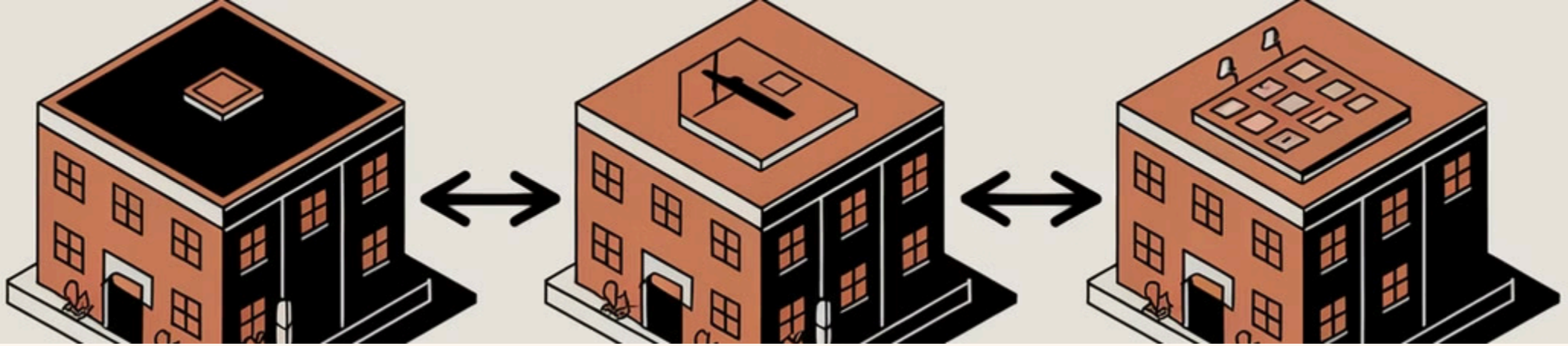


Aplicación de Agencia de Viajes con Microservicios

Presentación de la aplicación agencias de viajes basada en microservicios con SpringBoot, que permite gestionar reservas de vuelos y hoteles.

 **por Juan Carlos Filter Martín**



Arquitectura de Microservicios

1

Hoteles

Gestiona operaciones **CRUD** y **reservas mediante POST**. Al realizar una reserva, actualiza automáticamente la disponibilidad del hotel seleccionado.

2

Vuelos

Permite operaciones **CRUD** y **reservas mediante POST**. Al confirmar una reserva, actualiza la disponibilidad del vuelo seleccionado en tiempo real.

3

Reservas

Registra las reservas asociando vuelos y hoteles a usuarios específicos. Ofrece funcionalidades para listar reservas por nombre, ID o DNI, y permite eliminar reservas existentes.

Microservicios Eureka - Api Gateway - Security

Eureka Server

Actúa como servidor de descubrimiento para todos los microservicios.

Permite **visualizar los microservicios activos y sus puertos** asignados, facilitando la monitorización del sistema.

API Gateway

Centraliza todas las peticiones de los clientes, **unificando los microservicios** en un solo puerto.

Implementa la seguridad mediante Token JWT y gestiona el filtrado de endpoints según los permisos del usuario.

Security

Se encarga exclusivamente de la **generación y validación de tokens JWT**, garantizando la autenticación segura de los usuarios en el sistema.

Bases de Datos Independientes

Se ha implementado una arquitectura con bases de datos independientes para cada microservicio, garantizando el aislamiento y la autonomía de cada componente del sistema.



Base de Datos de Hoteles

Almacena información detallada sobre hoteles, habitaciones disponibles, tarifas y gestiona el historial de reservas del servicio de hoteles.



