EJERCICIO PARA CUBRIR GAPS

Tomar como referencia el caso práctico enviado en el proceso de selección y realizar las siguientes modificaciones y mejoras:

# Frontend

* Debe ser creado por medio de Optimus y tener una arquitectura microfrontend.
* Agregar autenticación por medio de tokens.
* Proteger rutas para accesos no autorizados por medio de Guards.
* Manejo de errores centralizado y personalizados.
* Manejos de observables.
* Las interfaces deben ser creadas utilizando los componentes del Design System.
* Manejo de estado centralizado (sesión storage, local storage).

# Backend

* Agregar autenticación a los microservicios basados en servicios de autenticación y autorización.
* Implementación correcta de manejo de errores.
* Implementar conceptos de herencia, composición, polimorfismo dentro de los servicios.
* Los servicios deben estar construidos con arquitectura Hexagonal y respetando el uso de las diferentes capas.
* Implementar una arquitectura orientada a eventos para su comunicación, utilizar Brokers como Kafka, Rabbit, etc.
* Implementar el patrón CQRS en su totalidad.
* Implementar el patrón CircuitBreaker en su totalidad.
* Implementar trazabilidad de Logs mediante: Splunk, ELK, Azure Monitor.
* Realizar un correcto uso de los niveles de logs teniendo en cuenta los diferentes ambientes.
* Debe contar con optimización de las APIs/EndPoints, implementando: paginación, uso de caché, compresión de respuestas.
* Aseguramiento de las APIs/EndPoints.
* Todos los APIs/EndPoints deben tener creadas las pruebas automatizadas de acuerdo con el estándar/arquetipo definido por Banco.

# General

* Deben crearse pruebas unitarias para microservicios y micrositios.
* La implementación del ejercicio debe estar alienada a los estándares de desarrollo definidos por Banco.
* Dentro del ejercicio se deben crear pipelines los cuales permitan la integración y el despliegue de los microservicios y micrositios.
* Todo el ejercicio debe ser desplegado en Contenedores mediante: minikube, oracle cloud infrasestructure (tiene un plan gratuito para pocs).
* Todo el código debe aplicar las prácticas de OWASP.

# Defensa

* Al finalizar se deberá realizar la defensa el ejercicio antes ChapterLeaders en un espacio de una hora.
* Se deberá mostrar el ecosistema funcionando.