

UNIVERSIDAD DON BOSCO
ESCUELA DE COMPUTACIÓN
DISEÑO Y PROGRAMACION DE SOFTWARE MULTIPLATAFORMA



FASE 1 DEL PROYECTO:

Tienda Online con Contaduría e Inventario Integrados

CATEDRÁTICA:

Karens Medrano

ESTUDIANTES:

Marlon David Martínez Montoya – MM210981

Karla Elizabeth Murillo Urrutia – MU192557

Gabriela Lourdes Rodríguez Parada – RP210191

Jessica Abigail Rodríguez Torres – RT200479

Roberto Alejandro Ventura Alvarenga – VA210756

Septiembre, 2024

Indice

<u>Introducción</u>	
<u>1. Perfil del proyecto</u>	
<u>1.1. Problema a Resolver</u>	
<u>1.2. Objetivo General</u>	
<u>1.3. Metas</u>	
<u>1.4. Resultados Esperados</u>	
<u>1.5. Metodología</u>	
<u>1.6. Área Geográfica y Beneficiarios del Proyecto</u>	
<u>1.7. Equipo Participante</u>	
<u>1.8. Posibles Fuentes de Financiamiento</u>	
<u>1.9. Factores Críticos de Éxito</u>	
<u>1.10. Duración</u>	
<u>1.11. Presupuesto</u>	
<u>2. Presentación del diseño</u>	
<u>2.1. Paleta de colores</u>	
<u>2.2. Iconografía</u>	
<u>2.3. Tipografía</u>	
<u>2.4. Diseños de pantalla UI/UX</u>	
<u>3. Lógica del negocio</u>	
<u>3.1. Servidores</u>	
<u>3.2. Lenguajes de programación y frameworks</u>	
<u>3.3. Bases de datos</u>	
<u>3.4. Arquitectura del software</u>	
<u>1. Tienda Online</u>	
<u>2. Facturación Electrónica</u>	
<u>3. Administración de Inventarios</u>	
<u>4. Contaduría</u>	
<u>3.5. Comunicación entre componentes</u>	
<u>3.5.1. Endpoints de autenticación</u>	
<u>3.5.2. Endpoints ventas</u>	
<u>3.5.3. Endpoints inventario</u>	
<u>3.5.4. Endpoints firma electrónica</u>	
<u>5. Sistema de Facturación con Capacidad de Emitir DTE</u>	
<u>3.6. Seguridad</u>	
<u>3.7. Gestión de sesiones y usuarios</u>	
<u>4. Herramientas de desarrollo</u>	
<u>5. Presupuesto</u>	
<u>6. Cronograma</u>	
<u>7. Bibliografía</u>	

Introducción

En el entorno competitivo actual, las pequeñas y medianas empresas (PYMEs) enfrentan desafíos significativos en la gestión eficaz de sus operaciones, desde la venta y procesamiento de pagos hasta el manejo de inventarios y cumplimiento de regulaciones fiscales. Estos desafíos pueden llevar a ineficiencias operativas, errores humanos y problemas legales si no se cuenta con un sistema adecuado y eficiente.

Para abordar estos problemas, nuestro proyecto se propone desarrollar una plataforma integral de tienda online que no solo facilite la venta de productos, sino que también integre herramientas avanzadas para el procesamiento de pagos, la facturación electrónica, y la gestión de inventarios y contaduría. Este sistema está diseñado para mejorar significativamente la eficiencia administrativa y operativa de las empresas, permitiendo una gestión más fluida y conforme a las normativas fiscales vigentes.

El desarrollo de esta plataforma se llevará a cabo utilizando la metodología ágil Scrum, lo que permitirá un enfoque flexible y colaborativo, asegurando que el sistema evolucione según las necesidades emergentes y los requisitos del usuario. La solución se implementará en dos versiones: una aplicación web y una aplicación móvil, ambas con interfaces intuitivas y fáciles de usar, asegurando una experiencia de usuario óptima.

Nuestro objetivo es proporcionar a las empresas una herramienta poderosa que no solo simplifique la gestión de ventas e inventarios, sino que también asegure el cumplimiento con las normativas fiscales a través de un sistema de facturación electrónica integrado. A través de esta plataforma, buscamos empoderar a las empresas para que puedan operar de manera más eficiente y adaptarse a los requisitos legales sin dificultades.

1. Perfil del proyecto

1.1. Problema a Resolver

Muchas pequeñas y medianas empresas enfrentan dificultades en la gestión de ventas, procesamiento de pagos y manejo de inventarios, además de cumplir con las regulaciones fiscales que requieren facturación electrónica. La falta de un sistema adecuado puede generar ineficiencias, errores humanos y complicaciones legales.

1.2. Objetivo General

El objetivo principal del proyecto es desarrollar una plataforma de tienda online que facilite la venta de productos, gestione el procesamiento de pagos y la facturación electrónica, y ofrezca herramientas de contaduría e inventario para mejorar la eficiencia administrativa y operativa de las empresas. Al final del proyecto, se espera que los usuarios puedan gestionar sus ventas y operaciones internas de manera más efectiva y cumpliendo con las normativas fiscales vigentes.

1.3. Metas

Para desarrollar un sistema con este nivel de complejidad, es necesario que al inicio del proyecto se establezcan cierto tipo de metas a tomar en cuenta, con el objetivo de crear el sistema de manera eficiente, para este caso las metas propuestas son:

- **Implementación y funcionalidad de venta y pago:** para la aplicación móvil esta funcionalidad será de mucha importancia, debido a que se tratará de implementar de una manera específica para que los usuarios a través de la aplicación puedan ver ciertos productos y puedan realizar la compra de manera funcional.
- **Desarrollo de facturación electrónica:** esta funcionalidad es de mucha relevancia para el sistema, ya que estas facturas serán procesadas por el Ministerio de Hacienda siempre y cuando la empresa que adquiera el sistema posea las credenciales correspondientes para dicho proceso.

