
- El sistema debe encriptar las contraseñas antes de almacenarlas

RF-02: Inicio de Sesión

- El sistema debe permitir a los clientes autenticarse con usuario y contraseña
- El sistema debe mantener la sesión activa mientras el cliente navega

RF-03: Cierre de Sesión

- El sistema debe permitir a los clientes cerrar su sesión de forma segura

1.2 Gestión de Productos

RF-04: Visualización del Catálogo

- El sistema debe mostrar todos los productos disponibles con:
    - Nombre del producto
    - Precio
    - Descripción breve
- El catálogo debe ser visible sin necesidad de autenticación

RF-05: Información Detallada de Productos

- Cada producto debe mostrar su precio en formato monetario
- Los productos deben tener una descripción opcional

1.3 Gestión del Carrito de Compras

RF-06: Agregar Productos al Carrito

- El sistema debe permitir agregar productos al carrito de compras
- Si el producto ya existe en el carrito, debe incrementar la cantidad
- Solo clientes autenticados pueden agregar productos al carrito

RF-07: Visualizar Carrito

- El sistema debe mostrar todos los productos en el carrito con:
    - Nombre del producto
    - Precio unitario
    - Cantidad
    - Subtotal por producto
- El sistema debe calcular y mostrar:
    - Subtotal general
    - IVA (16%)
    - Total a pagar

RF-08: Modificar Cantidad de Productos

- El sistema debe permitir aumentar o disminuir la cantidad de cada producto
- Si la cantidad llega a cero, el producto debe eliminarse automáticamente

RF-09: Eliminar Productos del Carrito

- El sistema debe permitir eliminar productos individuales del carrito

1.4 Gestión de Pedidos

RF-10: Crear Pedido

- El sistema debe permitir crear un pedido desde el carrito de compras
- El sistema debe solicitar (opcional):
    - Dirección de entrega (por defecto usa la del cliente)
    - Notas adicionales
- El sistema debe generar un folio único con formato "PED-AAAAMMDDHHMMSS"
- El pedido debe crearse con estado "PENDIENTE"
- El sistema debe calcular subtotal, IVA (16%) y total
- Al crear el pedido, el carrito debe vaciarse

RF-11: Consultar Historial de Pedidos

- El sistema debe mostrar todos los pedidos del cliente ordenados por fecha (más recientes primero)
- Cada pedido debe mostrar:
    - Folio
    - Fecha y hora
    - Estado
    - Total
    - Cantidad de productos

RF-12: Ver Detalle de Pedido

- El sistema debe mostrar información completa del pedido:
    - Folio, fecha, estado
    - Dirección de entrega
    - Notas adicionales
    - Lista de productos con cantidades y precios
    - Subtotal, IVA y total

RF-13: Cancelar Pedido

- El sistema debe permitir cancelar pedidos solo si están en estado "PENDIENTE"
- No se pueden cancelar pedidos en estado "ENTREGADO" o ya "CANCELADO"

1.5 Gestión de Facturas (Compras)

RF-14: Procesar Compra

- El sistema debe permitir procesar una compra desde el carrito
- El sistema debe generar un folio único con formato "FAC-AAAAMMDDHHMMSS"
- El sistema debe calcular subtotal, IVA (16%) y total
- Al procesar la compra, el carrito debe vaciarse
- El sistema debe guardar la factura asociada al cliente

RF-15: Consultar Historial de Facturas

- El sistema debe mostrar todas las facturas generadas por el cliente
- Cada factura debe mostrar su folio y total

---

2. REQUERIMIENTOS NO FUNCIONALES

2.1 Usabilidad

RNF-01: Interfaz Intuitiva

- La interfaz debe ser simple y fácil de usar para usuarios con conocimientos básicos de navegación web

RNF-03: Feedback Visual

- El sistema debe proporcionar retroalimentación visual inmediata cuando:
    - Se actualiza el contador del carrito

