| DUOC UC - ESCUELA DE INFORMATICA Y TELECOMUNICACIONES |
| --- |
| Especificación de Requisitos de Software |
| *Proyecto: Frame Picture* |
|  |
| **Revisión*: [0.1]*** |
| **[23/09/2025]** |

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

**Sergio Aaron**

**Ignacio Gomez**

**Contenido**

[**Ficha del documento 3**](#_heading=h.cbvosnltnhq5)

[**1. Introducción 4**](#_heading=h.4viq9vcqh075)

[1.1. Propósito 4](#_heading=h.91szejsst22s)

[1.2. Ámbito del Sistema 4](#_heading=h.e9m09fwz6n4m)

[1.3. Definiciones, Acrónimos y Abreviaturas 4](#_heading=h.rnh8ne2j4m9i)

[1.4. Referencias 4](#_heading=h.vlggpztvy9m2)

[1.5. Visión General del Documento 5](#_heading=h.dah35iz1643)

[**2. Descripción General 6**](#_heading=h.wn5ncsz79j3a)

[2.1. Perspectiva del Producto 6](#_heading=h.1hokl1hxbtn)

[2.2. Funciones del Producto 6](#_heading=h.dmitqnss70r)

[2.3. Características de los Usuarios 6](#_heading=h.i95zitr6ovwy)

[2.4. Restricciones 6](#_heading=h.6f9bh9q5pjsu)

[2.5. Suposiciones y Dependencias 7](#_heading=h.phjdde52dpu5)

[2.6. Requisitos Futuros 7](#_heading=h.apstz7f83wu5)

[**3. Requisitos Específicos 8**](#_heading=h.cjvlb4hblga4)

[3.1 Requisitos comunes de las interfaces 8](#_heading=h.ym8h32uk2kfp)

[*3.1.1 Interfaces de usuario (UI) 8*](#_heading=h.s9qxxycmoeby)

[*3.1.2 Interfaces de hardware 8*](#_heading=h.3ki0eu293mki)

[*3.1.3 Interfaces de software 8*](#_heading=h.gs53erb5rw9z)

[*3.1.4 Interfaces de comunicación 9*](#_heading=h.p7qqedwz2dq9)

[3.2 Requisitos funcionales 9](#_heading=h.atpes2vc5n4v)

[3.3 Requisitos no funcionales 10](#_heading=h.brd2meqc7i7n)

[*3.3.1 Requisitos de rendimiento 10*](#_heading=h.6yjn6fi3011w)

[*3.3.2 Seguridad 10*](#_heading=h.y42xnb4kf1oi)

[*3.3.3 Fiabilidad 10*](#_heading=h.3dd5ut1lqw1x)

[*3.3.4 Disponibilidad 11*](#_heading=h.sfjav9lpydho)

[*3.3.5 Mantenibilidad 11*](#_heading=h.twgk1napjgi3)

[*3.3.6 Portabilidad 11*](#_heading=h.1mibbc2pymuu)

[3.4 Otros Requisitos 11](#_heading=h.t83lbundwj2s)

# 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

El propósito de este documento es definir los requisitos funcionales y no funcionales para el sitio web **"Legacy-Frame"**. Este sistema servirá como una plataforma en línea moderna y accesible, dedicada a la venta de cuadros y marcos de alta calidad.

## 1.2. Ámbito del Sistema

El sistema "Legacy-Frame" es una aplicación web de front-end diseñada para ser

**Responsiva** (adaptable a computadores, tablets y móviles). El alcance de esta fase del proyecto incluye:

* Un catálogo digital para visualizar cuadros y marcos.
* Un sistema de gestión de usuarios (registro e inicio de sesión).
* Una sección informativa sobre la empresa y un formulario de contacto.

Las funcionalidades que requieren un back-end, como la pasarela de pago y la gestión de inventario, se simularán del lado del cliente usando

***localStorage*** o se marcarán como requisitos para futuras versiones.

## 1.3. Definiciones, Acrónimos y Abreviaturas

**ERS:** Especificación de Requisitos del Software.

**Front-end:** La parte de la aplicación web con la que el usuario interactúa.

**UI:** Interfaz de Usuario.

**HTML/CSS/JS:** Tecnologías base para el desarrollo web.

**localStorage:** Almacenamiento de datos del lado del cliente en el navegador.

## 1.4. Referencias

Documento de caso de estudio: ***Caso Frame Picture.docx*.**

Archivos del proyecto: ***Login.html, registro.html, molduras.html, contacta.html, registro.js*, etc.**

## 1.5. Visión General del Documento

Este documento detalla la perspectiva del producto, sus funciones y restricciones. Posteriormente, especifica los requisitos de interfaz, funcionales (lo que el sistema *hace*) y no funcionales (cómo el sistema *es*).

# 2. Descripción General

## 2.1. Perspectiva del Producto

El producto es un nuevo sitio web independiente que busca ser la tienda online líder en venta de cuadros y marcos a nivel nacional. Su misión es ofrecer productos de alta calidad que realcen los espacios de los clientes, a través de una plataforma accesible y confiable.