También es importante aclarar que esta funcionalidad, solo será implementada para el sistema al nivel de escritorio, es decir que desde la aplicación móvil no se podrán generar facturas electrónicas.

- **Optimizar la experiencia del usuario y usabilidad:** este aspecto se tiene que tomar en cuenta en todo el tiempo en el que se desarrollara la aplicación móvil y el sistema, ya que se tratara la manera de que sea un sistema completo y sencillo de utilizar.

1.4. Resultados Esperados

Como todo sistema, al momento de ser completado se espera que cumpla con ciertas características con el objetivo de tener resultados positivos para la empresa y para los usuarios que lo utilizan, ciertos resultados para este proyecto son:

- **Efectividad en la gestión del inventario:** creado el sistema se espera que los usuarios capacitados para el uso del sistema tengan la facilidad de poder realizar ciertas acciones con el inventario de los productos, acciones como ver los productos, agregar producto, eliminar productos y modificar los productos, todas estas acciones de manera fácil y funcional.
- **Facturación electrónica:** para el sistema a nivel de escritorio se espera que todos los documentos enviados al Ministerio de Hacienda cumplan con las normativas fiscales (siempre y cuando la empresa que posea este sistema de facturación tenga credenciales correspondientes con el Ministerio de Hacienda).

1.5. Metodología

Se utilizará la metodología ágil **Scrum**, la cual fomenta el trabajo en equipo y la mejora continua. A través de sprints, se verificará y entregará el desarrollo del proyecto en pequeñas partes hasta su conclusión exitosa.

1.6. Área Geográfica y Beneficiarios del Proyecto

No hay una limitación al área geográfica, ya que el proyecto se va a desarrollar o se puede adaptar a cualquier tipo de empresa según los requerimientos de cada una.

Los beneficiarios son todas aquellas micro, medianas y grandes empresas con las que tienen registro ante el ministerio de hacienda y/o un área contable, en la que presentan registros de los aspectos relacionados con dicha área.

1.7. Equipo Participante

- **Marlon David Martínez Montoya** – MM210981
- **Karla Elizabeth Murillo Urrutia** – MU192557
- **Gabriela Lourdes Rodríguez Parada** – RP210191
- **Jessica Abigail Rodríguez Torres** – RT200479
- **Roberto Alejandro Ventura Alvarenga** – VA210756

1.8. Posibles Fuentes de Financiamiento

- **Fondos propios:** Utilizar ahorros personales o capital propio de cada uno de los participantes del equipo para financiar el desarrollo del proyecto. Una vez completado el producto final, se podrá comercializar y vender a empresas interesadas, generando ingresos que compensen la inversión inicial.
- **Alianzas estratégicas:** Establecer acuerdos con otras empresas, inversionistas o entidades que puedan estar interesadas en el desarrollo del proyecto y estén dispuestas a financiarlo.

1.9. Factores Críticos de Éxito

Para poder garantizar que el proyecto se complete con éxito y cumpla con los objetivos establecidos se identifican ciertos factores fundamentales para esto:

- **Análisis de requerimientos:** Realizar un análisis detallado de las necesidades y expectativas de posibles usuarios finales y empresas para asegurar que el sistema cumpla con los requerimientos necesarios.
- **Diseño centrado en el usuario:** Crear una interfaz de usuario que sea intuitiva y amigable a la hora de usarla y que facilite la interacción y mejore la experiencia del usuario.
- **Integración eficiente del sistema:** Asegurar la integración correcta de la tienda online con el sistema de contaduría e inventario, así como la plataforma de pago y facturación electrónica.
- **Cumplimiento normativo:** Garantizar que el sistema de facturación electrónica cumpla con las regulaciones que establece la Ley de Comercio Electrónico establecido por la Asamblea Legislativa.
- **Seguridad y privacidad:** Implementar medidas de seguridad para proteger los datos sensibles de los usuarios y las transacciones financieras.
- **Pruebas y validación:** Realizar pruebas constantes del sistema en todas sus fases de desarrollo para poder identificar y corregir errores para así asegurar un funcionamiento óptimo del sistema.
- **Eficiencia operativa:** Optimizar los recursos disponibles tanto humanos como tecnológicos para maximizar la eficiencia durante el desarrollo del proyecto.

1.10. Duración

El proyecto está planificado para desarrollarse a lo largo de 10 semanas, las cuales estarán dentro del ciclo universitario. En la figura 1 se presenta el cronograma detallado de las actividades.

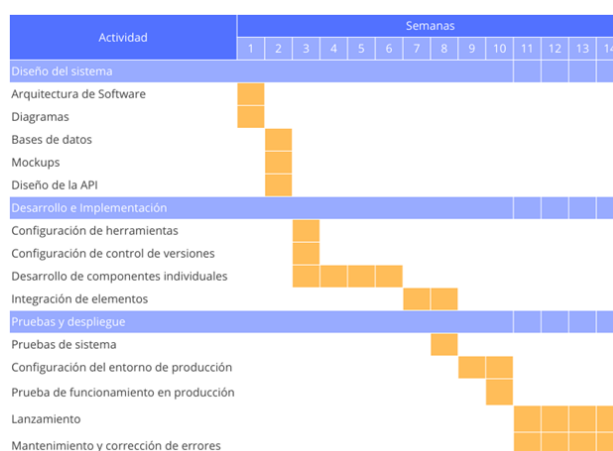


Figura 1. Cronograma

1.11. Presupuesto

En primer lugar, se debe tomar en cuenta el recurso humano: se contratarán 5 trabajadores durante las 10 semanas del proyecto, abarcando el pago de 7 quincenas. Serán 5 ingenieros en Ciencias de la Computación. A cada trabajador se le pagará por quincena \$500. Abarcando las 7 quincenas, \$3500 por trabajador.

