

DOCUMENTO DE RIESGOS

Arquitectos cognitivos:

- Jesús Sebastián Tocas Atarama
- Pablo Atahonero García de Blas

Nosotros, como arquitectos cognitivos, tuvimos una reunión con los Arquitectos Senior para debatir y cuestionar las siguientes decisiones de diseño que se han tomado.

13° Decisión de Diseño ADD-013: Implementación de la cesta como un microservicio y las localizaciones de terceros

Esta decisión arquitectónica también está relacionada con los requisitos funcionales: RF007-Funciones de la cesta.

En relación con esta decisión de diseño n° 13 de los ASS, los Arquitectos Cognitivos hemos meditado dichas decisiones y hemos localizado algunas desventajas o riesgos que conlleva hacer una cesta con el uso de una caché, el uso de una base de datos MongoDB o las localizaciones de microservicios de terceros.

Desventajas principales de usar una caché:

- Ocupa más espacio en el servidor.
- Cuando realizamos un cambio, los usuarios no lo ven hasta que se renueva la caché.
- Se requieren conocimientos avanzados para el máximo aprovechamiento de la caché.
- Pueden dejarlo vulnerable a piratas informáticos, porque es probable que la caché pueda almacenar información confidencial.

Desventajas de usar MongoDB:

- No permite el uso de los Joins, algo fundamental.

Inconvenientes a la hora de usar localizaciones de microservicios de terceros:

- Las actualizaciones no son frecuentes ni sencillas.
- Son difíciles de gestionar.
- Escalabilidad y rendimiento costosos.

Pese a los aspectos no beneficiosos que pueden aportar dichas soluciones, estamos de acuerdo con su uso para conseguir el mejor funcionamiento posible.

14° Decisión de Diseño ADD-014: Comunicación de los servicios con una API REST

Esta decisión arquitectónica también está relacionada con los requisitos funcionales:
RF002.1-Comunicar los servicios con una API REST y de forma asíncrona.

Hablando de esta decisión de diseño nº 14, los Arquitectos Cognitivos estamos de acuerdo con el uso de una API REST para la correcta comunicación entre microservicios. Eso sí, al usar esto, se pueden presentar una serie de adversidades para implementarlo, como pueden ser:

- Problemas al unir información detallada. Con REST se hacen peticiones que pueden traer más información de la que necesitamos.
- La seguridad podría ser un punto débil si no se sabe gestionar con cautela.
- No define tipos de datos.
- No hay estándares de respuesta.