



# **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

---

Segunda Aula Interativa

Prof. Raphael Gomide

# Nesta aula

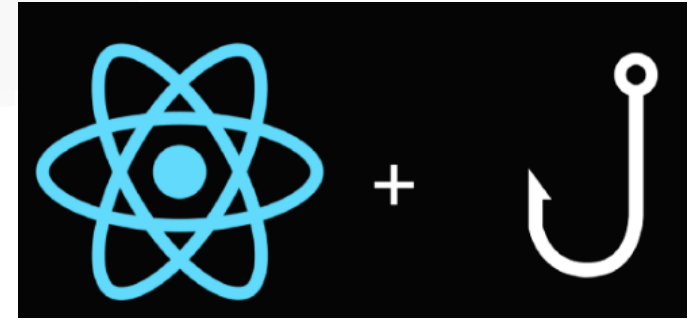


- ☐ Introdução aos React Hooks.
- ☐ Trabalho Prático.

# 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**.
- Não pretende (ainda) substituir totalmente os **Class 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** e **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**, **state** e/ou **efeitos colaterais**.
- Mais informações podem ser vistas [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 `componentDidUpdate`**.
  - Quando o **parâmetro** é um array vazio (`[]`), **useEffect** é **invocado apenas uma vez** – **semelhante a `componentDidMount`**.
  - Quando o parâmetro está preenchido com `[state1, state2, etc]`, **useEffect** é **invocado após a atualização** do valor de **qualquer uma das variáveis**.
- Quando **há retorno na função** – **useEffect** o utiliza para **eliminar recursos** – **semelhante ao `componentWillUnmount`**.
- Acompanhe uma demonstração do professor.

# Trabalho Prático



- Análise do Trabalho Prático.
- Dicas:
  - Apenas um aluno do grupo entrega o trabalho no sistema.
  - No campo de comentários, coloque o **nome dos integrantes do grupo em ordem alfabética**. Isso facilita muito a minha correção.
  - Faça o mesmo com o primeiro slide do trabalho, ou seja, coloque o nome dos integrantes em ordem alfabética.



# Conclusão



- ✓ Os React Hooks tornaram a escrita de componentes React muito mais declarativa.