

## PROYECTO INTEGRADO



Alejandro Toral Romero

## Abstract:

El proyecto consiste en una web de productos y servicios fotovoltaicos, en la cual se incluye tanto la parte del cliente como la parte del trabajador de la tienda desde la cual podrá gestionar el funcionamiento de la tienda, teniendo un control de pedidos, usuarios, ingresos entre otras cosas.

Dentro de las funciones que incluye la web los usuarios podrán comprar productos, solicitar servicios, escribir a atención al cliente en caso de dudas y ver la información relacionada con su cuenta.

## Inglés:

The project consists of a website for photovoltaic products and services, which includes both the client-facing part and the store employee part. The latter allows managing store operations such as order control, user management, income tracking, among other functionalities.

Among the website's features, users can purchase products, request services, contact customer support for inquiries, and view information related to their account.

# Índice

<b>1. Justificación.....</b>	<b>4</b>
1.1 Características Generales.....	4
1.2 Restricciones Generales.....	4
1.3 Aspectos a cubrir, junto con los que no se van a tratar.....	5
1.4 Estudio de las prestaciones de la herramienta que se propone frente a otras existentes en la misma categoría.....	5
<b>2. Justificación de la tecnología empleada.....</b>	<b>7</b>
<b>3. Requerimientos Hardware y Software.....</b>	<b>8</b>
<b>4. Análisis y diseño.....</b>	<b>9</b>
4.1 Diagrama Casos de uso y Diagrama de clases.....	9
4.1.1 Diagrama Casos de uso.....	9
4.1.2 Diagrama de clases.....	10
4.2 Base de Datos Relacional.....	11
<b>5. Implementación.....</b>	<b>12</b>
<b>6. Evaluación y prueba.....</b>	<b>13</b>
<b>10. Manual de Estilos.....</b>	<b>15</b>
10.1 Sketches.....	15
10.2 Criterios de accesibilidad.....	19
10.3 Criterios de usabilidad.....	19
10.4 Tipografía.....	20
10.5 Mapa de colores.....	21
10.6 Dispositivos/vistas para las que se ha diseñado el proyecto.....	22
<b>11 Software Utilizado.....</b>	<b>23</b>
<b>12 Mejoras posibles y aportaciones.....</b>	<b>24</b>
Funcionalidad:.....	24
Organización de Código:.....	25
Organización de la Información:.....	26

# 1. Justificación

## 1.1 Características Generales

La página web está diseñada para ofrecer una solución integral para la venta de productos fotovoltaicos y la solicitud de servicios relacionados. Sus características principales incluyen:

-Sistema de Registro y Login: Proceso de registro y autenticación para garantizar la seguridad de la información del usuario. Los usuarios pueden registrarse y autenticarse manualmente o utilizando autenticación de Google.

-Carrito de Compras y Gestión de Pedidos: Los usuarios pueden añadir productos a un carrito de compras y proceder a realizar pedidos, generando facturas automáticas que se envían por correo electrónico.

-Servicios Personalizados: Los usuarios pueden solicitar servicios de planificación, instalación y mantenimiento de sistemas fotovoltaicos.

-Panel de Control de Usuarios: Los usuarios pueden modificar su información personal, revisar su historial de pedidos y descargar facturas.

-Datos en Tiempo Real: Información actualizada sobre el rendimiento de las instalaciones fotovoltaicas del usuario.

-Panel de Control para Empleados: Gestión de empleados, clientes, productos, pedidos e ingresos diarios, así como un sistema de chat en tiempo real para atención al cliente.

## 1.2 Restricciones Generales

Al tratarse de una web ficticia, no se puede incluir una pasarela de pago para poder realizar los pedidos, lo cual en una página web real sí haría falta. En la web el usuario puede ver en tiempo real lo que está produciendo su instalación, para poder probarlo se ha hecho una simulación a pequeña escala lo cual no proporciona datos reales de lo que sería una instalación en una vivienda ya que para esto habría que crear un hardware que se conecte a una instalación real de una vivienda.

### **1.3 Aspectos a cubrir, junto con los que no se van a tratar**

Aspectos a cubrir:

- Venta de productos fotovoltaicos.
- Solicitud y gestión de servicios de planificación, instalación y mantenimiento.
- Registro y autenticación de usuarios, incluyendo autenticación de Google.
- Gestión de pedidos y generación de facturas.
- Panel de control para usuarios y empleados.
- Sistema de chat en tiempo real para atención al cliente.
- Gráficas de ingresos diarios y actuales.

Aspectos No Cubiertos:

- Logística de envío de productos.
- Gestión de Stock de productos
- Soporte para múltiples idiomas o monedas.
- Marketing y promoción de productos.
- Asistencia técnica detallada para productos fuera del ámbito de los servicios ofrecidos.

### **1.4 Estudio de las prestaciones de la herramienta que se propone frente a otras existentes en la misma categoría.**

-Especialización en Fotovoltaica:

Herramienta Propuesta: Diseñada específicamente para productos y servicios fotovoltaicos, proporcionando una experiencia personalizada y relevante.

Otras Soluciones: Plataformas como Shopify y WooCommerce no están especializadas y requieren personalización adicional.

-Integración de Servicios y Productos:

Herramienta Propuesta: Permite la compra de productos y la solicitud de servicios de planificación, instalación y mantenimiento en una sola plataforma.

Otras Soluciones: Generalmente enfocadas solo en la venta de productos, requieren integraciones adicionales para servicios especializados.

#### -Panel de Control Completo:

Herramienta Propuesta: Ofrece gestión integral de información personal, pedidos y datos en tiempo real para usuarios, y administración de clientes, productos y pedidos para empleados.

Otras Soluciones: Carecen de funcionalidades específicas para la gestión de servicios y datos en tiempo real sin extensiones.

#### - Autenticación Dual:

Herramienta Propuesta: Proporciona opciones de autenticación manual y mediante Google, mejorando la accesibilidad y seguridad.

Otras Soluciones: Pueden requerir configuraciones adicionales para integrar autenticación mediante Google.

#### -Atención al Cliente en Tiempo Real:

Herramienta Propuesta: Utiliza Firebase para ofrecer chat en tiempo real, mejorando la comunicación con el cliente.

Otras Soluciones: Dependientes de plugins o servicios externos, lo que puede incrementar la complejidad y el costo.

