|  |
| --- |
| DUOC UC - Escuela de informatica y telecomunicaciones |
| Especificación de Requisitos de Software |
| *Proyecto:* |
|  |
| **Revisión*: 1.0*** |
| **23/09/2025** |

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

Contenido

[Ficha del documento 3](#_Toc503382603)

[1. Introducción 4](#_Toc503382604)

[1.1. Propósito 4](#_Toc503382605)

[1.2. Ámbito del Sistema 4](#_Toc503382606)

[1.3. Definiciones, Acrónimos y Abreviaturas 4](#_Toc503382607)

[1.4. Referencias 4](#_Toc503382608)

[1.5. Visión General del Documento 4](#_Toc503382609)

[2. Descripción General 5](#_Toc503382610)

[2.1. Perspectiva del Producto 5](#_Toc503382611)

[2.2. Funciones del Producto 5](#_Toc503382612)

[2.3. Características de los Usuarios 5](#_Toc503382613)

[2.4. Restricciones 5](#_Toc503382614)

[2.5. Suposiciones y Dependencias 6](#_Toc503382615)

[2.6. Requisitos Futuros 6](#_Toc503382616)

[3. Requisitos Específicos 7](#_Toc503382617)

[3.1 Requisitos comunes de las interfaces 8](#_Toc503382618)

[3.1.1 Interfaces de usuario 8](#_Toc503382619)

[3.1.2 Interfaces de hardware 8](#_Toc503382620)

[3.1.3 Interfaces de software 8](#_Toc503382621)

[3.1.4 Interfaces de comunicación 8](#_Toc503382622)

[3.2 Requisitos funcionales 9](#_Toc503382623)

[3.3 Requisitos no funcionales 9](#_Toc503382624)

[3.3.1 Requisitos de rendimiento 9](#_Toc503382625)

[3.3.2 Seguridad 10](#_Toc503382626)

[3.3.3 Fiabilidad 10](#_Toc503382627)

[3.3.4 Disponibilidad 10](#_Toc503382628)

[3.3.5 Mantenibilidad 10](#_Toc503382629)

[3.3.6 Portabilidad 10](#_Toc503382630)

[3.4 Otros Requisitos 10](#_Toc503382631)

# Ficha del documento

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| *22/09/2025* | *1.0* | *Vicente Alfaro, Ivan Morales* | *Version inicial del documento ERS* |
|  |  |  |  |

Documento validado por las partes en fecha: 22/09/2025

|  |  |  |
| --- | --- | --- |
| Por el cliente |  | Por la empresa suministradora |
| [Firma] |  | [Firma] |
| Sr./Sra. |  | Sr./Sra. |

# 1. Introducción

En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software (ERS). Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

## 1.1. Propósito

El propósito de este documento es definir todo lo relacionado al proyecto del sitio web TumTum Clothing, desde una descripción general del producto, su perspectiva, funciones, requisitos funcionales y no funcionales, perfil de usuarios, entre otros.

## 1.2. Ámbito del Sistema

El sistema será una pagina web de comercio electrónico para la marca de ropa TumTum Clothing.

* **Lo que hará:** mostrar catalogo de productos, permitir registro/inicio de sesión de usuarios, vendedores y administrador de la tienda, gestionar carrito de compras, procesar pedidos, mostrar información de la marca, gestión de stock de productos.
* **Lo que no hará en esta fase:** integración avanzada con sistemas de contabilidad
* **Objetivos y beneficios:** aumentar ventas, mejorar visibilidad de la marca, facilitar interacción con clientes, facilitar la gestión de productos/stock de la tienda

## 1.3. Definiciones, Acrónimos y Abreviaturas

* ERS: Especificación de Requisitos de Software
* CRUD: Create, Read, Update, Delete
* UI: Interfaz de Usuario (User Interface)
* UX: Experiencia de Usuario (User Experience)
* LocalStorage: almacenamiento local del navegador (Web API)

