



Taller
Gratuito

¿Paradigmas de programación en Java?

Ing. Aristedes
Novoa

anovoa@galaxy.edu.pe





01

Paradigmas de Programación

02

POO/POA

03

Programación Funcional

04

Programación Reactiva

05

Casos Prácticos

ÍNDICE



Paradigmas

Programación Procedimental

PP



Procedimiento



- Cobol, clíper
- Dbf, c, pascal, c++
- Shell

Programación Orientada a Objetos

POO



Clase



Programación Orientada a Aspectos

POA



Aspecto



Programación Funcional

F(x)



Función

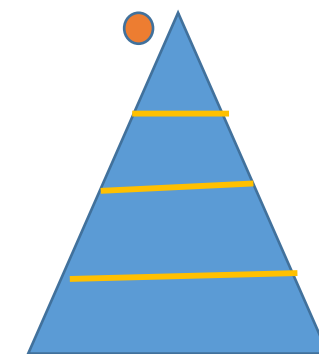


Programación Reactiva

Rx



Mono/Flux



Paradigmas de Programación



Es un **paradigma** de programación que usa **objetos** en sus **interacciones**. Busca que nuestra forma de programar sea mas cercana a las **cosas de la vida real**.



TypeScript

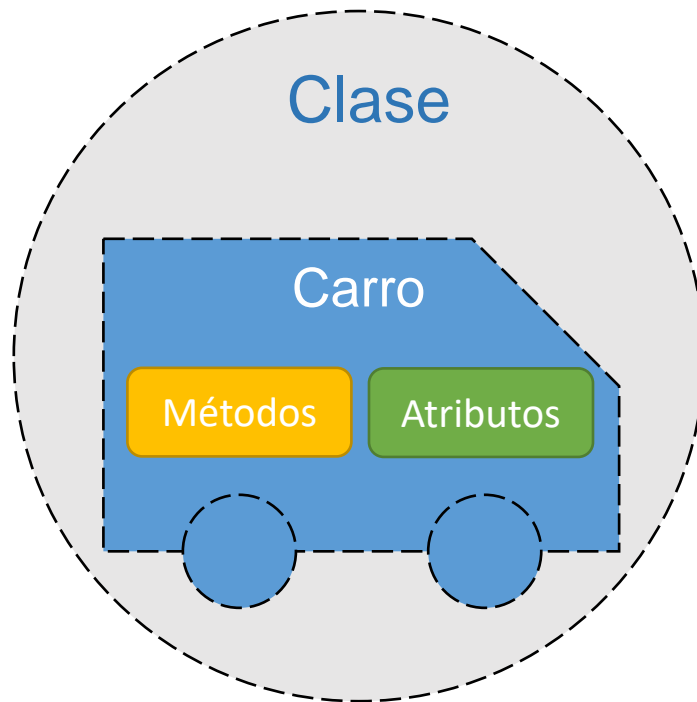
Dart





- 10 Temas

Conceptos fundamentales de POO





La programación funcional es un **paradigma declarativo** orientado a enfocarse en el **"qué"** se debe hacer y no en el **"cómo"** el cual corresponde al **enfoque imperativo**.

Cómo sabemos **Java es un lenguaje orientado a objetos**, sin embargo, en versiones recientes 8 en adelante podemos hacer uso de la programación funcional.