Por otro lado, se necesita infraestructura en la nube para la interconexión de la API con la web y la base de datos: máquinas virtuales para el alojamiento de la API, servicios de contenedores, almacenamiento, alojamiento de la base de datos y servicio de hosting.

Los costos de la nube se analizarán para una cantidad de 50 usuarios diarios de la plataforma en un periodo de 1 mes a partir del lanzamiento.

Recurso	Costo por unidad	Cantidad	Unidad	Total
Desarrolladores FullStack	\$ 2.500,00	5	trabajadores	\$ 12.500,00
Máquina Virtual en la nube		30	dias	\$ 69,12
Almacenamiento 1TB en la nube		30	dias	\$ 20,80
Base de datos en la nube		30	dias	\$ 363,14
Kubernetes		30	dias	\$ 84,24
Total				\$ 13.037,30

Figura 2. Presupuesto

2. Presentación del diseño

2.1. Paleta de colores

Se utilizará una paleta de colores con énfasis en tonos azulados, esto con el objetivo de transmitir una marca con innovación tecnológica que al mismo tiempo mantiene una cercanía con el cliente, transmitiéndole confianza y seguridad.

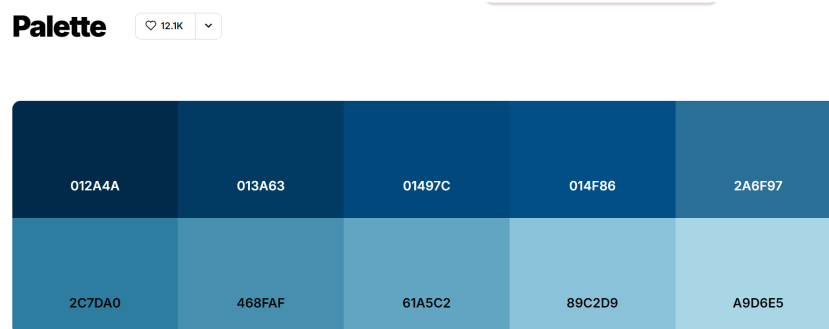


Figura 3. Paleta de colores

2.2. Iconografía

Para el presente proyecto, se utilizará la librería de iconos de React Native Vector Icons por su fácil implementación sobre React Native y su diseño fresco y amigable.

React Native Vector Icons es una librería utilizada en el desarrollo de aplicaciones móviles con React Native que proporciona un conjunto de íconos personalizables. La librería incluye íconos populares de fuentes como FontAwesome, Material Icons, Ionicons, y muchas otras, permitiendo agregar íconos escalables y personalizables fácilmente en aplicaciones. La fuente de la iconografía es FontAwesome 5.

En la figura 4 se muestran algunos ejemplos de la Iconografía.



Figura 4. Iconografía

2.3. Tipografía

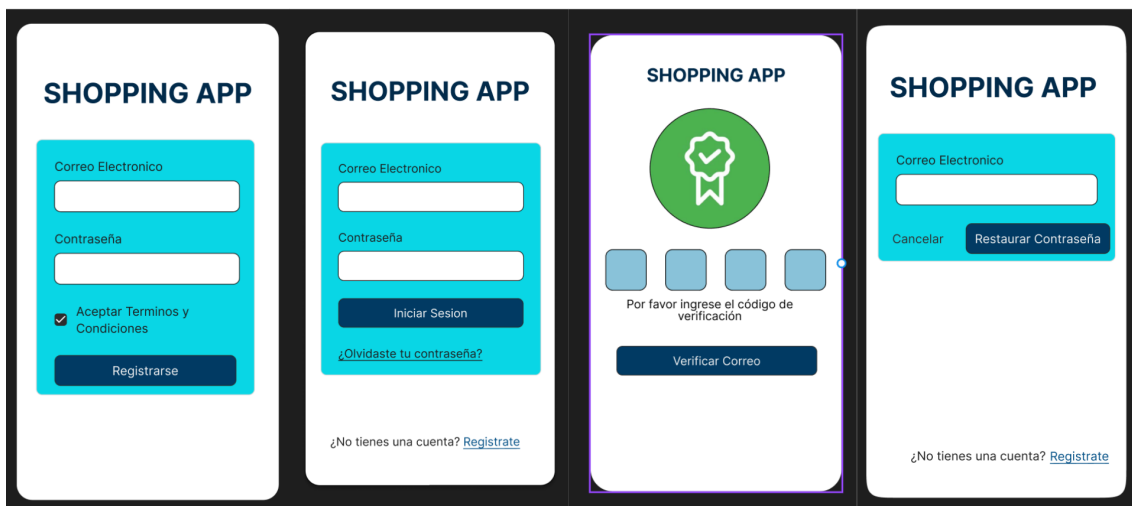
Se utilizarán las tipografías Roboto y Quicksand para toda la interfaz del software, sus diseños redondeados brindarán un carácter de cercanía de la página hacia los usuarios.

