**Universidad Tecnológica del Perú**

**Facultad de Ingeniería**

**Ingeniería de Sistemas e Informática**

**DESARROLLO DE UNA PLATAFORMA WEB PARA UNA TIENDA DE DISPOSITIVOS** **ELECTRÓNICOS**

**Avance 02**

**Curso:**  
Desarrollo web Integrado

**Integrantes:**

* Cusi Vargas Jesus Arturo U21225686
* Espino Aguilar Yadira Saely U21302193
* Espino Aguilar Yadira Saely U21302193
* Espino Aguilar Yadira Saely U21302193

**Docente:**

Jose Antonio Espinal Teves

**Sección:**

24698

**LIMA – PERÚ**

**2024**

**ÍNDICE**

[**1.**](#_gjdgxs) **INTRODUCCIÓN** 4

[**1.1.**](#_30j0zll) **Objetivos del Proyecto Web** 4

[**1.2.**](#_1fob9te) **Justificación del Proyecto** 5

[**1.3.**](#_3znysh7) **Estado del Arte (Estudios Relacionados)** 5

[1.3.1. Antecedentes 5](#_2et92p0)

[1.3.2. Problema 6](#_tyjcwt)

[1.3.3. Discusión 6](#_3dy6vkm)

[**1.4.**](#_1t3h5sf) **Descripción del Problema** 7

[**2.**](#_4d34og8) **MARCO TEÓRICO** 7

[**2.1.**](#_2s8eyo1) **Conceptos Fundamentales** 7

[2.1.1. Desarrollo web integrado (JAVA EE) 7](#_17dp8vu)

[2.1.2. Arquitectura DAO, MVC 8](#_3rdcrjn)

[2.1.3. Resumen 8](#_lnxbz9)

[**2.2.**](#_1ksv4uv) **Tecnologías Utilizadas** 8

[2.2.1. Frontend (HTML, CSS, JavaScript, Frameworks) 8](#_44sinio)

[2.2.2. Backend (Java EE, Servlets, JSP, APIs RESTful, Sprint Boot) 9](#_z337ya)

[2.2.3. Base de Datos (MySQL, MongoDB) 9](#_1y810tw)

[2.2.4. Herramientas de Desarrollo (Java,VsCode) 9](#_2xcytpi)

[**3.**](#_3whwml4) **ANÁLISIS Y DISEÑO DEL SISTEMA** 10

[**3.1.**](#_2bn6wsx) **Requerimientos del Sistema** 10

[3.1.1. Requerimientos Funcionales 10](#_qsh70q)

[3.1.2. Requerimientos No Funcionales 11](#_3as4poj)

[**3.2.**](#_1pxezwc) **Análisis de Casos de Uso** 12

[3.2.1. Diagrama de Casos de Uso 12](#_49x2ik5)

[3.2.2. Descripción de Casos de Uso (Larman) 12](#_2p2csry)

[**3.3.**](#_147n2zr) **Diseño del Sistema** 14

[3.3.1. Diagrama de Clases 14](#_3o7alnk)

[3.3.2. Diagrama de Entidad-Relación (ER) 15](#_23ckvvd)

[3.3.3. Diseño de la Base de Datos 16](#_ihv636)

[3.3.4. Diseño de la Interfaz de Usuario 16](#_32hioqz)

[17](#_1hmsyys)

[3.3.5. Cronograma 17](#_41mghml)

[**4.**](#_2grqrue) **DESARROLLO DEL PROYECTO WEB** 20

[**4.1.**](#_vx1227) **Estructura de Carpetas y Archivos (Git Hub)** 20

[4.1.1. Main/ - Código Fuente 20](#_3fwokq0)

[4.1.2. Branchs/ - Codigo Fuente 20](#_1v1yuxt)

[**4.2.**](#_4f1mdlm) **Implementación del Backend** 20

[4.2.1. Configuración del Servidor 20](#_2u6wntf)

[4.2.2. Desarrollo de Componentes 20](#_19c6y18)

[4.2.3. Integración con la Base de Datos 20](#_3tbugp1)

[4.2.4. Manejo de Sesiones y Autenticación 20](#_28h4qwu)

[**4.3.**](#_nmf14n) **Implementación del Frontend** 20

[4.3.1. Desarrollo de Plantillas HTML 20](#_37m2jsg)

[4.3.2. Estilos con CSS 20](#_1mrcu09)

[4.3.3. Interactividad con JavaScript 20](#_46r0co2)

[4.3.4. Consumo de APIs 20](#_2lwamvv)

[**4.4.**](#_111kx3o) **Pruebas del Sistema** 20

[4.4.1. Pruebas Unitarias 20](#_3l18frh)

[4.4.2. Pruebas de Integración 20](#_206ipza)

[4.4.3. Pruebas de Usuario (Usabilidad) 20](#_4k668n3)

**DESARROLLO DE UNA PLATAFORMA WEB PARA UNA TIENDA DE DISPOSITIVOS ELECTRÓNICOS**

# **INTRODUCCIÓN**

En el contexto actual de transformación digital, el comercio electrónico es crucial para el crecimiento y competitividad de las empresas, especialmente en el sector de dispositivos electrónicos. Este proyecto tiene como objetivo desarrollar una plataforma web eficiente para una tienda de dispositivos electrónicos, mejorando la experiencia de compra en línea y optimizando la gestión de inventarios y pedidos en tiempo real.