#### -Tecnologías Modernas:

Herramienta Propuesta: Utiliza React, Tailwind CSS y Laravel para una arquitectura moderna y eficiente.

Otras Soluciones: Menos flexibles y modernas, con posibles dificultades en personalización y mantenimiento.

#### -Gráficas de Ingresos:

Herramienta Propuesta: Incluye gráficos de ingresos diarios y actuales para monitoreo financiero en tiempo real.

Otras Soluciones: Ofrecen reportes financieros pero requieren herramientas adicionales para gráficos en tiempo real.

## 2. Justificación de la tecnología empleada

Las tecnologías principales utilizadas en el proyecto son React, Tailwind, Laravel y Firebase. Este conjunto de tecnologías se emplea para crear una web fácil de mantener y escalable.

### Frontend:

React: Framework utilizado con JavaScript nativo y Vite para el lanzamiento del proyecto en desarrollo y producción. Su capacidad para crear interfaces dinámicas y eficientes lo hace ideal para este proyecto.

#### Librerías utilizadas:

Axios: Para peticiones HTTP, con ventajas sobre jQuery o Fetch.

Tremor: Para la utilización de componentes Tailwind como gráficos.

React-OAuth: Para la autenticación de usuario con Google.

Firebase: Para crear un chat en tiempo real.

JWT-decode: Para el descifrado de JWT.

Validator: Para la validación de campos en los formularios.

### Backend:

Laravel: Framework PHP utilizado para crear una API propia, lo que permite agregar funciones de forma rápida y eficiente en el futuro. Para la generación de facturas PDF se ha utilizado FPDF y para el envío de correos PHPMailer.

Firebase: Utilizado para la creación de un chat en tiempo real, proporcionando mayor eficiencia en comparación con una base de datos MySQL.

### Diseño:

Tailwind CSS: Utilizado por su fácil mantenimiento y adaptabilidad para dispositivos, permitiendo un diseño responsivo y moderno.

### 3. Requerimientos Hardware y Software

#### Requerimientos de Software:

- Node.js: Necesario para la instalación de React y Tailwind mediante el gestor de paquetes NPM.
- PHP (versión 8.2): Requerido para el backend con Laravel 11, ya que versiones inferiores pueden tener conflictos.
- Composer: Para la gestión de dependencias y creación del proyecto de Laravel.
- Base de Datos MySQL: Para el almacenamiento de datos del sitio web.
- Firebase: Creación de una cuenta para obtener la API Key, necesaria para el sistema de chat en tiempo real.

#### Requerimientos de Hardware:

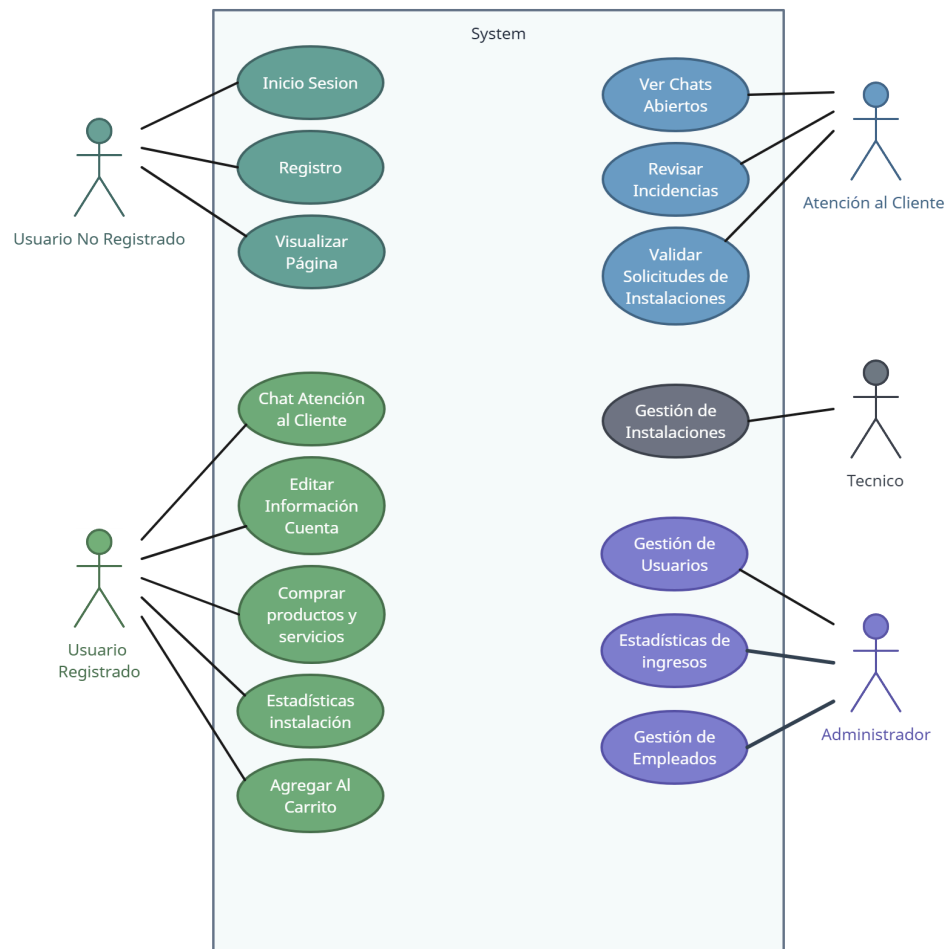
- Servidor Web: Para hostear la web y la api de laravel simultáneamente, con suficiente capacidad de procesamiento para manejar múltiples solicitudes concurrentes.
- Conexión a Internet: Estable y rápida, especialmente para el uso de Firebase y la transferencia de datos en tiempo real.