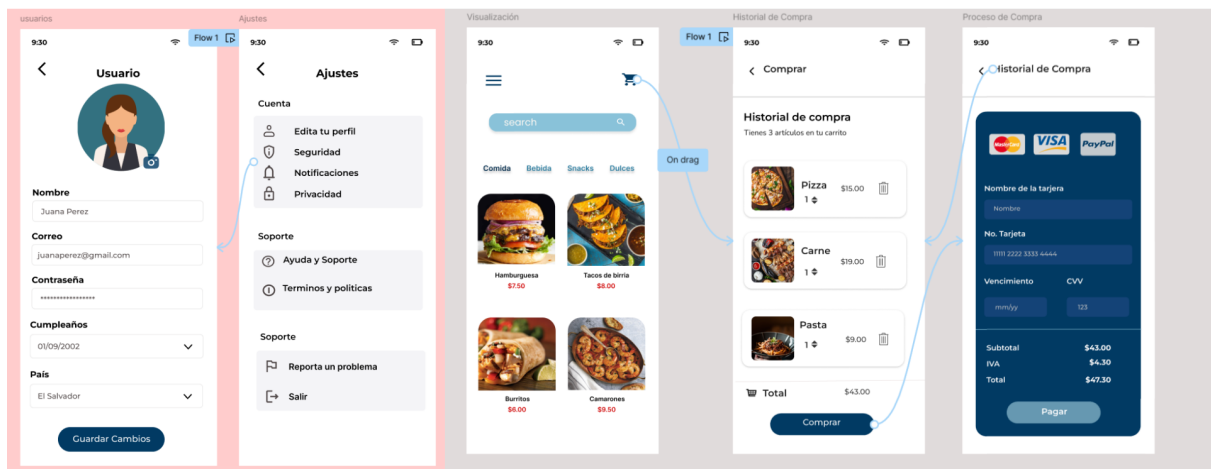
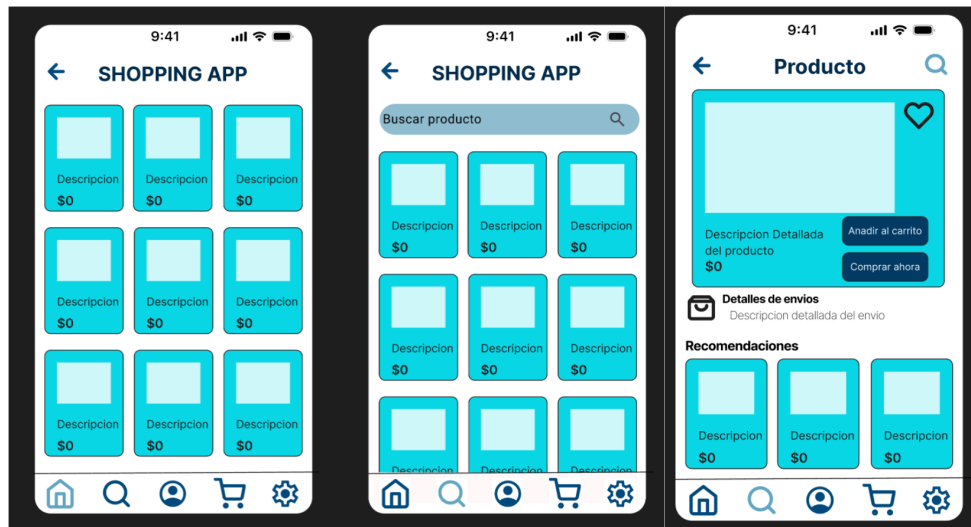
En la siguiente tabla se amplía la información del uso de estas tipografías dentro del sistema.

	Fuente	Tipo	Tamaño
Encabezados	Roboto	Regular 400	32px
Textos principales	Roboto	Regular 400	20px
Textos Secundarios	Roboto	Regular 400	16px
Botones	Quicksand	Bold 700	20px

2.4. Diseños de pantalla UI/UX

Presentamos las siguientes pantallas como prototipos del proyecto.





3. Lógica del negocio

3.1. Servidores

Inicialmente, el proyecto se desarrollará en un servidor local utilizando XAMPP (Apache), que incluirá el servicio de base de datos MySQL. Posteriormente, el proyecto será migrado a un entorno Docker, donde se implementarán los microservicios en contenedores, junto con el API Gateway Kong para la gestión y enrutamiento de las API.

3.2. Lenguajes de programación y frameworks

El proyecto se divide en dos sistemas:

- Para la aplicación web, se trabajará con React JS en el frontend.
- Para la aplicación móvil, se utilizará React Native en el frontend.

En cuanto al backend y la programación de los microservicios, se empleará el framework Laravel, que funciona con PHP para las API REST y Kong se utilizará como API Gateway.

3.3. Bases de datos

El proyecto utilizará dos bases de datos principales en MySQL para almacenar y consultar la información proveniente de las APIs de los microservicios. La primera base de datos estará dedicada a la lógica de autenticación, la segunda manejará la lógica de contabilidad, ventas e inventario.