La plataforma permitirá transacciones seguras y accesibles desde cualquier lugar, ampliando el alcance de la tienda y facilitando su expansión en el entorno digital. Utilizando tecnologías como Java EE para el backend y frameworks modernos en el frontend, la solución será escalable, segura y adaptable a futuros avances tecnológicos.

## **Objetivos del Proyecto Web**

**OBJ 1.-** Desarrollar una plataforma web que permita a los usuarios realizar compras de dispositivos electrónicos de manera segura, eficiente y accesible, optimizando los procesos de venta e inventario.

**OBJ 2.-** Mejorar la competitividad de la tienda mediante la implementación de tecnologías avanzadas que permitan la automatización de la gestión de inventarios y pedidos, garantizando la seguridad de los datos y la satisfacción del cliente.

## **Justificación del Proyecto**

El comercio electrónico ha experimentado un crecimiento significativo, impulsado por la transformación digital y el cambio en las preferencias de los consumidores hacia las compras en línea. En este contexto, la implementación de una plataforma web para una tienda de dispositivos electrónicos no solo optimiza la gestión de inventarios y pedidos, sino que también amplía el alcance de la tienda al permitir que los clientes realicen compras desde cualquier lugar y en cualquier momento​.

El proyecto responde a la necesidad de las empresas de adaptarse a las nuevas demandas del mercado, donde la velocidad, eficiencia y seguridad son claves para la satisfacción del cliente. La automatización de procesos, como la actualización de inventarios y el seguimiento de ventas, no solo reduce los errores humanos, sino que también permite una mejor toma de decisiones basada en datos en tiempo real​. De esta forma, se contribuye a mejorar la competitividad y sostenibilidad de la tienda, generando un impacto positivo en su crecimiento.

## **Estado del Arte (Estudios Relacionados)**

### **Antecedentes**

El comercio electrónico ha crecido de manera significativa a nivel mundial, impulsado por la digitalización y el cambio de hábitos de los consumidores. En Perú, su adopción en las PYMES ha permitido que las empresas mejoren su competitividad y amplíen sus mercados. Según estudios recientes, las plataformas e-commerce han sido clave para el crecimiento de las empresas, facilitando pagos y personalizando las ofertas para mejorar la experiencia del usuario​.

### **Problema**

La tienda de dispositivos electrónicos no cuenta con una plataforma de comercio electrónico, lo que limita su alcance y competitividad en un mercado cada vez más digitalizado. La ausencia de un sistema de venta en línea reduce su capacidad para captar a un público más amplio, especialmente a aquellos clientes que prefieren realizar sus compras de forma virtual. Además, la falta de automatización en la gestión de inventarios y pedidos genera ineficiencias operativas, errores en el control de stock y una dificultad para proporcionar información precisa y actualizada sobre los productos disponibles. Esto afecta negativamente la experiencia del cliente, reduce la satisfacción y compromete la posibilidad de fidelización a largo plazo.​

### **Discusión**

Las plataformas de comercio electrónico se han convertido en una necesidad para las empresas que buscan mantenerse competitivas en el entorno actual. El desarrollo de una plataforma web no solo optimiza los procesos internos, como la gestión de inventarios y pedidos, sino que también ofrece la posibilidad de implementar estrategias de marketing digital efectivas. Estas plataformas permiten a las empresas recopilar y analizar datos en tiempo real, lo que facilita la toma de decisiones más informadas y mejora la experiencia del cliente​.

En cuanto a las tecnologías, es fundamental que estas plataformas están construidas sobre sistemas robustos y escalables, como Java EE, que permite integrar tanto el frontend como el backend de manera eficiente. El uso de herramientas de automatización y tecnologías emergentes, como el IoT y la Inteligencia Artificial, también se han vuelto cruciales para mejorar la personalización de las ofertas y garantizar la seguridad de las transacciones​.

## **Descripción del Problema**

La tienda de dispositivos electrónicos enfrenta el reto de no contar con una plataforma de comercio electrónico, lo que limita su capacidad para llegar a un público más amplio que prefiere realizar compras en línea. Esto resulta en una desventaja competitiva en un mercado donde el comercio electrónico está en auge, afectando tanto las ventas como la expansión de la tienda más allá de su local físico​. Además, la gestión del inventario y los pedidos se realiza manualmente, lo que genera ineficiencias, errores en el control de stock y dificulta ofrecer información actualizada sobre los productos disponibles​.

La falta de automatización y de presencia en línea impide que la tienda implemente estrategias de marketing digital efectivas y aproveche herramientas como el análisis de datos, lo que podría mejorar tanto la experiencia del cliente como la optimización interna de los procesos​. En resumen, la ausencia de una plataforma digital no solo afecta la eficiencia operativa, sino que también limita el crecimiento y sostenibilidad de la tienda en un entorno de alta competencia.

