

Metodología de Sistemas I

Comisión 4

Docente: Cristian Halm.

Integrantes:

Telega Cristian

Cabrera Jonathan David

Schäffer Pablo Octavio

GitHub: <https://github.com/Cabrera-Jonathan-David/pixelPerfect>

**Contenido**

**1- Introducción**

1.1 Propósito

1.2 Ámbito del Sistema

**2- Descripción General del Sistema**

2.1 Perspectiva del Producto

2.2 Objetivos del Sistema

2.3 Herramientas Utilizadas

**3- Definición de Requisitos del Sistema**

3.1 Requisitos Funcionales

3.2 Requisitos No Funcionales

**4- Diagramas**

4.1 Diagrama de Casos de Uso

4.1.1 Especificación de caso de uso

4.1.2 Camino de éxito

4.1.3 Excepciones

4.2 Modelo Conceptual

#### **1. Introducción**

#### **1.1 Propósito**

Este documento describe los requisitos funcionales y no funcionales del sistema **Pixel Perfect**, un ecommerce especializado en artículos de computación.

El propósito es proporcionar una visión completa del sistema, detallando cómo los usuarios y administradores interactúan con las funcionalidades disponibles, y servir como referencia tanto para desarrolladores como para interesados en el desarrollo del sistema.

***1.2 Ámbito del Sistema***

El sistema **Pixel Perfect** es un ecommerce diseñado para ofrecer una experiencia eficiente tanto para compradores como para administradores.

**Incluye:**

* Gestión de usuarios y productos.
* Realización de compras tanto para usuarios registrados como invitados.
* Acceso a funcionalidades administrativas para gestionar pedidos y productos.

**Excluye:**

* Integración con pasarelas de pago reales.
* Servicios de soporte técnico o reparación de productos.

### ***2. Descripción General del Sistema***

***2.1 Perspectiva del Producto***

El sistema se basa en APIs simuladas mediante JSON Server, las cuales proporcionan datos organizados en:

* **Usuarios:** Gestiona credenciales y roles (*admin* o *client*).
* **Clientes:** Contiene información personal y de contacto.
* **Productos:** Ofrece un catálogo con detalles de cada producto.
* **Ventas:** Registra compras y pedidos realizados por los clientes.

El sistema vincula usuarios y clientes mediante el campo id\_cliente.

***2.2 Objetivos del Sistema***

1. Facilitar la compra de productos tecnológicos mediante una plataforma intuitiva.
2. Permitir una gestión eficiente de productos y pedidos para administradores.
3. Ofrecer opciones de compra rápida para usuarios invitados.

***2.3 Herramientas Utilizadas***

* **Frontend:** Angular (TypeScript, HTML, CSS).
* **Backend Simulado:** JSON Server.
* **Framework de Estilo:** Bootstrap.
* **Librerías Adicionales:** JavaScript.