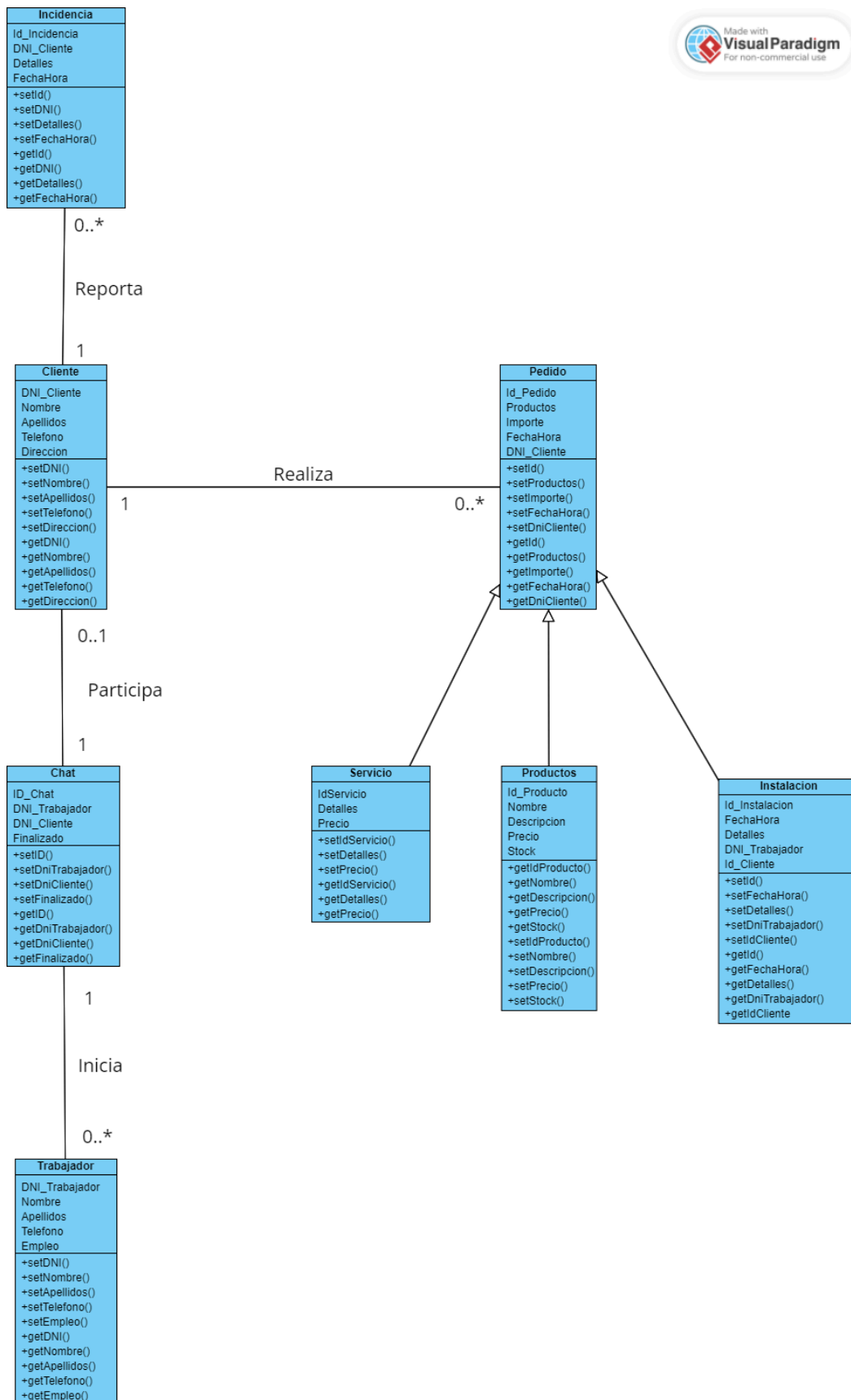
## 4. Análisis y diseño

### 4.1 Diagrama Casos de uso y Diagrama de clases.

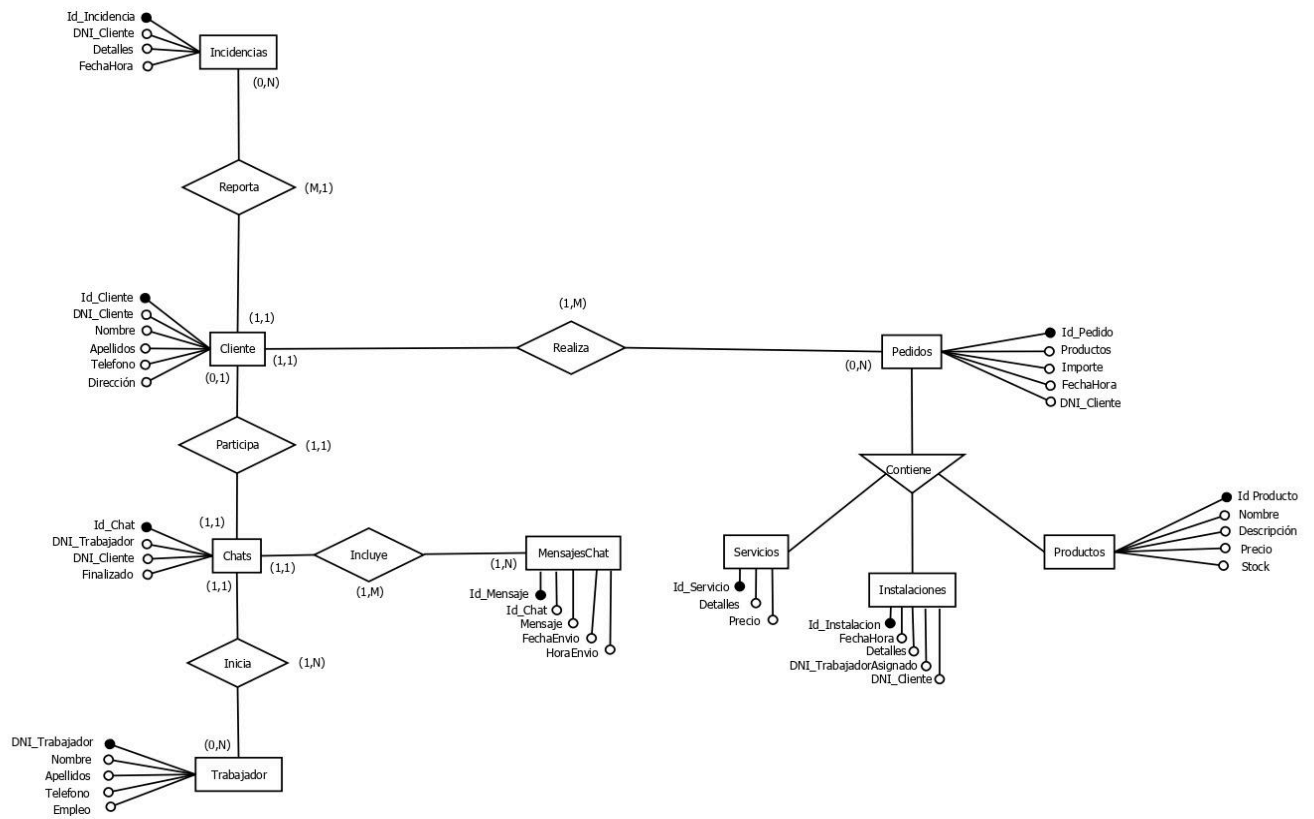
#### 4.1.1 Diagrama Casos de uso



## 4.1.2 Diagrama de clases



## 4.2 Base de Datos Relacional



## 5. Implementación

- Hojas de Estilo

Tailwind CSS: Utilizado para la creación de un diseño responsivo y moderno. Tailwind permite un fácil mantenimiento y una rápida adaptación a distintos dispositivos gracias a su sistema de clases utilitarias.

- Plantillas:

Componentes de React: Se utilizan componentes reutilizables de React para crear la estructura y funcionalidad de la interfaz de usuario. Estos componentes incluyen:

Header y Footer: Para la navegación y la estructura básica del sitio.

ProductList y ProductDetail: Para la visualización de productos.

ServiceRequestForm: Para solicitar servicios de planificación, instalación y mantenimiento.

- Formularios:

React Hook Form: Utilizado para la gestión de formularios en React, proporcionando una validación y manejo de formularios eficiente.

Registro y Login: Formularios para la autenticación de usuarios, tanto manual como con Google.

Carrito de Compras y Pago: Formularios para añadir productos al carrito y realizar el proceso de compra.

Solicitud de Servicios: Formularios para que los usuarios soliciten servicios específicos.

- Funciones de Envío de Datos:

Axios: Utilizado para realizar peticiones HTTP a la API de Laravel, manejando operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para productos, pedidos y usuarios.

Ejemplo: `axios.post('/api/orders', orderData)` para crear un nuevo pedido.

- Conexión:

API REST con Laravel: Laravel se utiliza para crear una API que maneja la lógica de negocio y la interacción con la base de datos.

Autenticación: Gestión de tokens JWT para la seguridad y autenticación de las peticiones.

- Consultas y Selección de la BD:

Eloquent ORM de Laravel: Utilizado para interactuar con la base de datos MySQL, facilitando la creación de consultas complejas y la gestión de relaciones entre tablas.

Ejemplo: `User::find($id)` para seleccionar un usuario específico de la base de datos.

- Chat en Tiempo Real:

Firebase: Implementación de un sistema de chat en tiempo real para la atención al cliente. Firebase proporciona una solución eficiente y escalable para la comunicación instantánea.

