



Francisco Gabriel Inda Lomeli 21300640

Ian Gael Osorno Gonzalez 21300651

**José Maximiliano Rodríguez Martínez
21300616**

23/06/2025

Presentación Final

Especificación de Requerimientos de Software (SRS)

1. Introducción

1.1 Propósito

Este documento describe los requisitos funcionales y no funcionales para el desarrollo de una tienda de tenis digital. La aplicación permitirá a los usuarios explorar el catálogo de productos, gestionar su carrito de compras, administrar el inventario y generar tickets de compra en formato imprimible.

También describe los requisitos para el desarrollo de una tienda de tenis digital con integración completa de base de datos, permitiendo:

- Exploración de catálogo con datos persistentes
- Gestión de carritos de compra vinculados a usuarios
- Actualización en tiempo real de inventario
- Generación de tickets con datos almacenados

1.2 Alcance

La tienda digital estará diseñada para operar en dispositivos móviles y computadoras de escritorio, utilizando Angular y TypeScript para el frontend, SQL para la base de datos y un sistema de generación de tickets en formato imprimible.

Frontend: Angular + TypeScript (interfaz web responsiva).

Backend: API RESTful (Node.js/Express o Spring Boot).

Base de Datos: *MySQL/PostgreSQL para:*

Productos

Usuarios

Pedidos

Inventario

Generación de Tickets: PDF/HTML con datos desde BD

1.3 Definiciones, Acrónimos y Abreviaturas

- **SRS:** Software Requirements Specification (Especificación de Requerimientos de Software)
- **UI:** User Interface (Interfaz de Usuario)
- **SQL:** Structured Query Language
- ORM:** TypeORM/Sequelize para mapeo objeto-relacional
- JWT:** Autenticación mediante tokens
- Transacción ACID:** Garantía de integridad en compras

1.4 Referencias

- Guía de estilo de Angular: Angular Style Guide

- Documentación de TypeScript: TypeScript Documentation
- SQL Standards: SQL:2016 Standard

1.5 Descripción General

El sistema proporcionará una plataforma digital para que los clientes exploren el catálogo de tenis, gestionen su carrito de compras y reciban tickets de compra para su registro.

2. Descripción General del Producto

2.1 Perspectiva del Producto

El sistema es una aplicación web que permitirá a los usuarios navegar por el catálogo de tenis, agregar productos al carrito, y generar tickets de compra tras la confirmación del pedido.

2.2 Funciones del Producto

- **Catálogo de Productos:** Exploración de tenis con descripciones, imágenes y precios.
- **Carrito de Compras:** Agregar, eliminar y actualizar productos en el carrito.
- **Inventario:** Administración de stock en tiempo real.
- **Generación de Tickets:** Creación de tickets de compra en formato imprimible.

Módulo	Funcionalidad	Tablas BD Relacionadas
Catálogo	Búsqueda/filtrado de productos	Productos, Categorías
Carrito	Gestión de ítems por usuario	Carritos, ItemsCarrito
Pedidos	Historial y tickets	Pedidos, DetallesPedido
Inventario	Control de stock automático	Inventario

2.3 Características del Usuario

- **Clientes:** Usuarios que exploran el catálogo y realizan compras.
- **Administradores:** Personal autorizado para gestionar el inventario y productos.

2.4 Restricciones

- Compatible con los navegadores más utilizados (Chrome, Firefox, Safari, Edge).
- Cumplimiento de regulaciones locales de protección de datos.

2.5 Suposiciones y Dependencias

- Se asume que los usuarios tendrán acceso a internet.
- La base de datos SQL se alojará en un servidor seguro.

3. Requisitos Específicos

Requerimientos Funcionales (RF)

1. Catálogo de Productos

- **RF1.1:** El sistema debe mostrar los productos disponibles, mostrando imagen, descripción, precio y unidades disponibles.
- **RF1.2:** El sistema debe permitir una vista rápida del producto , incluyendo un botón para añadirlo al carrito sin abandonar la página del catálogo.
- **RF1.3:** El sistema debe mostrar todos los productos en el catalogo.

