



Fundamentos de Arquitectura Cloud

Estudiante: Ismer Jiménez 26.675.538-4

Evaluación del módulo 2

Consigna del proyecto

Introducción

El presente documento describe la arquitectura técnica del sistema de reservas en la nube, desarrollado con una arquitectura basada en microservicios utilizando Node.js y Express. El objetivo principal es proporcionar un sistema escalable, modular y seguro para gestionar reservas de recursos.

Arquitectura General

La arquitectura está compuesta por cuatro microservicios principales:

- - Auth Service: gestiona la autenticación de usuarios.
- - Booking Service: permite crear, consultar y cancelar reservas.
- - Availability Service: administra la disponibilidad de los recursos.
- - API Gateway: actúa como punto de entrada único para el sistema.

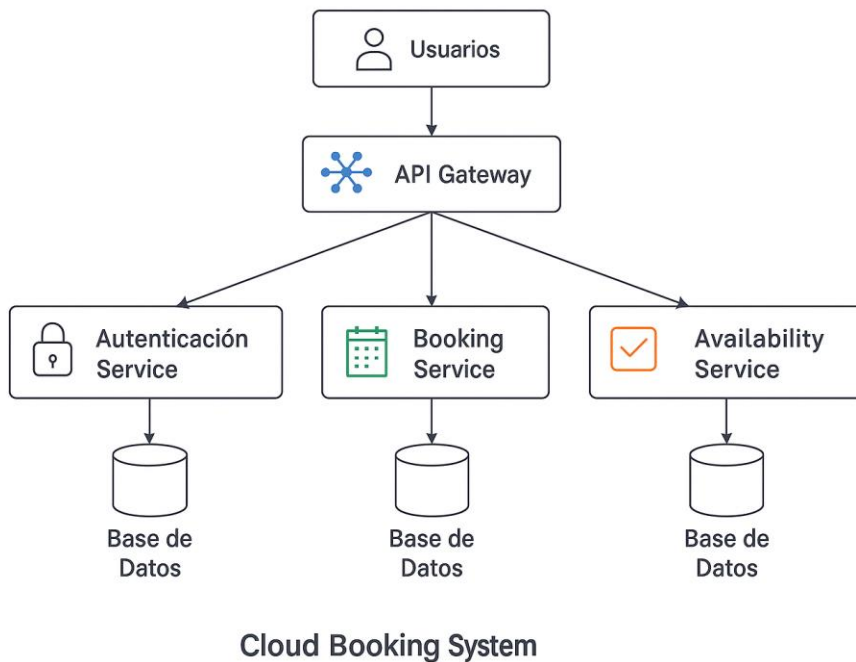


Figura 1: Diagrama de arquitectura del sistema de reservas

Tecnologías Utilizadas

- - Node.js y Express para los servicios backend.
- - Jest para pruebas unitarias.
- - GitHub para control de versiones.
- - Docker (planificado) para contenedores y despliegue.

Descripción de los Servicios

- **Auth Service:** Se encarga de la autenticación y validación de usuarios, con posibilidad de extenderlo usando JWT.
- **Booking Service:** Gestiona la lógica relacionada con reservas: creación, modificación y cancelación.
- **Availability Service** Valida y administra la disponibilidad de los recursos en tiempo real.
- **API Gateway** Canaliza todas las solicitudes hacia los microservicios correspondientes, permitiendo una entrada unificada.

Seguridad

Cada microservicio está diseñado para funcionar de forma aislada y segura. Se plantea el uso de autenticación basada en tokens (JWT) para proteger los endpoints y encriptar los datos sensibles del usuario.

Pruebas

Se implementaron pruebas unitarias básicas en cada microservicio usando Jest. Se recomienda incluir pruebas de integración y pruebas de carga utilizando herramientas como JMeter o k6.

Despliegue

El sistema puede ejecutarse localmente con Node.js. Se sugiere su futura dockerización y despliegue en plataformas como AWS o GCP para aprovechar autoescalado y balanceo de carga.

Conclusión

La arquitectura diseñada cumple con los requisitos de escalabilidad, modularidad y seguridad. Además, facilita el mantenimiento, la extensión y el despliegue en entornos cloud modernos.