

**Integrantes:**

* Josue Ticona
* Leandro Vergara
* Benjamín Norambuena
* Tamara Pizarro

# Índice

## 1.- Desarrollo de Microservicios

1.1.- Descripción de los microservicios desarrollados

1.2.- Diagrama de Arquitectura

1.3.- Capturas de código

1.4.- Capturas pruebas funcionales en Postman

## 2.- Base de datos y configuración

2.1.- Modelo entidad-relación implementado

2.2.- Capturas de las clases de entidades Java

2.3.- Capturas de la configuración de application.properties y de la conexión exitosa a la base de datos

## 3.- Integración de Servicios

3.1.- Descripción teórica del flujo de trabajo CI/CD

3.2.- Explicación del entorno de prueba: pruebas con Postman y, opcionalmente, interfaz de apoyo desarrollada por el docente para mostrar respuestas JSON

## 4.- Buenas prácticas y colaboración

4.1.- Captura del flujo de trabajo con Git

4.2.- Breve explicación del uso colaborativo del repositorio

4.3.- Ejemplos de buenas prácticas aplicadas

**5.- Conclusión y Reflexión**

**6.- Anexos**

# Desarrollo de Microservicios

## 1.1.- Descripción de los Microservicios desarrollados

* **Gestión de usuarios:** Crear, actualizar, desactivar y eliminar cuentas; asignar y modificar permisos; gestionar roles.
* **Gestión de cuenta:** Registro e inicio de sesión, actualizar información personal, direcciones de envío y detalles de pago.
* **Gestión de pedidos:** Supervisar pedidos, optimizar rutas de entrega, actualizar estado de los pedidos.
* **Realizar pedidos:** El usuario puede realizar pedidos de productos.
* **Gestión de tienda:** Se gestionará la tienda de la sucursal correspondiente.
* **Gestión proveedores:** Se gestionan los proveedores correspondientes.
* **Gestión de envío:** Se gestionarán los envíos realizados. Eliminar, actualizar.
* **Envío:** El usuario podrá elegir la dirección del envío de sus productos.
* **Carrito de compra:** Los productos seleccionados se añaden al carrito de compras.
* **Gestión de inventario:** Agregar, actualizar y eliminar productos; ajustar cantidades de stock.
* **Transacciones:** Registrar ventas, aplicar descuentos, emitir facturas electrónicas.

