

## Objetivos

- Entender los patrones adecuados para mejorar la calidad del código
- Comprender la importancia de los diseños arquitectónicos

## Material de trabajo

- ¿Por qué es mejor un ENUM que una constante?
- ¿Cuál es la importancia de delegar la responsabilidad a las clases?
- ¿Por qué es mejor dividir la funcionalidad en interfaces simples?

## Tareas

- ¿Por qué debo siempre crear un POJO para cada request?
- ¿Tengo librerías que me ayudan a convertir entidades en POJOS?
- ¿Cuál es el patrón factory?
- ¿Cuándo usar los enums es conveniente?
- ¿Puedo convertir los enums a diferentes objetos?

## Otros materiales

- <https://www.ilimit.com/blog/arquitecturas-monoliticas-o-arquitectura-de-microservicios-ventajas-e-inconvenientes/#:~:text=Arquitectura%20monol%C3%ADtica.-Los%20primeros%20programas&text=Este%20tipo%20de%20arquitectura%20se,en%20comparaci%C3%B3n%20con%20otras%20arquitecturas.>
- <https://aws.amazon.com/es/microservices/>
- <https://docs.microsoft.com/es-es/azure/architecture/guide/architecture-styles/microservices>
- <https://www.redhat.com/es/topics/microservices>
-