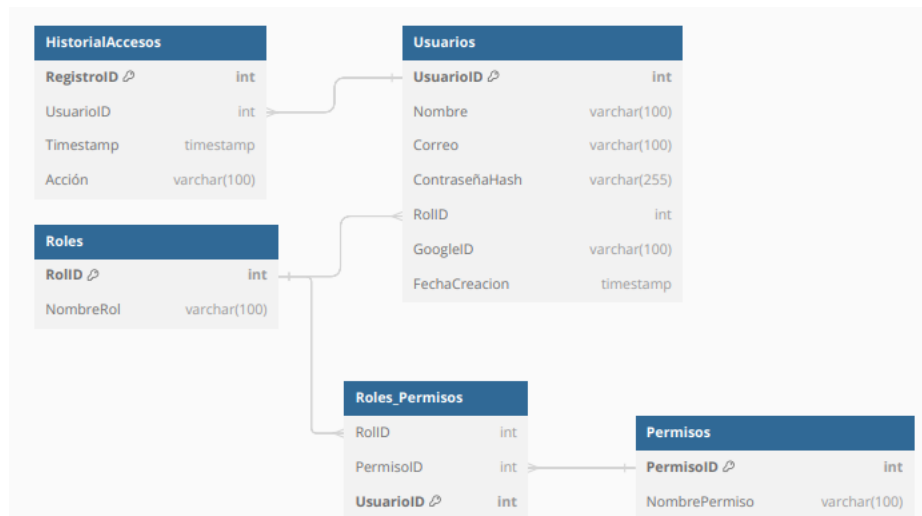


Figura 5. Base de datos de autenticación

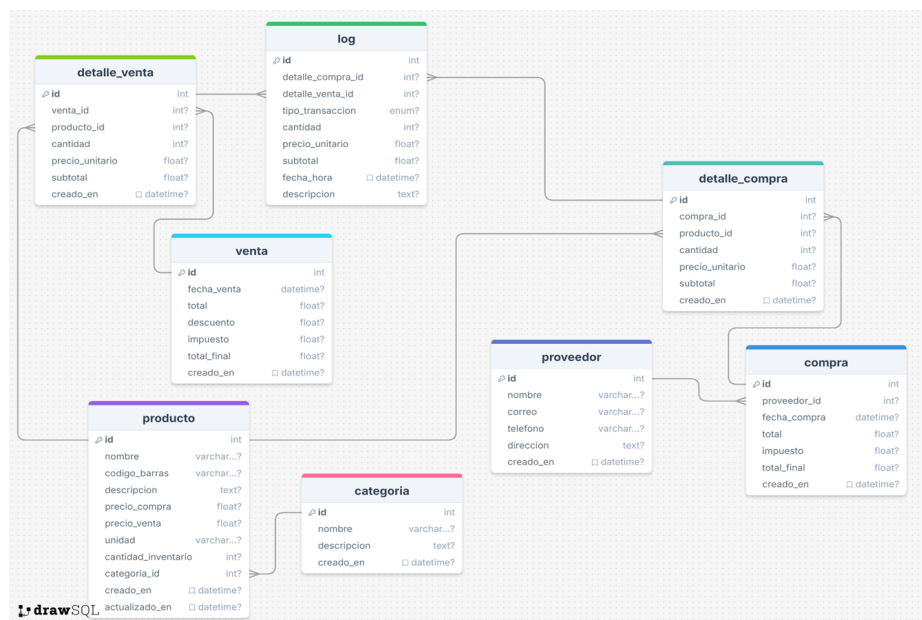


Figura 6. Base de datos de ventas, compras e inventariado

3.4. Arquitectura del software

Los módulos principales del proyecto son los siguientes:

1. Tienda Online

- **Gestión de Productos:** Registro y actualización de productos, incluyendo detalles como nombre, descripción, precio, imágenes y cantidad en stock.

- **Carrito de Compras:** Funcionalidad para que los clientes agreguen productos a su carrito, actualicen cantidades y procedan al pago.
- **Procesamiento de Pagos:** Integración con pasarelas de pago para procesar pagos en línea de forma segura.

2. Facturación Electrónica

- **Generación de Facturas:** Creación automática de facturas electrónicas cumpliendo con las regulaciones fiscales.
- **Envío de Facturas:** Envío de facturas electrónicas a los clientes por correo electrónico.

3. Administración de Inventarios

- **Control de Stock:** Actualización automática del inventario con cada venta realizada.
- **Notificaciones de Stock Bajo:** Alertas para reabastecer productos cuando el stock es bajo.

4. Contaduría

- **Registro de Transacciones:** Registro automático de todas las transacciones de ventas y pagos en el sistema contable.
- **Reportes Financieros:** Generación de reportes financieros.