Ejemplo: Uso de Firestore para almacenar y sincronizar mensajes en tiempo real.

- Uso de Fichero de Configuración:

Environment Variables: Uso de archivos `.env` tanto en el frontend como en el backend para gestionar configuraciones sensibles y variables de entorno.

Backend: Configuración de la base de datos, claves de API y otros parámetros sensibles.

Frontend: Configuración de claves de API de Firebase y la URL de la API de Laravel.

## 6. Evaluación y prueba

- Verificación de Campos y Realimentación al Usuario:

Validación de Formularios: Se implementó la validación de campos en tiempo real utilizando `React Hook Form` y `Validator`.

Ejemplo: Verificación de correos electrónicos, contraseñas, y otros datos de entrada del usuario.

Mensajes de Error y Confirmación: Se proporcionan mensajes de error claros y específicos para campos no válidos y confirmaciones de acciones exitosas, mejorando la experiencia del usuario.

- Bateria de Ejemplos:

- Registro y Login:

Registro de Usuario: Pruebas de registro con datos válidos e inválidos.

Login: Pruebas de autenticación manual y con Google, incluyendo casos de error (contraseña incorrecta, cuenta no existente).

- Carrito de Compras y Pedidos:

Añadir al Carrito: Pruebas de añadir y eliminar productos del carrito.

Proceso de Compra: Pruebas de finalización del pedido, generación de factura y envío de correo electrónico.

- Solicitud de Servicios:

Solicitar Servicio: Pruebas de solicitud de planificación, instalación y mantenimiento con datos válidos e inválidos.

Confirmación de Solicitud: Verificación de la recepción y registro de solicitudes en la base de datos.

- Panel de Control de Usuarios:

Modificar Información Personal: Pruebas de actualización de datos del usuario.

Visualización de Pedidos: Verificación de la correcta visualización del historial de pedidos y descarga de facturas en PDF.

- Panel de Control de Empleados:

Gestión de Empleados: Pruebas de agregar, editar y eliminar empleados.

Gestión de Clientes y Productos: Pruebas de agregar, editar y eliminar clientes y productos.

Visualización de Pedidos e Ingresos: Verificación de la correcta visualización de la lista de pedidos y gráficos de ingresos.