2. Carrito de Compras

- **RF2.1:** El cliente debe poder agregar productos al carrito desde el catálogo o la página de detalle del producto, especificando talla, color y cantidad.
- **RF2.2:** El cliente debe poder modificar cantidades.
- **RF2.3:** El sistema debe mostrar en tiempo real un resumen del carrito con nombre del producto, cantidad, precio unitario y total.

3. Inventario

- **RF3.1:** El sistema debe actualizar el stock de forma automática tras cada compra confirmada.
- **RF3.2:** El sistema debe mostrar los cambios en la base de datos.
- **RF3.4:** El sistema debe informar al cliente si un producto está agotado.
- **RF3.5:** El sistema debe manejar el stock de cada variante de un producto (por talla) de forma individual.

4. Proceso de Pago

- **RF4.1:** El sistema debe generar un ticket imprimible una vez que el cliente confirme su compra, incluyendo todos los detalles del pedido.
- **RF4.2:** el sistema debe de permitir una compra por medio de PAYPAL

5. Inicio de Sesión y Registro

- **RF5.1:** El sistema debe permitir a los usuarios registrarse con un correo electrónico y contraseña.
- **RF5.2:** El sistema debe validar que el correo electrónico sea único y tenga un formato correcto.
- **RF5.3:** El usuario debe poder iniciar sesión con correo y contraseña.
- **RF5.4:** El usuario debe poder recuperar su contraseña a través de una pregunta de seguridad.
- **RF5.5:** El usuario debe poder actualizar su perfil desde su cuenta.

Requerimientos No Funcionales (RNF)

1. Dominio y Accesibilidad

- **RNF1.1:** El sistema debe estar asociado a un dominio personalizado (www.sneakp-ulse.com) y renovarse cuando sea necesario

2. Seguridad de la Información

- **RNF2.1:** El sistema debe encriptar las contraseñas de los usuarios usando hash(para contraseñas)
- **RNF2.2:** El sistema debe cumplir con normativas del documento de aviso de privacidad.

3. Rendimiento y Usabilidad

- **RNF3.1:** El sistema debe responder a las acciones del usuario en menos de 2 segundos.
- **RNF3.2:** La interfaz debe ser responsiva e intuitiva
- **RNF3.3:** El sistema debe reaccionar con animaciones suaves y funciones interactivas que mejoren la experiencia del usuario.

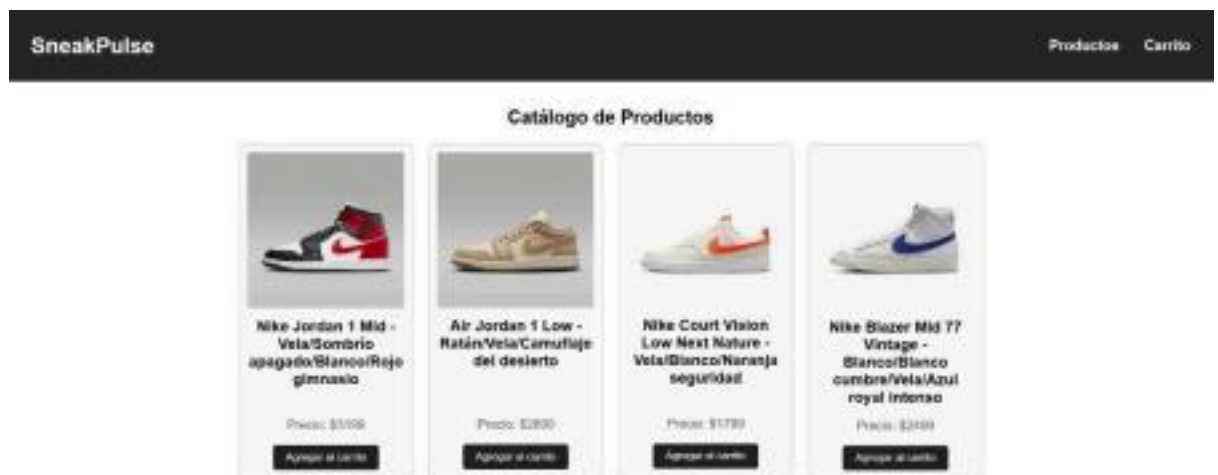
4. Mantenibilidad

- **RNF4.1:** El código del sistema debe estar bien documentado y seguir las buenas prácticas de desarrollo (Angular y TypeScript).
- **RNF4.2:** Las consultas a la base de datos deben estar optimizadas para responder en menos de 500ms (uso de índices y buenas prácticas SQL).

