

---

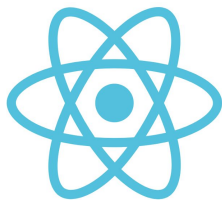
# Desenvolvimento WEB

-

Full Stack Completo: Java + React

---

# Um pouco de teoria



**React JS**



Semantic UI



JavaScript

## O que é React?

- O **React** (também denominado React.js ou ReactJS) **é uma biblioteca front-end JavaScript de código aberto com foco em criar interfaces de usuário em páginas web;**
  - Apenas uma biblioteca JavaScript;
  - Uma biblioteca JavaScript declarativa, eficiente e flexível para criar interfaces visuais.
- (facebook)
- Criado em 2011 pelo time do Facebook, o React surgiu com o objetivo de otimizar a atualização e a sincronização de atividades simultâneas no feed de notícias da rede social, entre eles chat, status, listagem de contatos e outros.

## React / React Native

- O React é uma biblioteca criada para construir telas de forma declarativa. Ou seja, o **React cria aplicações web para serem executadas em navegadores**, sejam eles para desktop, mobile ou qualquer outro dispositivo.
- Criada em 2015 pelo Facebook, o módulo React Native, que em conjunto com o React, possibilita o desenvolvimento de aplicativos para Android e iOS utilizando componentes de interface de usuário nativos de ambas plataformas, sem precisar recorrer ao HTML. Desta forma, podemos dizer que o **React Native é uma biblioteca voltada para a criação de aplicativos mobile em Android e iOS**.
- Em resumo, React e React Native são ferramentas diferentes, mas relacionadas. O React é usado para criar aplicações web, enquanto o React Native é usado para criar aplicações móveis. No entanto, ambos compartilham muitos conceitos e sintaxe, então é possível usar o conhecimento de um para aprender o outro.

## React e o JavaScript

- O React é uma biblioteca que utiliza a linguagem de programação JavaScript.
- Por essa razão, se você quer aprender a manipular esta biblioteca e extrair dela todo o seu potencial, **é preciso antes ter uma sólida base de conhecimento em JavaScript**, bem como em suas outras duas linguagens de marcação: HTML e CSS. Isso porque o React utiliza uma extensão alternativa ao JavaScript para descrever seus componentes. Esta sintaxe é chamada de **JSX**.

## JSX

- É uma extensão da sintaxe do JavaScript que nos **permite escrever HTML dentro do JavaScript**;
- O JSX é muito próximo do HTML;
- O JSX fornece uma sintaxe concisa e familiar para definir uma estrutura de árvore com atributos que não requerem o aprendizado de uma linguagem de templates ou abandonar o JavaScript. O JSX é fácil de ler e escrever.

- Vejamos um exemplo:

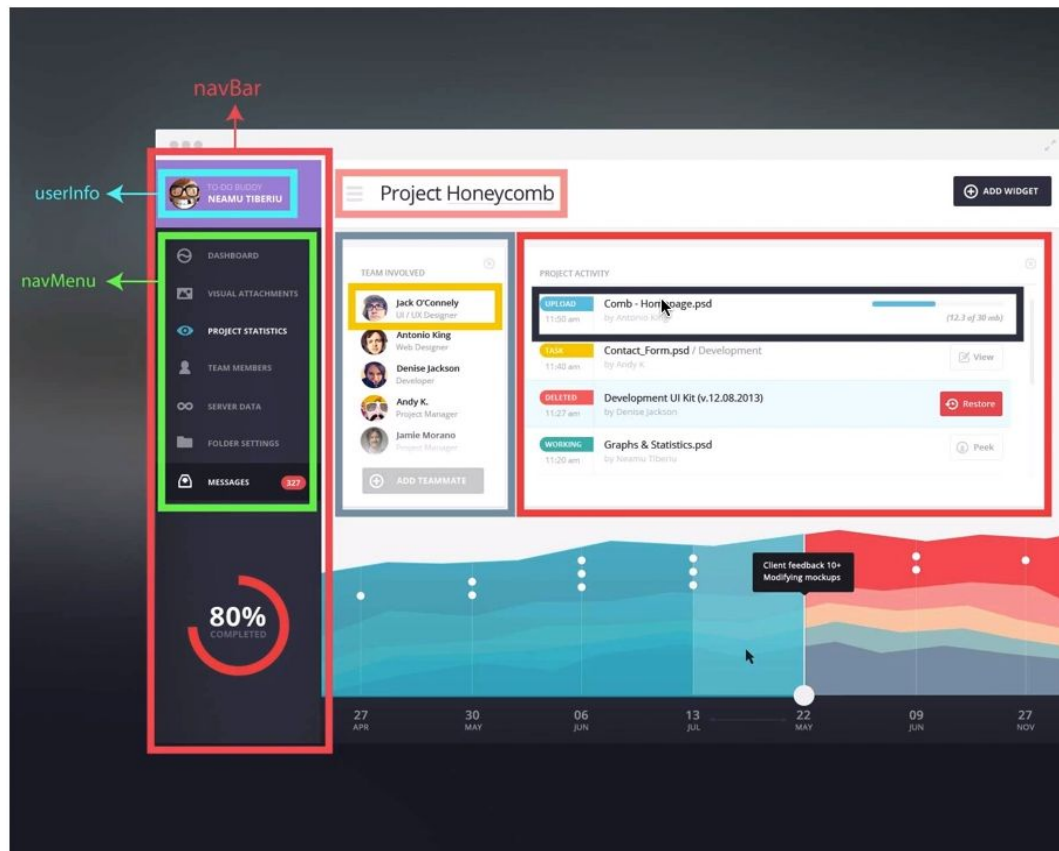
```
const element = <h1>Olá Mundo!</h1>
ReactDOM.render(
  element,
  document.getElementById('root')
);
```

- Neste exemplo, declaramos a variável `element` e a usamos dentro do JSX, renderizando-a com `ReactDOM.render()`.

# React JS



## Componentes



## props

- Abreviação de `properties`
- Usado para passagem de parâmetros entre um componente Pai e um componente Filho



## Componentes e props

```
import React from 'react'

export default function HelloWorld() {
  return (
    <h1>Alô Mundo!</h1>
  );
}
```

```
import React from 'react'

export default function HelloWithProps(props) {
  return (
    <h1>Alô {props.name} !</h1>
  );
}
```

```
import React from 'react'
import HelloWorld from './HelloWorld'
export default function App() {
  return (
    <div> <HelloWorld> </div>
    <div> <HelloWorld> </div>
  );
}
```

```
import React from 'react'
import HelloWithProps from './HelloWithProps'
export default function App() {
  return (
    <div> <HelloWithProps name="Roberto"> </div>
    <div> <HelloWithProps name="Alencar"> </div>
  );
}
```

## State (Estado)

### Componentes **stateless**

Componentes **sem** estado

São utilizados para a apresentação dos dados

Podem ser escritos como uma simples função

Normalmente são criados para serem utilizados em outras telas (cabeçalhos, rodapé de telas, menu e etc.)

```
...  
  
export default function HelloWithProps(props) {  
  return (  
    <h1>Alô {props.name} !</h1>  
  );  
}
```

## State (Estado)

*OBS.: O estado (state) está relacionado ao componente, não a aplicação.*

### Componentes **stateful**

#### Componentes **com** estado

Normalmente, são criados para a implementação das telas do sistema (formulários, telas que exibem dados retornados pela api e etc.)

```
...  
  
class ListProduto extends React.Component {  
  
  state = {  
    listaProduto: [],  
    titulo: "",  
    mostrarBarraPesquisa: false,  
    openModalFiltro: false,  
    listaNotificacoes: []  
  }  
  
  ...  
}
```

```
...  
  
export default function FormCliente () {  
  
  const { state } = useLocation();  
  
  const [idCliente, setIdCliente] = useState();  
  const [nome, setNome] = useState();  
  const [cpf, setCpf] = useState();  
  const [dataNascimento, setDataNascimento] = useState();  
  const [foneCelular, setFoneCelular] = useState();  
  
  ...  
}
```