# **MARCO TEÓRICO**

## **Conceptos Fundamentales**

### **Desarrollo web integrado (JAVA EE)**

Para crear aplicaciones web efectivas, es esencial entender algunos conceptos básicos. Estos incluyen cómo se estructuran las aplicaciones en la web, cómo diseñarlas para que sean fáciles de usar y cómo las tecnologías permiten que los usuarios y el servidor se comuniquen. En esta oportunidad usaremos el lenguaje Java para realizar este proyecto, ya que es un lenguaje muy flexible y robusta en nuestro proyecto.

### **Arquitectura DAO, MVC**

Java EE (Enterprise Edition) es una plataforma diseñada para desarrollar aplicaciones web grandes y complejas. Con Java EE, se pueden construir aplicaciones que manejan muchos usuarios y grandes volúmenes de datos de manera eficiente. Usa tecnologías como Servlets y JSP para gestionar cómo se muestran los datos en la web y APIs RESTful para facilitar la comunicación entre diferentes partes de la aplicación. Java EE también facilita la integración de varios componentes de software, lo que hace que las aplicaciones sean más organizadas y fáciles de mantener.

### **Resumen**

En resumen, para desarrollar aplicaciones web modernas, es fundamental entender los conceptos básicos y cómo aplicar ciertos patrones de diseño. Usar Java EE para el desarrollo integrado y patrones como DAO y MVC ayuda a construir aplicaciones que sean robustas y fáciles de mantener. Con estos conocimientos, podemos crear aplicaciones web que funcionen bien y satisfagan las necesidades de los usuarios.

## **Tecnologías Utilizadas**

### **Frontend (HTML, CSS, JavaScript, Frameworks)**

El frontend es lo que los usuarios ven e interactúan en una página web. Utilizamos HTML para construir la estructura de las páginas, CSS para darle estilo y diseño, y JavaScript para agregar interactividad. Frameworks como React, Angular o Vue.js ayudan a desarrollar interfaces de usuario más dinámicas y eficaces, haciendo el proceso de desarrollo más ágil y organizado.

### **Backend (Java EE, Servlets, JSP, APIs RESTful, Sprint Boot)**

El backend es el cerebro detrás de la aplicación web. Aquí es donde se maneja la lógica y los datos. Java EE es una plataforma que ayuda a construir el backend de aplicaciones grandes. Usamos Servlets y JSP para gestionar cómo se procesan y muestran los datos. Las APIs RESTful permiten que diferentes partes de la aplicación se comuniquen entre sí. Spring Boot es un marco de trabajo que simplifica la creación de servicios web, haciéndolos más rápidos de desarrollar y más fáciles de gestionar.

### **Base de Datos (MySQL, MongoDB)**

Las bases de datos son esenciales para almacenar y organizar los datos. MySQL es una base de datos relacional que maneja datos estructurados de manera eficiente. MongoDB, en cambio, es una base de datos NoSQL que permite almacenar datos en un formato más flexible, ideal para datos que no encajan en un esquema rígido.

### **Herramientas de Desarrollo (Java,VsCode)**

Las herramientas de desarrollo nos ayudan a escribir y gestionar el código de nuestras aplicaciones. Java es un lenguaje de programación popular para el desarrollo de backend, conocido por su robustez y versatilidad. Visual Studio Code (VSCode) es un editor de código que hace que la programación sea más fácil y rápida, gracias a sus muchas extensiones y funcionalidades que mejoran la productividad que en nuestro proyecto nos beneficia con su facilidad de código.

# **ANÁLISIS Y DISEÑO DEL SISTEMA**

## **Requerimientos del Sistema**

### **Requerimientos Funcionales**

1. **Registro y Autenticación de Usuarios:**

* Los usuarios podrán registrarse en el sistema con una cuenta personal.
* Los usuarios podrán iniciar sesión y cerrar sesión en su cuenta.
* Los usuarios podrán recuperar su contraseña si la olvidan.

1. **Gestión de Productos:**

* Solo los administradores deben poder añadir, editar y eliminar productos tecnológicos (celulares, laptops, etc.).
* Los productos tienen atributos como nombre, descripción, precio, y fotos.
* Los usuarios verán una lista de productos, filtrarlos y buscar productos específicos.

1. **Carrito de Compras:**

* Los usuarios podrán añadir productos al carrito de compras.
* Los usuarios deben poder ver y modificar los productos en su carrito (añadir o eliminar artículos, actualizar cantidades).

1. **Proceso de Compra:**

* Los usuarios podrán proceder al pago y completar la compra.
* El sistema deberá generar un resumen de la compra y un número de orden.
* Los usuarios recibirán una confirmación de la compra por correo electrónico.

1. **Métodos de Pago:**

* El sistema debe permitir pagos mediante tarjetas de crédito/débito y otros métodos de pago en línea (como PayPal).

1. **Gestión de Pedidos:**

* Los administradores deben poder ver y gestionar los pedidos realizados por los usuarios.

1. **Opiniones y Reseñas:**

* Los usuarios podrán dejar opiniones y reseñas sobre los productos que compraron.

