

Especificación de Requisitos de Software

Proyecto: Shibuya Store

Revisión: [99.99]

[23-09-2025]

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
25/09		Diego zamora Gabriel Sanchez	Presentación del proyecto

Documento validado por las partes en fecha:

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 presente documento tiene como propósito definir los requisitos del sistema ShibuyaStore, una página web orientada a la exhibición de figuras de anime.

Está dirigido tanto a los desarrolladores responsables de la implementación, como al cliente que solicita el proyecto, con el fin de tener un marco de referencia común sobre el alcance y funcionamiento esperado del software

1.2. Ámbito del Sistema

En esta subsección:

- Se hará de manera más modular cada validación y hojas de estilos css
- el sistema te dejara navegar en distintas páginas de la misma mostrando los productos dejando agregarlos al carrito de la página web te mostrará un login en donde puedes iniciar o crear una cuenta aunque por el momento no contamos con una base de datos y no guarda la información
- los beneficios que posee actualmente la página web son una estructura sólida y con capacidades de mejorar, un objetivo de esta página es llegar a personas que quieren comprar figuras de anime a un precio accesible y de calidad premium. como meta espero que esta pagina quede terminada en su totalidad con base de datos y backend

1.3. Definiciones, Acrónimos y Abreviaturas

ERS: Especificación de Requisitos de Software.

UI: Interfaz de Usuario.

HTML: Lenguaje de Marcado de Hipertexto.

CSS: Hojas de Estilo en Cascada

1.4. Referencias

- IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications.
- Plantilla ERS entregada (DSY1104 Evaluación Parcial 1 - ERS).
- Maqueta HTML/CSS entregada por el cliente (archivos fuente proporcionados).

1.5. Visión General del Documento

Este documento sigue la estructura de la plantilla IEEE 830 recibida. Contiene:

- Ficha del documento y control de versiones.
- Introducción y alcance del sistema.
- Descripción general del producto y su contexto.
- Requisitos específicos (interfaces, requisitos funcionales y no funcionales).
- Anexos y posibles casos de uso (pendientes según alcance futuro).

El documento prioriza la trazabilidad y la verificabilidad de los requisitos, indicando cuando un requisito es aplicable en la etapa de maqueta o queda para etapas futuras.

2. Descripción General

El producto es independiente, diseñado como una página web estática en su primera versión. Futuras etapas podrán incluir integración con bases de datos y sistemas de pago en línea. ---

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

Perspectiva: ShibuyaStore es una maqueta independiente que representa el front-end de una tienda de figuras de anime. En la versión actual es un producto aislado (sin integraciones externas). En futuras iteraciones se espera que forme parte de un ecosistema e-commerce con backend, base de datos y pasarelas de pago.

Relación con otros sistemas (actualmente no implementados):

- Backend RESTful para gestión de inventario y usuarios (futuro).
- Pasarela de pagos (futura).
- Servicio de envío/logística (futuro).

2.2. Funciones del Producto

Resumen de funciones (alto nivel):

1. Presentación de catálogo: mostrar listados de productos con imagen, nombre, precio estimado y etiqueta.
2. Ficha de producto: mostrar descripción detallada, múltiples imágenes, especificaciones y estado (stock estimado solo como atributo visual).
3. Carrito (simulado): permite añadir/quitar productos y ver un resumen temporal en sesión (no persistente).
4. Páginas informativas: 'Quiénes somos', 'Contacto', 'Políticas' (contenido estático).
5. Navegación y filtrado visual básico (por categoría o etiqueta) implementado en el frontend.

Estas funciones están descritas a un nivel de producto; los detalles de implementación se definirán cuando se plantee la versión con backend.

2.3. Características de los Usuarios

Usuarios generales: Personas interesadas en figuras de anime. No requieren conocimientos técnicos avanzados.

Administrador (futuro): Encargado de cargar o actualizar productos en el sistema cuando exista backend, este debe tener un curso de administración de empresas y debe contar con un conocimiento avanzado en programación con HTML5, CSS, JavaScript y bases de datos.

2.4. Restricciones

- El sistema está desarrollado únicamente con HTML y CSS (con posible uso básico de JavaScript).
- No existen integraciones con bases de datos ni medios de pago en la versión actual.
- Debe poder visualizarse correctamente en navegadores modernos.
- Políticas de la empresa.
- Limitaciones del hardware.
- Conocimiento en lenguaje(s) de programación.
- Requisitos de habilidad.
- Consideraciones acerca de la seguridad.

2.5. Suposiciones y Dependencias

Suposiciones:

- Se asume que el cliente proveerá los contenidos (imágenes, descripciones) o dará autorización para su uso.

- Se asume que la maqueta se alojará en un servidor web convencional o en hosting estático (GitHub Pages, Netlify, Vercel, etc.).

Dependencias:

- Navegadores compatibles (Chrome, Firefox, Edge, Safari).
- Recursos estáticos entregados por el cliente (imágenes optimizadas, logos, textos legales).
- Conectividad a internet para acceso al sitio.

2.6. Requisitos Futuros

Posibles mejoras y requisitos a considerar en versiones posteriores:

- Implementación de backend con API REST y base de datos para persistencia de productos y usuarios.
- Sistema de autenticación y administración de usuarios.
- Integración de pasarelas de pago.
- Módulo de gestión de inventario y pedidos.
- Internacionalización (i18n) y soporte multi-moneda.