4. Interfaces Externas

4.1 Interfaz de Usuario (UI)

- Navegación clara con menús accesibles.
- Diseño responsivo adaptable a diferentes dispositivos.



[Volver](#)

Carrito de compras



El carrito está vacío

[Volver](#)

Carrito de compras

Nike Jordan 1 Mid - Vela/Sonriso apagado/Blanco/Rojo gimnasio

Cantidad: 2

\$6390.00

[Eliminar](#)

Air Jordan 1 Low - Resán/Vela/Camouflage del desierto

Cantidad: 1

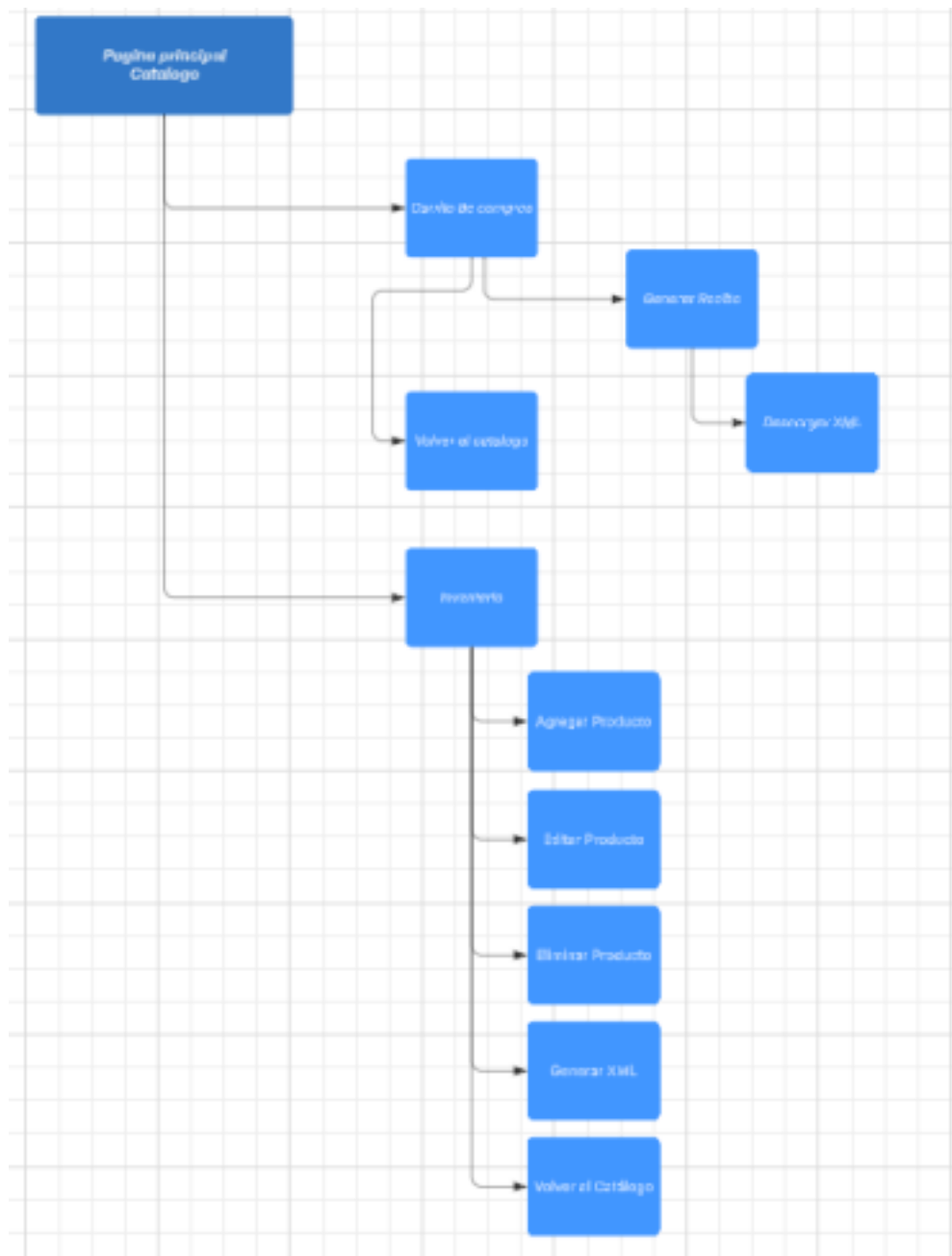
\$2690.00