### **Requerimientos No Funcionales**

1. **Usabilidad:**

* La página web será intuitiva y fácil de usar para los usuarios.
* Debe ser accesible desde diferentes dispositivos (computadoras, tablets, smartphones).

1. **Escalabilidad:**

* El sistema debe ser capaz de manejar un aumento en el número de productos y usuarios sin requerir rediseño significativo.

1. **Mantenimiento:**

* El código debe ser modular y bien documentado para facilitar el mantenimiento y la actualización futura.

1. **Compatibilidad:**

* La página debe ser compatible con los navegadores más utilizados (Chrome, Firefox, Safari, Edge).

## **Análisis de Casos de Uso**

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

### **Descripción de Casos de Uso (Larman)**

1. **Caso de Uso: Registro de Usuario**

**Nombre:** Registro de Usuario

**Actor:** Usuario

**Propósito:** Permitir a los usuarios crear una cuenta en el sistema para realizar compras.

**Descripción:**

* El usuario accede a la página de registro.
* El usuario ingresa su nombre, correo electrónico y una contraseña.
* El usuario envía el formulario de registro.
* El sistema valida la información y crea una nueva cuenta.
* El usuario recibe un correo de confirmación y puede iniciar sesión.

**Precondiciones:** El usuario debe tener acceso a internet y un correo electrónico válido.

**Postcondiciones:** El usuario tiene una cuenta registrada y puede iniciar sesión en el sistema.

**Escenario Principal:** Creación de una cuenta.

**Escenarios Alternativos:**

Si el correo electrónico ya está registrado, el sistema muestra un mensaje de error.

1. **Caso de Uso: Añadir Producto al Carrito**

**Nombre:** Añadir Producto al Carrito

**Actor:** Usuario

**Propósito:** Permitir a los usuarios añadir productos a su carrito de compras.

**Descripción:**

* El usuario navega por la lista de productos.
* El usuario selecciona un producto para ver los detalles.
* El usuario hace clic en "Añadir al Carrito."
* El sistema añade el producto al carrito y actualiza el conteo de artículos.
* El usuario puede ver el carrito con el producto añadido.

**Precondiciones:** El usuario debe estar en la página de un producto.

**Postcondiciones:** El producto se añade al carrito del usuario y está disponible para la compra.

**Escenario Principal:** Compra de un producto.

**Escenarios Alternativos:**

Si el producto está fuera de stock, el sistema muestra un mensaje informando de la falta de disponibilidad.

## **Diseño del Sistema**

### **Diagrama de Clases**

### **Diagrama de Entidad-Relación (ER)**

### **Diseño de la Base de Datos**

### **Diseño de la Interfaz de Usuario**

### 

### **Cronograma**

**Semana 1-2: Introducción**

* **Objetivos del Proyecto Web:** Definir qué se pretende lograr con el proyecto.
* **Justificación del Proyecto:** Explicar la necesidad y el valor del proyecto.
* **Estado del Arte:** Revisar estudios y proyectos similares.
* **Descripción del Problema:** Identificar el problema que se está abordando.

**Semana 3-4: Marco Teórico**

* **Conceptos Fundamentales:** Detallar la arquitectura y tecnologías clave.
* **Tecnologías Utilizadas:** Explicar las herramientas y tecnologías que se utilizarán en el proyecto.

**Semana 5-8: Análisis y Diseño del Sistema**

* **Requerimientos del Sistema:** Especificar los requisitos funcionales y no funcionales.
* **Casos de Uso:** Crear diagramas y descripciones de los casos de uso del sistema.
* **Diseño del Sistema:** Desarrollar diagramas de clases, ER, y el diseño de la base de datos e interfaz.

**Semana 9-13: Desarrollo del Proyecto Web**

* **Estructura de Carpetas y Archivos:** Organizar el código en GitHub.
* **Backend:** Configurar el servidor, desarrollar componentes y gestionar la base de datos.
* **Frontend:** Desarrollar la interfaz de usuario, incluyendo HTML, CSS, JavaScript y APIs.

**Semana 14: Pruebas del Sistema**

* Realizar pruebas unitarias, de integración y de usabilidad para asegurar que todo funcione correctamente.

**Semana 15: Despliegue y Mantenimiento**

* Preparar y desplegar el sistema en un entorno de producción simulado y configurar el servidor. Planificar el mantenimiento y futuras actualizaciones.