## 1.4. Referencias

1. Plantilla IIEE 830 DUOCUC (documento base)
2. Anexo 1: Instruciones del producto (Tienda Online) – documento entregado por la asignatura
3. Manual-javascript – documento entregado por el profesor
4. CursoCSS3 – documento entregado por el profesor
5. Etiquetas\_HTML5 – documento entregado por el profesor
6. HTML5-CSS3 – documento entregado por el profeso

## 1.5. Visión General del Documento

Este documento detalla el propósito, alcance, perfil de usuario, restricciones y requisitos del sistema TumTum Clothing. Incluye también requisitos funcionales, no funcionales, supuestos y requisitos futuros.

# 2. Descripción General

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto**.** Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender.

Normalmente, esta sección consta de las siguientes subsecciones: Perspectiva del producto, funciones del producto, características de los usuarios, restricciones, factores que se asumen y futuros requisitos.

## 2.1. Perspectiva del Producto

El sistema funcionara de manera independiente, pero podrá integrarse en el futuro con pasarelas de pago(Mercado Pago, paypal u otra) y Almacenamiento externo (base de datos). Sera accesible desde navegadores web en dispositivos móviles y de escritorio

## 2.2. Funciones del Producto

* Home: Menú, logo, productos destacados.
* Registro/Login: Formularios con validaciones en JavaScript.
* Catalogo de productos con filtros y búsqueda; detalle de producto
* Carro de compras con persistencia en LocalStorage
* Contacto con formulario validado (nombre, correo, comentario)
* Panel administrativo (CRUD) para productos y usuarios, protegido por autenticación y roles

## 2.3. Características de los Usuarios

* Cliente: Usuario que navega, se registra, inicia sesión y compra productos. Conocimientos básicos de navegación web.
* Administrador: persona con acceso total al panel administrativo para gestionar productos y usuarios.
* Vendedor: usuario con permisos limitados (visualizar productos y ordenes).
* Visitante: usuario no registrado que puede navegar el catalogo y ver contenido publico.

## 2.4. Restricciones

* Desarrollo con tecnologías web estándar: HTML5, CSS3 y JavaScript puro.
* Repositorio remoto en GitHub con control de versiones.
* Variaciones obligatorias en formularios mediante JavaScript.
* Responsive Design y accesibilidad básica (alt en imágenes, labels en formularios).
* Uso de HTTPS en el entorno de producción.

## 2.5. Suposiciones y Dependencias

* Se asume disponibilidad de imágenes y descripciones de productos por parte del cliente.
* Dependencias: hosting, servicio de dominio, navegador moderno.
* En esta entrega, el carrito persistirá en LocalStorage; en entregas posteriores se podrá migrar a base de datos

## 2.6. Requisitos Futuros

* Integración con pasarelas de pago reales (MercadoPago, PayPal, etc.)
* Integración con sistemas de logística
* Integración con base de datos

# 3. Requisitos Específicos

## 3.1 Requisitos comunes de las interfaces

### 3.1.1 Interfaces de usuario

Las interfaces serán páginas web con barra de navegación superior, área central de contenido, tarjetas de producto y footer. El estilo será coherente con la identidad de TumTum Clothing (paleta de colores, tipografía y uso de imágenes de producto)

### 3.1.2 Interfaces de hardware

**Dispositivo Cliente (Usuarios – Clientes, Vendedores, Administradores)**

* **Descripción:** Computadoras personales, notebooks, tablets y smartphones.
* **Propósito del interfaz:** Permitir el acceso a la tienda en línea desde cualquier dispositivo.
* **Definición del interfaz:**
  + Navegador web instalado (Chrome, Firefox, Edge, Safari).
  + Compatibilidad con pantallas táctiles (tablets, smartphones).
  + Resoluciones mínimas soportadas: 360x640 px (móviles) hasta 1920x1080 px (desktop).
  + Configuración: conexión a Internet estable.

