

Comunicação entre componentes



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>



FREE WEEKEND

VUE 3 MASTERCLASS

Master Vue.js and the ecosystem by building a real-world application from scratch.

RESERVE YOUR SPOT

We are unlocking our best video course, priced at \$329, on April 23-24.

Watch all available Vue 3 Masterclass lessons for free.

Learn how to build production-grade Vue.js applications using the ecosystem of library and plugins. Including best practices, application architecture, and tips & tricks for the enterprise.

Vue 3 Masterclass

Se liga aí!!!

Nos dias 23 e 24 de abril vai está acontecendo gratuitamente um master class fornecido pelos mantenedores do Vue. Para se cadastrar basta acessar o link:

<https://vueschool.io/free-weekend/?friend=vuejs>

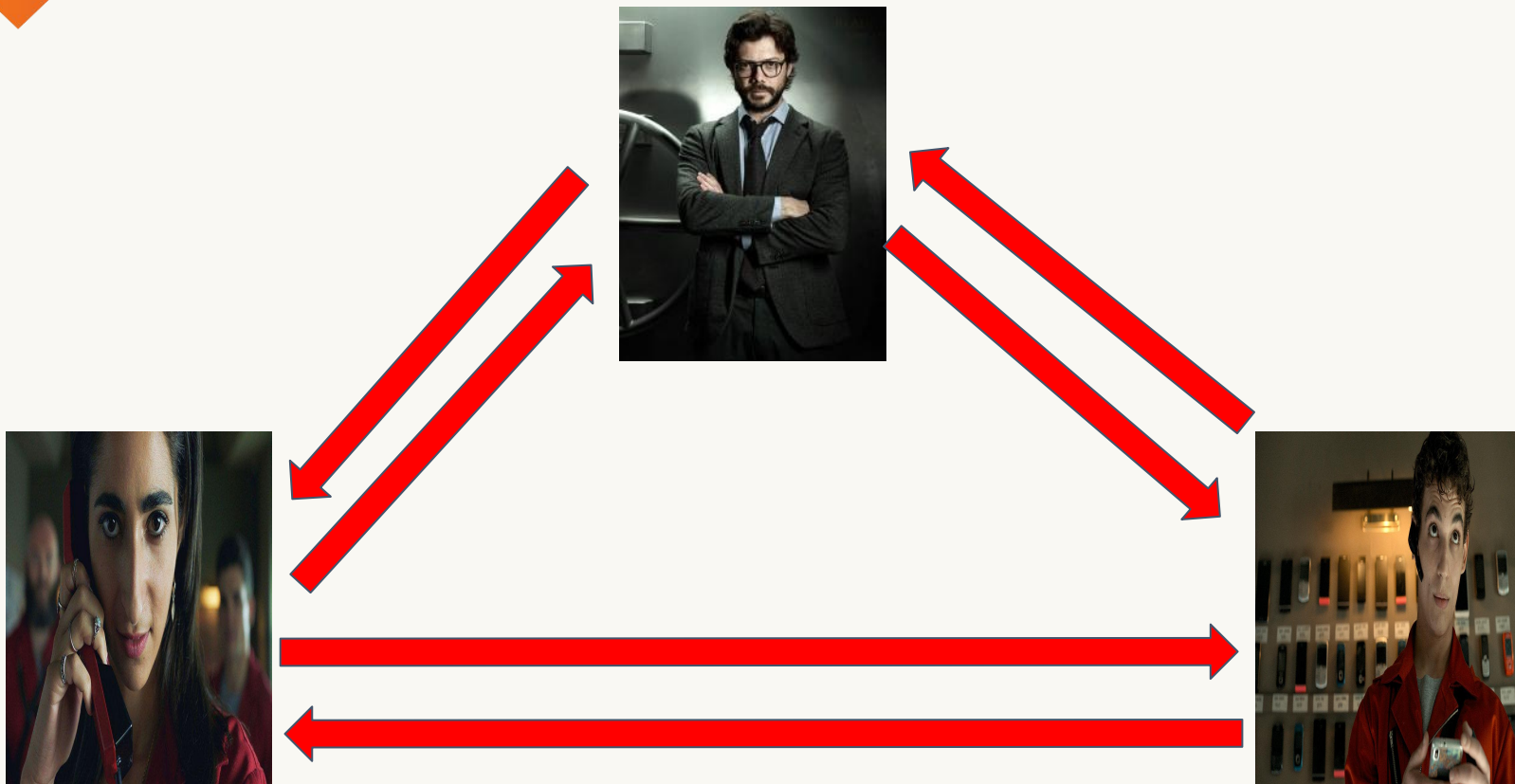
AGENDA

- Propriedades (props)
- Casos de uso
- Como declarar um propriedade
- Tipos de dados
- Validação de propriedade