**Semana 16: Evaluación y Resultados**

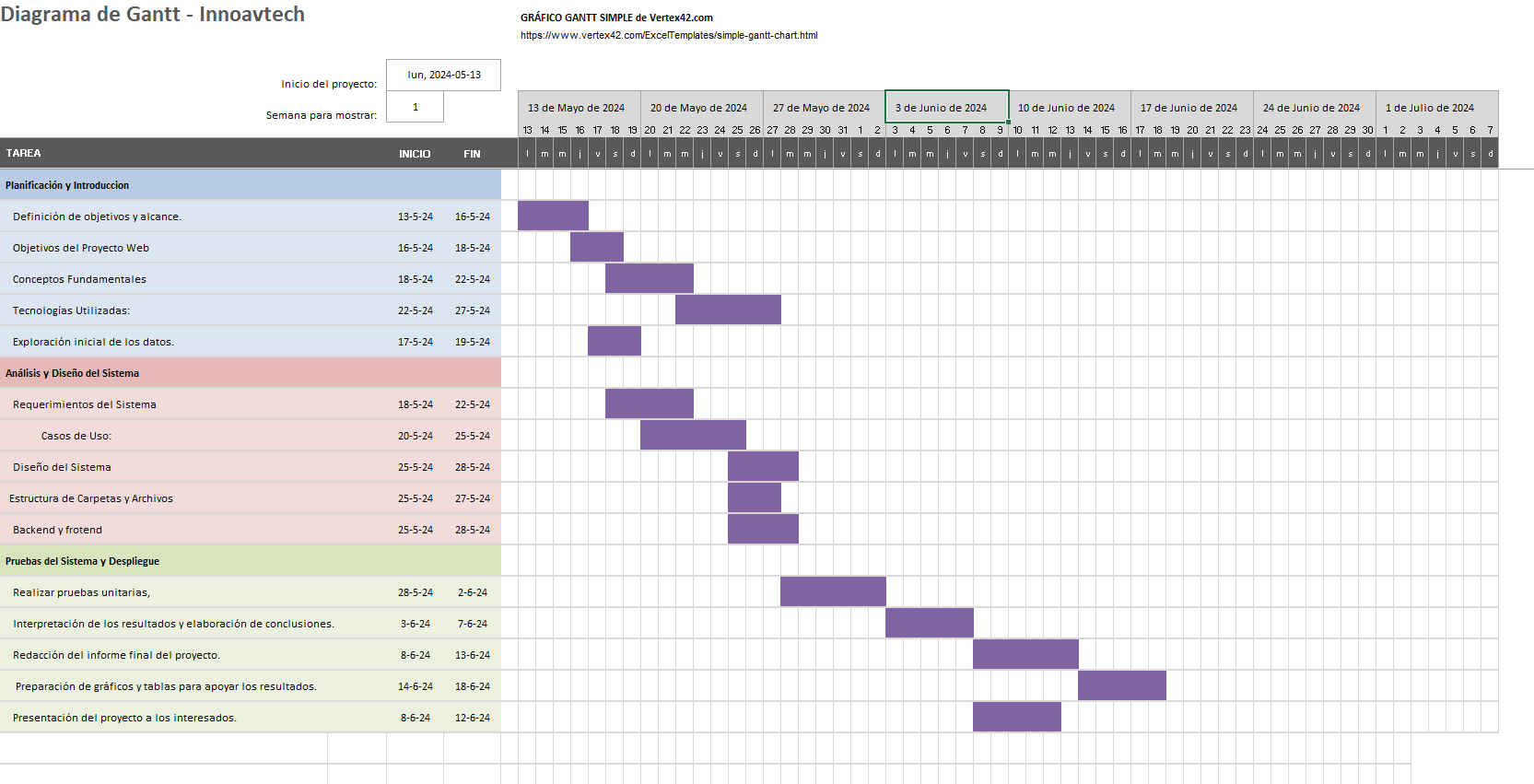
* Evaluar el rendimiento y seguridad del sistema, revisar los resultados obtenidos y compararlos con los objetivos iniciales.

**Semana 16: Conclusiones y Recomendaciones**

* Redactar conclusiones generales, identificar limitaciones del proyecto y proporcionar recomendaciones para futuros trabajos.

**Semana 16: Referencias y Apéndices**

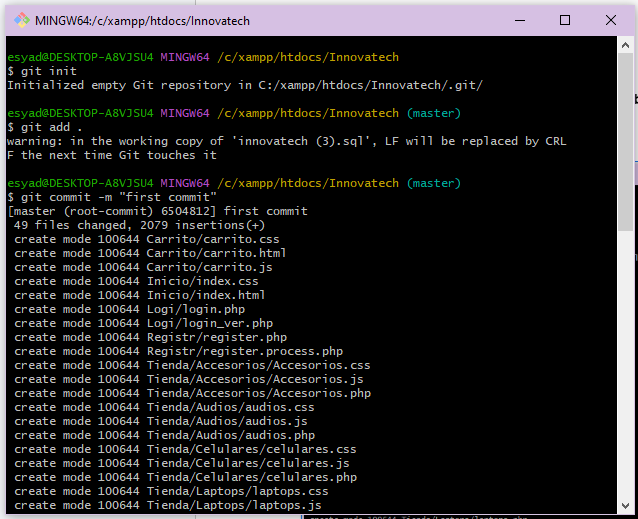
* Incluir toda la bibliografía, sitios web consultados, documentación técnica y cualquier apéndice relevante como código fuente y manuales de usuario.

