

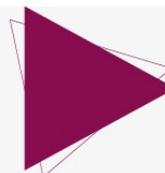
Angular e Observables com RxJS



THE
DEVELOPER'S
CONFERENCE

DOMINGOS TERUEL

- ▶ FullStack Developer @ Softplan/Construtech Ventures
- ▶ +15 anos como desenvolvedor
- ▶ Evangelista open source
- ▶ Contribuidor dos projetos openSUSE, PHP, Wordpress, e +
 - ▶ Core PHPSC  phpsc.com.br
- ▶ Contato: mingomax@dteruel.net.br
- ▶ Online: @mingomax



Tópicos

- Angular
- RxJS
- Live coding



**Angular é uma plataforma que facilita a
construção de aplicativos web**

Conceitos

- Dependency injection
- Single-pass change detection
- Declarative Templates

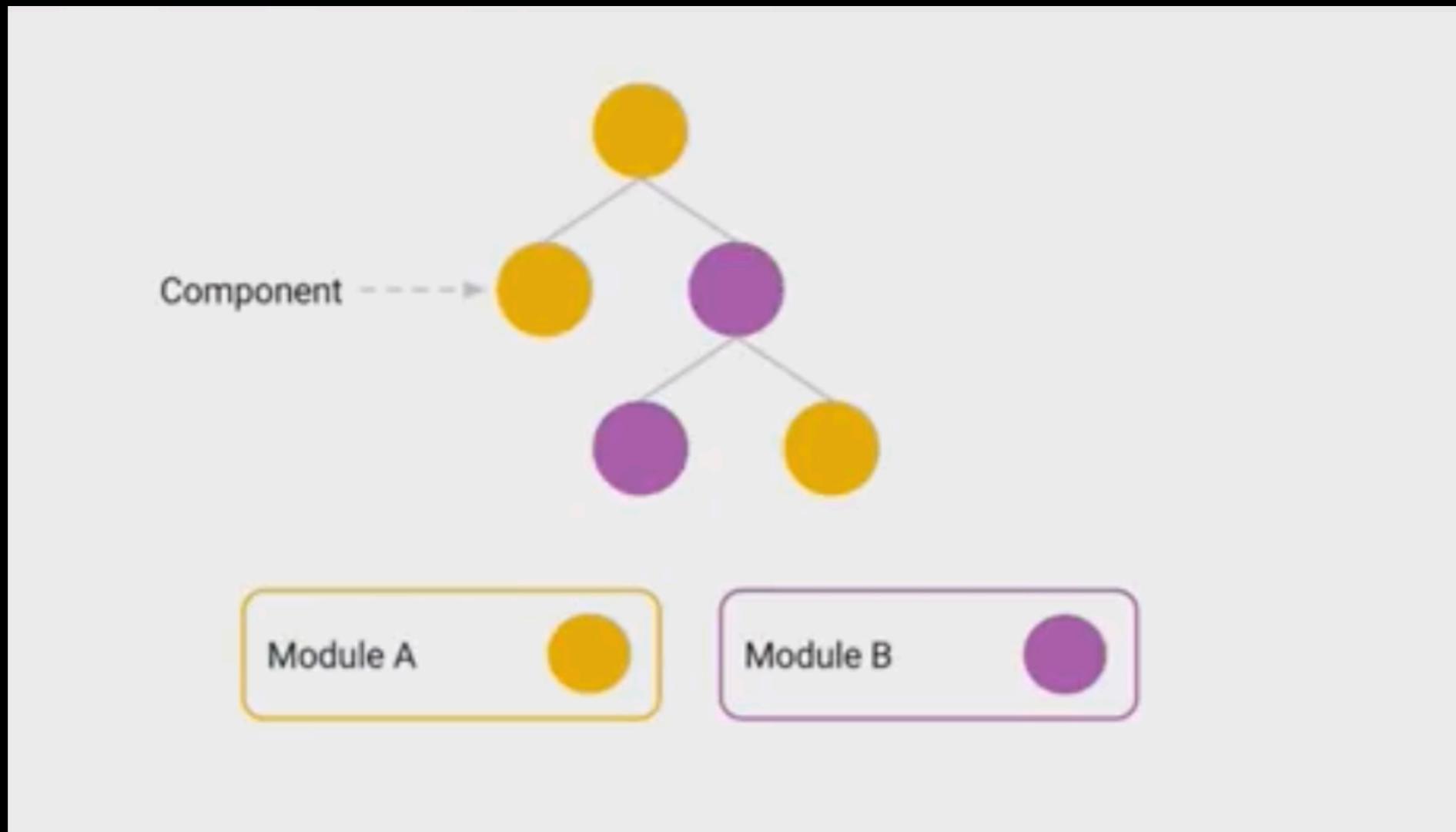


Dependency Injection

- Prove acesso de forma fácil a camada de serviços onde você precisar



Change Detection



Declarative templates

- Descreve como sua aplicação deve parecer para o seu cliente



Interpolação de texto

```
<!-- Use property value from person object from component instance. -->  
<h2>Profile for {{person.name}}</h2>  
  
<!-- Use method on person object from component instance. -->  
<h2>Profile for {{person.getBiography()}}</h2>  
  
<!-- Use property on person object from component instance. -->  
<img alt="Photo of {{person.name}}" ...>
```

