Manual Tecnico

Nombre: Sergio Rolando Hernández Pérez
carnet: 201931555

Este proyecto fue diseñado como un sistema basado en microservicios para la gestión de hoteles, restaurantes, reservaciones y promociones, entre otros servicios, en una plataforma llamada "Comer y Dormir". Cada uno de los nueve microservicios representa una funcionalidad independiente del sistema, con una arquitectura basada en servicios RESTful que permite la escalabilidad y modularidad del sistema.

El desarrollo se llevó a cabo en **Fedora 40**, con una máquina de **8GB de RAM**, utilizando **Docker** como plataforma de contenedorización para cada microservicio. A pesar de los intentos iniciales, se decidió no implementar un servicio de **discovery service** junto con el **API Gateway** debido a problemas técnicos y de tiempo con **Docker**. Adicionalmente, aunque se consideró un sistema de autenticación completo, por cuestiones de tiempo y enfoque, solo se implementó el encriptado de contraseñas de usuarios utilizando **Encoder**.

Objetivos del Sistema

1. **Modularidad**: Se buscó diseñar una solución que permitiera agregar o eliminar microservicios sin impactar al resto del sistema.
2. **Escalabilidad**: El uso de microservicios facilita la escalabilidad tanto horizontal como vertical del sistema, permitiendo que cada servicio se despliegue independientemente.
3. **Separación de Responsabilidades**: Cada microservicio maneja una responsabilidad única, desde la gestión de reservas hasta la generación de reportes.
4. **Despliegue Independiente**: Todos los microservicios fueron diseñados y desplegados usando **Docker**.

Descripción de los Microservicios

A continuación, se describe cada uno de los microservicios implementados, junto con su funcionalidad y los puertos en los que operan.

1. User Microservice (Puerto 8081)

Este microservicio se encarga de la gestión de usuarios del sistema. Aquí se crean, actualizan y eliminan los perfiles de usuarios. Además, se implementó el encriptado de contraseñas utilizando **BCrypt** para proteger los datos sensibles de los usuarios. Los roles principales gestionados son:

- **CLIENT**

- **EMPLOYEE**
- **MANAGER**

La autenticación completa fue descartada por problemas de tiempo, pero se logró implementar la encriptación de contraseñas como medida de seguridad.

2. Hotel Microservice (Puerto 8082)

Este microservicio gestiona la creación y mantenimiento de hoteles. Permite:

- Registrar hoteles.
- Modificar información de hoteles.
- Asignar habitaciones a cada hotel.

Cada hotel tiene su propio conjunto de habitaciones, que pueden tener diferentes tipos, como **SUITE** o **NORMAL**.

3. Reservation Microservice (Puerto 8083)

Este servicio es fundamental en la plataforma, permitiendo a los usuarios:

- Hacer reservaciones de habitaciones en hoteles.
- Modificar o cancelar reservaciones.
- Realizar el proceso de **check-in** y **check-out** de manera autónoma.

El sistema asegura que las habitaciones asignadas estén disponibles y sincronizadas con los datos del **Hotel Microservice**.

4. Restaurant Microservice (Puerto 8084)

El **Restaurant Microservice** se encarga de gestionar restaurantes asociados a los hoteles. Las funcionalidades principales incluyen:

- Crear y gestionar restaurantes.
- Registrar platillos en el menú de cada restaurante.
- Actualizar el precio de los platillos.

Cada restaurante puede estar asociado a un hotel o funcionar de manera independiente.

5. Order Microservice (Puerto 8085)

Este microservicio gestiona las órdenes de los usuarios en los restaurantes. Las órdenes pueden ser pagadas mediante diferentes métodos y cada transacción queda registrada para posterior consulta. Las funciones clave son:

- Crear órdenes.

- Cancelar o modificar órdenes.
- Registrar el pago de las órdenes.