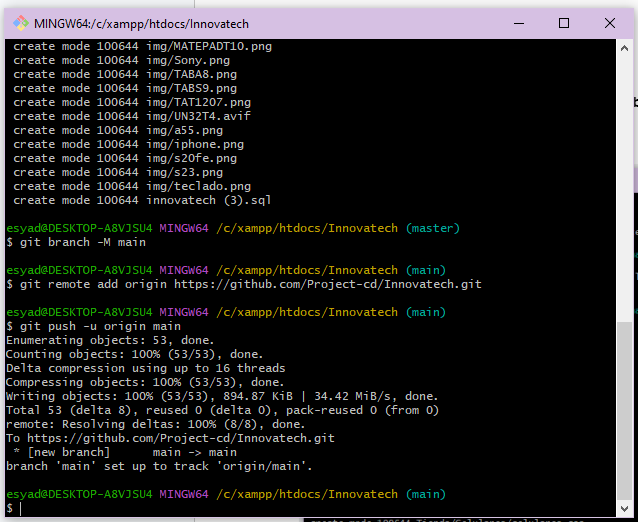
**Diagrama de Gantt**

# **DESARROLLO DEL PROYECTO WEB**

## **Estructura de Carpetas y Archivos (Git Hub) YADIRA**

### **Main/ - Código Fuente**





### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

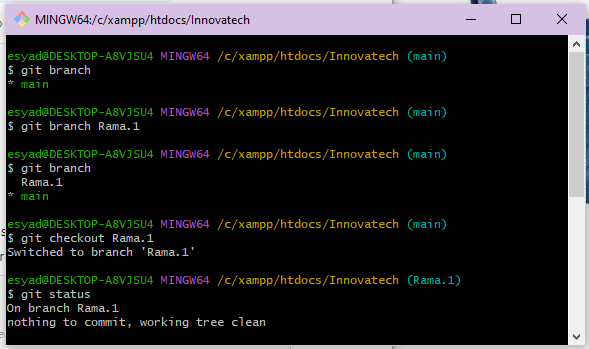
### 

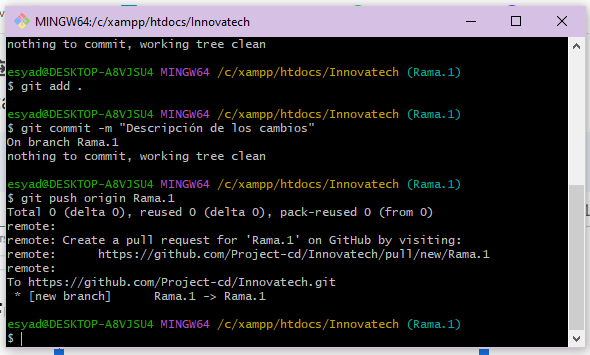
