Logotipo

El contenido generado por IA puede ser incorrecto.

**Especificación de Requisitos del Software (ERS)**

**Proyecto:**

Frontend para Tienda Online "HuertoHogar"

**Integrantes:**

Cristofer San Martín

**Profesor:**

Fernando Enrique Sepúlveda Campos

**Fecha entrega:**

Viernes 09/09/2025

Índice

1. Introducción

* [1.1. Propósito del Documento](#_1.1._Propósito_del)
* [1.2. Alcance del Proyecto](#_1.2._Alcance_del)
* [1.3. Descripción General del Producto](#_1.3._Descripción_General)

1. Requerimientos del Software

* [2.1. Requerimientos Funcionales (RF) 📝](#_2.1._Requerimientos_Funcionales)
* [2.2. Requerimientos No Funcionales (RNF) ✨](#_2.2._Requerimientos_No)

1. Herramientas y Tecnologías 🛠️

* [Lenguajes de Programación:](#_Lenguajes_de_Programación:)
* [Librerías / Frameworks:](#_Librerías_/_Frameworks:)

1. Propuestas del Proyecto

* [4.1. Arquitectura del Frontend](#_4.1._Arquitectura_del)
* [4.2. Propuesta Visual](#_4.2._Propuesta_Visual)
* [4.3. Estrategia de Simulación de Backend](#_4.3._Estrategia_de)

# 1. Introducción

## 1.1. Propósito del Documento

El propósito de este documento es definir y especificar los requisitos funcionales y no funcionales, las herramientas tecnológicas y las propuestas de diseño para el desarrollo de la interfaz de usuario (frontend) del proyecto "HuertoHogar". Este ERS servirá como guía fundamental para el equipo de desarrollo durante el ciclo de vida del proyecto.

## 1.2. Alcance del Proyecto

El alcance de este proyecto se limita exclusivamente al **desarrollo del frontend** de la aplicación web "HuertoHogar". Esto incluye la creación de la estructura HTML, el diseño visual con CSS y toda la interactividad del lado del cliente con JavaScript puro. No se incluye el desarrollo del backend, base de datos ni la configuración de servidores. La comunicación con el backend será **simulada** en el cliente para demostrar la funcionalidad completa de la interfaz.

## 1.3. Descripción General del Producto

"HuertoHogar" es una tienda online diseñada para vender productos frescos del campo directamente a los consumidores. La plataforma permitirá a los usuarios registrarse, navegar por un catálogo de productos, añadirlos a un carrito de compras y simular el proceso de pago. El objetivo es ofrecer una experiencia de usuario limpia, intuitiva y que refleje los valores de frescura y calidad de la marca.

# 2. Requerimientos del Software

## 2.1. Requerimientos Funcionales (RF) 📝

### RF-001: Registro de Usuarios

* + El sistema debe permitir a un nuevo usuario registrarse a través de un formulario.
  + El formulario solicitará: Nombre Completo, Correo Electrónico y Contraseña.
  + Se implementará validación en tiempo real para todos los campos, indicando al usuario si los datos son correctos (borde verde) o incorrectos (borde rojo y mensaje de error).

### RF-002: Validación de Datos de Registro

* + **Nombre:** Debe tener un mínimo de 3 caracteres.
  + **Correo:** Debe tener un formato de email válido.
  + **Contraseña:** Debe tener entre 8 y 50 caracteres, e incluir al menos una mayúscula, una minúscula, un número y un carácter especial.

### RF-003: Autenticación de Usuarios (Login)

* + El sistema debe proveer un formulario de inicio de sesión para usuarios registrados.
  + La autenticación se simulará con un usuario predefinido (juan.perez@example.com) y una contraseña segura (HuertoHogar\_25!).
  + El sistema mostrará un mensaje de error si las credenciales son incorrectas.

### RF-004: Visualización de Catálogo de Productos

* + La aplicación debe mostrar una lista de productos disponibles, cada uno con su imagen, nombre, precio y descripción.
  + Debe existir una opción para filtrar los productos por categoría (Frutas, Verduras, etc.).
  + Se debe incluir una barra de búsqueda para encontrar productos por nombre.

### RF-005: Gestión del Carrito de Compras 🛒

* + Los usuarios deben poder agregar productos al carrito desde la página del catálogo.
  + Se debe poder visualizar el contenido del carrito, modificar la cantidad de cada producto y eliminarlos.
  + El sistema debe calcular y mostrar el subtotal de la compra en tiempo real.

### RF-006: Simulación de Proceso de Compra

* + En la página del carrito, un botón de "Finalizar Compra" simulará el cierre del pedido, mostrando un mensaje de confirmación al usuario y vaciando el carrito.

## 2.2. Requerimientos No Funcionales (RNF) ✨

### RNF-001: Usabilidad y Diseño

* + La interfaz debe ser intuitiva, atractiva y fácil de navegar.
  + El diseño debe seguir la paleta de colores y tipografías definidas en la propuesta visual del caso de estudio.

### RNF-002: Rendimiento

* + La carga inicial del sitio y la navegación entre páginas debe ser rápida.
  + Las validaciones de formularios y las actualizaciones del carrito deben ocurrir en tiempo real sin demoras perceptibles para el usuario.

### RNF-003: Compatibilidad

* + El sitio web debe ser compatible con las últimas versiones de los navegadores web modernos (Google Chrome, Mozilla Firefox, Microsoft Edge).

# ## 3. Herramientas y Tecnologías 🛠️

## Lenguajes de Programación:

* + **HTML5:** Para la estructura semántica y el contenido de las páginas web.
  + **CSS3:** Para el diseño, la maquetación y los estilos visuales. Se utilizará el modelo **Flexbox** para el layout.
  + **JavaScript (ECMAScript 6+):** Para la lógica, la interactividad, la manipulación del DOM y las validaciones de formularios.

## Librerías / Frameworks:

* + No se utilizarán librerías o frameworks externos (ej. React, Vue, jQuery). Todo el desarrollo se realizará con **JavaScript puro (Vanilla JS)** para cumplir con los objetivos académicos del proyecto.
* **Herramientas de Desarrollo:**
  + **Editor de Código:** Visual Studio Code (recomendado).
  + **Control de Versiones:** Git y GitHub para el versionado del código y la entrega del proyecto.
  + **Navegador Web:** Herramientas de desarrollador de Google Chrome para depuración.

# 4. Propuestas del Proyecto

## 4.1. Arquitectura del Frontend

Se propone una arquitectura modular y organizada, separando la estructura (HTML), la presentación (CSS) y la lógica (JavaScript) en carpetas distintas (/html, /css, /js). Esto facilita el mantenimiento, la escalabilidad y la colaboración en el código.

## 4.2. Propuesta Visual

El diseño se basará estrictamente en la "Propuesta de Diseño Visual" del caso de estudio, utilizando:

* **Colores:** Fondo #F7F7F7, acentos en Verde Esmeralda, Amarillo Mostaza y Marrón.
* **Tipografía:** Montserrat para el cuerpo de texto y Playfair Display para los encabezados, importadas desde Google Fonts.

## 4.3. Estrategia de Simulación de Backend

Dado que el proyecto es solo frontend, toda la data será "hardcodeada" (escrita directamente en el código).

* **Productos:** Se almacenarán en un arreglo de objetos en un archivo data.js.
* **Estado de Sesión y Carrito:** Se gestionará a través de la API de **LocalStorage** del navegador. Esto permite que el carrito de compras y el estado de "logueado" del usuario persistan entre recargas y diferentes páginas del sitio.