



# A aula interativa de **Desenvolvimento Reativo** começará em breve!

Atenção:

- 1) Acesse a aula com NOME COMPLETO, para que sua frequência seja computada.
- 2) Mantenha o microfone DESLIGADO, abrindo-o apenas em momentos de interatividade.
- 3) Mantenha seu vídeo sempre ATIVADO.

# Desenvolvimento Reativo

---

Primeira Aula Interativa

Prof. Raphael Gomide

# Nesta aula



- ❑ Orientação sobre a Atividade Prévia.
- ❑ Indicações:
  - Angular
  - React
  - Vue
- ❑ Introdução aos React Hooks.

## Orientação sobre a Atividade Prévia

- Criar projetos com **Angular**, **React** e **Vue**.
- Pode ser o mesmo projeto ou projetos distintos.
- Mais que um “Hello, world” e menos que um “App completo”.
- Sugestão: [CodeSandBox](#) ou Github/Gitlab.
- Forma de entrega: PDF com prints de tela + descrição breve de cada projeto + links.

## Indicações – **Angular**



Loiane Groner

<https://twitter.com/loiane>



Michelli Brito

<https://twitter.com/MichelliBrito4>



William Grasel

<https://twitter.com/willgmbr>

## Indicações – React



Dan Abramov

[https://twitter.com/dan\\_abramov](https://twitter.com/dan_abramov)

Willian Justen

[https://twitter.com/Willian\\_justen](https://twitter.com/Willian_justen)



Kent C. Dodds

<https://twitter.com/kentcdodds>

## Indicações – Vue



Evan You

<https://twitter.com/youyuxi>

Fabio Vedovelli

<https://twitter.com/vedovelli74>



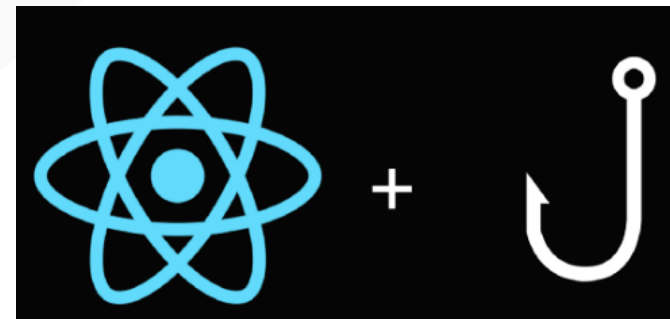
Maximilian Schwarzmüller

<https://twitter.com/maxedapps>

# Introdução aos React Hooks



- Criados pelo Facebook no fim de 2018.
- Fornece uma escrita de código ainda mais declarativa.
- Utiliza **closures** e **array destructuring**.
- Permite a utilização de **estado** em **Functional Components**.
- Principais hooks: **useState** e **useEffect**.



[Fonte](#)



# O hook “useState”

- Visa substituir **this.state** e **this.setState** de **Class Components**.
- Escrita mais declarativa.
- Utiliza array destructuring.
- Sintaxe padrão: `const [variable, setVariable] = useState(0);`
- Na instrução acima, **variable** representa a **variável** de **estado**.
- Na instrução acima, **setVariable** representa a **função atualizadora**.
- As **funções atualizadoras** só atuam na **variável** para a qual “apontam”.
- **Não** há mais o *merge* de **this.setState**.

# O hook “useEffect”



- Visa substituir **componentDidMount**, **componentDidUpdate** e **componentWillUnmount** de **Class Components**.
- Escrita mais declarativa.
- Com **useEffect**, não há mais o conceito de montagem do componente (**mounting**) e atualização do componente (**updating**).
- **useEffect** tem um **modelo mental diferente** dos métodos de **ciclo de vida** – a ideia de **useEffect** é sincronizar todo o DOM conforme os valores de **props** e **state**. Mais informações [aqui](#).

# O hook “useEffect”



- **useEffect** permite utilizar um parâmetro extra, conhecido como array de dependências (dependency array ou, simplesmente, **deps**).
  - Quando **não há** o parâmetro, **useEffect** é **invocado após qualquer atualização** – **semelhante a componentDidMount**.
  - Quando o **parâmetro** é [], **useEffect** é **invocado apenas uma vez** – **semelhante a componentDidMount**.
  - Quando o parâmetro está preenchido com [**state1, state2, etc**], **useEffect** é **invocado após a atualização de estado de qualquer uma das variáveis**.
- Quando **há retorno na função** – **useEffect** utiliza o retorno para **eliminar recursos** – **semelhante ao componentWillUnmount**.
- Acompanhe uma demonstração do professor.

# Prática com React Hooks



- Acompanhe o professor.

# Conclusão



- ✓ Tanto o Angular quanto o React e o Vue são excelentes ferramentas para Front End e aplicam os conceitos do Desenvolvimento Reativo.
- ✓ Foi feita uma sugestão de conteúdo de qualidade para Angular, React e Vue.
- ✓ Os React Hooks tornaram a escrita de componentes React muito mais declarativa.