

Especialización en Back End I

Patrones de diseño

Objetivo

Identificar en cada punto qué patrones de diseño podrían aplicarse en la arquitectura de microservicios propuesta y hacer un diagrama integral de la arquitectura con los patrones aplicados.

- Ejercitación grupal
- Nivel de complejidad: bajo 🔥

Consigna

¿Recuerdan la flamante fintech que nos contactó para que los asesoremos técnicamente con la arquitectura de microservicios que planteaban implementar? Bueno, volvieron "en forma de fichas" y a tocar la puerta del departamento de arquitectura de DH Soluciones.

Como era de esperarse, decidieron lanzarse a implementarla con apenas pocos conocimientos y los problemas no tardaron en aparecer. Entre ellos, podemos citar:

- La complejidad de realizar pruebas, implementar y de comprender qué resuelve (para el negocio) cada uno de los componentes del sistema distribuido. Al mismo tiempo que la dificultad para emular ambientes cuando intentan ejecutar decenas o cientos de microservicios en un mismo entorno de desarrollo local.
- Problemas para contar con una única configuración (centralizada) del universo de microservicios de la compañía, y que permita que al actualizarla impacte en aquellos microservicios afectados.
- Los obstáculos que encuentran para orquestar la gran cantidad de microservicios. Nos dijeron que cuentan con alrededor de 15, ¡pero no están seguros de la cantidad!



- La visibilidad que requieren algunos servicios hacia el exterior (dominio público) y aquellos otros que necesitan ocultar porque son de uso meramente interno, como lo son aquellos que acceden a datos "sensibles" de los clientes (datos privados de identificación, movimientos financieros, inversiones en curso, etc.).
- Pérdida de trazabilidad de errores de los servicios. Nos explican que algo recurrente que les sucede es que el sistema —una vez que intenta hacer las transferencias a otra cuenta— falla al momento de volver a autenticar con el usuario. Lo mismo si se intentara con un código de confirmación.
- Por último, encuentran que tienen que recurrir a miles de registros, de distintas fuentes, e intentar "cruzarlos" para ver por qué se producen errores o comportamientos anómalos en el sistema.

En línea con nuestro trabajo anterior, esta vez los vamos a volver a ayudar, pero apoyándonos sobre los patrones de diseño que conocemos, asociándolos con los distintos problemas arriba citados.

¡Éxitos!