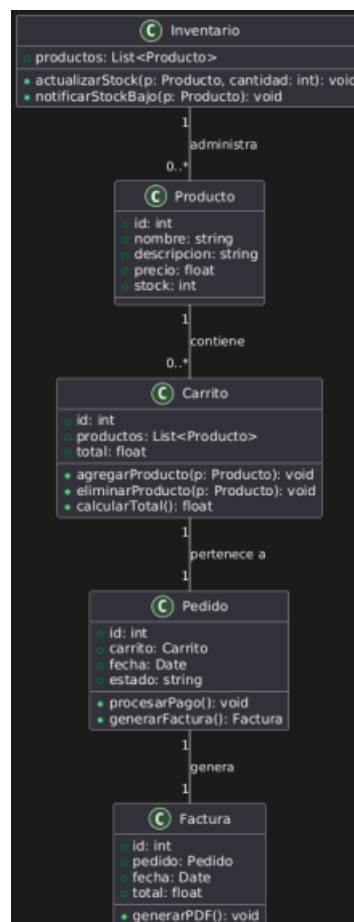


Figura 7. Diagrama de clases

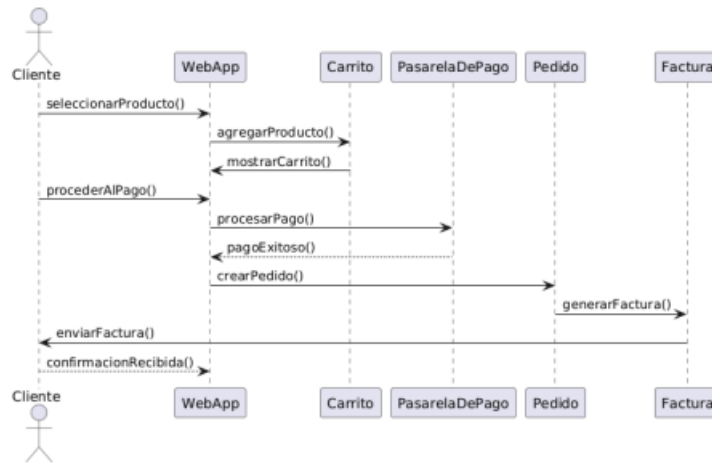


Figura 8. Diagrama de secuencia

3.5. Comunicación entre componentes

3.5.1. Endpoints de autenticación

- **Inicio de Sesión**
 - **POST /api/auth/login**: Iniciar sesión de un usuario en el sistema.
 - **POST /api/auth/google-login**: Iniciar sesión de un usuario utilizando Google OAuth.
- **Restablecimiento de Contraseña**
 - **POST /api/auth/reset-password**: Solicitar el restablecimiento de la contraseña.
 - **PUT /api/auth/change-password**: Cambiar la contraseña del usuario.
- **Cierre de Sesión**
 - **POST /api/auth/logout**: Cerrar la sesión del usuario.
- **Gestión de Roles y Permisos**
 - **POST /api/roles**: Crear un nuevo rol.
 - **PUT /api/roles/{rolID}**: Modificar un rol existente.
 - **DELETE /api/roles/{rolID}**: Eliminar un rol existente.
 - **GET /api/roles**: Obtener la lista de roles y sus permisos.
 - **GET /api/roles/{rolID}**: Obtener los detalles de un rol específico.
- **Gestión de Usuarios**
 - **POST /api/usuarios**: Crear un nuevo usuario.
 - **PUT /api/usuarios/{usuarioID}**: Modificar un usuario existente.
 - **DELETE /api/usuarios/{usuarioID}**: Eliminar un usuario existente.
 - **GET /api/usuarios**: Obtener la lista de usuarios.
 - **GET /api/usuarios/{usuarioID}**: Obtener los detalles de un usuario específico.
- **Seguridad y Auditoría**
 - **GET /api/auditoria/historial**: Obtener el historial de accesos.

3.5.2. Endpoints ventas

1. Ventas

- GET /api/ventas: Obtener todas las ventas.
- GET /api/ventas/{id}: Obtener una venta específica.
- GET /api/ventas/fecha/{fecha}: Obtener ventas por fecha específica.
- POST /api/ventas: Crear una nueva venta.
- PUT /api/ventas/{id}: Actualizar una venta.
- DELETE /api/ventas/{id}: Eliminar una venta.

2. Detalles de Venta

- GET /api/detalles_venta: Obtener todos los detalles de venta.
- GET /api/detalles_venta/{id}: Obtener un detalle de venta específico.
- GET /api/detalles_venta/venta/{venta_id}: Obtener detalles de venta por ID de venta.
- POST /api/detalles_venta: Crear un nuevo detalle de venta.
- PUT /api/detalles_venta/{id}: Actualizar un detalle de venta.
- DELETE /api/detalles_venta/{id}: Eliminar un detalle de venta.

3. Log

Endpoints Generales para la Tabla logs

- GET /api/logs: Obtener todos los registros de transacciones.
- GET /api/logs/{id}: Obtener un registro de transacción específico por su ID.
- GET /api/logs/tipo/{tipo_transaccion}: Obtener registros de transacciones por tipo (compra, venta, devolución).
- GET /api/logs/fecha/{fecha}: Obtener registros de transacciones por fecha específica.
- GET /api/logs/fecha/{fecha_inicio}/{fecha_fin}: Obtener registros de transacciones entre un rango de fechas.
- POST /api/logs: Crear un nuevo registro de transacción.
- PUT /api/logs/{id}: Actualizar un registro de transacción existente.
- DELETE /api/logs/{id}: Eliminar un registro de transacción específico por su ID.

3.5.3. Endpoints inventario

1. Categorías

- GET /api/categorias: Obtener todas las categorías.
- GET /api/categorias/{id}: Obtener una categoría específica.
- POST /api/categorias: Crear una nueva categoría.
- PUT /api/categorias/{id}: Actualizar una categoría.
- DELETE /api/categorias/{id}: Eliminar una categoría.

2. Productos

- GET /api/productos: Obtener todos los productos.

- GET /api/productos/{id}: Obtener un producto específico.
- GET /api/productos/categoria/{categoria_id}: Obtener productos por categoría.
- GET /api/productos/inventario/bajo: Obtener productos con inventario bajo (menor a un límite específico).
- POST /api/productos: Crear un nuevo producto.
- PUT /api/productos/{id}: Actualizar un producto.
- DELETE /api/productos/{id}: Eliminar un producto.

3. Log

Endpoints Generales para la Tabla logs

- GET /api/logs: Obtener todos los registros de transacciones.
- GET /api/logs/{id}: Obtener un registro de transacción específico por su ID.
- GET /api/logs/tipo/{tipo_transaccion}: Obtener registros de transacciones por tipo (compra, venta, devolución).
- GET /api/logs/fecha/{fecha}: Obtener registros de transacciones por fecha específica.
- GET /api/logs/fecha/{fecha_inicio}/{fecha_fin}: Obtener registros de transacciones entre un rango de fechas.
- POST /api/logs: Crear un nuevo registro de transacción.
- PUT /api/logs/{id}: Actualizar un registro de transacción existente.
- DELETE /api/logs/{id}: Eliminar un registro de transacción específico por su ID.