[Eliminar](#)

Subtotal: \$9080.00

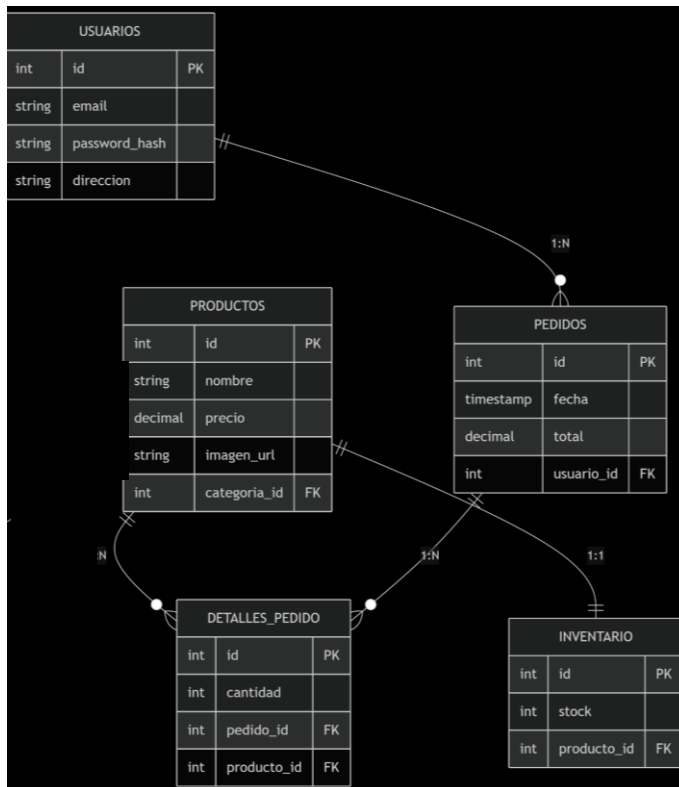
Total: \$10784.52

[Generar Pedido](#)



4.2 Interfaz de Base de Datos

- Base de datos SQL para almacenamiento seguro de productos, usuarios y pedidos.
- Consultas optimizadas para minimizar los tiempos de respuesta.



• 4.3 Interfaz de Generación de Tickets

- Estructura estándar para la generación de tickets de compra.
- Validación para asegurar la correcta generación de tickets.