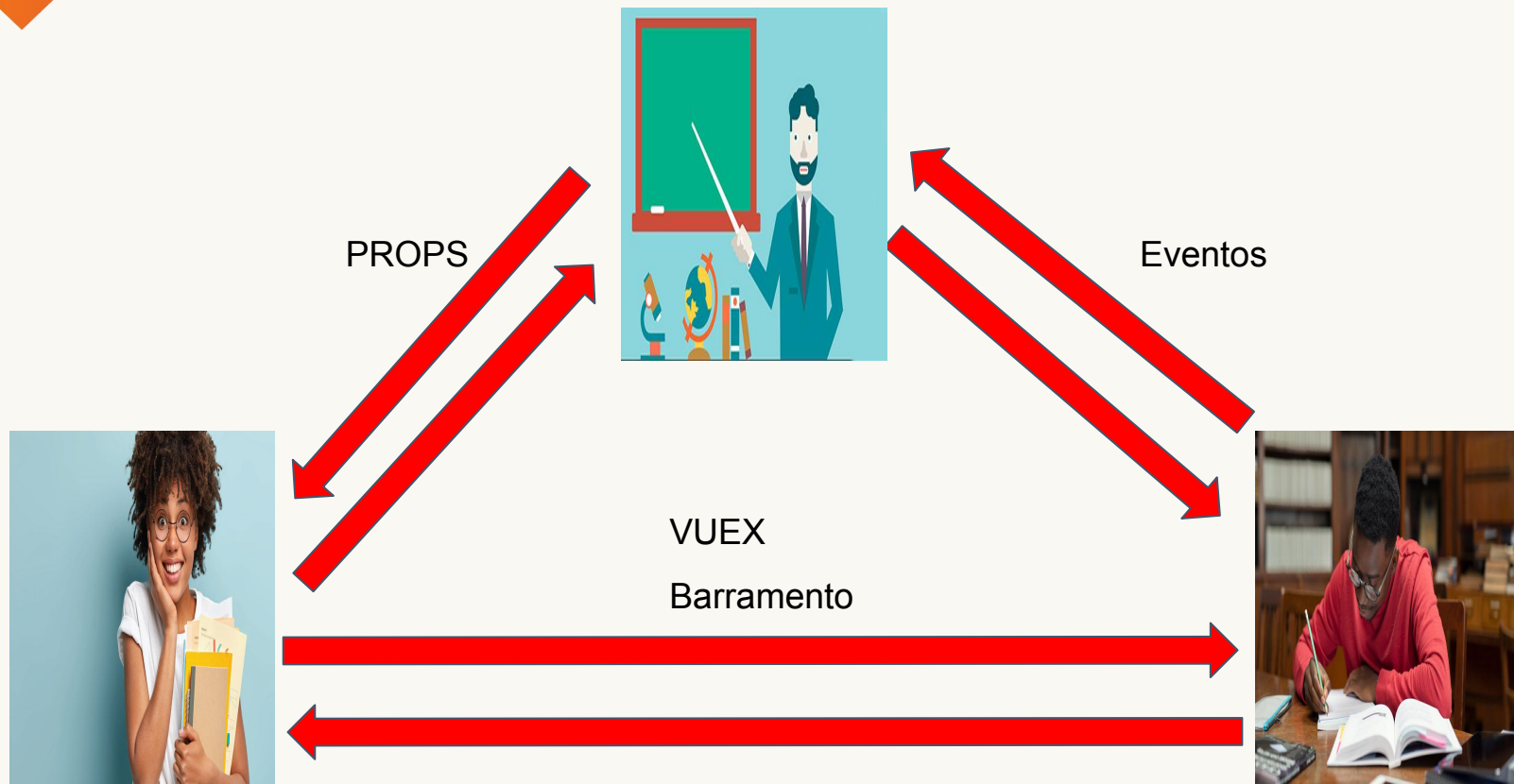
Introdução



Introdução



Introdução



Propriedades (props) de um componente são valores externos que são passados para o componente na hora da sua instanciação.

Existem duas formas declarar uma prop:

- declarando dentro do atributo props do componente como uma lista.
- declarando dentro do atributo props do componente como um objeto.

- `props: ['nome']`

- ```
props: {
 nome: {
 type: String,
 default: 'André',
 require: true
 }
},
```

Perceba que ao declarar uma propriedade como um objeto, é possível adicionar alguns atributos para a mesma.

```
props: {
 nome: {
 type: String,
 default: 'André',
 require: true
 }
},
```

- É possível dizer que a propriedade receba um valor de um tipo específico com o atributo Type.
- É possível atribuir um valor padrão para a propriedade com o atributo default
- É possível dizer se a propriedade é obrigatória ou não para a instânciação do componente.



# Props

Para passar um valor como atributo para o componente é simples. Basta na chamada do componente, renderização no html, fazer um bind no atributo de mesmo nome da propriedade declarada.

Ficou confuso ? Deixa eu te explicar!

```
props: {
 nome: {
 type:String,
 default: 'André',
 require: true
 }
},

</div>
<hr>
<Aluno :nome="nome"/>
<Display />
</div>
</template>
```

Vamos usar a propriedade nome declara nos exemplos anteriores.

Basta que na chamada do componente no HTML seja passado como atributo, fazendo o bind, o valor que é desejado passar para o componente.

Se ligou ?



Massa, até agora vimos a comunicação do componente pai para o componente filho. Mas como o filho pode se comunicar com o pai? É isso que veremos agora!

A comunicação do componente filho para o componente pai se dá através da emissão de eventos. Onde esses eventos serão ouvidos pelo componente pai e a partir daí é gerado algum resultado.

Então como emitir esses eventos?

Lembra que vimos em aula que frameworks trazer trechos de códigos embarcados nele?

Pronto, iremos utilizar o método `$emit`, da instância atual do Vue para propagar que determinada ação deve acontecer ou que uma determinada variável deve mudar.

Para emitir um evento utiliza-se a seguinte instrução:

```
this.$emit('data', this.times)
```



O primeiro parâmetro do método você deve passar o nome que será atribuído ao evento.

O segundo parâmetro do método você deve passar o valor que é desejado propagar para o componente pai.

Para ouvir os eventos propagados pelos componentes filhos também é fácil, basta utilizar o @ seguido do nome do evento.

Deixa eu te mostrar:

```
<button type="button" @click="data">Adicionar</button>
```

# Vamos à Prática!!!





A propriedade watch, como diz o nome fica “assistindo” ou monitorando o estado de uma variável e executa uma ação ao detectar uma mudança no seu estado.









**Vamos por a mão na massa!**





# Dúvidas?





# DEVinHouse

Parcerias para desenvolver a sua carreira

**OBRIGADO!**



<LAB365>