

Especificación de Requisitos de Software

Proyecto: Verdes - souvenir

Revisión: [99.99]

[Seleccionar fecha]

Contenido

FICHA DEL DOCUMENTO.....	3
1. INTRODUCCIÓN.....	4
1.1. PROPÓSITO.....	4
1.2. ÁMBITO DEL SISTEMA.....	4
1.3. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS.....	4
1.4. REFERENCIAS.....	4
1.5. VISIÓN GENERAL DEL DOCUMENTO.....	4
2. DESCRIPCIÓN GENERAL.....	5
2.1. PERSPECTIVA DEL PRODUCTO.....	5
2.2. FUNCIONES DEL PRODUCTO.....	5
2.3. CARACTERÍSTICAS DE LOS USUARIOS.....	5
2.4. RESTRICCIONES.....	5
2.5. SUPOSICIONES Y DEPENDENCIAS.....	6
2.6. REQUISITOS FUTUROS.....	6
3. REQUISITOS ESPECÍFICOS.....	7
3.1 REQUISITOS COMUNES DE LAS INTERFACES.....	8
3.1.1 <i>Interfaces de usuario</i>	8
3.1.2 <i>Interfaces de hardware</i>	8
3.1.3 <i>Interfaces de software</i>	8
3.1.4 <i>Interfaces de comunicación</i>	8
3.2 REQUISITOS FUNCIONALES.....	9
3.3 REQUISITOS NO FUNCIONALES.....	9
3.3.1 <i>Requisitos de rendimiento</i>	9
3.3.2 <i>Seguridad</i>	10
3.3.3 <i>Fiabilidad</i>	10
3.3.4 <i>Disponibilidad</i>	10
3.3.5 <i>Mantenibilidad</i>	10
3.3.6 <i>Portabilidad</i>	10
3.4 OTROS REQUISITOS.....	10

Ficha del documento

Fecha	Revisión	Autor	Modificación

Documento validado por las partes en fecha:

Por el cliente

Por la empresa suministradora

[Firma]

[Firma]

Sr./Sra.

Sr./Sra.

1. Introducción

1.1. Propósito

Este documento tiene como Propósito describir los requisitos funcionales y no funcionales del sistema Verdes-Souvenir. El sistema es una tienda en línea de souvenirs ecológicos, desarrollada con React para el frontend y Spring Boot con Java para el backend, conectándose a una base de datos MySQL. Permite a los usuarios navegar por productos, registrarse, iniciar sesión, realizar compras y enviar mensajes de contacto, así como gestionar información desde un panel de administración

1.2. Ámbito del Sistema

- Nombre: Verdes - Souvenir.
- Qué hará: El sistema será una tienda online que se dividirá en dos grandes áreas:
 1. **Parte pública (Tienda):** permitirá a los usuarios navegar por los productos, ver información de la empresa, revisar el blog, enviar mensajes de contacto y gestionar un carrito de compras.
 2. **Parte privada (Administración):** permitirá a los administradores gestionar los recursos principales del sistema. En esta sección se podrán:
 - **Usuarios:** mostrar, crear, editar y eliminar usuarios registrados.
 - **Productos:** mostrar, crear, editar y eliminar productos disponibles en la tienda.
 - **Mensajes:** visualizar los mensajes enviados desde el formulario de contacto.
- **Beneficios:** el sistema ofrece una plataforma de comercio electrónico funcional con persistencia real de datos en una base de datos (MySQL), ofreciendo una experiencia cercana a una tienda real.
- **Objetivos:**
 - Desarrollar una aplicación web funcional utilizando React (frontend) y Spring Boot (backend).
 - Permitir la gestión básica de usuarios, productos y mensajes desde un panel administrativo.
 - Garantizar que la aplicación sea usable y clara, tanto para clientes como para administradores.
- **Metas:** entregar un sistema académico completo que cumpla con los requisitos mínimos de una tienda online y que sea fácilmente extensible hacia un sistema más robusto en el futuro.

1.3. Definiciones, Acrónimos y Abreviaturas

- **ERS:** Especificación de Requerimientos de Software.
- **API REST:** Interfaz de programación de aplicaciones utilizada para la comunicación entre el frontend (React) y el backend (Spring Boot).
- **JPA (Jakarta Persistence API):** Especificación de Java para gestionar la persistencia de datos en bases de datos relacionales.
- **CRUD:** Create, Read, Update, Delete (Crear, Leer, Actualizar, Eliminar), operaciones aplicadas tanto a usuarios como a productos en la parte de administración.
- **Carrito:** módulo que permite a los clientes simular la compra de productos seleccionados en la tienda.
- **Dashboard:** vista principal del administrador donde se concentran accesos a módulos de gestión.