## 2.2. Funciones del Producto

Las funciones clave implementadas en el front-end son:

1. **Gestión de Cuentas de Usuario:** Creación de cuentas con validaciones específicas y autenticación.
2. **Catálogo de Productos:** Visualización de un catálogo extenso con opciones de filtrado.
3. **Carrito de Compras (Simulado):** Gestión de productos seleccionados por el usuario.
4. **Comunicación:** Formulario de contacto directo con la empresa.

## 2.3. Características de los Usuarios

El usuario objetivo es el público general interesado en arte y decoración, sin necesidad de conocimientos técnicos avanzados para navegar el sitio.

## 2.4. Restricciones

**Tecnología:** El sistema está desarrollado exclusivamente con HTML5, CSS3 y JavaScript.

**Persistencia de Datos:** Toda la información del usuario se almacena en el ***localStorage*** del navegador, lo que implica que los datos no son centralizados.

**Seguridad:** Las validaciones se realizan del lado del cliente. No se implementan medidas de seguridad de back-end.

## 2.5. Suposiciones y Dependencias

**Suposición de Conectividad:** Se asume que el usuario final tendrá una conexión a internet activa para poder cargar las librerías externas (Bootstrap, Font Awesome) desde sus respectivos CDN.

**Dependencia de Navegador:** El sistema depende de que el usuario utilice un navegador web moderno que soporte HTML5, CSS3 y JavaScript (ES6), y que tenga la funcionalidad de ***localStorage*** habilitada.

**Dependencia de Librerías Externas:** La correcta visualización y funcionamiento de la interfaz dependen de la disponibilidad de los servicios CDN de Bootstrap y Font Awesome. Una caída de estos servicios podría afectar la apariencia del sitio.

## 2.6. Requisitos Futuros

**RF-F01:** Integración de una **pasarela de pago segura** para procesar compras reales.

**RF-F02:** Desarrollo de un **panel de administración** para que el personal de "Legacy-Frame" pueda gestionar productos, precios e inventario.

**RF-F03:** Implementación de un sistema de **seguimiento de envíos** en tiempo real.

**RF-F04:** Permitir a los usuarios registrados dejar **reseñas y calificaciones** en los productos que han comprado.

# 3. Requisitos Específicos

## 3.1 Requisitos comunes de las interfaces

### 3.1.1 Interfaces de usuario (UI)

* La interfaz será limpia, profesional y responsiva. Los componentes clave incluyen una barra de navegación fija, formularios interactivos con validación en tiempo real, un catálogo de productos visual mediante tarjetas y un footer con información de contacto.

### 3.1.2 Interfaces de hardware

El sistema debe ser compatible con dispositivos de entrada estándar como teclado y mouse. Además, debe poder conectarse y ser completamente funcional en **dispositivos táctiles móviles (smartphones y tablets)**. Esto se logra a través del diseño responsivo, que adapta los componentes de la UI para una correcta visualización y una interacción táctil sencilla (botones más grandes, menús adaptables, etc.).

### 3.1.3 Interfaces de software

El producto se integra con dos productos de software de terceros para su correcto funcionamiento:

1. **Bootstrap 5.3:**
   * **Descripción:** Framework de front-end para el diseño de sitios web responsivos y mobile-first.
   * **Propósito del interfaz:** Se utiliza para la maquetación general del sitio, la creación de la grilla responsiva, y el estilo de componentes como botones, formularios, modales y la barra de navegación.
   * **Definición del interfaz:** Se integra mediante la inclusión de su hoja de estilos CSS y su archivo JavaScript a través de un enlace CDN en el ***<head>*** y al final del ***<body>*** de los archivos HTML.
2. **Font Awesome 6.0:**
   * **Descripción:** Librería de iconos vectoriales.
   * **Propósito de la interfaz:** Se utiliza para mostrar todos los iconos presentes en el sitio web (íconos de redes sociales, de contacto, de interfaz, etc.).
   * **Definición de la interfaz:** Se integra mediante un enlace a su hoja de estilos CSS vía CDN en el ***<head>*** de los archivos HTML.

### 3.1.4 Interfaces de comunicación

En su estado actual como prototipo de front-end, el sistema no establece comunicaciones activas con otros sistemas. La única comunicación que realiza es a través del protocolo **HTTPS** para solicitar los archivos de las librerías externas (Bootstrap y Font Awesome) desde sus respectivos servidores CDN al navegador del cliente.