## 1.2.- Diagrama de Arquitectura

Diagrama de microservicios

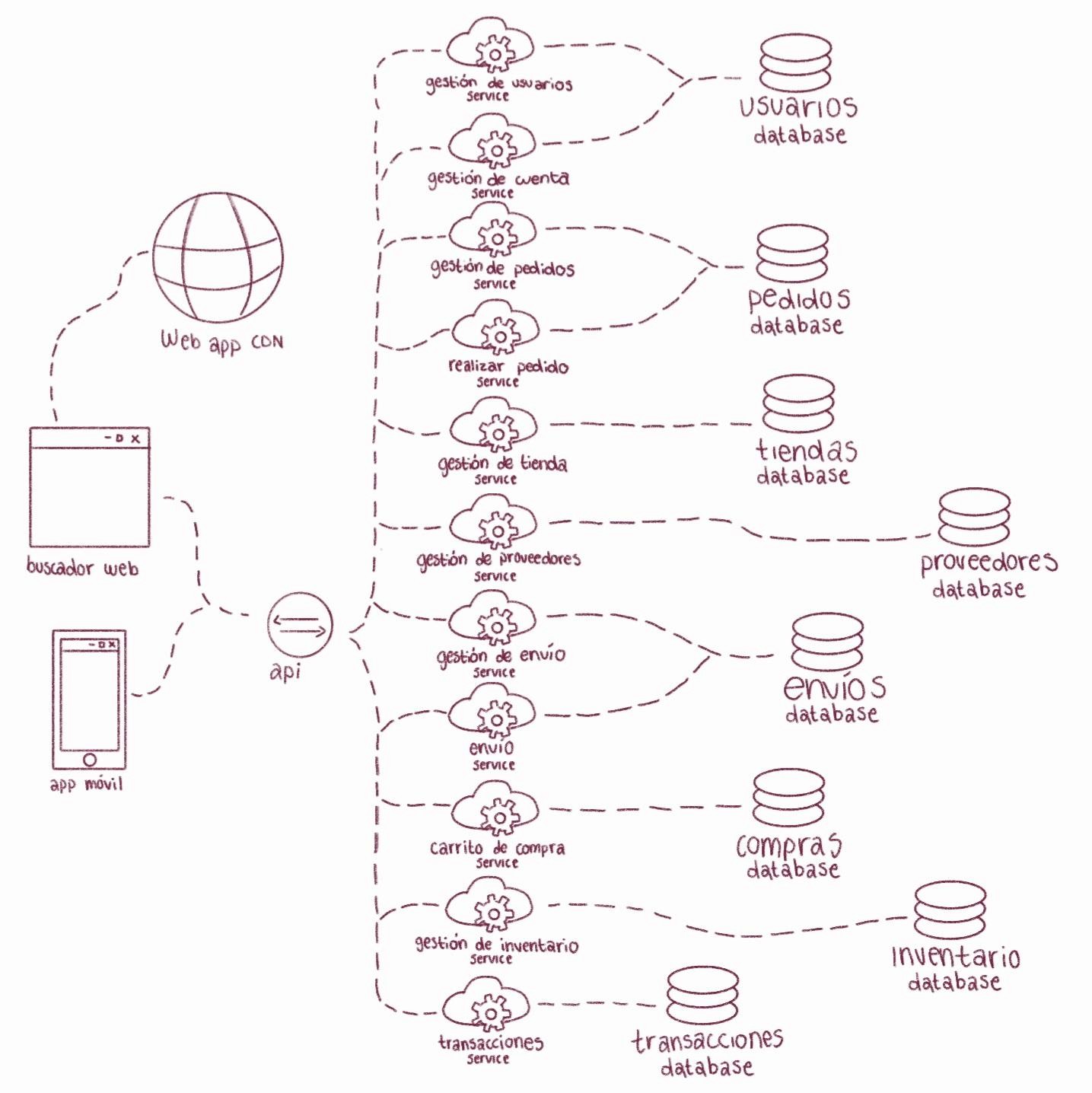


Diagrama de despliegue

