



Ficha del documento

Alumnos: Camilo Poblete y Dane Miranda

Nombre del Sistema: Court – Plataforma Web de Venta de Artículos de Tenis

Fecha: 12/12/2025

Autores: Equipo de Desarrollo Court

Propósito: Especificación de Requisitos del Software (ERS)

1. Contenidos

| | |
|---|----------|
| Ficha del documento | 1 |
| 1. Contenidos | 2 |
| 2. Introducción | 3 |
| 2.1. Propósito | 3 |
| 2.2. Ámbito del Sistema | 3 |
| 2.3. Definiciones, Acrónimos y Abreviaturas | 3 |
| 2.4. Referencias | 3 |
| 2.5. Visión General del Documento | 3 |
| 3. Descripción General | 4 |
| 3.1. Perspectiva del Producto | 4 |
| 3.2. Funciones del Producto | 4 |
| 3.3. Características de los Usuarios | 4 |
| 3.4. Restricciones | 4 |
| 3.5. Suposiciones y Dependencias | 5 |
| 4. Requisitos Específicos | 5 |
| 4.1. Requisitos de Interfaces | 5 |
| 4.2. Requisitos Funcionales | 5 |
| 4.3. Requisitos No Funcionales | 5 |
| 4.4. Otros Requisitos | 6 |

2. Introducción

2.1. Propósito

El propósito de este documento es describir los requisitos funcionales y no funcionales del sistema Court, una plataforma web desarrollada para gestionar la compra y venta de artículos de tenis en la comuna de Maipú. Este documento está dirigido a desarrolladores, administradores del sistema y partes interesadas.

2.2. Ámbito del Sistema

El sistema Court permitirá a clientes visualizar productos, registrarse, iniciar sesión, gestionar un carrito de compras y realizar pagos. El administrador podrá gestionar usuarios y productos mediante un panel de control con funciones CRUD.

2.3. Definiciones, Acrónimos y Abreviaturas

- CRUD: Crear, Leer, Actualizar y Eliminar
- ERS: Especificación de Requisitos de Software
- FrontEnd: Interfaz del usuario desarrollada en React
- BackEnd: Servicios y lógica del servidor desarrollada en Spring Boot

2.4. Referencias

- IEEE 830 – Estándar de Requerimientos de Software

2.5. Visión General del Documento

Este documento presenta una visión estructurada del sistema Court, describiendo su contexto, funcionalidades, restricciones y requisitos detallados para su desarrollo.

3. Descripción General

3.1. Perspectiva del Producto

El sistema Court funciona de manera independiente con arquitectura cliente-servidor. El FrontEnd en React consume servicios REST del BackEnd en Spring Boot con una base de datos MySQL.

3.2. Funciones del Producto

- Autenticación de clientes y administradores.
- Visualización de catálogo de productos.
- Carrito de compras y generación de boletas.
- Gestión de productos por parte del administrador.
- Gestión de usuarios (clientes y administradores).

3.3. Características de los Usuarios

- Clientes: usuarios comunes, con acceso al catálogo y compras.
- Administradores: gestionan productos y usuarios.

3.4. Restricciones

- El sistema requiere conexión a internet.
- Las funcionalidades administrativas están restringidas a usuarios con el rol correspondiente.
- Debe ejecutarse en navegadores modernos.

3.5. Suposiciones y Dependencias

- Se asume disponibilidad del servidor BackEnd.
- El sistema depende de la base de datos MySQL.
- Se asume que los administradores utilizarán correctamente la plataforma.

4. Requisitos Específicos

4.1. Requisitos de Interfaces

La plataforma contará con una interfaz web responsive. Existirán vistas separadas para clientes y administradores.

4.2. Requisitos Funcionales

- 01: El sistema debe permitir registrar clientes y administradores.
- 02: El sistema debe permitir iniciar sesión dependiendo del rol.
- 03: El sistema debe permitir que el administrador gestione productos (CRUD).
- 04: El cliente debe poder visualizar el catálogo de productos.
- 05: El cliente debe poder agregar productos al carrito.
- 06: El sistema debe generar una boleta luego de un pago exitoso.
- 07: El administrador debe poder desactivar usuarios.

4.3. Requisitos No Funcionales

- F01: El sistema debe responder en menos de 1 segundo en la mayoría de las solicitudes.
- 02: El sistema debe ser seguro y proteger datos sensibles.
- 03: La interfaz debe ser accesible desde dispositivos móviles.
- 04: El sistema debe ser mantenable y escalable.

4.4. Otros Requisitos

N/A