2.2 Rendimiento

RNF-04: Tiempo de Respuesta

- Las operaciones básicas (agregar al carrito, ver productos) deben completarse en menos de 2 segundos
- La carga inicial del catálogo debe completarse en menos de 3 segundos

RNF-05: Capacidad

- El catálogo debe poder gestionar hasta 1000 productos sin degradación notable del rendimiento

2.3 Seguridad

RNF-06: Autenticación

- El sistema debe utilizar Spring Security para la gestión de autenticación

RNF-07: Encriptación de Contraseñas

- Las contraseñas deben almacenarse encriptadas utilizando BCrypt
- Nunca se deben mostrar contraseñas en texto plano

RNF-08: Autorización

- Solo los usuarios autenticados pueden acceder a:
    - Carrito de compras
    - Historial de pedidos
    - Historial de facturas
- Las páginas públicas (catálogo, registro, login) deben ser accesibles sin autenticación

RNF-09: Protección de Datos

- Cada cliente solo debe poder ver sus propios pedidos y facturas

- El sistema debe validar que un cliente no pueda acceder a información de otros clientes

2.4 Mantenibilidad

RNF-10: Arquitectura en Capas

- El sistema debe seguir el patrón MVC (Modelo-Vista-Controlador)
- La lógica de negocio debe estar separada de la capa de presentación

RNF-11: Código Documentado

- El código debe seguir convenciones de nomenclatura estándar de Java
- Las clases y métodos complejos deben estar documentados

2.5 Disponibilidad

RNF-12: Operación Continua

- El sistema debe estar disponible 24/7 para consultas del catálogo
- Se permite una ventana de mantenimiento semanal de máximo 2 horas

RNF-13: Manejo de Errores

- El sistema debe capturar y registrar excepciones
- Los errores deben mostrarse al usuario de forma amigable sin exponer detalles técnicos

2.6 Compatibilidad

RNF-14: Navegadores Web

- El sistema debe ser compatible con:
    - Google Chrome
    - Mozilla Firefox
    - Safari
    - Microsoft Edge

RNF-15: Base de Datos

- El sistema debe utilizar Oracle como gestor de base de datos
- La conexión debe realizarse mediante JDBC

2.7 Portabilidad

RNF-17: Independencia de Plataforma

- El sistema debe poder ejecutarse en cualquier sistema operativo que soporte Java 17 o superior
- El sistema debe ser independiente del servidor de aplicaciones (embebido con Spring Boot)

2.9 Aspectos Legales

RNF-18: Cálculo de Impuestos

- El IVA debe calcularse al 16% según la normativa fiscal vigente
- El sistema debe mostrar claramente el desglose de subtotal, IVA y total

RNF-19: Registro de Transacciones

- Todas las compras (facturas) deben quedar registradas con:
    - Folio único
    - Fecha y hora exacta
    - Cliente asociado
    - Detalle de productos y montos