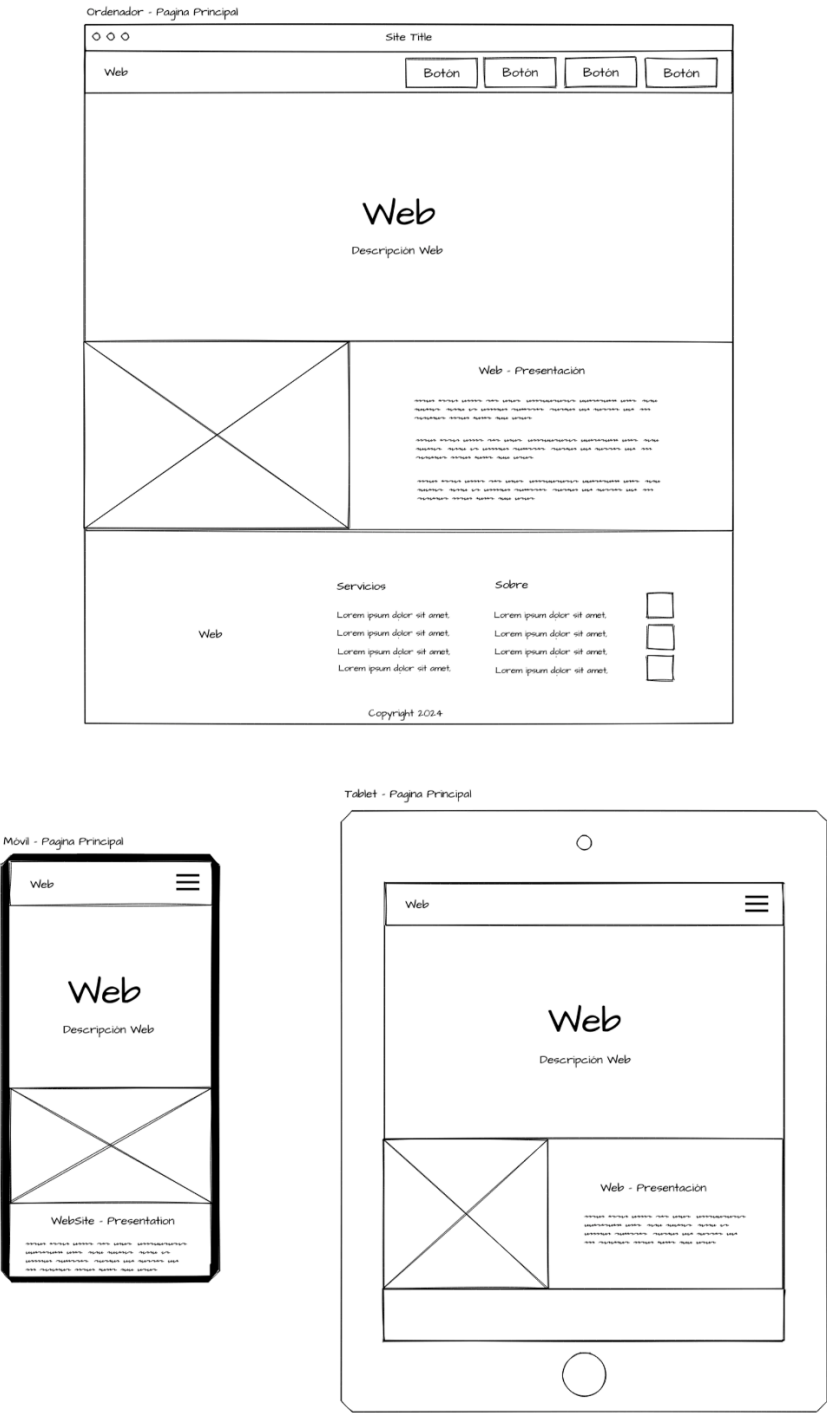
- Chat en Tiempo Real:

Comunicación: Pruebas de envío y recepción de mensajes entre clientes y empleados en tiempo real.