### 

### 

### 

### **Branchs/ - Código Fuente**





## **Implementación del Backend GILBERT**

### **Configuración del Servidor**

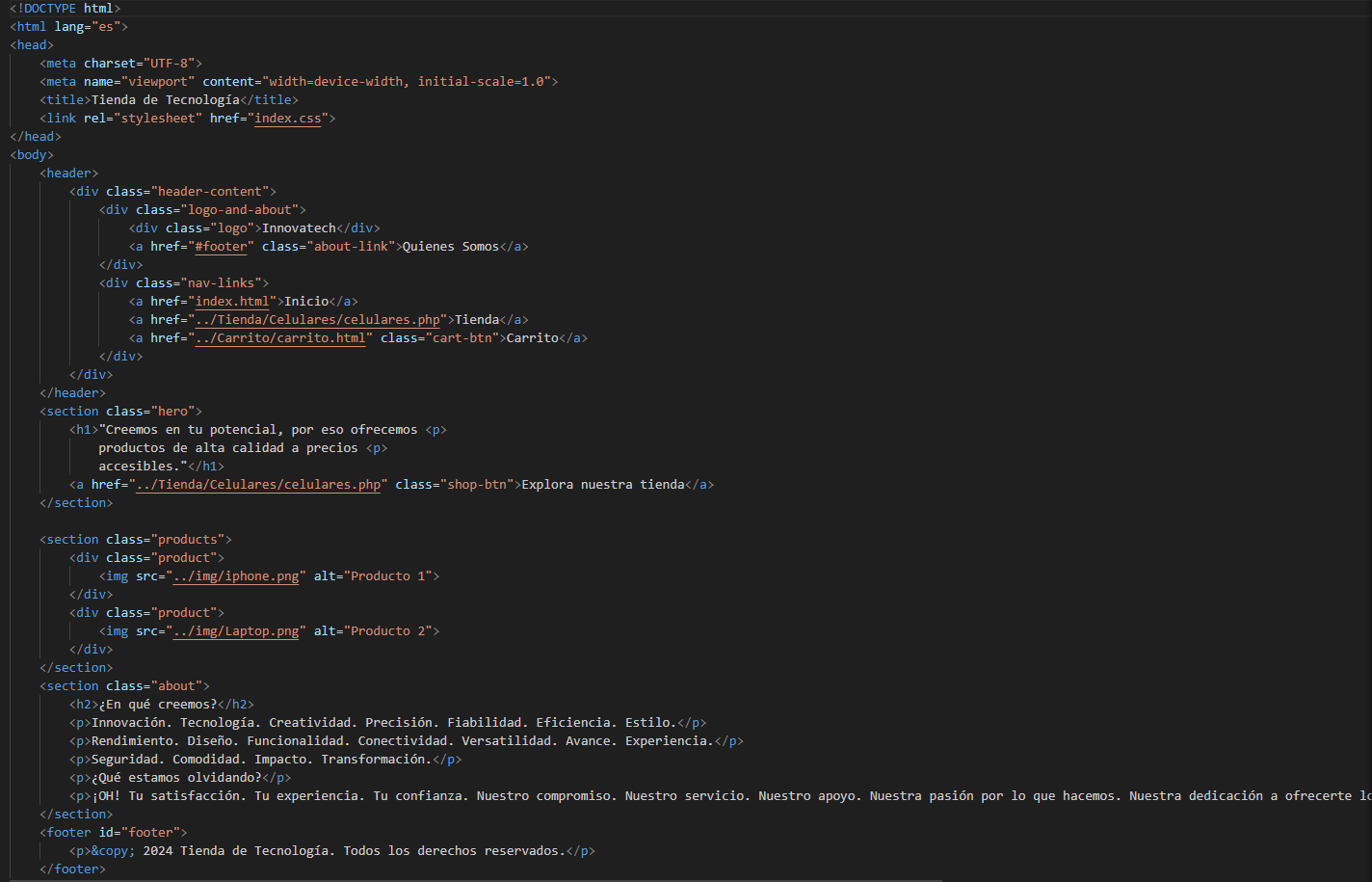
### **Desarrollo de Componentes**

### **Integración con la Base de Datos**

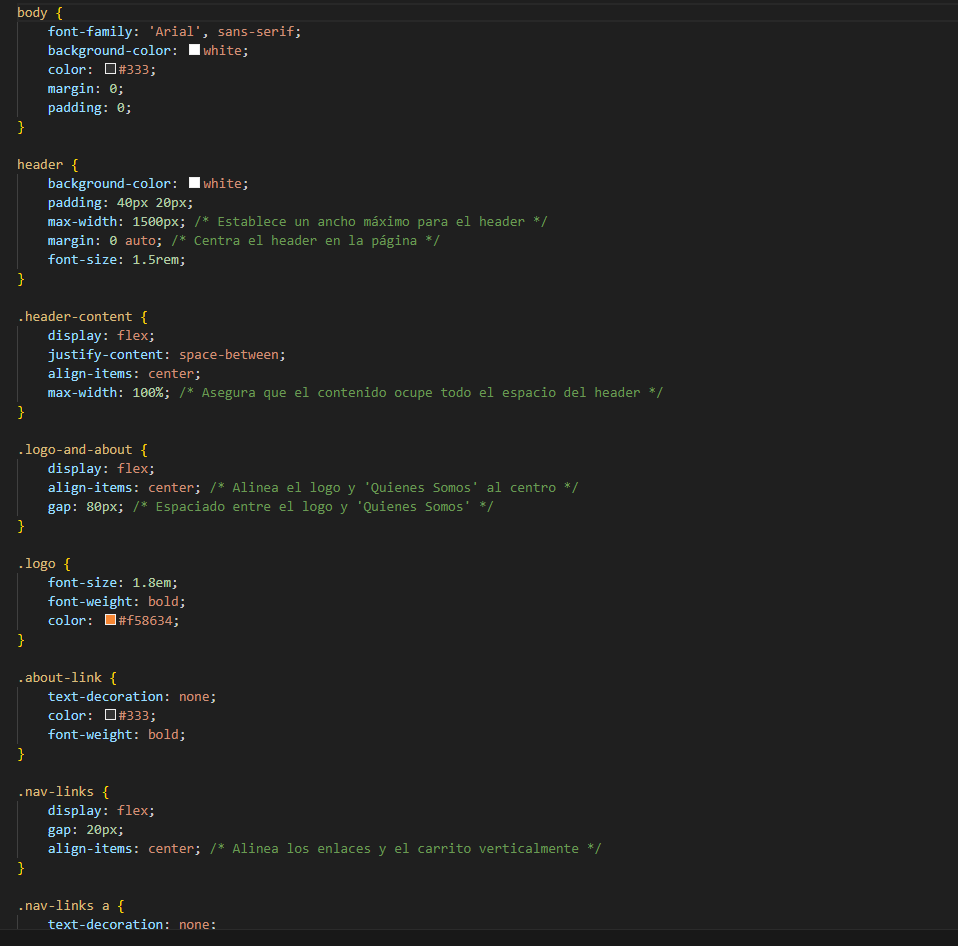
### **Manejo de Sesiones y Autenticación**