**Servidor Web / Hosting**

* **Descripción:** Equipo o servicio de alojamiento remoto donde se almacenan los archivos de la página (HTML, CSS, JS, imágenes).
* **Propósito del interfaz:** Servir los recursos de la aplicación web y atender solicitudes HTTP/HTTPS.
* **Definición del interfaz:**
  + Procesa peticiones HTTP/HTTPS desde clientes.
  + Configuración: debe soportar estándares HTML5, CSS3 y JavaScript.
  + Capacidad para manejar al menos 50 conexiones simultáneas (según requisitos no funcionales).

**Dispositivos de Entrada/Salida**

* **Descripción:** Teclado, mouse, pantalla táctil.
* **Propósito del interfaz:** Interacción directa con la interfaz gráfica del sistema.
* **Definición del interfaz:** Entrada de datos en formularios (registro, login, contacto, administrador).
  + Configuración: soporte para teclado físico y virtual (móviles).

**Almacenamiento Local (Cliente)**

* **Descripción:** Disco duro / memoria interna del dispositivo del usuario.
* **Propósito del interfaz:** Uso de LocalStorage para guardar temporalmente datos del carrito de compras.
* **Definición del interfaz:** Espacio mínimo requerido ~5MB por navegador (limitación estándar de LocalStorage).

### 3.1.3 Interfaces de software

**Producto 1: Navegadores Web (Chrome, Firefox, Edge, Safari)**

* **Descripción:** Aplicaciones cliente donde los usuarios acceden a la tienda online.
* **Propósito del interfaz:** Permitir la visualización y ejecución del HTML, CSS y JavaScript de la web.
* **Definición del interfaz:** Interpretación de código HTML5, CSS3 y JS. Entrada: solicitudes HTTP(S). Salida: renderizado visual de la página y ejecución de scripts.

**Producto 2: LocalStorage (Web API de JavaScript)**

* **Descripción:** API del navegador que permite almacenar datos de manera local en el dispositivo del cliente.
* **Propósito del interfaz:** Guardar y recuperar la información del carrito de compras y datos temporales de usuario.
* **Definición del interfaz:** Clave/valor en formato JSON. Entrada: localStorage.setItem(key, value). Salida: localStorage.getItem(key).

**Producto 3: GitHub (repositorio remoto de control de versiones)**

* **Descripción:** Plataforma de control de versiones basada en Git.
* **Propósito del interfaz:** Gestionar el código fuente del proyecto, asegurar la colaboración entre desarrolladores y mantener el historial de cambios.
* **Definición del interfaz:** Uso de comandos Git (push, pull, commit, branch). Interfaz en formato texto (mensajes de commit, issues, documentación Markdown).

### 3.1.4 Interfaces de comunicación

**Comunicación Cliente–Servidor**

* **Descripción:** El navegador del usuario (cliente) se comunica con el servidor donde está alojada la tienda online.
* **Protocolo:** HTTP/HTTPS.
* **Propósito:** Transferencia de páginas HTML, archivos CSS, scripts JavaScript e imágenes.
* **Formato de datos:** Texto plano (HTML, CSS, JS), binarios (imágenes), JSON (en futuras integraciones con APIs).

**Comunicación con Almacenamiento Local (LocalStorage)**

* **Descripción:** El navegador se comunica con el almacenamiento local del propio dispositivo.
* **Protocolo:** API Web LocalStorage (JavaScript).
* **Propósito:** Guardar y recuperar datos del carrito de compras en el cliente.
* **Formato de datos:** Clave–valor en formato texto o JSON.

**Comunicación con Repositorio GitHub (para desarrollo y colaboración)**

* **Descripción:** El proyecto se sincroniza con un repositorio remoto en GitHub.
* **Protocolo:** HTTPS / SSH (dependiendo de la configuración de Git).
* **Propósito:** Control de versiones, almacenamiento del código fuente y colaboración en equipo.
* **Formato de datos:** Archivos de texto (HTML, CSS, JS, MD), binarios (imágenes).