# 10. Manual de Estilos

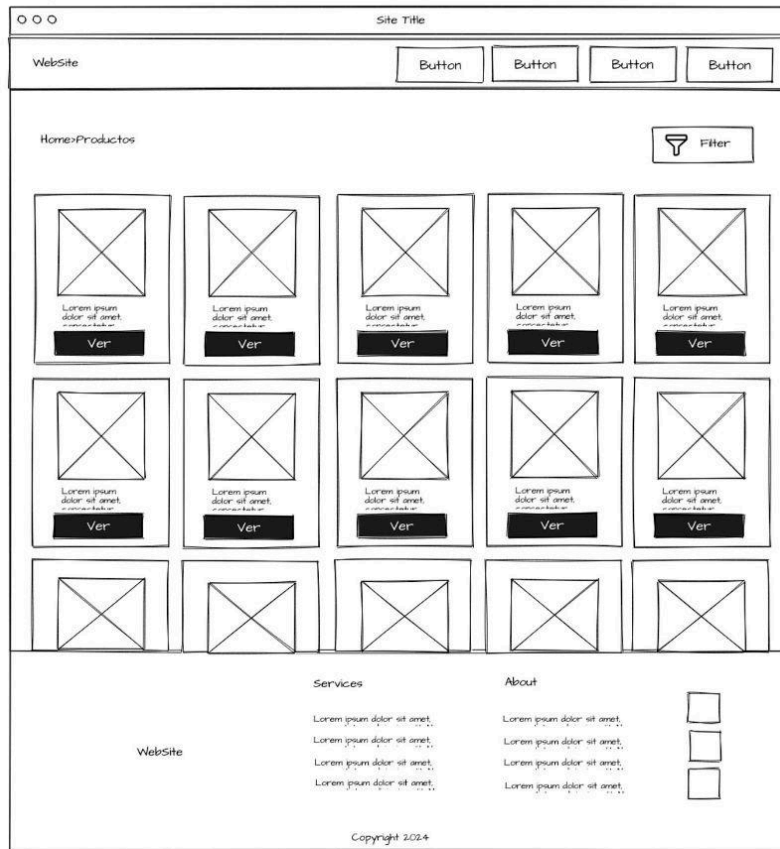
## 10.1 Sketches

### Página Principal

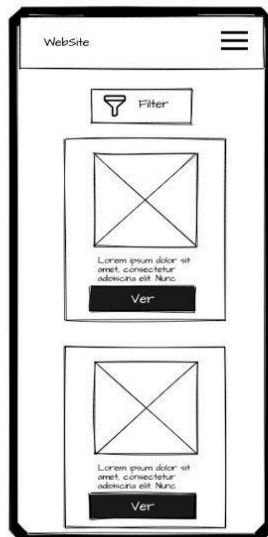


# Página Productos

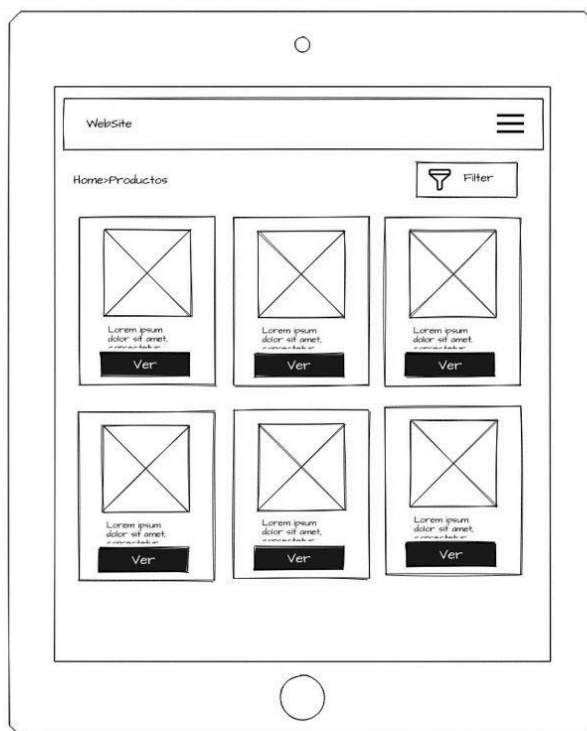
Desktop - Products



Mobile - Products

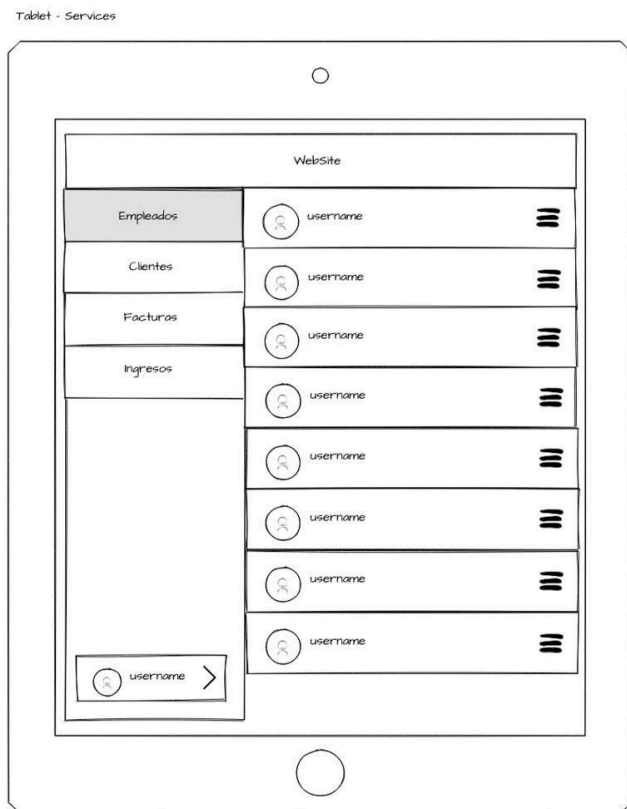
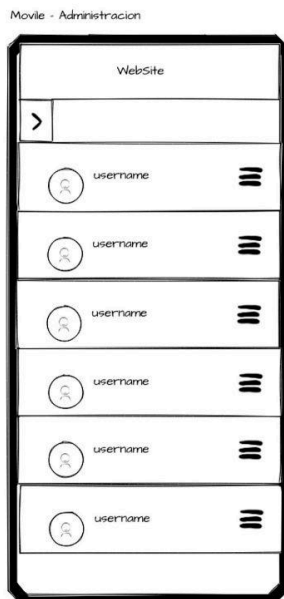
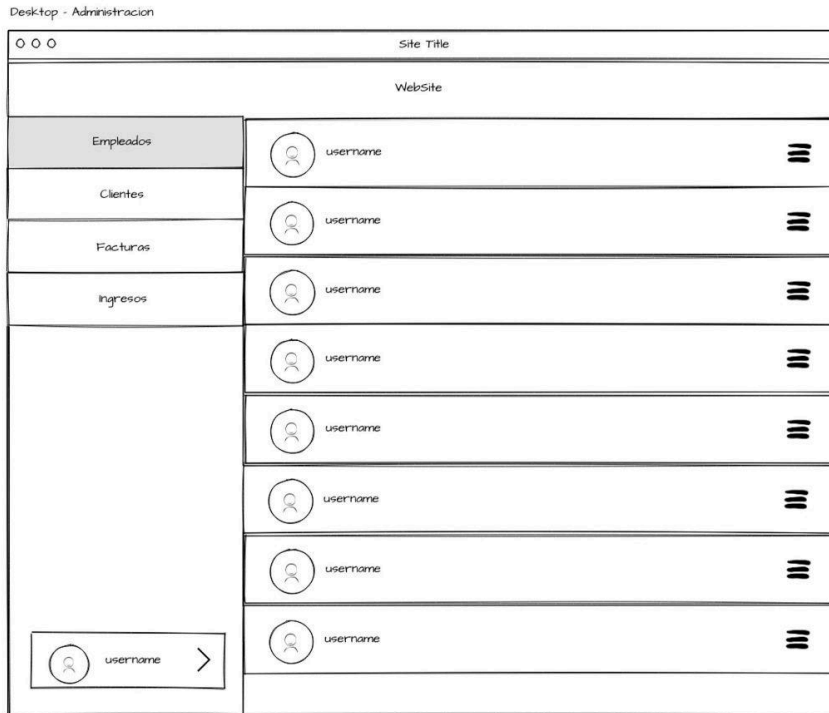


