

---

# programação reativa

por que usar?



bio.

## Jeferson Cruz

- Especialista Java/Mobile
- Officeless
- Fundador da empresa **BLACKFISH LABS**

# organização da talk

## → Por que?

Desafio da Engenharia de Software Atualmente

## → Para que?

Casos de uso onde a Programação Reativa pode fazer sentido

## → O que é Programação Reativa?

Conceitos/Código

## → Exemplo: Android (RxJava/RxAndroid)

## → Exemplo: Spring Boot 2 (Reactor)

—

**Mais uma 'buzzword' ou algo  
realmente útil?**

---

# Manifesto reativo 2014

<http://www.reactivemanifesto.org/pt-BR>

—

# Por quê?

aplicações orientadas a alto  
volumes de dados em tempo  
real

- Usuários querem seus dados AGORA!
- Confirmar pedidos AGORA!
- Jogos online, precisam responder AGORA!

**Agora, agora,  
agora!**



# —

## Para que?

casos de uso onde a  
programação reativa pode  
fazer sentido



→ **Eventos de UI**

Movimentos do mouse, clicks de botões, etc ..

→ **Chamadas para serviços externos (REST, RPC, etc.)**

Operações realizadas sobre HTTP são bloqueantes por natureza

→ **Consumo de mensagens**

Processamento de mensagens com alta concorrência

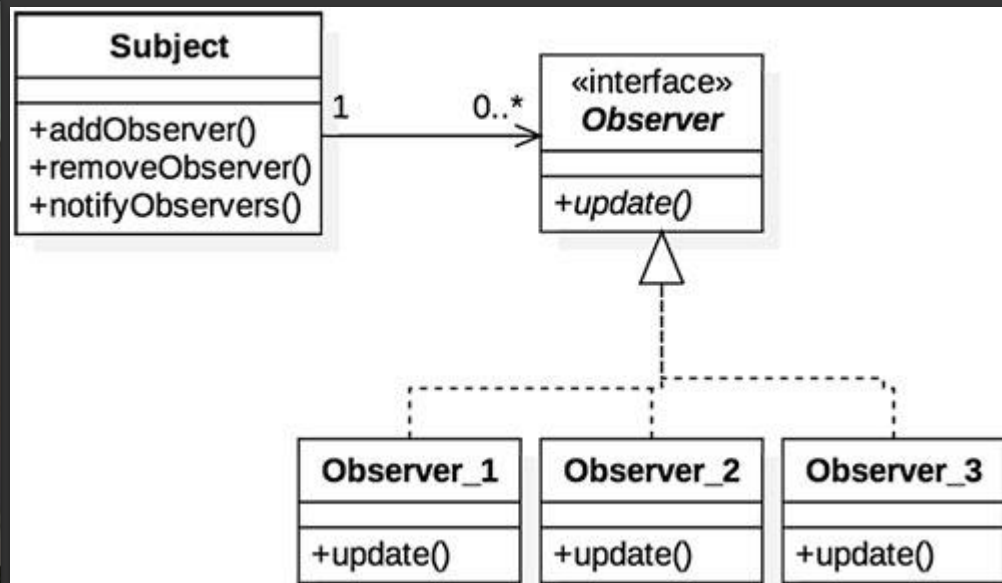
→ **Abstração sobre o processamento assíncrono**

Frameworks

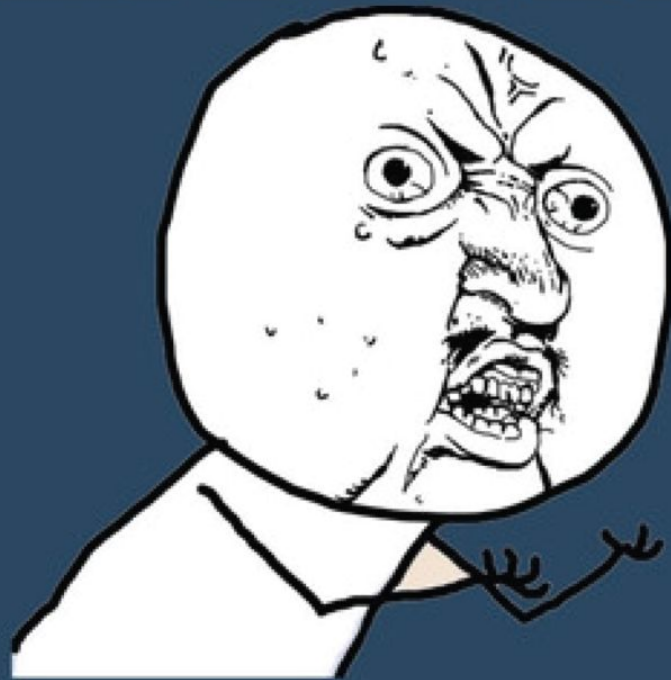
# O que é programação reativa?

versão estendida e mais poderosa do padrão Observer

# Padrão de Projeto Observer



**BLA BLA BLA**



**SHOW ME THE CODE**

## Fluxos de dados, potencialmente assíncronos


4 lines (3 sloc) | 107 Bytes

...

```
1 Observable.just(1, 2, 3, 4, 5, 6)
2   .filter(integer -> integer % 2 == 0)
3   .subscribe(System.out::println)
```

## Combinação com consumo/roteamento de eventos

4 lines (3 sloc) | 107 Bytes



```
1 Observable.just(1, 2, 3, 4, 5, 6)
2   .filter(integer -> integer % 2 == 0)
3   .subscribe(System.out::println)
```

# Operadores de streams

- criação
- transformação
- filtro
- combinação
- tratamento de erros
- utilitários
- métodos para operações condicionais
- operações matemáticas
- conexão de streams e vários outros.

## Propagação de estado

4 lines (3 sloc) | 107 Bytes

```
1 Observable.just(1, 2, 3, 4, 5, 6)
2   .filter(integer -> integer % 2 == 0)
3   .subscribe(System.out::println)
```



# Android (RxJava/RxAndroid)



## Notas

26 out

- Imprimir apostilas  
=====
- Kotlin  
Swift

26 out

- Fazer sketch

26 out

- Publicar slides meetup  
=====
- DevMT



## Spring Boot 2 (Reactor)



—

Códigos, Exemplos, etc ...



[github.com/JefersonOC/meetup](https://github.com/JefersonOC/meetup)

# Perguntas?



# Muito Obrigado!



[blackfishlabs.github.io](http://blackfishlabs.github.io)

[github.com/JefersonOC](https://github.com/JefersonOC)