Base de Datos de Vuelos

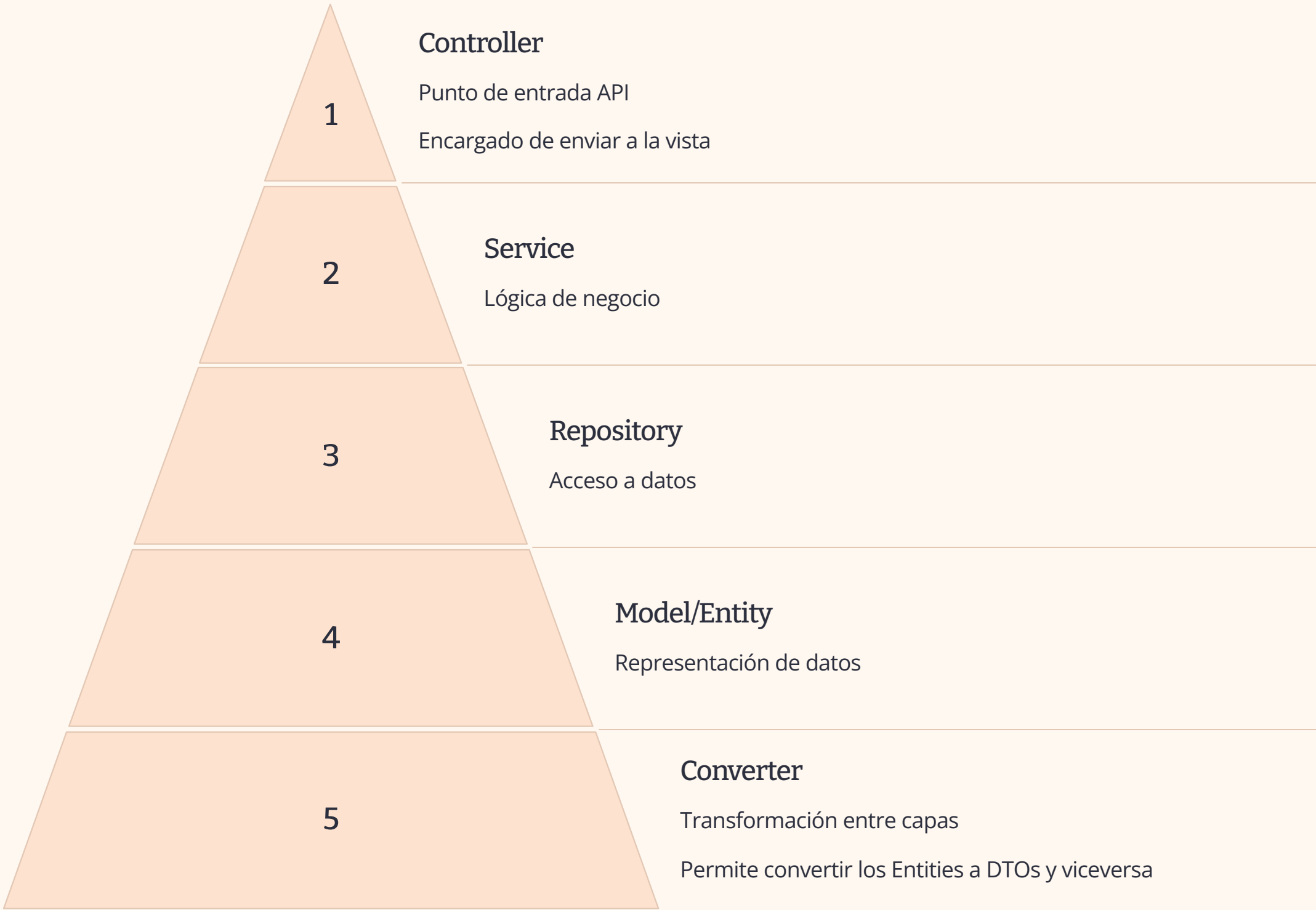
Contiene datos de aerolíneas, rutas, asientos disponibles y mantiene registros de todas las reservas de vuelos realizadas.



Base de Datos de Reservas

Centraliza las referencias a reservas de hoteles y vuelos, manteniendo la integridad referencial entre los distintos servicios del sistema.

Estructura interna de los Microservicios



Cada microservicio sigue una arquitectura en capas bien definida. El Controller maneja las peticiones HTTP y realiza validaciones básicas. El Service implementa la lógica de negocio mientras que el Repository gestiona el acceso a la base de datos. Los modelos y entidades representan los datos, y los Converters facilitan la transformación entre DTOs y Entities.