Tablet - Products



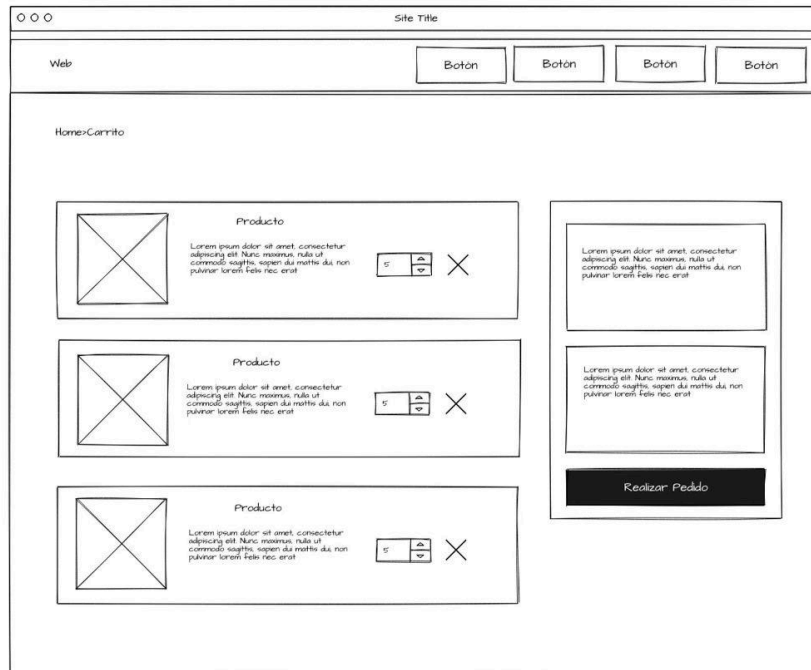


## Página Administracion

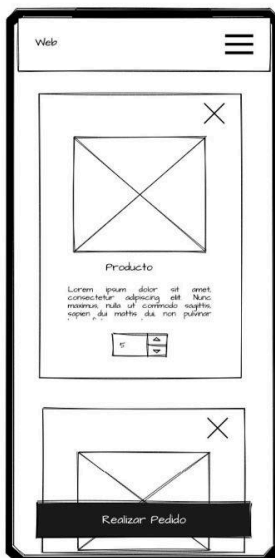


# Página Carrito

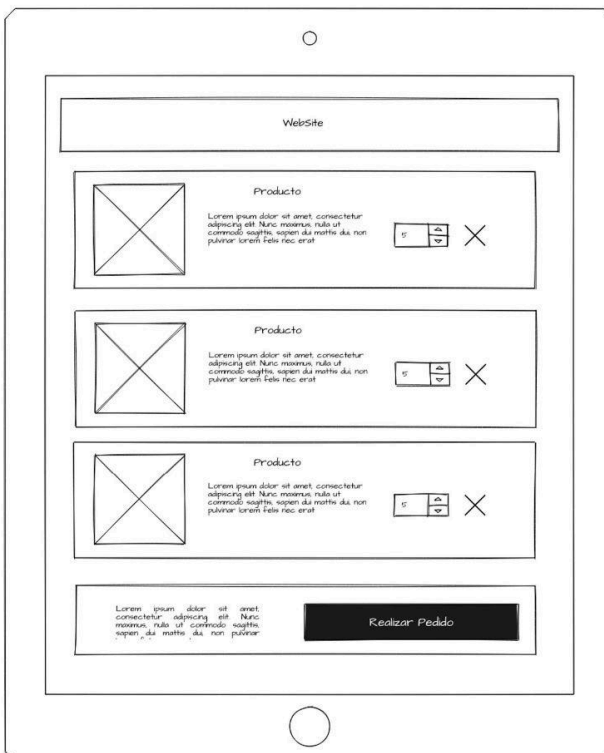
Ordenador - Carrito



Mobile - Administracion



Tablet - Services



## **10.2 Criterios de accesibilidad**

**Contraste de Color:** Se utilizan combinaciones de color que cumplen con los estándares de contraste para facilitar la lectura de texto y la percepción visual de elementos importantes.

**Texto Alternativo en Imágenes:** Todas las imágenes relevantes incluyen atributos alt descriptivos, facilitando su comprensión por parte de usuarios que utilizan tecnologías de asistencia como lectores de pantalla.

**Navegación por Teclado:** Todas las funcionalidades principales son accesibles y utilizables a través del teclado, asegurando que los usuarios puedan navegar sin depender exclusivamente del ratón.

**Etiquetas Semánticas:** Se utilizan etiquetas HTML semánticas adecuadamente (<nav>, <header>, <footer>, etc.) para mejorar la navegación y comprensión estructural del contenido por parte de los usuarios y tecnologías asistivas.

**Compatibilidad con Tecnologías de Asistencia:** La web ha sido probada con lectores de pantalla y otras tecnologías de asistencia para asegurar su compatibilidad y facilidad de uso para usuarios con discapacidades visuales o motoras.

## **10.3 Criterios de usabilidad**

**Facilidad de Navegación:** La estructura de navegación es clara y consistente en todas las páginas, facilitando a los usuarios encontrar rápidamente la información y funcionalidades que necesitan.

**Claridad y Concisión:** El contenido se presenta de manera clara y concisa, utilizando un lenguaje sencillo y directo para asegurar la comprensión rápida y precisa de los productos y servicios fotovoltaicos.

**Feedback Visual y Auditivo:** Se proporciona feedback visual y auditivo claro cuando los usuarios interactúan con la web, como confirmaciones de acciones, mensajes de error y estado de las operaciones.

**Eficiencia en el Flujo de Tareas:** Los procesos de compra, solicitud de servicios y gestión de cuentas están diseñados para ser intuitivos y eficientes, minimizando la cantidad de pasos necesarios y haciendo uso de formularios prellenados y opciones por defecto cuando sea posible.

**Diseño Responsivo:** La web es completamente responsiva, adaptándose de manera óptima a diferentes dispositivos y tamaños de pantalla, asegurando una experiencia consistente desde móviles hasta desktops.

**Pruebas de Usuario:** Se han realizado pruebas de usuario para evaluar la experiencia y recopilar retroalimentación sobre la navegación, claridad del contenido y facilidad de uso general de la web.

**Acceso a la Información:** Se facilita el acceso rápido a información importante como políticas de privacidad, términos y condiciones, y contacto, garantizando transparencia y confianza para los usuarios.

## 10.4 Tipografía

- **Roboto:**

**Descripción:** Roboto es una familia de fuentes sans-serif desarrollada por Google. Es ampliamente reconocida por su legibilidad y versatilidad, lo que la hace adecuada para interfaces web modernas.

**Uso:** Roboto ha sido seleccionada para proporcionar un aspecto limpio y profesional a los textos de la página, asegurando una buena legibilidad en diferentes tamaños y dispositivos.

- **system-ui:**

**Descripción:** system-ui es una fuente genérica del sistema, que toma la tipografía nativa del sistema operativo y del navegador. Es adaptable y se adapta a las preferencias del usuario y a las configuraciones del sistema.

**Uso:** system-ui se ha implementado para garantizar una experiencia coherente y accesible para los usuarios, aprovechando las ventajas de la tipografía nativa del entorno de visualización.