## 3.2 Requisitos funcionales

**RF-001: Registro de Nuevo Usuario**

* **Descripción:** Un visitante puede crear una cuenta en "Legacy-Frame", siempre que cumpla con las condiciones.
* **Entradas:** Nombre, Apellido, Email, Teléfono, Dirección, Contraseña y fecha de nacimiento.
* **Proceso:**
  1. El usuario completa el formulario en ***registro.html***.
  2. El sistema debe validar que el usuario es  
      **mayor de 18 años** a partir de la fecha de nacimiento ingresada. Esta validación se realiza con JavaScript.
  3. El sistema valida en tiempo real la  
      **fortaleza de la contraseña** (largo, mayúsculas, etc.).
  4. Al enviar, se verifica que el email no exista previamente en ***localStorage***.
  5. Si la validación es exitosa, los datos se guardan en ***localStorage***.
* **Salidas:** Un mensaje de éxito y redirección a la página de login.

**RF-002: Descuento por Código de Registro**

* **Descripción:** Ofrecer un descuento a usuarios que se registren con un código especial.
* **Entradas:** Campo opcional "Código de Descuento" en el formulario de registro.
* **Proceso:**
  1. Durante el registro, si el usuario ingresa el código **"HappyFrame01"**, se almacena una marca especial en su perfil de ***localStorage*** (ej. ***descuento: true***).
  2. Esta marca se utilizará en el futuro para aplicar un 10% de descuento en el carrito de compras.
* **Salidas:** Almacenamiento del beneficio de descuento en los datos del usuario.

**RF-003: Visualización y Filtrado de Catálogo de Productos**

* **Descripción:** Cualquier visitante puede ver y filtrar los productos ofrecidos.
* **Entradas:** Clic en la sección "Molduras" y selección de filtros.
* **Proceso:**
  1. El sistema renderiza ***molduras.html***, mostrando todos los productos.
  2. El usuario puede usar controles (ej. botones o un menú desplegable) para  
      **filtrar los productos por tipo** (ej. "Cuadrados", "Circulares") y por tamaño.
  3. El filtrado se simula con JavaScript, ocultando o mostrando las tarjetas de producto que no cumplen con el criterio seleccionado.
* **Salidas:** Una vista del catálogo actualizada que solo muestra los productos filtrados.

**RF-004: Funcionalidad del Carrito de Compras (Simulado)**

* **Descripción:** Permitir al usuario agregar, modificar y eliminar productos de su carrito de compras.
* **Entradas:** Clic en el botón "Agregar al Carrito" en un producto.
* **Proceso:**
  1. Al agregar un producto, este se añade a un array en ***localStorage***.
  2. Se debe poder abrir una vista del carrito donde se listen los productos, se puedan cambiar las cantidades o eliminar artículos.
  3. El sistema debe mostrar un  
      **resumen del carrito con precios detallados y el total**.
* **Salidas:** Una interfaz que muestra el contenido del carrito y el monto total a pagar.

## 3.3 Requisitos no funcionales

### 3.3.1 Requisitos de rendimiento

Las páginas del sitio web deben cargar en menos de 3 segundos en una conexión de banda ancha estándar. Las imágenes de productos deben estar optimizadas para la web.

### 3.3.2 Seguridad

* **Validación de Datos:** Se implementa validación de formularios del lado del cliente (usando JavaScript) para prevenir la entrada de datos incorrectos antes de ser procesados.
* **Restricciones de Almacenamiento:** No se emplean técnicas criptográficas. Se debe destacar que el almacenamiento de datos de usuario, incluyendo contraseñas, en ***localStorage*** **no es una práctica segura** y se utiliza únicamente con fines de prototipado.
* **Logs:** No se registran ficheros con "logs" de actividad, ya que toda la operación ocurre en el navegador del cliente.

### 3.3.3 Fiabilidad

Se espera que el sistema funcione sin errores críticos de JavaScript que impidan al usuario completar las funcionalidades principales (registro, visualización de productos, contacto) en los navegadores soportados. El tiempo medio entre incidentes debe ser alto, asumiendo un uso normal de la aplicación.

### 3.3.4 Disponibilidad

Una vez alojado en un servicio de hosting comercial, se exigirá una disponibilidad del **99.5%** del tiempo, permitiendo breves ventanas de mantenimiento planificado fuera de horas punta.

### 3.3.5 Mantenibilidad

**Responsable:** Las tareas de mantenimiento y actualización del código serán realizadas por el equipo de desarrollo (en este caso, el estudiante).

**Frecuencia:** No se requiere mantenimiento programado regular. Las actualizaciones se realizarán según surjan nuevas necesidades funcionales o se detecten errores. No se contempla la generación automática de estadísticas.

### 3.3.6 Portabilidad

* **Dependencia del Servidor:** El sistema tiene un **0% de componentes y código dependiente del servidor**, ya que es una aplicación puramente de front-end.
* **Portabilidad Alta:** Al estar desarrollado en HTML, CSS y JavaScript, el sitio es altamente portable. Puede ser alojado en cualquier servidor web estándar (Apache, Nginx, etc.) sin necesidad de configuración especial.
* **Independencia del S.O.:** El sistema es independiente del sistema operativo del servidor y del cliente, funcionando a través del navegador web.

## 3.4 Otros Requisitos

No se identifican otros requisitos para la fase actual del proyecto.