1.4. Referencias

- Documento de **Especificación de Requerimientos de Software (ERS)** entregado por el docente.
- Material de clases del curso.
- Sitio oficial de [MDN Web Docs](https://developer.mozilla.org/es/docs/) como referencia técnica para el desarrollo en HTML, CSS y JavaScript.

1.5. Visión General del Documento

Este documento consta de:

- Un área de **definición del negocio**, donde se establece el propósito, alcance y contexto del sistema.
- Un área de **especificación de requisitos**, que detalla los requisitos funcionales y no funcionales del sistema.
- Una descripción de las **interfaces de usuario**, donde se definen las vistas principales de clientes y administradores.
- Un área de **anexos**, donde se incluyen formularios de caso de uso y capturas de pantalla de las vistas implementadas

-

2. Descripción General

2.1. Perspectiva del Producto

El sistema Verdes-Souvenir es un producto software independiente, desarrollado como una aplicación web con arquitectura cliente-servidor.

Utiliza una base de datos MySQL para la persistencia de datos y se comunica a través de una API REST interna.

El producto se divide en parte pública (tienda) y parte privada (administración) . Interactúa con un servidor de base de datos (MySQL) y expone endpoints REST.

2.2. Funciones del Producto

A grandes rasgos, el sistema provee las siguientes funciones:

- **Clientes (usuarios finales):**
 - Registrarse con datos personales validados (RUN, nombre, correo, región, comuna, contraseña, etc.).
 - Iniciar sesión con credenciales.
 - Navegar por productos y añadirlos al carrito.
 - Enviar mensajes de contacto a la tienda.
 - Revisar secciones informativas (Home, Blog, Nosotros).
- **Administrador:**
 - **Usuarios:** mostrar, crear, editar y eliminar usuarios.
 - **Productos:** mostrar catálogo, crear nuevos productos y editar/eliminar productos existentes.
 - **Mensajes:** visualizar los mensajes enviados desde el formulario de contacto.

2.3. Características de los Usuarios

El sistema contempla **dos tipos principales de usuarios**:

- **Cliente:**
 - Perfil: persona que navega en la tienda.
 - Experiencia técnica: nivel básico de uso de navegadores web.
 - Necesidades: registro, inicio de sesión, compras simuladas, contacto con la empresa.
- **Administrador:**
 - Perfil: encargado de gestionar la tienda en línea.
 - Experiencia técnica: nivel medio de uso de PC, con conocimientos básicos de gestión de información.
 - Necesidades: gestión de usuarios, productos, mensajes y pedidos.

2.4. Restricciones

El sistema se encuentra limitado por los siguientes factores:

- **Políticas de la empresa:** el sistema está desarrollado únicamente con fines académicos, no comerciales, y no contempla despliegue en un entorno productivo real.
- **Limitaciones de hardware:** requiere un equipo con navegador moderno (Google Chrome, Mozilla Firefox, Microsoft Edge) y conexión a Internet únicamente para descargar los recursos si se aloja en GitHub Pages.
- **Interfaces externas:** no se conecta con aplicaciones, APIs ni bases de datos externas; toda la lógica se ejecuta en el cliente.
- **Operaciones paralelas:** al estar basado en LocalStorage, el sistema no permite que múltiples usuarios trabajen simultáneamente sobre los mismos datos desde distintos equipos.
- **Lenguajes de programación:** desarrollado en **HTML, CSS y JavaScript puro**, sin frameworks de backend.
- **Persistencia de datos:** la información se guarda en **LocalStorage**, lo que implica:
 - Los datos se pierden si se borran los datos del navegador.
 - La información no es accesible desde distintos navegadores o dispositivos.
- **Seguridad:** no hay cifrado de contraseñas ni medidas avanzadas de protección contra ataques (ej. SQL Injection o XSS).
- **Funciones de auditoría:** no existe un historial de cambios o bitácora para auditar operaciones de usuarios o administradores.
- **Control de acceso:** el sistema diferencia entre administrador y cliente, pero no contempla múltiples roles con distintos permisos (ej. cajero, supervisor, etc.).
- **Escalabilidad:** el sistema no está diseñado para soportar un número elevado de usuarios ni grandes volúmenes de datos debido al uso exclusivo de LocalStorage.

2.5. Suposiciones y Dependencias

- Se asume que los usuarios utilizarán correos con dominios permitidos: **@duoc.cl**, **@profesor.duoc.cl**, **@gmail.com**.
- Se asume que los administradores son designados manualmente dentro del sistema.
- El sistema depende del correcto funcionamiento del navegador y su soporte a LocalStorage.
- Cualquier cambio hacia una infraestructura de servidor y base de datos real implicaría rediseñar gran parte del sistema.

2.6. Requisitos Futuros

A futuro, el sistema podría ampliarse con las siguientes mejoras:

- Integrar un sistema de **pagos en línea** real.
- Implementar un **sistema de reportes avanzados** para el administrador.
- Permitir integración con servicios externos (ej. envío de correos automáticos).