3.5.4. Endpoints firma electrónica

1. Cuenta de Contribuyente con el Ministerio de Hacienda

- **Endpoint:** POST /api/hacienda/contribuyente/cuenta
- **Descripción:** Crear y verificar una cuenta de contribuyente con acceso al ambiente de pruebas.

2. Certificado de Firma Electrónica para Pruebas

- **Endpoint:** GET /api/hacienda/certificado/firma
- **Descripción:** Obtener el certificado de firma electrónica para el ambiente de pruebas desde la plataforma de factura.gob.sv.

3. Solución Gratuita para Firmar Documentos Tributarios Electrónicos

- **Endpoint:** POST /api/hacienda/firmador
- **Descripción:** Configurar el servicio para firmar los Documentos Tributarios Electrónicos, ya sea en Windows o dentro de un Docker.

4. Credenciales para el Uso del Sistema de Transmisión de DTE

- **Endpoint:** POST /api/hacienda/credenciales
- **Descripción:** Autenticarse contra la API de Seguridad utilizando usuario y contraseña para firmar y transmitir un DTE.

5. Sistema de Facturación con Capacidad de Emitir DTE

- **Endpoint:** PUT /api/hacienda/sistema/facturacion

- **Descripción:** Ajustar el sistema de facturación interno para cumplir con los requisitos funcionales y técnicos para la generación y transmisión de DTE al Ministerio de Hacienda.

3.6. Seguridad

Para garantizar la seguridad en el proyecto se implementarán las siguientes medidas de seguridad:

1. Autenticación y Autorización

- **Autenticación Segura:** Se emplearán mecanismos de autenticación robustos como OAuth 2.0 y JWT para asegurar que solo los usuarios autenticados tengan acceso a las APIs del sistema.
- **Autorización:** Se establecerán roles y permisos para controlar el acceso a diferentes áreas del sistema, utilizando controles de acceso basados en roles (RBAC) y atributos (ABAC) para gestionar las autorizaciones.

2. Protección de Datos Sensibles

- **Cifrado de Contraseñas:** Las contraseñas se almacenarán utilizando algoritmos de hashing seguros como bcrypt o Argon2 para evitar la exposición de credenciales.

3. Seguridad en la API

- **Validación de Entradas:** Se implementarán técnicas de validación y saneamiento de entradas para prevenir ataques de inyección y otros ataques basados en entradas maliciosas.
- **Rate Limiting:** Se establecerán límites en la cantidad de solicitudes permitidas para cada usuario, protegiendo el sistema contra ataques de denegación de servicio (DDoS).
- **Autenticación de API:** Se utilizarán tokens de acceso para asegurar que las APIs solo sean utilizadas por clientes autorizados.

4. Protección Contra Amenazas Comunes

- **Cross-Site Scripting (XSS):** Se implementarán medidas para prevenir XSS, incluyendo la codificación adecuada de datos de entrada y la validación de datos.
- **Cross-Site Request Forgery (CSRF):** Se utilizarán tokens CSRF para proteger contra ataques de falsificación de solicitudes.
- **Inyección SQL:** Se emplearán consultas parametrizadas y ORM (Object-Relational Mapping) para prevenir inyecciones SQL.

5. Monitoreo y Registro

- **Monitoreo de Seguridad:** Se implementarán herramientas de monitoreo para detectar y alertar sobre actividades sospechosas o anomalías en tiempo real.
- **Registro de Eventos:** Se mantendrán registros detallados de eventos de seguridad y accesos para facilitar auditorías y el análisis de incidentes.

6. Seguridad en el Entorno de Despliegue

- **Configuración Segura:** Los servidores y servicios en la nube estarán configurados siguiendo las mejores prácticas de seguridad y evitando configuraciones predeterminadas inseguras.
- **Actualizaciones y Parcheo:** Se mantendrán todos los componentes del sistema actualizados con los últimos parches de seguridad para proteger contra vulnerabilidades conocidas.

3.7. Gestión de sesiones y usuarios

1. Administrador del sistema

Descripción: Encargado de la configuración y mantenimiento del software.

Objetivos: Configurar el software, gestionar usuarios.

Tareas:

- Podrá agregar, eliminar y modificar a todos los usuarios del sistema.
- Tendrá acceso al historial completo de modificaciones.
- Será capaz de iniciar un proceso de restablecimiento de contraseña para un usuario.
- Podrá agregar, eliminar y modificar todos los productos del inventario.
- Podrá configurar los detalles de la facturación electrónica.
- Podrá editar, procesar y cancelar compras.

Restricciones:

- El administrador no podrá acceder a datos sensibles de los usuarios.
- El administrador a pesar de que podrá reestablecer contraseñas no podrá ver las contraseñas anteriores.
- El administrador solo podrá autenticarse con una autenticación multifactor.
- El administrador no podrá editar las facturas electrónicas enviadas a la API.
- El administrador no podrá editar o borrar historiales y logs.

2. Cliente

Descripción: Usuario que realiza compras en el ecommerce y gestiona su propia información.

Objetivos:

- Realizar compras de productos.
- Administrar su cuenta y pedidos.

Tareas:

- Podrá buscar y visualizar productos en el ecommerce.
- Podrá agregar productos al carrito de compras y realizar pedidos.
- Podrá consultar el estado de sus pedidos y historial de compras.
- Podrá modificar su información personal y dirección de envío.
- Podrá realizar devoluciones y solicitar reembolsos.
- Podrá gestionar sus preferencias de notificaciones y suscripciones.