Recibo Generado

```

<?xml version="1.0" encoding="UTF-8"?>
<compra>
  <producto id="1" cantidad="2">
    <nombre>Nike Jordan 1 Mid - Vela/Sombrío apagado/Blanco/Rojo gimnasio</nombre>
    <precio>3199</precio>
  </producto>
  <producto id="2" cantidad="1">
    <nombre>Air Jordan 1 Low - Ratán/Vela/Camouflage del desierto</nombre>
    <precio>2899</precio>
  </producto>
  <producto id="3" cantidad="1">
    <nombre>Nike Court Vision Low Next Nature - Vela/Blanco/Naranja seguridad</nombre>
    <precio>1799</precio>
  </producto>
  <producto id="4" cantidad="1">
    <nombre>Nike Blazer Mid 77 Vintage - Blanco/Blanco cumbre/Vela/Azul royal intenso</nombre>
    <precio>2499</precio>
  </producto>
  <subtotal>10396.00</subtotal>
  <iva>1663.36</iva>
  <total>12059.36</total>
  <tienda>Gracias por tu compra en ChinoSneakers</tienda>
</compra>

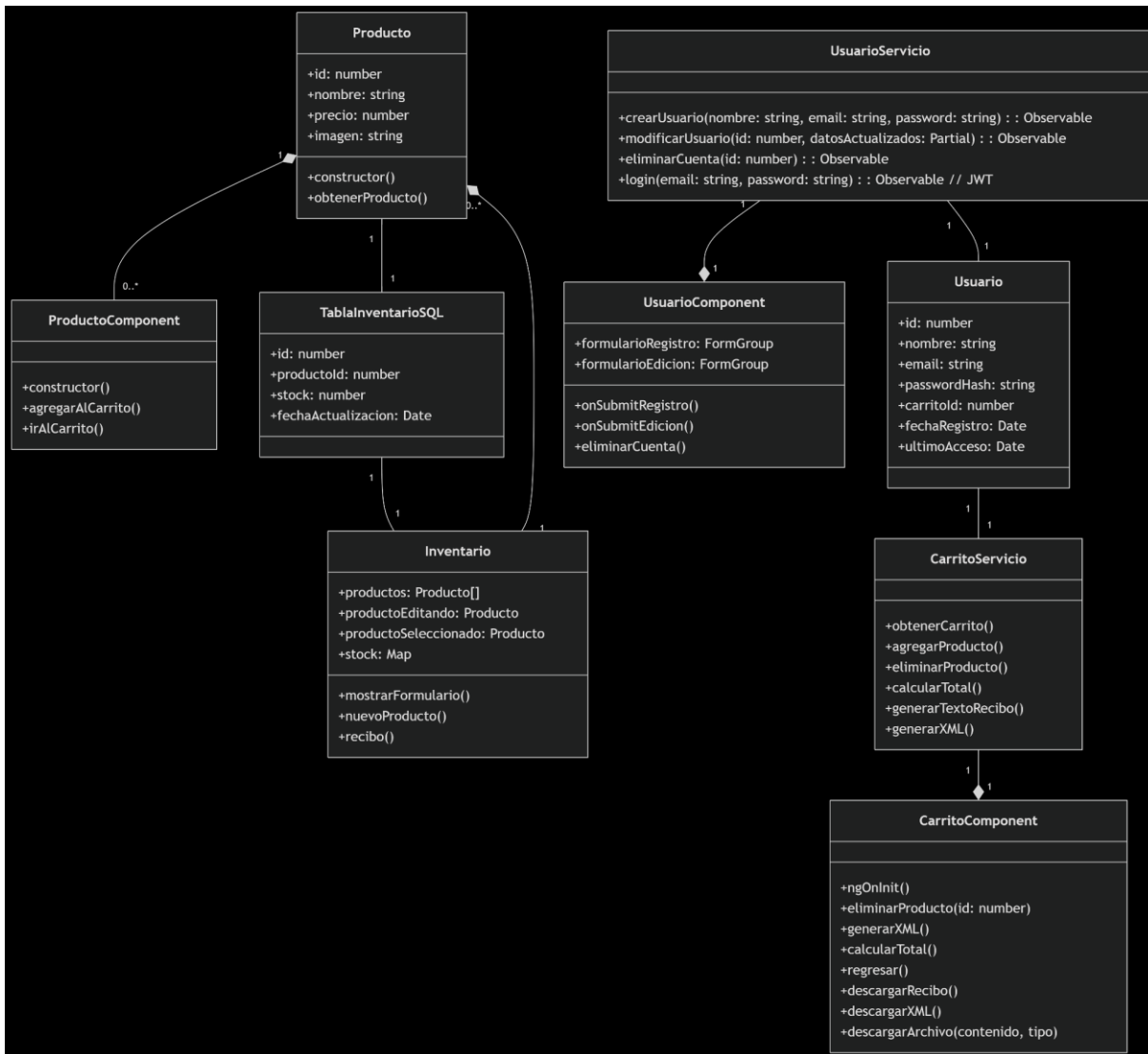
```

Descargar XML

5. Requisitos de Diseño del Sistema

5.1 Arquitectura del Sistema

El sistema seguirá una arquitectura cliente-servidor con un frontend en Angular y TypeScript, y un backend encargado de la gestión de datos e inventario.