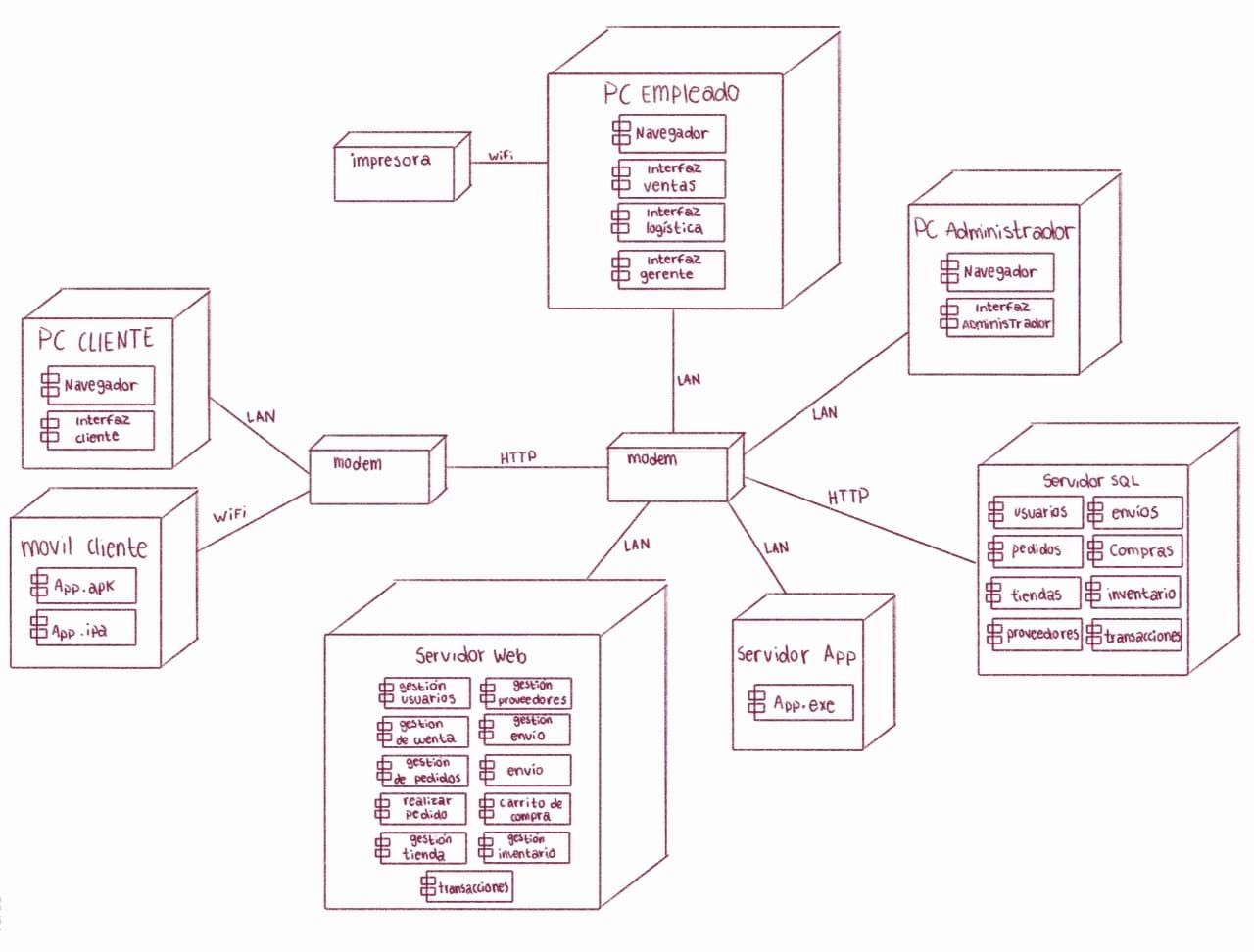


Diagrama casos de uso

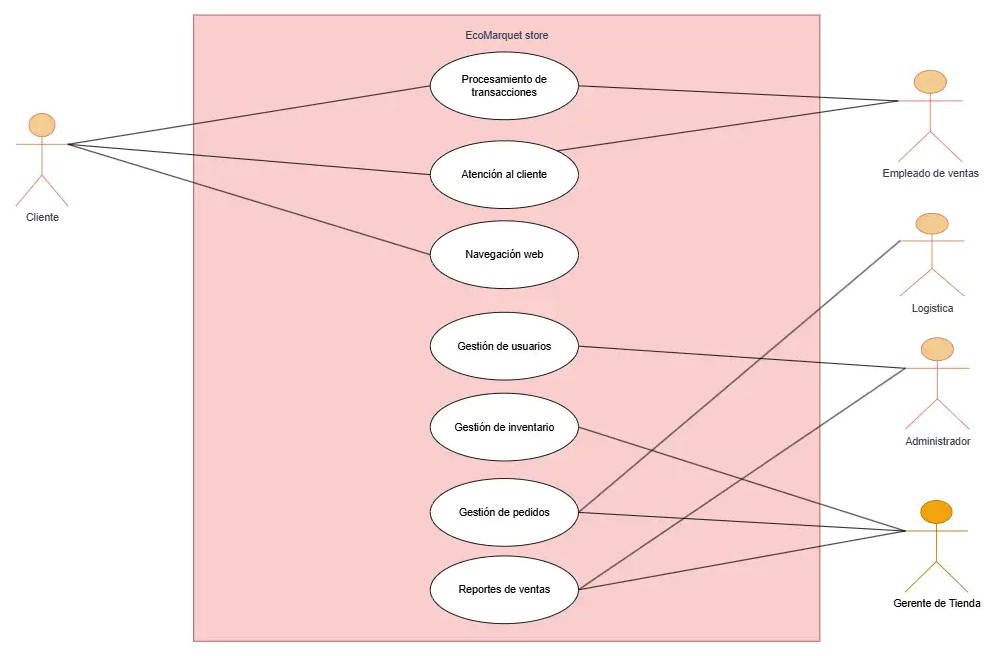
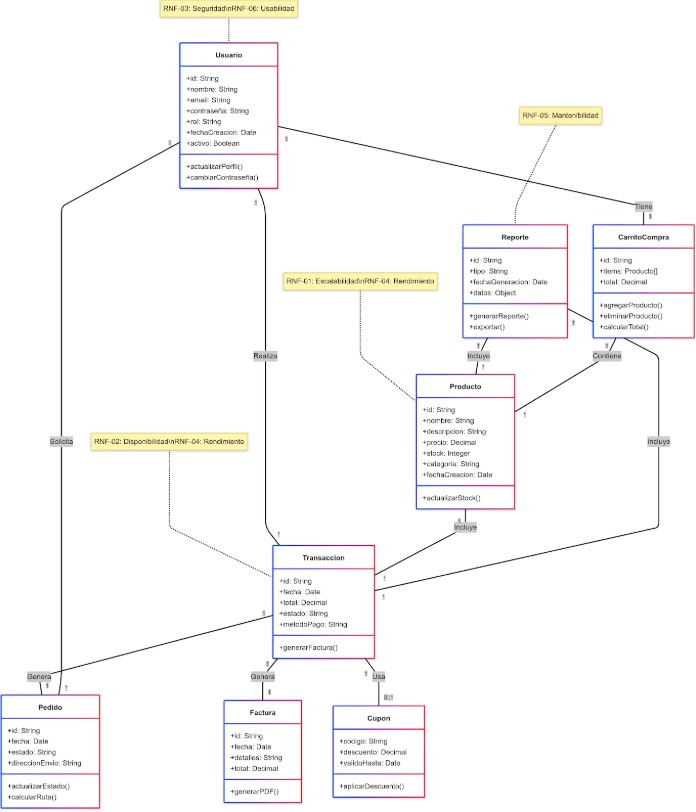