## 10.5 Mapa de colores

Negro (Texto Principal, elementos contraste)



RGB: `rgb(0,0,0)`

Hexadecimal: `#000000`

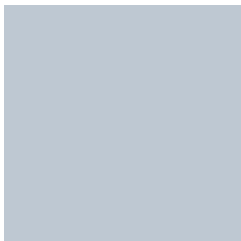
Azul (Menús y botones)



RGB: `rgb(31,90,128)`

Hexadecimal: `#1f5a80`

Azul Claro (Fondo de algunas secciones)



RGB: `rgb(193,202,214)`

Hexadecimal: `#c1cad6`

Blanco (Fondo de página y espacios en blanco)

RGB: `rgb(255,255,255)`

Hexadecimal: `#ffffff`

## **10.6 Dispositivos/vistas para las que se ha diseñado el proyecto**

El proyecto ha sido diseñado para ser accesible y funcional en múltiples dispositivos y vistas, incluyendo:

Escritorio: Diseño optimizado para pantallas de escritorio estándar, asegurando una experiencia visual completa y funcional.

Tablet: Adaptación para tabletas, garantizando que la interfaz sea intuitiva y utilizable en pantallas más pequeñas y táctiles.

Móvil: Enfoque especial en la experiencia móvil, con diseño responsivo para smartphones, asegurando que la navegación y la interacción sean fluidas y efectivas.

### **Tecnología Utilizada**

El desarrollo de la web se ha realizado utilizando Tailwind CSS, una herramienta de framework CSS utility-first. Tailwind CSS facilita la construcción y personalización rápida de interfaces de usuario mediante clases predefinidas que cubren desde el diseño hasta los estilos responsivos. Esta elección permite un desarrollo eficiente y mantenible, optimizando la gestión de estilos y la adaptabilidad a diferentes dispositivos sin necesidad de escribir CSS personalizado extensivo.

# 11 Software Utilizado

## Frontend:

- React: Framework utilizado para la creación de interfaces de usuario dinámicas y eficientes.
- Tailwind CSS: Framework CSS utility-first para estilos rápidos y mantenibles.
- Vite: Herramienta de construcción y desarrollo rápido para aplicaciones web modernas en JavaScript.
- Axios: Biblioteca para realizar peticiones HTTP desde el frontend a la API de backend.
- JWT-decode: Utilizado para decodificar tokens JWT en el frontend.
- Tremor: Para gráficas y componentes de react
- react-icons: Para los iconos utilizados

## Backend:

- Laravel: Framework PHP para el desarrollo de la API backend, proporcionando estructura y funcionalidad.
- PHP (versión 8.2): Lenguaje de programación utilizado junto con Laravel.
- Composer: Gestor de dependencias para PHP, utilizado para la gestión de paquetes y librerías.
- FPDF: Biblioteca PHP para la generación de archivos PDF, utilizada para las facturas automáticas.
- PHPMailer: Utilizado para el envío de correos electrónicos desde el backend.
- XAMPP: Paquete de software que incluye Apache, MySQL, PHP y Perl para entornos de desarrollo local.
- phpMyAdmin: Herramienta de administración de MySQL a través de la web, utilizada para gestionar la base de datos.
- Postman: Herramienta utilizada para probar y documentar APIs.

## Base de Datos:

- MySQL: Sistema de gestión de bases de datos relacional utilizado para almacenar datos del sitio web.
- Firebase: Utilizado para el sistema de chat en tiempo real, complementando la funcionalidad de MySQL.

#### Otras Herramientas:

- Autenticación Google: Para que los usuarios se registren en la web con google.
- Git/Git Bash (Windows): Sistema de control de versiones y terminal utilizado para el seguimiento de cambios en el código fuente.
- Node.js: Plataforma necesaria para ejecutar React y Tailwind a través de npm.
- npm: Gestor de paquetes utilizado para la instalación de dependencias en Node.js.
- Visual Paradigm: Herramienta de modelado utilizada para los diagramas de casos de uso y clases.

#### Herramientas de Comunicación y Colaboración:

- Hostinger: Utilizado para el alojamiento web, proporcionando servicios de hosting para la aplicación.

## 12 Mejoras posibles y aportaciones

#### Funcionalidad:

- Optimización del Proceso de Compra:

Implementar un proceso de compra más fluido y intuitivo, reduciendo el número de pasos y mejorando la experiencia del usuario desde la selección de productos hasta la finalización del pedido.

- Mejora en la Gestión de Servicios:

Ampliar las opciones de solicitud de servicios fotovoltaicos, ofreciendo configuraciones personalizadas y opciones adicionales durante el proceso de solicitud.

- Integración de Recomendaciones de Productos:

Implementar un sistema de recomendación de productos basado en el historial de compras y preferencias del usuario para mejorar la personalización y promover ventas cruzadas.



- Mejoras en el Panel de Control de Empleados:

Facilitar herramientas avanzadas de gestión de clientes y productos para empleados, incluyendo filtros avanzados, exportación de datos y estadísticas detalladas.

- Implementación de Funcionalidades Adicionales de Chat:

Ampliar las capacidades del chat en tiempo real para incluir soporte multimedia, transferencias de archivos y respuestas automatizadas para mejorar la atención al cliente.

### **Organización de Código:**

- Refactorización y Mejora de la Arquitectura de Componentes:

Revisar y refactorizar componentes de React para mejorar la legibilidad, mantenibilidad y rendimiento del código, utilizando patrones como contenedores y presentadores.

- Implementación de Pruebas Automatizadas:

Ampliar la cobertura de pruebas unitarias y de integración en el frontend y backend utilizando herramientas como Jest, React Testing Library y PHPUnit para garantizar la estabilidad del código.

- Optimización del Código Backend:

Mejorar la eficiencia y escalabilidad del código backend en Laravel mediante la optimización de consultas SQL, el uso de caché y la implementación de buenas prácticas de desarrollo.

- Documentación Detallada del Código y API:

Documentar exhaustivamente el código, incluyendo comentarios claros en el código fuente y una documentación completa de la API para facilitar su mantenimiento y la integración con otros sistemas.

## **Organización de la Información:**

- Mejora en la Organización de la Base de Datos:

Revisar y optimizar la estructura de la base de datos MySQL para mejorar la eficiencia en la recuperación de datos y garantizar la integridad de la información almacenada.

- Implementación de Sistemas de Análisis y Reportes:

Desarrollar sistemas de análisis de datos y generación de reportes para obtener insights sobre el comportamiento de los usuarios, ventas y rendimiento del sitio web.

- Personalización de Contenidos Dinámicos:

Utilizar técnicas de personalización de contenido basadas en datos del usuario para ofrecer experiencias más relevantes y personalizadas, aumentando la retención y fidelización de clientes.