5.2 Diagrama de Componentes

- **Frontend (Angular, TypeScript):** Interfaz de usuario para la tienda digital.
- **Backend (Node.js, SQL):** Gestión de datos y lógica de negocio.
- **Generador de Tickets:** Módulo para la generación de tickets de compra.
- **Base de Datos (SQL):** Almacenamiento de datos de productos y pedidos.

Casos de Uso

1. Explorar el catálogo de productos

- **Actor:** Cliente
- **Descripción:** El cliente navega por la tienda y visualiza los productos disponibles con sus imágenes, descripciones y precios.
- **Flujo principal:**
 1. El cliente accede a la tienda digital.
 2. El sistema muestra el catálogo de productos.
 3. El cliente puede desplazarse por la lista de productos.
 4. (Opcional) El cliente puede filtrar productos por categoría o buscar un producto específico.

2. Agregar productos al carrito

- **Actor:** Cliente
- **Descripción:** El cliente selecciona productos y los agrega al carrito para su compra.
- **Flujo principal:**
 1. El cliente visualiza un producto en el catálogo.
 2. El cliente selecciona la cantidad deseada.
 3. El cliente presiona el botón "Agregar al carrito".
 4. El sistema añade el producto al carrito y muestra un mensaje de confirmación.

3. Modificar el carrito de compras

- **Actor:** Cliente
- **Descripción:** El cliente puede cambiar la cantidad de productos en el carrito o eliminarlos antes de realizar la compra.
- **Flujo principal:**
 1. El cliente accede al carrito de compras.
 2. El sistema muestra los productos agregados y sus cantidades.
 3. El cliente puede:
 - Aumentar o disminuir la cantidad de un producto.
 - Eliminar un producto del carrito.
 4. El sistema actualiza el carrito y muestra los cambios.

4. Confirmar compra y realizar pago

- **Actor:** Cliente
- **Descripción:** El cliente revisa su carrito, confirma la compra y procede al pago.
- **Flujo principal:**
 1. El cliente accede al carrito y revisa su pedido.
 2. El cliente presiona el botón "Finalizar compra".
 3. El sistema solicita la información de pago y dirección de entrega (si aplica).
 4. El cliente ingresa los datos requeridos y confirma la compra.
 5. El sistema procesa el pago y genera un ticket de compra.
 6. (Opcional) El cliente puede descargar el ticket o recibirlo por correo electrónico.

5. Actualizar el inventario

- **Actor:** Sistema (con supervisión del Administrador)
- **Descripción:** El sistema actualiza automáticamente el stock de los productos comprados.
- **Flujo principal:**
 1. El cliente realiza una compra.
 2. El sistema descuenta la cantidad comprada del inventario.
 3. Si un producto tiene un stock bajo, el sistema notifica al administrador.

6. Administrar el catálogo de productos

- **Actor:** Administrador
- **Descripción:** El administrador puede agregar, modificar o eliminar productos del catálogo.
- **Flujo principal:**
 1. El administrador accede al panel de administración.
 2. El sistema muestra la lista de productos actuales.
 3. El administrador puede:
 - Agregar un nuevo producto.
 - Modificar los datos de un producto existente.
 - Eliminar un producto del catálogo.
 4. El sistema guarda los cambios y actualiza el catálogo.

7. Generar ticket de compra

- **Actor:** Sistema / Cliente
- **Descripción:** Tras realizar una compra, el sistema genera un ticket con los detalles de la transacción.
- **Flujo principal:**
 1. El cliente finaliza una compra.
 2. El sistema genera un ticket con los productos comprados, el total pagado y la fecha.
 3. El cliente puede descargar el ticket o recibirlo por correo electrónico.