6. Promotion Microservice (Puerto 8086)

El **Promotion Microservice** permite a los gerentes de los hoteles y restaurantes crear promociones para atraer más clientes. Estas promociones pueden ser aplicadas tanto a habitaciones de hoteles como a platillos en los restaurantes. Las funcionalidades incluyen:

- Creación de promociones.
- Envío de promociones a los usuarios.
- Modificación o eliminación de promociones activas.

Este microservicio fue diseñado para trabajar en conjunto con los servicios de hoteles y restaurantes, ofreciendo a los clientes descuentos o ventajas en sus reservas o consumos.

7. Report Microservice (Puerto 8087)

Este microservicio se encarga de la generación de reportes sobre las diferentes actividades dentro del sistema, como ingresos, pagos de empleados, consumos en restaurantes, y reservaciones en hoteles. Algunas de las funcionalidades clave incluyen:

- Generación de reportes financieros.
- Reportes de los consumos de clientes.
- Listados de empleados y pagos.

Los reportes se generan en formatos PDF y pueden ser descargados por los administradores del sistema.

8. Feedback Microservice (Puerto 8088)

Este servicio permite a los usuarios dejar reseñas y comentarios sobre los hoteles y restaurantes que han visitado. Las funcionalidades principales incluyen:

- Crear reseñas de hoteles y restaurantes.
- Gestionar comentarios para dar retroalimentación tanto positiva como negativa a los administradores.

Este microservicio está diseñado para mejorar la experiencia del usuario y ayudar a los hoteles y restaurantes a mejorar sus servicios.

9. Payroll Microservice (Puerto 8089)

Este microservicio es responsable de la gestión de pagos a los empleados del sistema. Cada hotel o restaurante puede registrar a sus empleados y programar los pagos. Las funciones incluyen:

- Registrar empleados y sus salarios.
- Procesar pagos periódicos.
- Generar reportes de pago para cada establecimiento.

Arquitectura de Despliegue

La arquitectura de microservicios fue diseñada y desplegada utilizando **Docker**. Cada microservicio tiene su propio contenedor Docker y base de datos independiente, lo que permite una mayor flexibilidad y facilidad de mantenimiento. El uso de **Docker Compose** facilitó la gestión de todos los servicios, permitiendo que se iniciaran, detuvieran y reiniciaran de manera coordinada.

Cada microservicio tiene su propia base de datos asociada (por ejemplo, `user_db`, `hotel_db`, `reservation_db`), con contraseñas y configuraciones propias para asegurar la integridad de los datos.