---

3. RESTRICCIONES TÉCNICAS

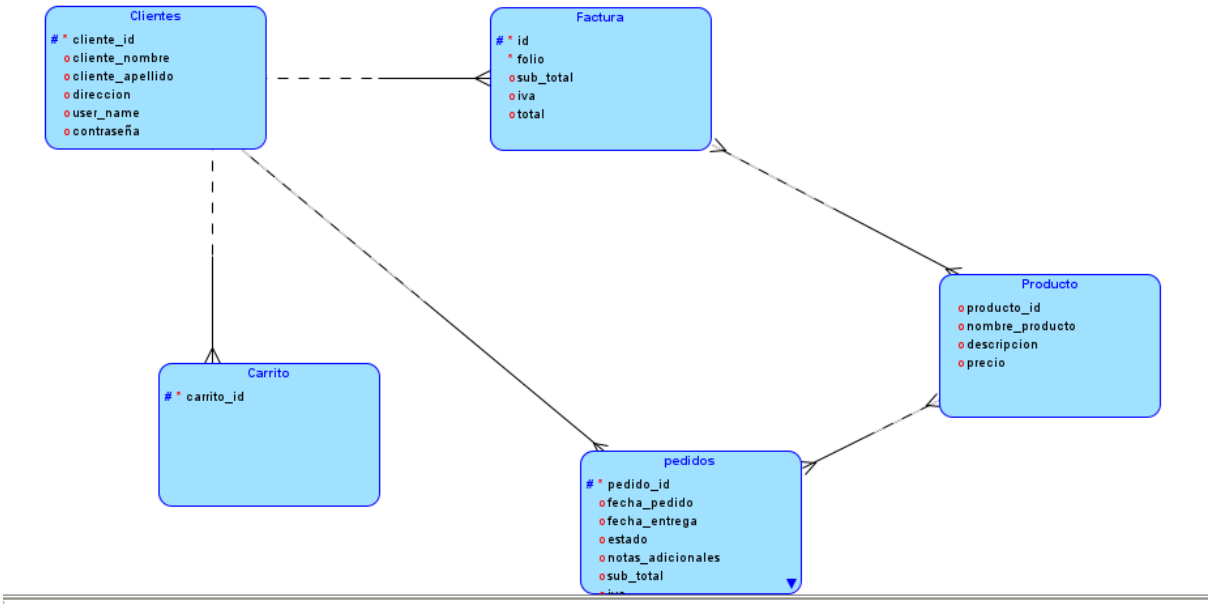
3.1 Tecnologías Obligatorias

- **Backend**: Spring Boot 3.x
- **Seguridad**: Spring Security con BCrypt
- **Base de Datos**: Oracle
- **ORM**: JPA/Hibernate
- **Frontend**: HTML5, CSS3, JavaScript vanilla
- **Servidor**: Tomcat embebido (puerto 8081)