Diagrama de clases



**1.3.- Capturas de código (endpoints, servicios, repositorios)**

**1.4.- Capturas pruebas funcionales en Postman**

# 2. Base de datos y configuración

**2.1.- Modelo entidad-relación implementado**

**2.2.- Capturas de las clases de entidades Java**

**2.3.- Capturas de la configuración de application.properties y de la conexión exitosa a la base de datos**

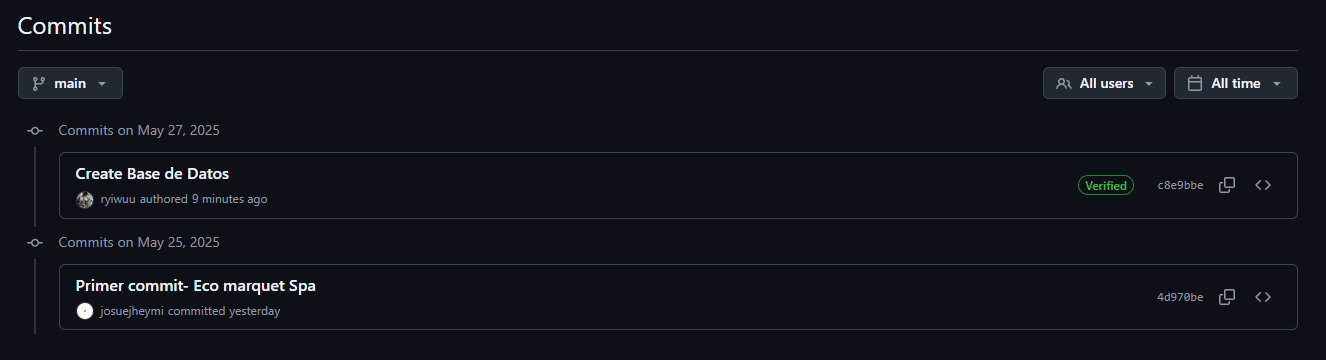
# 3. Integración de Servicios

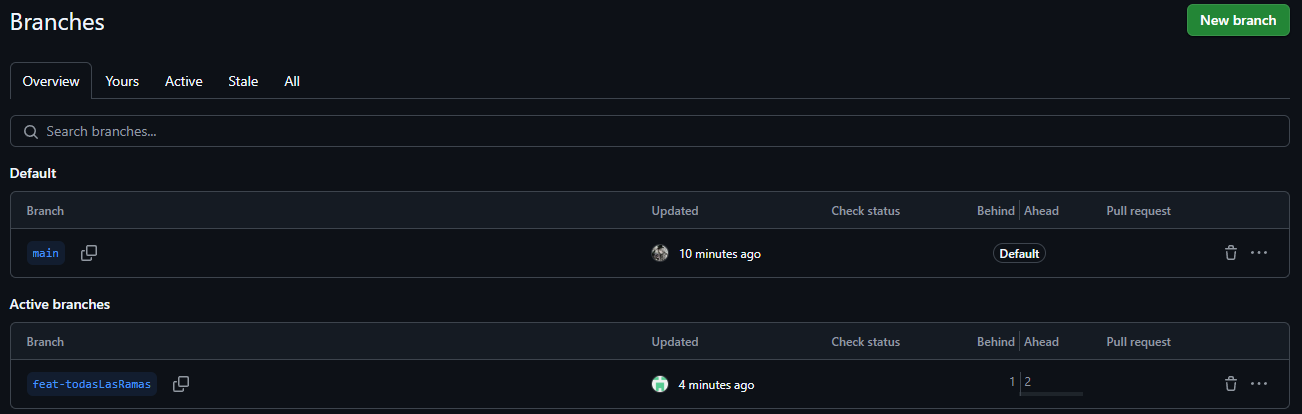
**3.1.- Descripción teórica del flujo de trabajo CI/CD**

**3.2.- Explicación del entorno de prueba: pruebas con Postman y, opcionalmente, interfaz de apoyo desarrollada por el docente para mostrar respuestas JSON**

# 4. Buenas prácticas y Colaboración (GIT)

**4.1.- Captura del flujo de trabajo con Git (commits, ramas, integraciones)**

****

****

## 4.2.- Breve explicación del uso colaborativo del repositorio

Se utilizo GitHub para trabajar en conjunto y mantener cada sección que se elaboro consistente uno con el otro.

Además de retroalimentarse uno al otro cuando fuere necesario.

**4.3.- Ejemplos de buenas prácticas aplicadas (nombres claros, modularidad, uso de anotaciones, REST)**

# 5. Conclusión y Reflexión

**6. Anexos (enlaces a repositorio, video de prueba si aplica)**

[**https://github.com/josuejheymi/ecomarquet-vm**](https://github.com/josuejheymi/ecomarquet-vm)