### ***3. Definición de Requisitos del Sistema***

#### **3.1 Requisitos Funcionales**

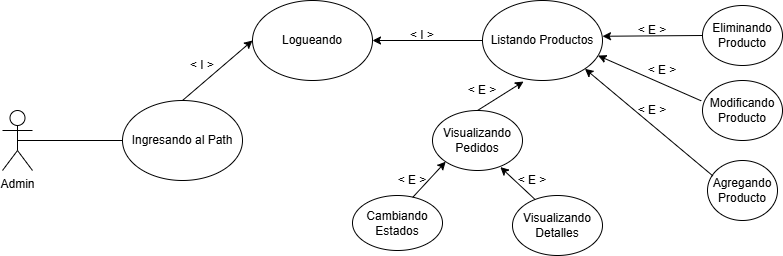
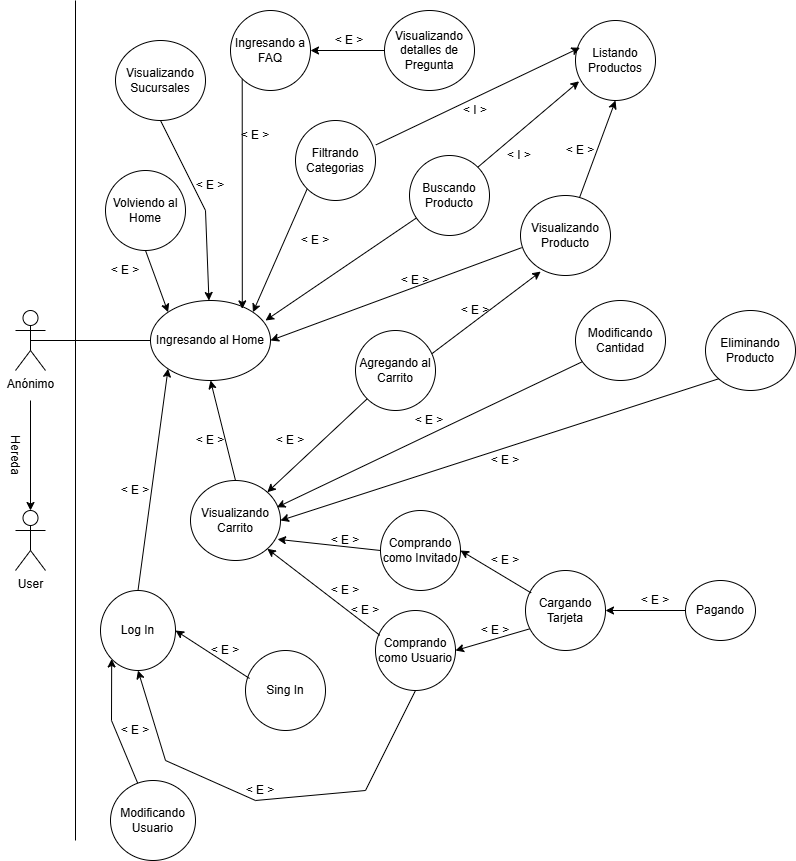
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Nombre** | **Descripción** | **Usuario** |
| **RF-001** | Registrar usuario | El sistema deberá permitir que los clientes no registrados creen una cuenta de usuario con su correo, un nombre de usuario y contraseña. Además, deberá autenticar que no exista ninguno de los mismos, además de información personal de contacto. | Anónimo |
| **RF-002** | Iniciar sesión | El sistema deberá permitir que los usuarios ingresen a su cuenta a través de su nombre de usuario y contraseña. Los mismos serán verificados previo al acceso. | Anónimo  Administrador |
| **RF-003** | Cerrar Sesión | El sistema deberá permitir a un usuario ya logueado cerrar la sesión desde la barra de navegación y ser redirigido a la página principal. | Usuario |
| **RF-004** | Modificar información de usuario | El sistema deberá permitir que un usuario registrado y logueado pueda modificar su información personal, verificando que los nuevos datos cumplan con los parámetros especificados. | Usuario |
| **RF-005** | Visualizar Productos | El sistema deberá presentar los productos almacenados en la base de datos, permitiendo que los usuarios puedan solicitar los detalles de los mismos o añadirlos directamente al carrito de compras. | Usuario  Anónimo |
| **RF-006** | Visualizar detalle del producto | El sistema deberá mostrar la información completa de un producto seleccionado, además de permitir añadirlo al carrito de compras. | Usuario  Anónimo |
| **RF-007** | Filtrar productos por categoría | El sistema deberá permitir mostrar únicamente los productos que correspondan a una categoría seleccionada por el usuario, con la opción de ver sus detalles o añadirlos al carrito de compras. | Usuario  Anónimo |
| **RF-008** | Búsqueda de productos por coincidencia | El sistema deberá mostrar los productos que contengan palabras clave ingresadas por el usuario, permitiendo ver los detalles o añadirlos al carrito de compras. | Usuario  Anónimo |
| **RF-009** | Modificar Carrito | El sistema deberá permitir modificar las cantidades de los productos en el carrito de compras, pudiendo agregar, restar o eliminar completamente un producto. | Usuario  Anónimo |
| **RF-010** | Validar datos de pago | El sistema deberá recibir la información de una tarjeta de débito y verificar dichos datos. | Usuario  Anónimo |
| **RF-011** | Crear Productos | El sistema deberá permitir crear un nuevo producto mediante un formulario que reciba toda la información requerida y una imagen a través de una URL. | Administrador |
| **RF-012** | Modificar Productos | El sistema deberá permitir modificar la información de un producto, verificando que los nuevos datos sean válidos. | Administrador |
| **RF-013** | Visualizar Almacén de Productos | El sistema deberá mostrar todos los productos almacenados, permitiendo modificar o eliminar la información de los mismos. | Administrador |
| **RF-014** | Generar Pedido | El sistema deberá almacenar los pedidos de los usuarios luego de verificar que los datos personales, las credenciales de pago y la existencia de stock sean correctos, guardando la información necesaria para el envío de los productos. | Usuario  Anónimo |
| **RF-015** | Visualizar Pedidos | El sistema deberá mostrar todos los pedidos confirmados, con la información del usuario que realizó la compra, los productos adquiridos y el estado del pedido, permitiendo modificar este último. | Administrador |
| **RF-016** | Visualizar preguntas frecuentes | El sistema deberá mostrar una lista de preguntas frecuentes y permitir al usuario ver la respuesta de cada una al seleccionarla, con la opción de ocultarla nuevamente. | Usuario  Anónimo |
| **RF-017** | Visualizar Locales | El sistema deberá mostrar información sobre los distintos locales asociados a la página. | Usuario  Anónimo |