3. Requisitos Específicos

3.1 Requisitos comunes de las interfaces

3.1.1 Interfaces de usuario

- El sistema contará con una **interfaz web responsiva** desarrollada en HTML5, CSS3 y JavaScript.
- La navegación se hará mediante un menú superior (para la tienda pública) y enlaces de acceso rápido (para administración).
- La interfaz utilizará colores sobrios (fondo claro en vistas de cliente, fondo oscuro en panel de administración).
- Las vistas principales incluyen:
 - **Home:** pantalla de bienvenida.
 - **Productos:** catálogo con productos destacados y opciones de agregar al carrito.
 - **Blog:** artículos informativos.
 - **Nosotros:** información de la empresa y el equipo.
 - **Contacto:** formulario para enviar mensajes.
 - **Carrito:** vista con productos seleccionados.
 - **Panel de administración:** módulos de Usuarios, Productos, Órdenes, Mensajes y Configuración.

3.1.2 Interfaces de hardware

- El sistema debe poder visualizarse correctamente en computadores de escritorio, notebooks

3.1.3 Interfaces de software

- El sistema no depende de otros softwares externos ni bases de datos.
- Persistencia de datos mediante **LocalStorage** del navegador.
- No se integra con servidores externos ni APIs de terceros.

3.1.4 Interfaces de comunicación

- Al ser un sistema local, no requiere protocolos de comunicación adicionales.
- los protocolos de comunicación utilizados fueron tanto WhatsApp y GitHub

3.2 Requisitos funcionales

RF-01. Registro de usuarios

- El sistema debe permitir registrar nuevos usuarios validando: RUN, nombre, apellidos, correo, contraseña, región, comuna y dirección.
- Solo se aceptarán correos de dominio @duoc.cl, @profesor.duoc.cl, @gmail.com.

RF-02. Inicio de sesión

- El sistema debe validar credenciales.
- Debe redirigir a **home.html** si es cliente y a **dashboard.html** si es administrador.

RF-03. Gestión de usuarios (Administrador)

- Mostrar lista de usuarios en tabla con todos sus datos.
- Editar usuarios directamente desde la tabla.
- Eliminar usuarios desde la tabla.
- Crear nuevos usuarios desde un formulario similar al registro.

RF-04. Gestión de mensajes

- Almacenar mensajes enviados desde contacto.html en LocalStorage.
- Mostrar mensajes en el panel de administración en forma de tabla.
- Permitir eliminarlos si es necesario.

RF-05. Gestión de productos (Administrador)

- Mostrar los productos en una tabla del panel de administración.
- Crear nuevos productos ingresando nombre, descripción, precio e imagen.
- Editar productos existentes desde la tabla.
- Eliminar productos cuando ya no estén disponibles.

RF-06. Carrito de compras (Cliente)

- Permitir agregar productos desde la vista productos.html.
- Visualizar productos en carrito.html.
- Simular compra (sin pasarela de pago real).

RF-07. Navegación

- Menú principal visible en todas las páginas públicas.
- Enlaces hacia Home, Productos, Blog, Nosotros, Contacto, Inicio de sesión y Carrito.
- En el panel administrador, enlaces hacia Usuarios, Productos, Órdenes, Mensajes y Configuración.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

El 95% de las páginas deben cargarse en menos de **2 segundos** en un navegador estándar.

El sistema debe poder gestionar al menos **50 usuarios y 100 productos** almacenados en LocalStorage sin afectar su desempeño.

3.3.2 Seguridad

- Solo los administradores tendrán acceso al panel de administración.
- Diferenciación de roles (cliente/administrador) mediante tipoUsuario.
- Restricción de correos permitidos en el registro.

3.3.3 Fiabilidad

- La información permanecerá disponible mientras el usuario no borre datos de navegación.
- El sistema debe funcionar sin errores críticos en navegadores compatibles.

3.3.4 Disponibilidad

Disponible el 100% del tiempo mientras el usuario tenga acceso a un navegador moderno.

3.3.5 Mantenibilidad

- El sistema debe permitir agregar más regiones, comunas y productos editando directamente el archivo JavaScript o HTML correspondiente.
- El código está documentado para facilitar futuras modificaciones.

3.3.6 Portabilidad

- El sistema debe ser ejecutable en navegadores modernos (Chrome, Firefox, Edge).
- Compatible con Windows, Linux y Apple (vía navegador).
- No requiere instalación de software adicional.

3.4 Otros Requisitos

- El sistema debe contar con un **administrador por defecto** (usuario de prueba) creado automáticamente si no existen usuarios en LocalStorage.
- Todos los formularios deben mostrar mensajes de validación claros al usuario.
- Los productos y usuarios deben manejarse bajo operaciones **CRUD** básicas.
- El sistema debe mantener una interfaz visual uniforme en todas las vistas.