

Proyecto AgenciaViajes: Sistema de Microservicios

Bienvenidos a la presentación del proyecto AgenciaViajes, un sistema completo de microservicios diseñado para gestionar hoteles, vuelos y reservas en una agencia de viajes. Esta solución tecnológica aprovecha las ventajas de Spring Boot, Spring Cloud, H2 Database, Eureka Server y API Gateway para crear una arquitectura escalable y mantenible.

A lo largo de esta presentación, exploraremos la arquitectura del proyecto, los módulos principales, sus funcionalidades y cómo interactúan entre sí para proporcionar un sistema robusto de gestión de viajes.



por Ruiz Rosa Óscar

Microservices

API Gateway
Microka Server

Eureka Server

API Gateway

Camverty

Microservices

Microservices

Arquitectura del Sistema

1

ApiGateway

Punto de entrada único para todas las solicitudes del cliente. Enruta las peticiones a los microservicios correspondientes a través del puerto 8080.

2

EurekaServer

Servidor de descubrimiento que registra y localiza los microservicios disponibles. Opera en el puerto 8761 y no se registra a sí mismo ni busca otros servicios.

3

Microservicios

Tres servicios especializados: ServiceHotel (puerto 8081), ServiceVuelo (puerto 8082) y ServiceReserva (puerto 8083), cada uno con responsabilidades específicas.

Módulo ApiGateway

Función Principal

Actúa como punto de entrada único para todas las solicitudes, simplificando la interacción del cliente con los microservicios y proporcionando una capa de abstracción.

Rutas Configuradas

Gestiona tres rutas principales:
/hotels/** para ServiceHotel,
/flights/** para ServiceVuelo y
/reservations/** para ServiceReserva.

Dependencias Clave

Utiliza spring-cloud-starter-gateway para la gestión de rutas y spring-cloud-starter-netflix-eureka-client para la comunicación con Eureka.



Módulo ServiceHotel

1

GET /hotels

Recupera la lista completa de hoteles disponibles en el sistema, permitiendo a los usuarios explorar todas las opciones de alojamiento.

2

POST /hotels

Crea un nuevo registro de hotel con detalles como nombre, categoría, precio y disponibilidad, ampliando las opciones para los clientes.

3

GET /hotels/{id}

Obtiene información detallada de un hotel específico mediante su identificador único, facilitando la consulta de información concreta.

4

PUT & DELETE

Permite actualizar la información de un hotel existente o eliminarlo del sistema cuando sea necesario, manteniendo la base de datos actualizada.



Módulo ServiceVuelo

Gestión Básica

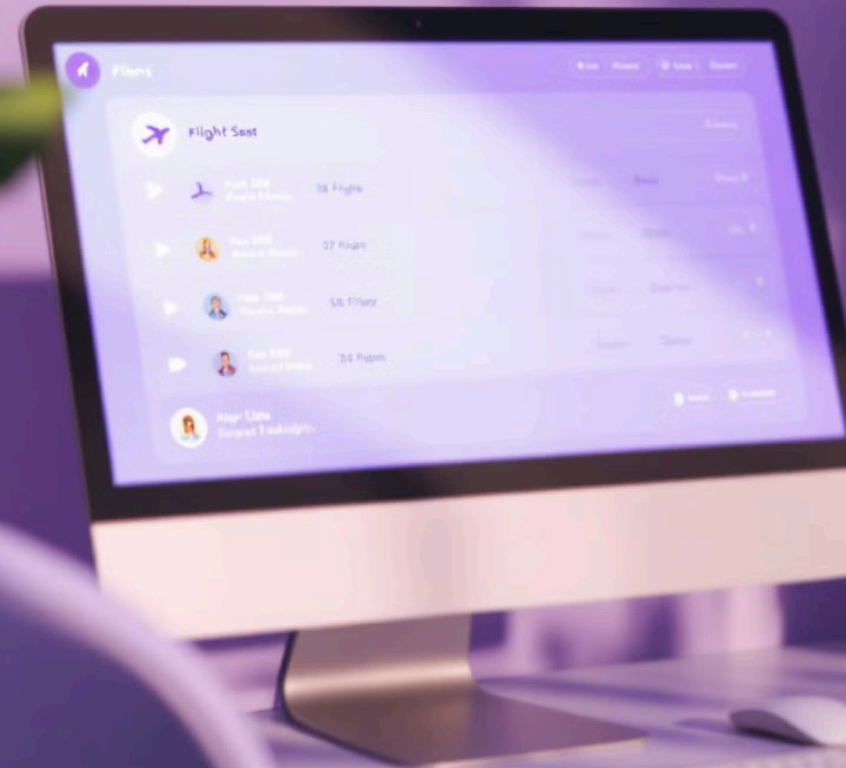
Implementa operaciones CRUD completas (GET, POST, PUT, DELETE) para administrar la información de vuelos en el sistema.