#### **3.2 Requisitos No Funcionales**

|  |  |
| --- | --- |
| **ID** | **Descripción** |
| **RNF-001** | El sistema deberá ser implementado como una aplicación web. |
| **RNF-002** | El sistema verificará las credenciales del usuario para permitir o denegar accesos a distintas partes de la página. |
| **RNF-003** | El sistema verificará que no exista duplicidad de datos primarios, como nombres de usuario o correos electrónicos, informando sobre la existencia de los mismos en caso de detectarla. |
| **RNF-004** | El sistema deberá ser desarrollado utilizando software libre, empleando el lenguaje de programación JavaScript, el framework Angular y el estándar HTML para el diseño de las páginas web. Se usará Bootstrap para brindar estilos, asegurando que el código HTML sea compatible con los navegadores comerciales existentes. |
| **RNF-005** | El sistema deberá ser diseñado utilizando la arquitectura Modelo-Vista-Controlador (MVC). |
| **RNF-006** | El sistema deberá utilizar la herramienta json-server para almacenar y recuperar datos mediante llamados a una API REST, facilitando el desarrollo y las pruebas de la aplicación. |
| **RNF-007** | El sistema deberá ser responsivo, adaptando el tamaño y diseño de las páginas según la resolución de la pantalla en la que se visualiza. |

### ***4. Diagramas***

#### **4.1 Diagrama de Casos de Uso**



#### **4.1.1 Especificación de caso de uso**

A continuación, se presenta una especificación de caso de uso, referida a cuando un Anónimo se registre como usuario.

|  |  |
| --- | --- |
| **Registrar usuario** | |
| **Identificador** | RF-001 |
| **Nombre** | Registrar usuario |
| **Analistas** | -Telega Cristian  -Cabrera Jonathan David  -Schäffer Pablo Octavio |
| **Versión** | 1.0 |
| **Actores** | Anónimo |
| **Prioridad** | Alta |
| **Objetivos** | Permitir a un usuario anónimo crear una cuenta que almacene sus datos personales. |
| **Precondiciones** | * Poseer una cuenta de correo electrónico no registrada. * Ser mayor de edad. * Postcondiciones * Almacenar la nueva cuenta creada y redirigir al usuario a la página de inicio. |
| **Postcondiciones** | Almacenar la nueva cuenta creada y redirigir al usuario a la página de inicio. |

**4.1.2 Camino de éxito**

|  |  |  |
| --- | --- | --- |
| **Paso** | **Acción del actor** | **Acción del Sistema** |
| 1 | Ingresar información en los campos de "Información de cuenta". | Verifica que no existan el nombre de usuario y el correo electrónico ingresados, que la contraseña cumpla con los caracteres solicitados y que el campo "Repetir contraseña" coincida. |
| 2 | Ingresar información en los campos de "Información personal". | Verifica que los campos "Nombre" y "Apellido" contengan solo letras, que el "DNI" sea numérico y que la "Fecha de nacimiento" indique que el usuario es mayor de 18 años. |
| 3 | Ingresar información en los campos de "Información de contacto". | Verifica que los campos "Código de área" y "Número de teléfono" contengan solo números. |
| 4 | Ingresar información en los campos de "Información de dirección". | Verifica que el campo "Altura" sea numérico y que "Ciudad", "Provincia" y "País" contengan solo letras. |
| 5 | Seleccionar el botón "Registrarse". | Verifica que todos los campos obligatorios estén completos, almacena los datos bajo un identificador generado aleatoriamente y redirige al usuario a la página de inicio. |

**4.1.3 Excepciones**

|  |  |  |
| --- | --- | --- |
| **Nº** | **Acción de actor** | **Acción de Sistema** |
| 1- EXC | El anónimo ingresa un nombre de usuario que ya está en uso. | El sistema muestra un error de validación de campo, haciendo invalido el botón de “Registrarse” |
| 2- EXC | El anónimo ingresa un correo electrónico que ya está en uso. | El sistema muestra un error de validación de campo, haciendo invalido el botón de “Registrarse” |
| 3- EXC | El anónimo ingresa una contraseña. | El sistema muestra un error de validación de campo, especificando que validación no fue cumplida, haciendo invalido el botón de “Registrarse” |
| 4- EXC | El anónimo repite la contraseña. | El sistema muestra un error de validación de campo, haciendo invalido el botón de “Registrarse” |
| 5- EXC | El anónimo ingresa su información personal. | El sistema muestra un error de validación de campo, haciendo invalido el botón de “Registrarse” |
| 6- EXC | El anónimo ingresa su información de contacto | El sistema muestra un error de validación de campo, haciendo invalido el botón de “Registrarse” |
| 7- EXC | El anónimo ingresa su información de dirección. | El sistema muestra un error de validación de campo, haciendo invalido el botón de “Registrarse” |

#### **4.2 Modelo Conceptual**

