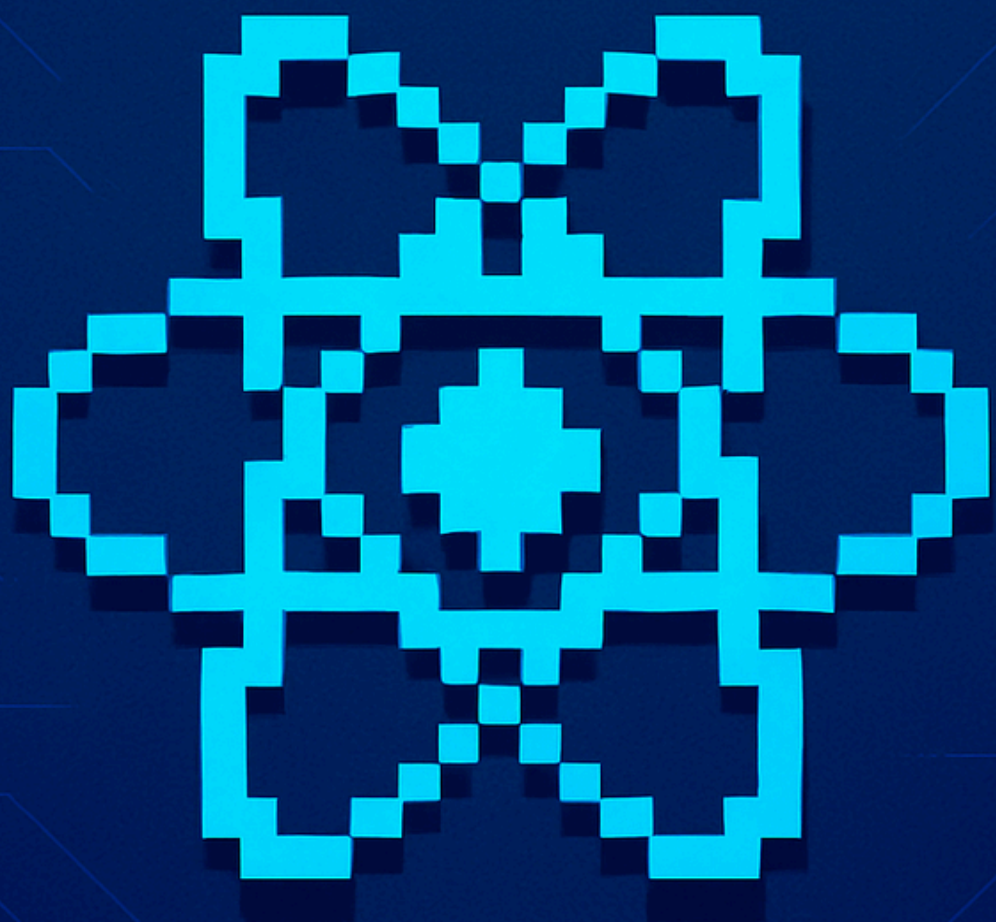


REACT

PRIMEIRA FASE



Um guia leve e prático
para **dominar** seus
primeiros códigos na
versão 19.2

Sobre o autor



Eu sou o Vinícius, desenvolvedor front-end e curioso por tudo que envolve tecnologia, interfaces e aprendizado constante.

Criei este ebook para quem está começando a explorar o **React**, uma das bibliotecas mais usadas do mundo.

Aqui você vai entender o básico, descobrir os conceitos que sustentam o React e conhecer um novo recurso introduzido na versão **19.2**: o componente `<Activity />`, que ajuda a controlar partes da interface de forma mais simples e eficiente.

Bora jogar a primeira fase?



Sumário

1. Introdução ao React	2
2. Conceitos fundamentais	3
3. O novo componente <Activity />	6
4. Conclusão	7



Introdução ao React

O **React** é uma biblioteca JavaScript criada pelo Facebook (hoje Meta) para construir **interfaces de usuário dinâmicas e reutilizáveis**.