## 3.2 Requisitos funcionales

[RF-01] Registro de usuarios con validación de correo y contraseña.  
[RF-02] Inicio de sesión con validación en tiempo real.  
[RF-03] Visualizar catálogo de productos con filtros.  
[RF-04] Añadir productos al carrito y persistir en localStorage.  
[RF-05] Enviar mensajes desde formulario de contacto validado.  
[RF-06] CRUD de productos en panel administrador con validaciones.  
[RF-07] CRUD de usuarios con validaciones de RUN, correo y dirección dinámica.  
[RF-08] Roles de acceso: Administrador, Vendedor y Cliente.

## 3.3 Requisitos no funcionales

### 3.3.1 Requisitos de rendimiento

* sistema debe cargar la página principal en **menos de 3 segundos** en conexiones estándar (10 Mbps).
* Debe soportar al menos **50 usuarios concurrentes** sin degradar el rendimiento.
* El **95% de las interacciones del usuario (carga de productos, añadir al carrito, validación de formularios)** deben completarse en **menos de 1 segundo**.
* El carrito de compras debe poder almacenar hasta **50 productos por usuario** sin afectar la experiencia.

### 3.3.2 Seguridad

* **Validaciones en todos los formularios** mediante JavaScript (longitudes, dominios de correo, contraseñas, RUN, etc.).
* **Uso obligatorio de HTTPS** para proteger las comunicaciones cliente–servidor.
* **Contraseñas encriptadas** (o simuladas en esta fase con hash básico).
* **Restricciones de acceso según roles** (Administrador, Vendedor, Cliente).
* **Protección de integridad de datos críticos**: los productos y usuarios solo pueden ser modificados desde el panel de administración.

### 3.3.3 Fiabilidad

* El sistema debe tener un **tiempo medio entre fallos (MTBF) de al menos 24 horas** en condiciones normales.
* No debe presentar **errores críticos en navegación estándar** (ver productos, añadir al carrito, enviar formularios).
* En caso de error (ej. pérdida de conexión), el sistema debe **mostrar mensajes claros al usuario** y permitir la recuperación.

### 3.3.4 Disponibilidad

* La aplicación debe estar disponible **24/7 siempre que el servicio de hosting esté activo**.
* El sistema debe mantener una **disponibilidad mínima del 99% mensual**, excluyendo tiempos de mantenimiento programado.

### 3.3.5 Mantenibilidad

* El código debe estar **documentado y versionado en GitHub** con mensajes de commit claros.
* La arquitectura debe ser **modular** (HTML, CSS, JS separados).
* Las tareas de mantenimiento estarán a cargo del **equipo de desarrollo**, con revisiones mensuales.
* Deben realizarse pruebas semanales para verificar el correcto funcionamiento de formularios, carrito y autenticación.

### 3.3.6 Portabilidad

* El sistema debe ser compatible con los navegadores modernos: **Chrome, Firefox, Edge, Safari**.
* Debe adaptarse automáticamente a dispositivos móviles y tablets (**diseño responsive**).
* El código debe ser portable a otros entornos de hosting estándar (servidores Apache/Nginx).
* Uso exclusivo de **HTML5, CSS3 y JavaScript**, tecnologías independientes del sistema operativo.

## 3.4 Otros Requisitos

**Accesibilidad Web**

* Todas las imágenes deben tener texto alternativo (*alt*) descriptivo.
* Formularios deben contar con etiquetas (*labels*) asociadas a sus campos para facilitar la navegación por lectores de pantalla.
* Contraste de colores debe cumplir estándares de accesibilidad (WCAG AA mínimo).

**Cumplimiento Legal**

* El sistema debe cumplir con la **Ley de Protección de Datos Personales en Chile**.
* Los datos de usuarios solo podrán utilizarse con fines de compra y contacto.

**Estándares de Desarrollo**

* Código fuente estructurado según buenas prácticas (HTML5 semántico, CSS modular, JS organizado).
* Commits en GitHub deben incluir mensajes claros en español.