

Hoja de trabajo No. 5

DDD: Patrones de comunicación entre Contexts

Realizar: El modelo realizado para el bounded context de cada grupo, debe ser ahora relacionado con los bounded context de los demás grupos.

Realizarse: en grupos de 4 alumnos.

Problema a resolver:

En la hoja de trabajo anterior, cada grupo ha desarrollado un context que funciona en forma independiente. Se ha simulado que se recibe información necesaria de otros contextos, por medio de stubs¹, y ahora ya debe realizar la comunicación “real”.

Trabajo a realizar:

Recuerde que para que el sistema sea resiliente², se utiliza el **patrón de diseño Event Sourcing**³ para comunicarse entre los distintos contexts. Esta comunicación debe ser por medio de un sistema de mensajería, como por ejemplo el message broker Rabbitmq⁴ o similar.

Con todos estos elementos a considerar, cada modelo debe utilizar una arquitectura hexagonal, para aislar el dominio/aplicación de toda la infraestructura a utilizar.

Su sistema debe **ser capaz de funcionar**, aunque sea de forma “limitada”, cuando alguno de los otros contextos no este operacional. Se esperaría que en cuanto el contexto que no esté operacional este activo nuevamente, se podría procesar la información necesaria.

Considere además utilizar el patrón CQRS⁵ para poder separar las necesidades de actualización de datos en su modelo, de las consultas que deban realizarse rápidamente.

NOTA IMPORTANTE: considere la utilización de una herramienta de control de la configuración, como Puppet, Chef, Ansible, etc. Ya que un tema muy

¹ Software stub: a temporary substitute for a yet-to-be-developed code.

https://en.wikipedia.org/wiki/Method_stub

² <http://dle.rae.es/?id=WA5onlw>

³ <http://docs.geteventstore.com/introduction/4.0.0/event-sourcing-basics/>

⁴ <https://www.rabbitmq.com/>

⁵ <https://martinfowler.com/bliki/CQRS.html>

importante es que se deben construir en el futuro varios ambientes, tales como el desarrollo, pruebas, producción y estas herramientas hacen la vida más fácil: “la infraestructura como código”.

Y, podría ser que se le facilite la vida si utiliza algún sistema de contenedores, como por ejemplo Docker, para hacer más fácil el despliegue de su aplicación.

Entregable:

1. Creación del ambiente de desarrollo en AWS, que incluya el framework del lenguaje a usar, la base de datos y el message broker.
2. Documento en PDF que muestre los patrones (puede ser el código) utilizados para la comunicación con los otros contextos.
3. Evidencias, tales como videos o screeshots, que muestren la operación de su contexto y como se comunica con los otros contextos.
4. Descripción de la funcionalidad esperada si los otros contextos no están activos y los mecanismos de cómo se procede al reactivarse esos contextos.
5. Dirección al repositorio de github donde reside su proyecto.

Fecha de entrega:

Jueves 29 de agosto.