|  |
| --- |
| DUOC UC - ESCUELA DE INFORMATICA Y TELECOMUNICACIONES |
| Especificación de Requisitos de Software |
| *Proyecto: Verdes - souvenir* |
|  |
| **Integrantes: Jonathan Larraguibel, Eugenio Parada** |
| **Fecha: 26/9/2025** |

|  |
| --- |
| Especificación de Requisitos según estándar de IEEE 830. |

**Contenido**

[**Ficha del documento 3**](#_heading=h.29ve2bg61y5b)

[**1. Introducción 4**](#_heading=h.pf6gua4yhp0m)

[1.1.](#_heading=h.h6cmtrtcshbx) Propósito 4

[1.2.](#_heading=h.kmokx9b8bbs6) Ámbito del Sistema 4

[1.3.](#_heading=h.3v1pduukboe6) Definiciones, Acrónimos y Abreviaturas 4

[1.4.](#_heading=h.un1ehmx2e6qw) Referencias 4

[1.5.](#_heading=h.ynzh0zz0m0gj) Visión General del Documento 4

[**2.**](#_heading=h.6rpi2qonufpf) **Descripción General 5**

[2.1.](#_heading=h.splgdba24g2x) Perspectiva del Producto 5

[2.2.](#_heading=h.7v17y71kxvg2) Funciones del Producto 5

[2.3.](#_heading=h.arhaog9qakp8) Características de los Usuarios 5

[2.4.](#_heading=h.48dtmsa2x92t) Restricciones 5

[2.5.](#_heading=h.45mqzhinrs52) Suposiciones y Dependencias 6

[2.6.](#_heading=h.p5xmmkg7aeen) Requisitos Futuros 6

[**3.**](#_heading=h.horrpwcsz4ey) **Requisitos Específicos 7**

[3.1](#_heading=h.4bb1tgltx9hm) Requisitos comunes de las interfaces 8

[*3.1.1*](#_heading=h.2nfnctci4ued) *Interfaces de usuario 8*

[*3.1.2*](#_heading=h.35nfx5ihd8pr) *Interfaces de hardware 8*

[*3.1.3*](#_heading=h.njre4l9guk50) *Interfaces de software 8*

[*3.1.4*](#_heading=h.mmqqlsncj7zu) *Interfaces de comunicación 8*

[3.2](#_heading=h.2n3dx6oathhx) Requisitos funcionales 9

[3.3](#_heading=h.twcm89ju491c) Requisitos no funcionales 9

[*3.3.1*](#_heading=h.svhexgd7umj0) *Requisitos de rendimiento 9*

[*3.3.2*](#_heading=h.gfbb2yw782kx) *Seguridad 10*

[*3.3.3*](#_heading=h.8qdn4gqaui1h) *Fiabilidad 10*

[*3.3.4*](#_heading=h.qa1mkwd5ehj8) *Disponibilidad 10*

[*3.3.5*](#_heading=h.acfox0om62on) *Mantenibilidad 10*

[*3.3.6*](#_heading=h.d2okb1vel0d5) *Portabilidad 10*

[3.4](#_heading=h.wy2w4em5yq6y) 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

En este documento se describirá de manera detallada los requisitos funcionales y no funcionales del sistema Verdes-Souvenir. El sistema busca simular una tienda en línea de souvenirs, permitiendo 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

1. **Parte tienda:** se 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 administración:** se 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 contacto.
* **Beneficios:** el sistema facilita la simulación de un comercio electrónico, con validaciones y se guardaran los datos mediante LocalStorage así ofreciendo una experiencia cercana a una tienda real.
* **Objetivos:**
* Desarrollar una aplicación web funcional utilizando tecnologías HTML, CSS y JavaScript.
* Permitir la gestión de usuarios, productos y mensajes desde un panel administrativo.
* Garantizar que la aplicación sea usable y clara, tanto para clientes como para administradores.
* **Metas:** la meta es entregar un sistema completo que cumpla con los requisitos 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.
* **LocalStorage:** almacenamiento local en el navegador utilizado para guardar datos de usuarios, productos, mensajes y pedidos.
* **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/) 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 académica. No depende de sistemas externos, ya que utiliza **LocalStorage** del navegador para almacenar información de usuarios, productos, mensajes y pedidos.

El producto se divide en dos áreas:

* **Parte tienda:** accesible a cualquier usuario, permite la navegación por las secciones principales (Home, Productos, Blog, Nosotros y Contacto), el registro, inicio de sesión y simulación de compras a través del carrito.
* **Parte administración:** accesible solo a administradores, contiene módulos de gestión de usuarios, productos, órdenes y mensajes.

El sistema no interactúa con otros softwares ni APIs externas, lo que lo convierte en un producto autocontenido.

## 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:

* Sustituir LocalStorage por una **base de datos relacional** (ej. MySQL o PostgreSQL).
* Implementar autenticación segura con **cifrado de contraseñas**.
* Integrar un sistema de **pagos en línea** real.
* Implementar un **sistema de reportes avanzados** para el administrador.
* Crear un **módulo de inventario** con control de stock.
* 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 GitHup

## 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](http://duoc.cl), @[profesor.duoc.cl](http://profesor.duoc.cl), @[gmail.com](http://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.