Sua ideia principal é **dividir a interface em pequenos blocos chamados componentes**, cada um responsável por uma parte específica da tela.

Essa abordagem mudou o jeito de pensar o front-end: em vez de atualizar a página inteira, o React **reage apenas às partes que mudam**, tornando a experiência **mais rápida, previsível e eficiente**.

Com o tempo, o React evoluiu, ganhando novos recursos que facilitam o desenvolvimento, como o `<Activity />`, apresentado na versão 19.2, que você vai conhecer mais adiante neste ebook.

Conceitos Fundamentais

Para começar a entender o React, é importante conhecer três conceitos básicos: **componentes, props e estado**.

Eles são o coração de qualquer aplicação React, e entender como funcionam é o primeiro passo para criar interfaces dinâmicas.

1. Componentes

Componentes são como “peças” que formam a sua interface. Cada um tem uma função e pode ser reutilizado quantas vezes quiser.

Exemplo:

```
function Botao() {  
  return <button>Clique aqui</button>;  
}  
  
export default Botao;
```

Conceitos Fundamentais

2. Props

As props (propriedades) são informações que você envia para um componente. Elas permitem **personalizar o comportamento** ou o texto de cada um.

Exemplo:

```
function Saudacao({ nome }) {  
  return <h2>Olá, {nome}!</h2>;  
}
```

Usando o componente:

```
<Saudacao nome="Vinícius" />  
<Saudacao nome="Maria" />
```

A prop *nome* muda o conteúdo de forma dinâmica, sem precisar criar vários componentes diferentes.

Conceitos Fundamentais

3. Estado (State)

O **estado** é o que permite que um componente **reaja** a interações do usuário. Quando o estado muda, o React atualiza automaticamente a tela.

Exemplo:

```
import { useState } from "react";

function Contador() {
  const [contador, setContador] = useState(0);

  return (
    <div>
      <p>Você clicou {contador} vezes</p>
      <button onClick={() => setContador(contador + 1)}>
        Clique aqui
      </button>
    </div>
  );
}
```

Aqui usamos o **hook** `useState` para criar uma variável `contador` e uma função que a atualiza (`setContador`).

💡 **Dica do Dev:** pense em cada **componente** como um **personagem de um jogo**. Ele tem suas **características** (*props*), suas **emoções e energia** (*estado*), e reage de forma diferente conforme o que acontece na partida (*interações do usuário*).

O novo componente `<Activity />`

A versão **19.2 do React** trouxe um novo componente chamado `<Activity />`, que permite **mostrar e esconder partes da interface** sem perder o estado interno dos componentes.

Em outras palavras: ele **pausa** um pedaço da sua aplicação, **limpa os efeitos ativos** (como listeners ou animações), mas **mantém tudo guardado** para quando você quiser exibir de novo, como se colocasse um jogo em *pause* e depois voltasse exatamente de onde parou.

Exemplo:

```
import { Activity } from 'react';

<Activity mode={isShowingSidebar ? "visible" : "hidden"}>
  <Sidebar />
</Activity>
```

No exemplo acima, o componente `<Sidebar />` fica visível quando `isShowingSidebar` é verdadeiro, e “escondido” quando é falso.

Enquanto está oculto, o React **remove os efeitos ativos**, mas **mantém o estado** do componente para restaurá-lo depois.



Conclusão

Chegamos ao fim da **primeira fase** dessa jornada com o React!

Você conheceu os fundamentos essenciais : **componentes, props e estado**, e descobriu o novo componente `<Activity />`, que torna mais fácil controlar o que aparece ou é pausado na sua interface, sem perder o estado do que já foi feito.

Com isso, o React continua mostrando por que é uma das ferramentas mais queridas do front-end: **simples, flexível e sempre evoluindo**.

Agora que você já passou pelo tutorial, é hora de colocar a mão no código e continuar explorando novas fases, testando, criando e experimentando suas próprias ideias.

Eu sou o Vinícius, e este foi o *React — Primeira Fase*.
Até o próximo nível!