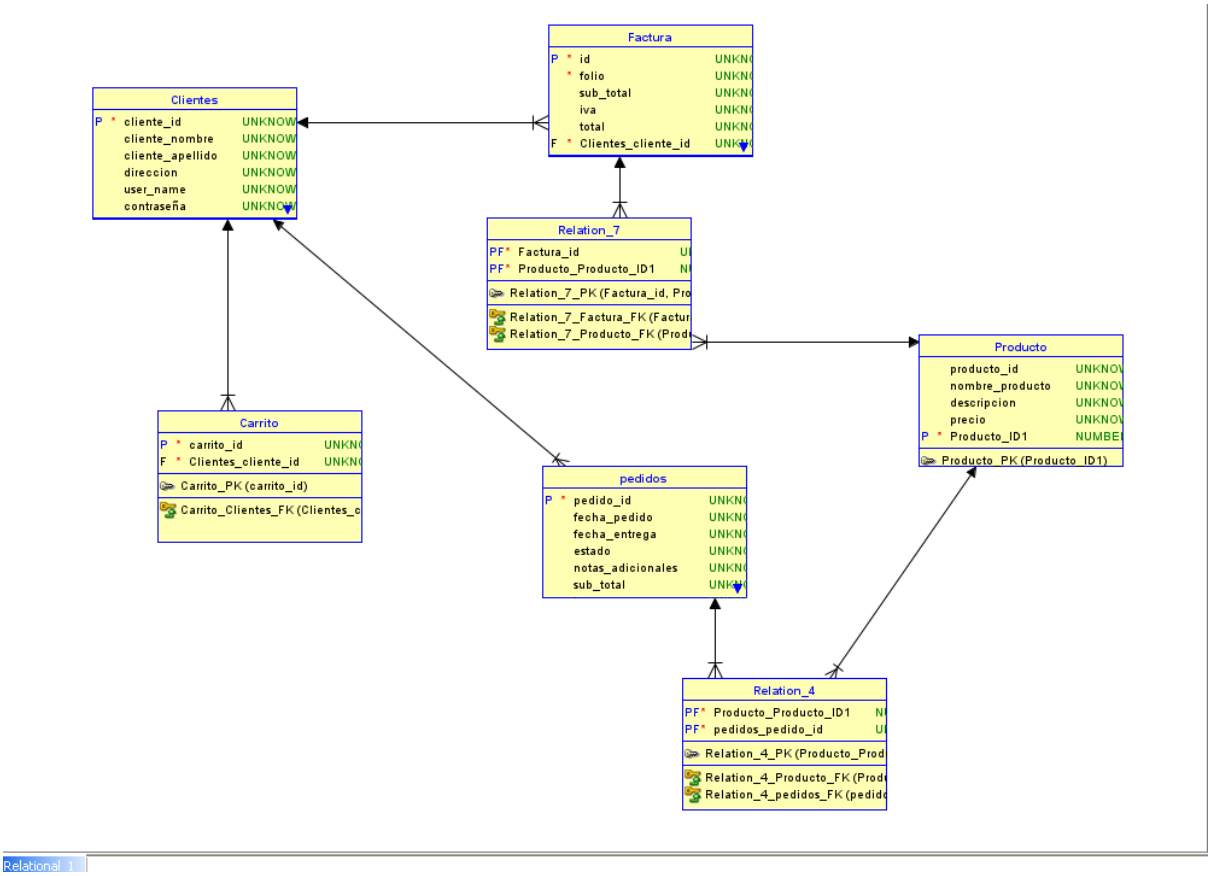
3.2 Restricciones de Negocio

- El carrito es individual por usuario (no compartido)
- Un cliente debe estar registrado y autenticado para realizar compras
- Los pedidos cancelados no se pueden reactivar
- Las facturas no se pueden cancelar una vez generadas
- El IVA se aplica a todos los productos sin excepción

Modelo conceptual



Modelo Lógico de la base de datos



A. ARQUITECTURA EN CAPAS (LAYERS/TIERS)

- Diagrama visual de 6 capas
- Descripción detallada de cada capa:
    1. **Capa de Presentación** (HTML/CSS/JS)
    2. **Capa de Seguridad** (Spring Security)
    3. **Capa de Controladores** (REST APIs)
    4. **Capa de Lógica de Negocio** (Validaciones y cálculos)
    5. **Capa de Acceso a Datos** (Repositories)
    6. **Capa de Persistencia** (Oracle SQL)
- Responsabilidades y tecnologías de cada capa

B. FRONTEND / BACKEND

- **Arquitectura Cliente-Servidor** con diagramas
- **Frontend (Client-Side)**:
    - Responsabilidades y tecnologías
    - Patrón SPA
    - Ejemplos de código
- **Backend (Server-Side)**:
    - Java
    - Springboot
    - Tomcat
    - Maven
    - Oracle SQL
- **Comunicación Frontend ↔ Backend**:
    - Diagramas de flujo
    - Ejemplos de Request/Response
    - Formato JSON

C. ESTRUCTURA MODULAR DEL SISTEMA

Organización de Paquetes/Módulos:

    - **Módulo de Entidades** (models/)
    - **Módulo de Repositorios** (repos/)
    - **Módulo de Controladores** (controllers/)
    - **Módulo de Servicios** (services/)
    - **Módulo Frontend** (static/)
- **Dependencias entre Módulos** con diagramas

Lógica y reglas del negocio

Utilizamos los endpoints para procesar las solicitudes que se hacen desde el navegador, asi podemos controlar cuando se hace una solicitud, creando un endpoint para cada acción necesaria. Para agregar una nueva función solo hace falta crear un nuevo endpoint para hacer las solicitudes y agregar su parte en el front que mande a llamarla

Descripción Interfaz de la aplicación

7. Descripción de reglas de seguridad(acceso/operación)

Utilizamos SpringSecurity para autenticar a los usuarios y darles permisos para entrar a ver poder agregar cosas al carrito, hacer las comprar y ver su historial de compras, tambien se utiliza una encriptación básica en las contraseñas, proporcionada por springSecurity

9. Conclusión

Este proyecto para aprender a manejar de una manera distinta las entidades de base de datos desde un lenguaje de programación y darle uso desde una aplicación web

Referencias

Spring Boot :: Spring Boot
java - How to connect to oracle database using spring boot - Stack Overflow
Spring Security