

ADR001 - SIGEA

Ordoñez Silva, Yonel Jr.

Indice

ADR 001: [Selección de Arquitectura de Microservicios Orientado a Eventos para el SIGEA]	1
Estado	1
Contexto	1
Decisión	2
Consecuencias	2
Positivas	2
Negativas	2
Alternativas consideradas	2

ADR 001: [Selección de Arquitectura de Microservicios Orientado a Eventos para el SIGEA]

Estado

[Propuesto]

Contexto

Se esta desarrollando un sistema que digitaliza y automatiza la gestión integral de actividades de extensión universitaria como cursos, talleres, conferencias y diplomados.

Se requiere de una arquitectura que permita que el sistema escale me manera rapida, que facilite las modificaciones y que responda a eventos externos como pagos o seguimientos.

Decisión

Se decide utilizar el patron de arquitectura de **Microservicios** junto con el patron de arquitectura **Orientada a Eventos**.

Los servicios aun no se definen y se espera que se definan en el ADR 002

Consecuencias

Positivas

- *Alta escalabilidad*, importante si queremos manejar eventos multitudinarios o escalar a diferentes universidades.
- *Flexibilidad*, para poder incorporar o cambiar las necesidades de los usuarios.
- *Mantenibilidad*, nos permite modificar de manera sencilla un componente si asi lo requiere, sin afectar en gran medida al funcionamiento de las demas.
- *Respuesta a Eventos*, vital para el manejo de pagos, notificaciones y seguimientos.

Negativas

- *Alta sobrecarga inicial al equipo*, debido al poco conocimiento previo de nuestro equipo en este tipo de arquitecturas (Microservicio y Orientado a Eventos), la complicada definicion clara de servicios y como cada uno se comunicara con los demas.
- *Conocimientos en CI/CD y de buenas practicas de desarrollo* por parte del equipo de developers.

Alternativas consideradas

- *Monolitica*: Nos permitiria desarrollar de forma mas facil, pero con el tiempo surgirian complicaciones tanto de escalabilidad como de mantenibilidad, ya que no soportaria varios usuarios simultaneamente, y añadir nuevas funcionalidades implicaria modificar toda la estructura.
- *Capas*: Permitiria separar responsabilidades y mantener a cada componente aislado, pero seria un problema con el tiempo ya que pueden surgir nuevos requisitos que impliquen la adicion de mas capas, lo cual puede complicar la comunicacion y gestion de cada una de ellas.
- *Microservicios*: Por si sola brinda una alta escalabilidad y mantenibilidad, por mas que sea dificil de comprender y estructurar de acuerdo al proyecto nos brinda mayores beneficios a largo plazo. A pesar de todas estas bondades, la arquitectura de Microservicios no es suficiente por si sola, ya que dependemos demasiado de los eventos externos y como respondemos a ellos o los coordinamos para el exito de nuestro proyecto.