Reducción de Plazas

Ofrece un endpoint especializado (PUT /flights/{id}/reduce/{plazas}) para actualizar automáticamente las plazas disponibles cuando se realizan reservas.

Filtrado por Disponibilidad

Permite buscar vuelos con un número específico de plazas disponibles (GET /flights/available/{plazas}), optimizando la experiencia de búsqueda.





Módulo ServiceReserva



Operaciones CRUD

Implementa endpoints completos para crear, leer, actualizar y eliminar reservas, proporcionando una gestión integral de las reservas de viajes.



Integración con Vuelos

Se comunica con ServiceVuelo para actualizar automáticamente las plazas disponibles cuando se realiza una nueva reserva, manteniendo la consistencia de datos.



WebFlux

Utiliza Spring WebFlux para comunicaciones reactivas entre servicios, mejorando la eficiencia y la capacidad de respuesta del sistema.

Ejemplos de Uso

Crear un Hotel

POST /hotels

```
{  
  "nombre": "Hotel Ejemplo",  
  "categoria": "5 estrellas",  
  "precio": 150.0,  
  "disponible": true  
}
```

Crear un Vuelo

POST /flights

```
{  
  "compania": "Aerolínea Ejemplo",  
  "fecha": "2024-12-01",  
  "precio": 300.0,  
  "plazasDisponibles": 100  
}
```

Crear una Reserva

POST /reservations

```
{  
  "nombreUsuario": "Juan Pérez",  
  "dni": "12345678",  
  "hotelId": 1,  
  "vueloId": 1  
}
```

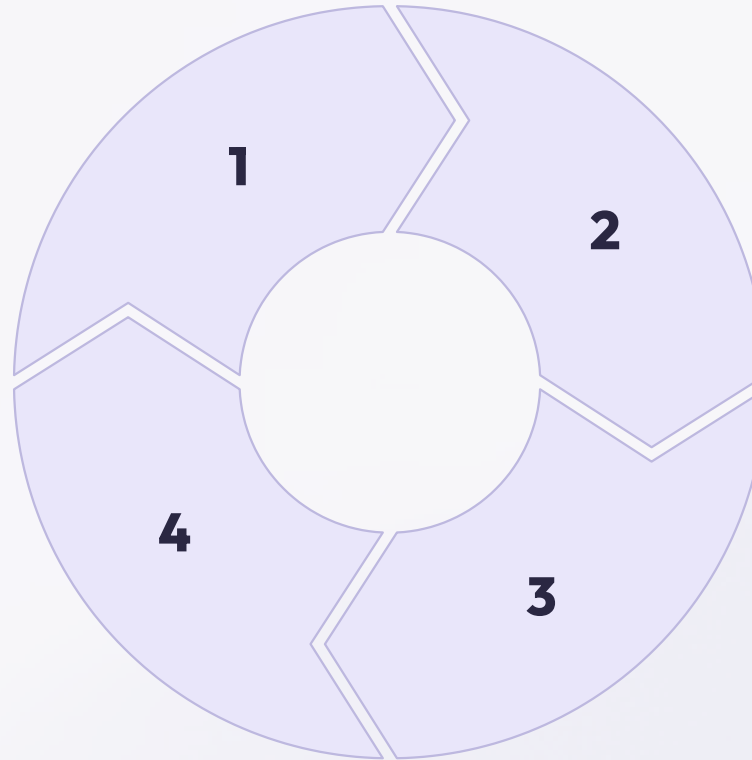
Conclusiones y Próximos Pasos

Beneficios Actuales

Escalabilidad gracias a la arquitectura de microservicios

Mejoras Futuras

Autenticación, bases de datos persistentes y pruebas automatizadas



Mantenimiento

Facilidad de mantenimiento y actualización de componentes individuales

Integración

Comunicación eficiente entre servicios mediante Eureka y API Gateway

El proyecto AgenciaViajes demuestra las ventajas de una arquitectura de microservicios bien diseñada. Para ejecutarlo, clone el repositorio y lance los módulos en orden: primero EurekaServer, luego los servicios y finalmente ApiGateway. Acceda a través de <http://localhost:8080>.