## **Implementación del Frontend Jesus**

### **Desarrollo de Plantillas HTML**

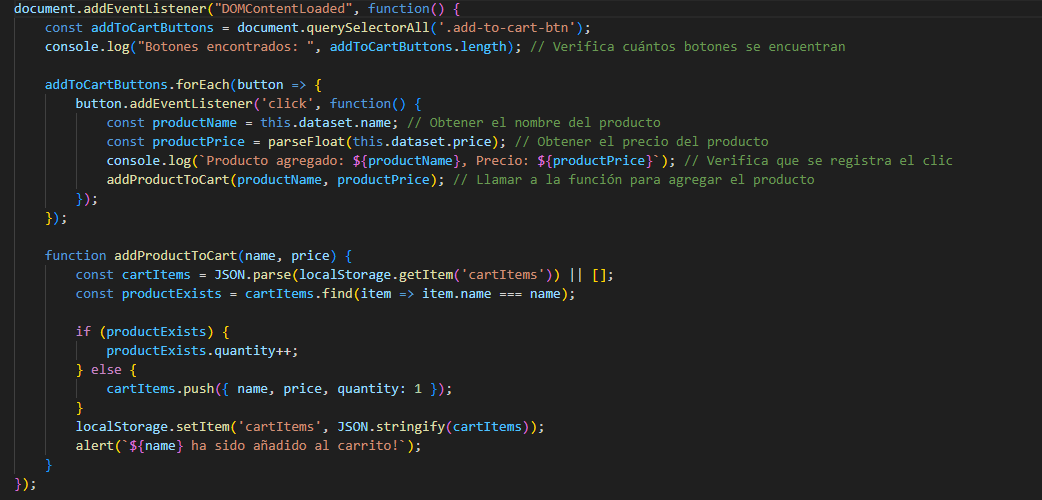


### **Estilos con CSS**

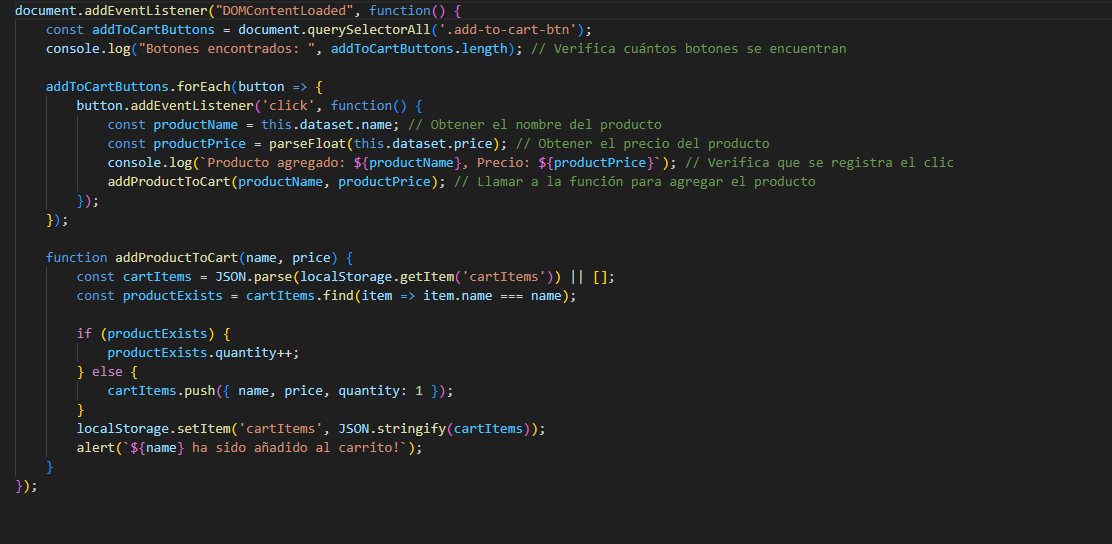
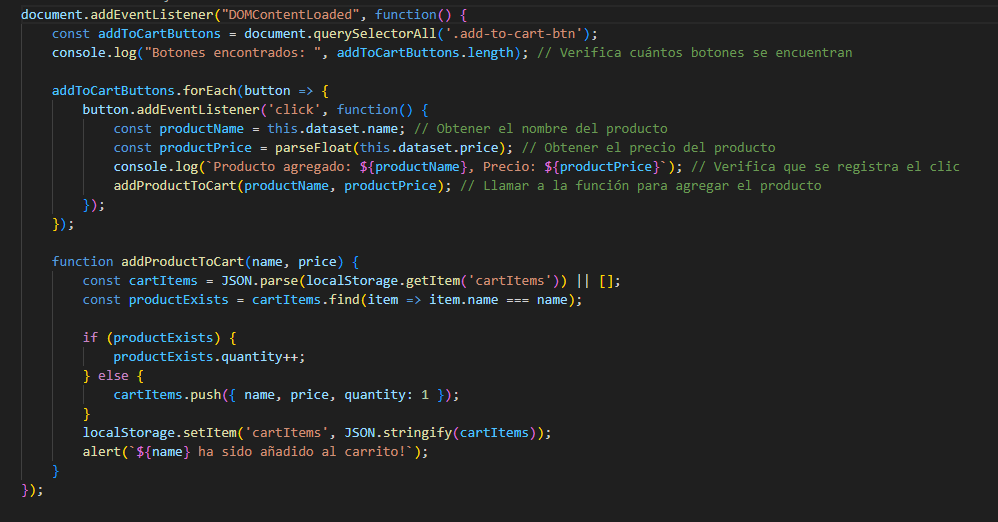


### **Interactividad con JavaScript**

Accesorios



CELULARES



### **Consumo de APIs**

## **Pruebas del Sistema**

### **Pruebas Unitarias**

### **Pruebas de Integración**

### **Pruebas de Usuario (Usabilidad)**

**5. REFERENCIA**