3. Requisitos Específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

- El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.
- Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.
- Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.
- Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:
 - **Corrección:** La ERS es correcta si y sólo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.

- **No ambiguos:** Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.
- **Completo:** Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
- **Consistentes:** Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.
- **Clasificados:** Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
- **Verificables:** La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc. Introducen ambigüedad en los requisitos. Requisitos como "en caso de accidente la nube tóxica no se extenderá más allá de 25Km" no es verificable por el alto costo que conlleva.
- **Modificables:** La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos facilitan enormemente esta tarea.
- **Trazables:** La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica que componentes del sistema son los que realizan el requisito R.

3.1 Requisitos comunes de las interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

3.1.1 Interfaces de usuario

Requisitos del interfaz de usuario (según plantilla):

- La aplicación consistirá en páginas web responsivas con una barra de navegación superior y un área principal de contenido.
- Se especifica estilo visual coherente con la temática anime (paleta de colores, tipografías y uso de

-
iconografía adecuada).

- Elementos mínimos: header con logo y menú, área de catálogo en rejilla, ficha de producto con galería de imágenes, footer con datos de contacto.
- La implementación deberá usar CSS externo para estilos y mantener semántica HTML (etiquetas <nav>, <main>, <article>, <figure>, <figcaption>, <footer>).
- Debe considerarse buenas prácticas de accesibilidad (atributos alt en imágenes, estructura de encabezados, contraste mínimo de texto).

3.1.2 Interfaces de hardware

Características lógicas y de compatibilidad:

- El sistema debe ser usable en dispositivos con pantalla táctil (smartphones y tablets).
- No se requiere hardware adicional; el sistema funcionará en cualquier dispositivo con navegador moderno.

3.1.3 Interfaces de software

- Integraciones y formatos (versión actual):
 - Producto software principal: Navegador web (Chrome, Firefox, Edge, Safari).
 - Propósito del interfaz: renderizar el contenido HTML/CSS/JS.
 - Definición del interfaz: peticiones HTTP para archivos estáticos; contenido en UTF-8; imágenes en formatos JPG/PNG/WebP.
 - En la versión actual no se integra con otros productos software de terceros. Para futuras integraciones (backend/API) se definirá: formato JSON, endpoints REST y esquema de autenticación.

-

3.1.4 Interfaces de comunicación

Requisitos de comunicación:

- En la etapa de maqueta no existen comunicaciones con sistemas externos.
- Protocolo esperado para despliegue: HTTPS (recomendado) y HTTP para pruebas locales.
- En versiones futuras, la comunicación entre front-end y back-end deberá usar HTTPS y JSON sobre REST.

3.2 Requisitos funcionales

Definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.

-

En ellas se incluye:

- Comprobación de validez de las entradas
- Secuencia exacta de operaciones
- Respuesta a situaciones anormales (desbordamientos, comunicaciones, recuperación de errores)
- Parámetros
- Generación de salidas
- Relaciones entre entradas y salidas (secuencias de entradas y salidas, fórmulas para la conversión de información)
- Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)

Los requisitos funcionales pueden ser divididos en sub-secciones.

3.2.1 Requisito funcional 1

Requerimiento funcional 1: Crear pedido en el sistema.

Actores: Garzón

Descripción: El garzón debe poder registrar los diferentes productos que pide el cliente, de acuerdo al menú disponible.

3.2.2 Requisito funcional 2

3.2.3 Requisito funcional 3

3.2.4 Requisito funcional n

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

- Tiempo objetivo de carga: menos de 3 segundos para la vista de catálogo en una conexión estándar (10 Mbps) y dispositivo de gama media.
- La interacción de añadir/quitar del carrito debe reflejarse en la interfaz en menos de 200 ms para una experiencia responsiva.
- Imágenes optimizadas (WebP/compresión) para reducir peso y mejorar tiempos de carga.

3.3.2 Seguridad

Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requisitos pueden especificar:

- Empleo de técnicas criptográficas.
- Registro de ficheros con “logs” de actividad.
- Asignación de determinadas funcionalidades a determinados módulos.
- Restricciones de comunicación entre determinados módulos.
- Comprobaciones de integridad de información crítica.

3.3.3 Fiabilidad

Especificación de los factores de fiabilidad necesaria del sistema. Esto se expresa generalmente como el tiempo entre los incidentes permisibles, o el total de incidentes permisible.

3.3.4 Disponibilidad

- Se espera que la página esté disponible siempre que el hosting esté activo; para entornos de producción se recomienda del 99%.
- Para la maqueta, la disponibilidad depende del proveedor de hosting estático.

3.3.5 Mantenibilidad

La persona que debe dar mantenimiento al sistema puede ser tanto el Administrador (Solo si este tiene experiencia con HTML, CSS, JavaScript y bases de datos) o puede realizarlo una persona externa contratada (desarrollador) pero que debe cumplir con una experiencia en el manejo de esos lenguajes.

El mantenimiento del proyecto debe realizarse completamente mínimamente 1 vez al mes.

3.3.6 Portabilidad

Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:

- Porcentaje de componentes dependientes del servidor.
- Porcentaje de código dependiente del servidor.
- Uso de un determinado lenguaje por su portabilidad.
- Uso de un determinado compilador o plataforma de desarrollo.

- - Uso de un determinado sistema operativo.

3.4 Otros Requisitos

Cualquier otro requisito.