Bind de propiedades

```
<!-- Bind [property] to the template express property. -->  
<img [src]="person.photoUrl" ...>  
  
<!-- Yes, this is valid HTML syntax. -->  
<input [value]="person.emailAddress">  
  
<!-- This also works on your own components! -->  
<birthday-card [date]="person.birthday">
```

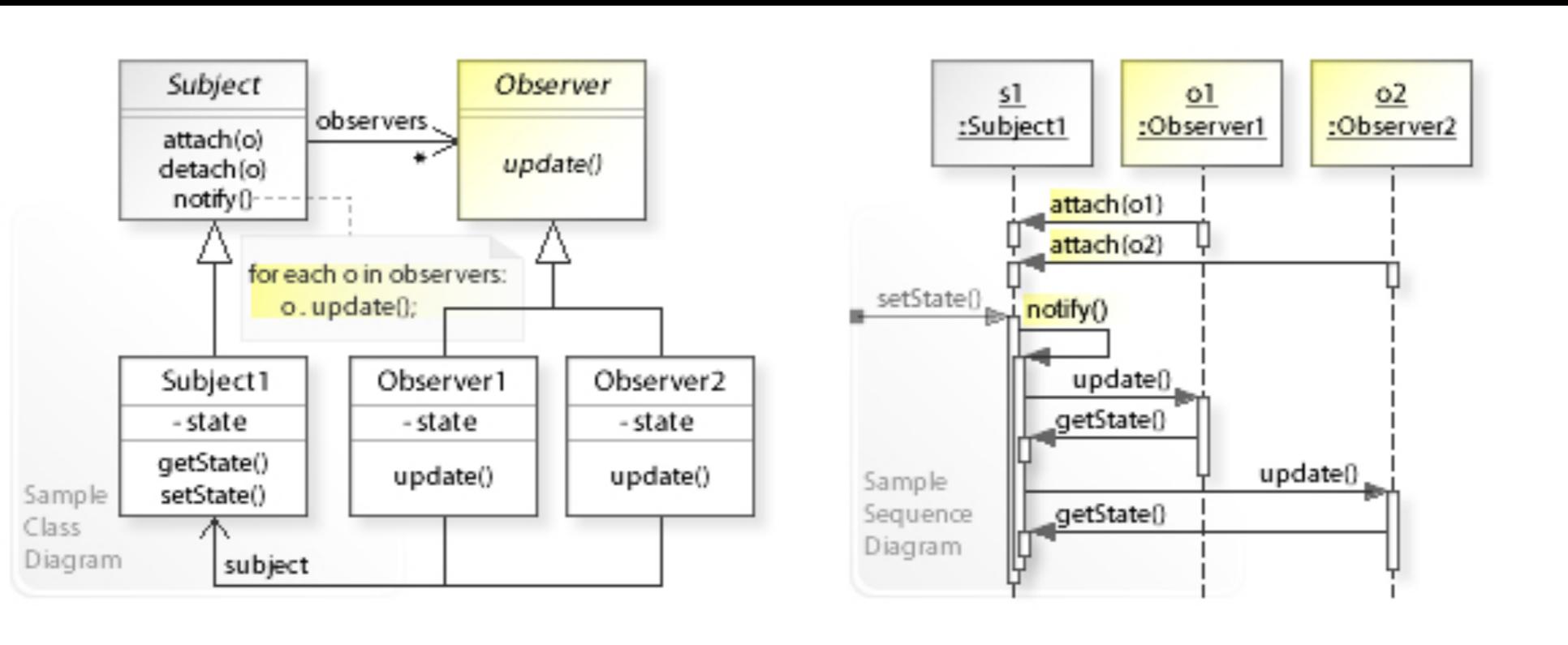


Reactive Extensions (Rx)



Princípios da programação funcional para o Javascript

Observer pattern



Conceito

- **Observable**
 - emite itens ou envia notificações para seus observadores(**observers**) chamando os métodos dos observadores.

Conceito

- **Subject**
 - É como conectamos um **observer**(observador) a um **Observable**. O observador implementa algum subconjunto dos seguintes métodos

Conceito

- **onNext:** Um **Observable** chama esse método sempre que o **Observable** emite um item. Este método toma como parâmetro o item emitido pelo **Observable**.
- **onError:** Um **Observable** chama esse método para indicar que não conseguiu gerar os dados esperados ou encontrou algum outro erro. Não fará mais chamadas para **onNext** ou **onCompleted**. O método **onError** usa como parâmetro uma indicação do que causou o erro.
- **onCompleted:** Um **Observable** chama esse método depois de ter chamado **onNext** pela última vez, caso não tenha encontrado nenhum erro.

O que faço com
Observable? E onde entra
Angular nessa história?

Live coding

PERGUNTAS?



<http://louc.me/46Xutn>