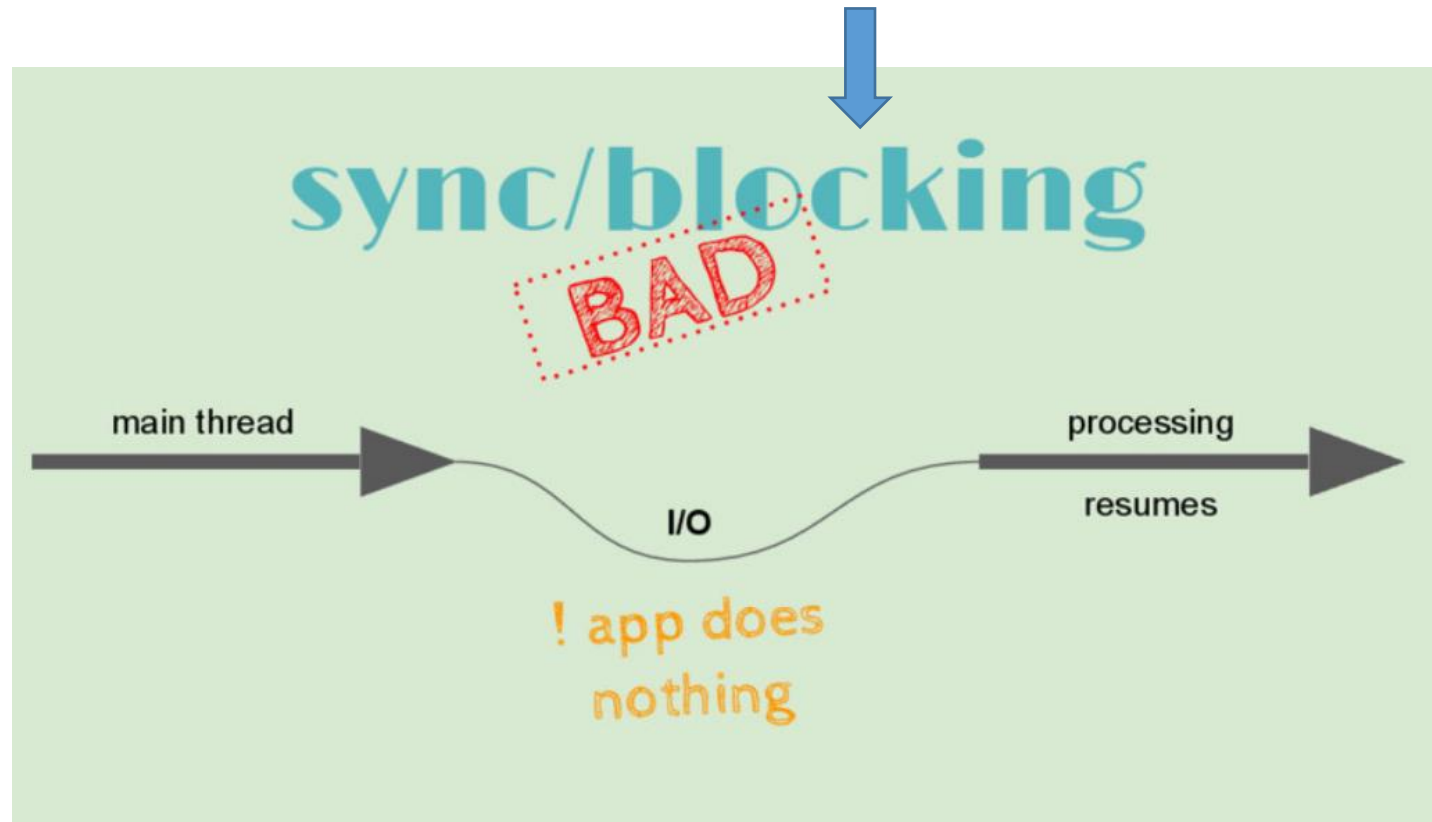
La programación funcional nos permite desarrollar software mucho **limpio**, **más legible**, **fácil de testear**, **eficiente**, **mantener**, **mejorar**, **evolucionar**.

■ Programación Funcional



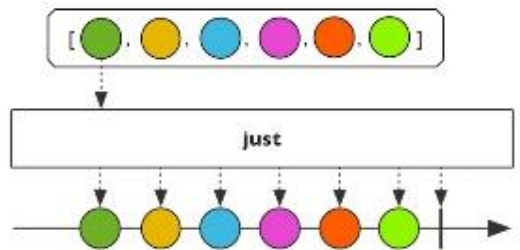
“Las arquitecturas de software de **ayer** simplemente no satisfacen las demandas de **hoy**”.



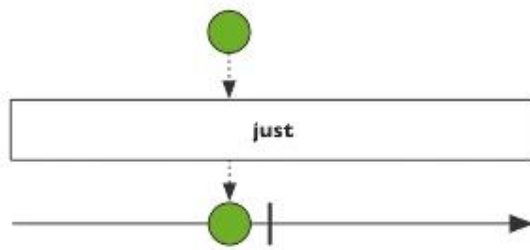




- Más flexibles, están ligeramente acoplados y son escalables.
- Tolerantes a fallos.
- Altamente receptivos
- Ofrecen a los usuarios retroalimentación interactiva y efectiva.
- Más fáciles de desarrollar y susceptibles de cambiar.
- Responsivos, resilientes, elásticos y manejados por **mensajes**.



```
Flux<String> seq =  
    Flux.just("foo", "bar", "foobar");
```



```
Mono<String> data = Mono.just("foo");
```

```
package pe.edu.galaxy.training.java.spring.boot.webfluxthymeleaf.repository;  
  
import org.springframework.data.mongodb.repository.ReactiveMongoRepository;  
  
public interface TallerRepository extends ReactiveMongoRepository<Taller, String>{  
    Flux<Taller> findByNombre(String nombre);  
    Mono<Taller> findById(String id);  
}
```

■ Mono vs Flux



GALAXY
TRAINING