



MISO

Maestría en Ingeniería de Software

Entrega 4 y 5

2025-2

Prueba de concepto (experimentación)

Contexto

- Finalmente, los ingenieros quieren ver una prueba de concepto (POC) de la arquitectura de solución que su grupo está proponiendo. Dado que su arquitectura debe satisfacer la visión del negocio, no olvide que **su experimentación debe probar que la arquitectura propuesta nos va a ayudar a escalar el negocio a nivel global.**
- Su experimentación debe validar un escenario de calidad por cada atributo de calidad definido en las entregas anteriores. Es decir, su **experimentación debe validar 3 escenarios de calidad.**
- Para lograr lo anterior, los ingenieros han definido las siguientes condiciones:
 1. La arquitectura debe seguir los principios de microservicios basados en eventos. Por tal motivo, la comunicación entre los servicios **debe hacerse usando comandos y eventos.**
 2. Sea claro en la definición de los eventos de acuerdo al escenario de calidad que desea satisfacer ¿Evento de integración o con carga de estado? ¿Por qué? Elabore en el diseño del esquema, desde la tecnología hasta la evolución de los mismos ¿Avro o Protobuf? ¿Event Stream Versioning? Justifique sus decisiones.
 3. Para probar las capacidades de escalado, los ingenieros esperan que usted desarrolle al menos **4 microservicios.** Cabe aclarar que **NO se espera tener los microservicios completamente desarrollados,** solo los comandos, consultas e infraestructura necesaria (tablas, tópicos, repositorios, etc) para satisfacer los escenarios de calidad.
 4. Dada la naturaleza de la comunicación por comandos y eventos, usted debe usar un Broker de eventos. Los ingenieros desean que usted use Apache Pulsar.

Contexto

5. En su experimentación debe ser claro que patrones y tácticas para el almacenamiento de los datos sus microservicios van a usar: ¿descentralizado o híbrido? ¿Por qué?
6. En términos de patrones para el almacenamiento, decida si va usar un modelo clásico CRUD o Event Sourcing. Recuerde que no necesariamente todos los servicios deben usar el mismo patrón de almacenamiento. Es su decisión definir que servicios pueden usar una u otra.
7. Dado la complejidad y longitud de los procesos de negocio, los ingenieros de EDA requieren que usted valide tácticas y patrones para el manejo de transacciones largas. Para tal propósito, se espera que implemente un patrón de sagas. Es decisión de su equipo definir si seguir un modelo de coreografía u orquestación.
8. Para la interacción con sistemas externos y UI, los ingenieros desean que desarrolle un BFF que sirva como base para el API (este puede contar como uno de los 4 servicios).
9. Finalmente, debe poder desplegar sus servicios en la plataforma de preferencia. Justifique el porqué de su decisión.
10. Su documento debe presentar los resultados cuantitativos y cualitativos de su experimentación.
11. Basado en los resultados de su experimentación, debe refinar el mapa de contextos del TO-BE presentado en la primera entrega del curso y los puntos de vistas presentados en la segunda entrega.
12. Los servicios deben ser desarrollados en Python, dado que es el lenguaje que se usa en la compañía.

Sobre la entrega

1. Este proyecto se presentará en dos entregas: una entrega parcial y una entrega final.
2. En la entrega parcial se debe presentar como mínimo 4 microservicios con comunicación asíncrona (comandos y eventos) usando Apache Pulsar y usando un modelo clásico CRUD o Event Sourcing para las bases de datos. En pocas palabras, **la primera entrega debe cumplir con los puntos 1 a 6 descritos en el contexto + punto 9 (despliegue)**.
3. La entrega final pretende refinar la arquitectura, incluyendo el manejo de transacciones largas por medio de la implementación de un coordinador de Sagas, desarrollo e integración de un BFF para la comunicación con externos, despliegue del sistema completo, consolidación de resultados de la experimentación y refinamiento de los puntos de vista y mapa de contexto. En pocas palabras, **la última entrega debe incluir lo presentado en la entrega parcial más los puntos 7 a 11**.
4. Su grupo debe **grabar un video en formato mp4, mov, wmv o avi de NO más de 45 minutos PARA CADA UNA de las entregas**, presentando todos los ítems enumerados anteriormente.
5. Los tutores están en la potestad de ver su código y solicitar una demostración en caso de verlo necesario.
6. Las contribuciones en el código deben ser equitativas. Esta contribución debe reflejarse en los commits y pull requests registrados en su software de control de versionamiento.
7. Por último, **asegúrese de que su código funciona**. Código que no corre **es equivalente nota de cero**.
8. Para cada una de las entregas no olvide incluir en el README o documento anexo, la descripción de actividades por miembro (es decir que hizo cada uno y como colaboró al equipo).