Stack Tecnológico

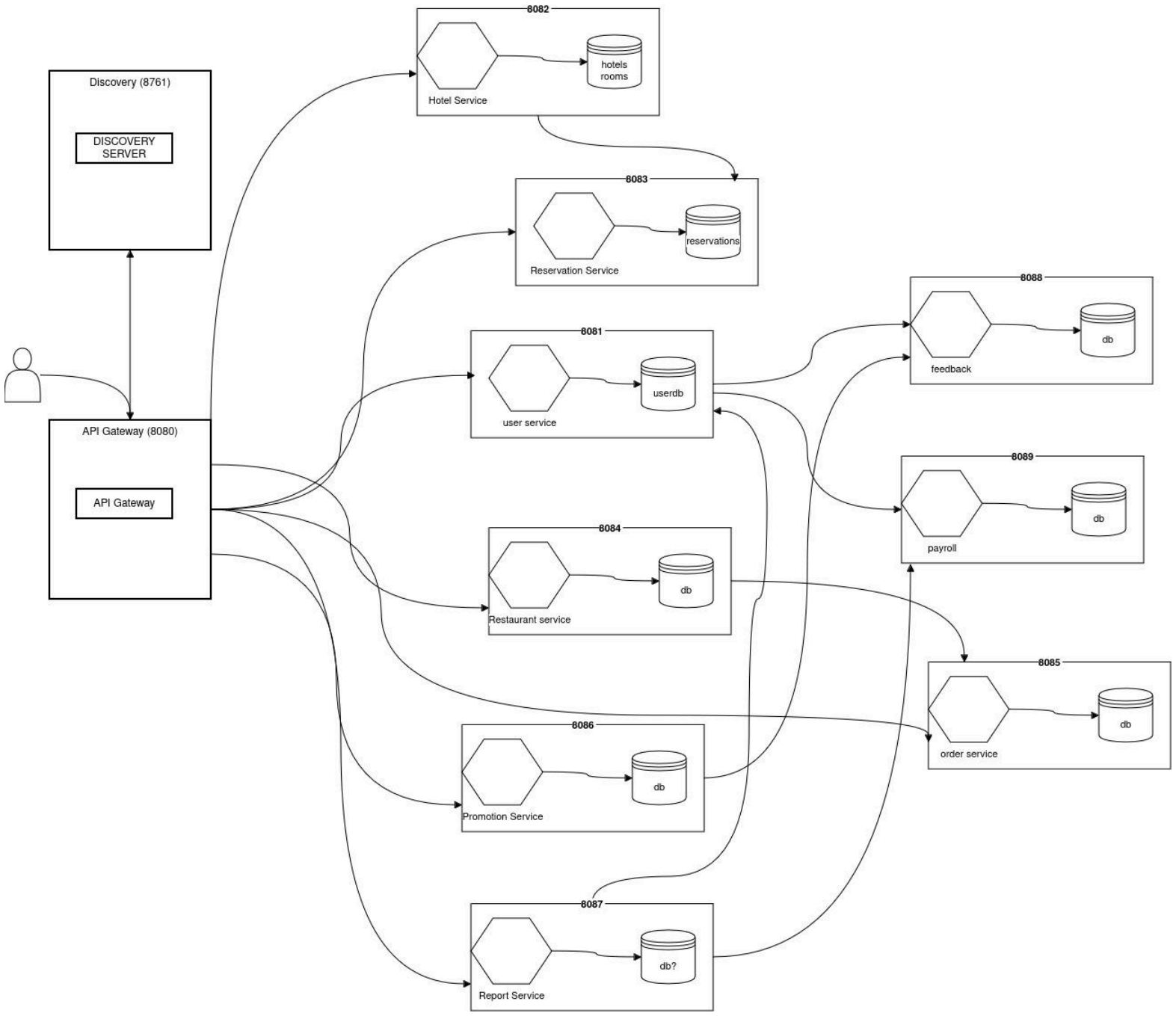
- **Lenguaje**: Java 21
- **Framework**: Spring Boot
- **Base de Datos**: MySQL
- **Contenerización**: Docker y Docker Compose
- **Sistema Operativo**: Fedora 40
- **Seguridad**: Encriptado de contraseñas con passwordEncoder
- **Documentación y Pruebas**: Swagger

Problemas y Decisiones Técnicas

A lo largo del desarrollo del proyecto, se encontraron algunos desafíos que resultaron en la modificación de ciertos aspectos del diseño original:

1. **Discovery Service y API Gateway**: Inicialmente, se planeó utilizar un servicio de descubrimiento junto con el API Gateway para la orquestación de microservicios. Sin embargo, debido a problemas con la integración de **Docker**, este plan fue descartado, y se optó por un enfoque más simple, donde cada microservicio es llamado directamente desde el **API Gateway**.
2. **Autenticación**: Se planificó implementar un sistema completo de autenticación con **JWT**. Sin embargo, debido a la falta de tiempo, se decidió no implementar el sistema de autenticación completo. En su lugar, solo se implementó el encriptado de contraseñas para proteger la información de los usuarios.


CAROLINA ACADEMIA GUATEMALENSIS
LITTERAS ORBIS CONSPICUA


Discovery (8761)
DISCOVERY SERVER
API Gateway (8080)
API Gateway
8082
hotels
rooms
Hotel Service
8083
reservations
Reservation Service
8081
userdb
user service
8084
db
Restaurant service
8086
db
Promotion Service
8087
db?
Report Service
8088
db
feedback
8089
db
payroll
8085
db
order service