Dependencias Adicionales



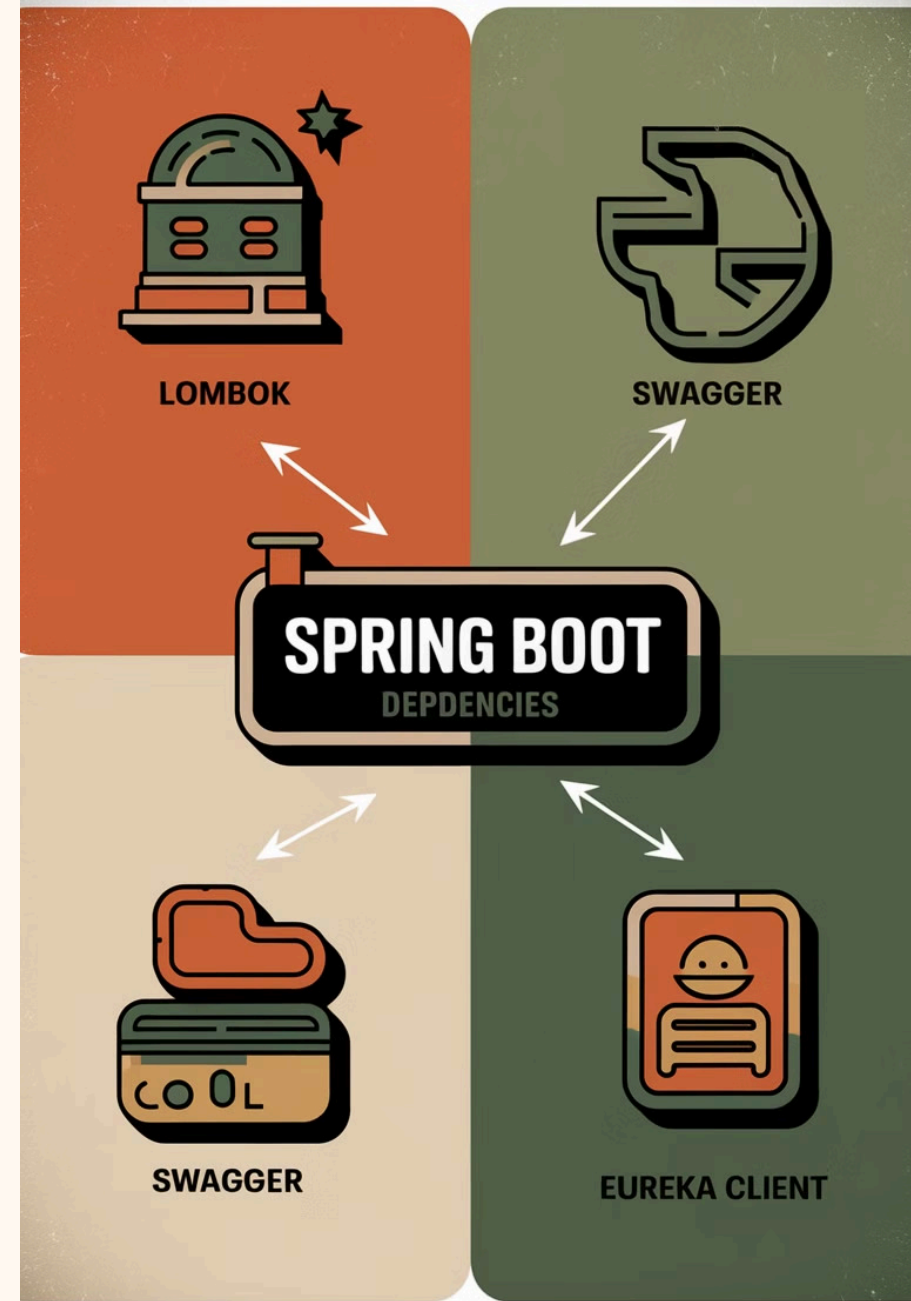
Lombok

Librería de Java que **reduce el código mediante anotaciones como @Getter, @Setter y @AllArgsConstructor**, generando automáticamente métodos sin necesidad de escribirlos manualmente.



Swagger

Permite probar APIs REST de forma interactiva, **similar a Postman, proporcionando una interfaz web** generada automáticamente para facilitar las pruebas y documentación.



Frontend con Angular y Bootstrap

Para el desarrollo del frontend he utilizado **Angular**, un framework potente y versátil que permite la creación de aplicaciones web dinámicas de una sola página (SPA).

Angular

Framework de JavaScript mantenido por Google que facilita el desarrollo de aplicaciones web escalables mediante componentes reutilizables, servicios y enrutamiento avanzado.

Bootstrap

Se ha integrado **Bootstrap** como framework CSS para proporcionar estilos consistentes, componentes responsive y una interfaz de usuario moderna sin necesidad de escribir CSS personalizado extensivo.



Desafíos y Soluciones

Problema Inicial

El mayor desafío fue **implementar la seguridad**. Inicialmente se intentó con Basic Auth, pero surgieron **problemas al filtrar endpoints**, resultando en errores 401 Unauthorized.

Solución Final

Tras varias pruebas y ajustes, **se logró implementar correctamente la autenticación JWT** con filtrado de endpoints, permitiendo un control de acceso seguro y eficiente.

1

2

Cambio de Enfoque

Se decidió **cambiar a JWT**, aunque no se había cubierto en clase, lo que requirió investigación adicional y aprendizaje autónomo para implementar correctamente esta solución.

3

Se consiguió recoger el token pero igual que con Basic Auth, este **también dio problemas** a la hora de **aplicar los filtros** a los endpoints.