Restricciones:

- El cliente no podrá acceder a datos de otros usuarios o productos que no estén disponibles para la compra.
- El cliente no podrá modificar el inventario de productos.
- El cliente no podrá cancelar pedidos después de haber sido procesados.
- El cliente no podrá acceder a funcionalidades administrativas o de configuración del sistema.

3. Trabajador (No Administrador)

Descripción: Usuario que realiza tareas operativas y de soporte dentro del ecommerce, pero sin acceso a funciones de administración.

Objetivos:

- Gestionar operaciones diarias y atender a clientes.
- Apoyar en la administración de pedidos y productos.

Tareas:

- Podrá consultar y actualizar el estado de los pedidos.
- Podrá procesar devoluciones y gestionar reembolsos.
- Podrá actualizar la disponibilidad de productos en el inventario.
- Podrá asistir a clientes con consultas y problemas a través de soporte.
- Podrá gestionar y actualizar información sobre productos según directrices del administrador.

Restricciones:

- El trabajador no podrá acceder a la configuración del sistema o la facturación electrónica.
- El trabajador no podrá agregar, eliminar o modificar usuarios del sistema.
- El trabajador no podrá editar o eliminar registros históricos y logs.
- El trabajador no podrá acceder a datos sensibles de los clientes o realizar cambios en la configuración de seguridad del sistema.
- El trabajador no podrá realizar cambios en la configuración de facturación o políticas de precios.

4. Herramientas de desarrollo

Las herramientas que se han visto contempladas en el desarrollo del proyecto son las siguientes:

- **XAMPP:** Utilizado como servidor local que incluye Apache y MySQL para el desarrollo inicial del proyecto.
- **Docker:** Para el despliegue del proyecto en contenedores, incluyendo los microservicios y el API Gateway Kong.
- **Kong:** API Gateway utilizado para gestionar y asegurar el tráfico de APIs.
- **React JS:** Utilizado para el desarrollo del frontend de la aplicación web.
- **React Native:** Utilizado para el desarrollo del frontend de la aplicación móvil.
- **Laravel:** Framework para el backend y la programación de microservicios, trabajando con PHP para las APIs REST.
- **MySQL:** Sistema de gestión de bases de datos utilizado para almacenar y consultar la información de autenticación, contabilidad, ventas, inventario y facturación electrónica.
- **API del Ministerio de Hacienda de El Salvador:** Para realizar pruebas con la facturación electrónica y transmitir documentos tributarios electrónicos (DTE).
- **Herramientas de Monitoreo de Seguridad:** Implementadas para detectar y alertar sobre actividades sospechosas o anomalías.
- **Herramientas de Pruebas de Penetración:** Para identificar y corregir vulnerabilidades en el sistema.
- **Herramientas de Revisión de Código:** Para realizar análisis estático y detectar problemas de seguridad durante el desarrollo.

5. Presupuesto

En primer lugar, se debe tomar en cuenta el recurso humano: se contratarán 5 trabajadores durante las 10 semanas del proyecto, abarcando el pago de 7 quincenas. Serán 5 ingenieros en Ciencias de la Computación. A cada trabajador se le pagará por quincena \$500. Abarcando las 7 quincenas, \$3500 por trabajador.

Por otro lado, se necesita infraestructura en la nube para la interconexión de la API con la web y la base de datos: máquinas virtuales para el alojamiento de la API, servicios de contenedores, almacenamiento, alojamiento de la base de datos y servicio de hosting.

Los costos de la nube se analizarán para una cantidad de 50 usuarios diarios de la plataforma en un periodo de 1 mes a partir del lanzamiento.

Recurso	Costo por unidad	Cantidad	Unidad	Total
Desarrolladores FullStack	\$ 2.500,00	5	trabajadores	\$ 12.500,00
Máquina Virtual en la nube		30	dias	\$ 69,12
Almacenamiento 1TB en la nube		30	dias	\$ 20,80
Base de datos en la nube		30	dias	\$ 363,14
Kubernetes		30	dias	\$ 84,24
Total				\$ 13.037,30

Figura 10. Presupuesto

6. Cronograma

El proyecto está planificado para desarrollarse a lo largo de 10 semanas, las cuales estarán dentro del ciclo universitario. En la figura 1 se presenta el cronograma detallado de las actividades.

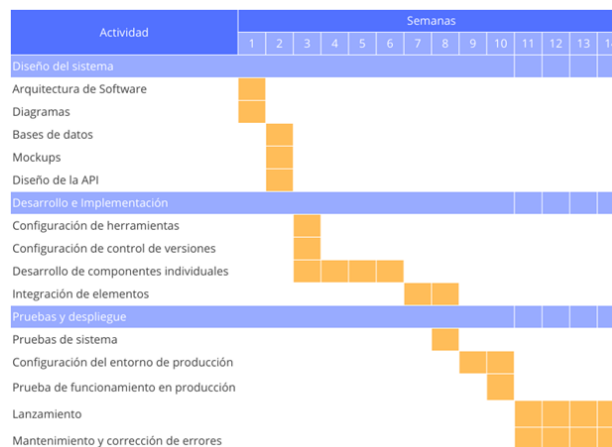


Figura 9. Cronograma

7. Bibliografía

Ministerio de Hacienda de El Salvador. (2023).

Facturación Electrónica: Documentos Tributarios Electrónicos (DTE). <https://factura.gob.sv>

Oblador, S. (n.d.). react-native-vector-icons. *npm*. <https://www.npmjs.com/package/react-native-vector-icons>

Kong Inc.(n.d.). *Register Now*. Retrieved from <https://konghq.com/>