## Casos de uso

- Empresas que possuem suas interfaces digitais construídas com React.JS são:

- Dropbox
- Tesla
- Airbnb
- Netflix
- Buscapé
- Hotmart
- iFood
- Reddit
- BBC
- Cloudfare, entre outros.



TESLA



NETFLIX



## Para estudar em casa !!!

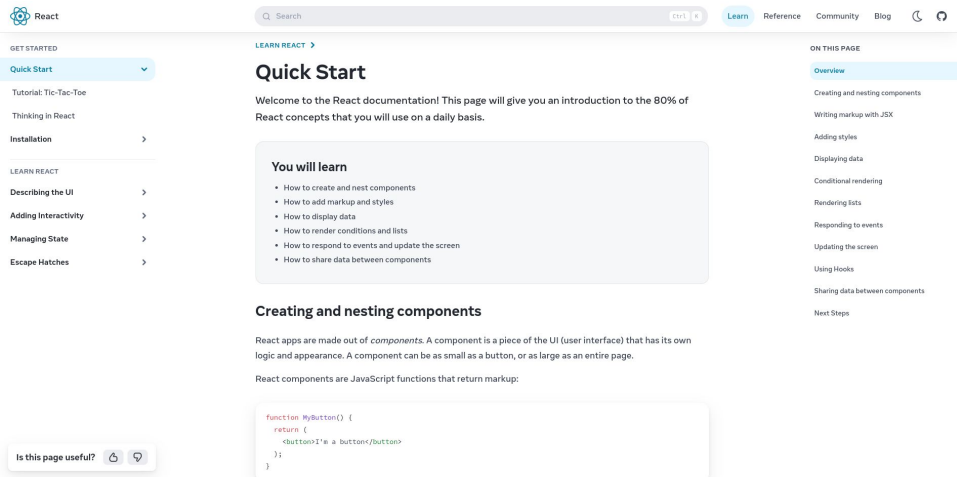
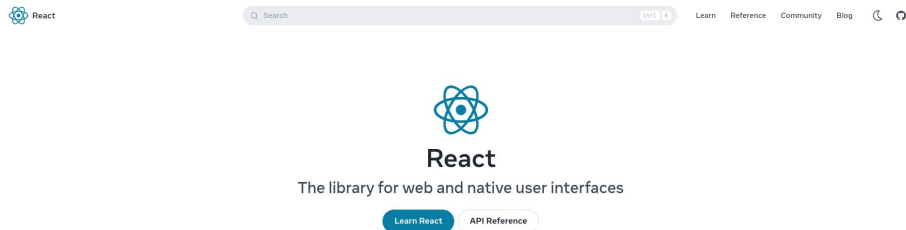
- Sugestão para revisão de JavaScript:
  - [https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Language\\_Overview](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Language_Overview)
- Sugestão para estudar React:
  - <https://legacy.reactjs.org/docs/hello-world.html>
  - <https://pt-br.reactjs.org/tutorial/tutorial.html>
  - <https://www.taniascascia.com/getting-started-with-react/>

# React JS



## Para estudar em casa !!!

- Página do React explicações sobre a biblioteca:
  - <https://react.dev/learn>



### Create user interfaces from components

React lets you build user interfaces out of individual pieces called components. Create your own React components like `Thumbnail`, `LikeButton`, and `Video`. Then combine them into entire screens, pages, and apps.

# Dúvidas



The background features two large, solid green abstract shapes. One is a semi-circle on the left side, and the other is a more complex, organic shape on the right side.

Obrigado !