

# Educação Profissional Paulista

Técnico em  
**Desenvolvimento  
de Sistemas**

# Linguagens de programação *front-end*

## **Frameworks e bibliotecas JavaScript**

Aula 1: Introdução ao React

Código da aula: [SIS]ANO2C3B2S8A1

Linguagens de  
programação  
*front-end*

## Mapa da Unidade 2 Componente 3

semana

11

Testes e qualidade  
de código.

semana

12

Performance e otimização.

semana

9

Estilização e layout.

semana

8

**Você está aqui!**

Frameworks e  
bibliotecas  
JavaScript.

semana

14

PWA e aplicações  
modernas.

Linguagens de  
programação  
*front-end*

Mapa da  
Unidade 2  
Componente 3

**Você está aqui!**

*Frameworks* e bibliotecas  
JavaScript

**Aula 1: Introdução ao React**

Código da aula: [SIS]C3B2S8A1

8



## Objetivos da aula

- Compreender e utilizar os conceitos básicos do React.



## Recursos didáticos

- Recurso audiovisual para exibição de vídeos e imagens.
- Caderno e lápis.
- Laboratório de informática.



## Duração da aula

50 minutos.



## Habilidades técnicas

- Implementar e gerenciar estados e eventos no React.



## Habilidades socioemocionais

- Desenvolver a curiosidade ao explorar novas tecnologias de desenvolvimento *front-end*.



## Relembre

# Componentes do React

Componentes são blocos reutilizáveis que permitem a construção da interface de usuário em aplicações React. Eles encapsulam a lógica e o layout de uma parte da interface.

Existem dois tipos principais de componentes:

- ▶ **Componentes funcionais:** são funções que recebem *props* e retornam elementos JSX. Eles são mais simples e geralmente não possuem estado.
- ▶ **Componentes de classe:** são baseados em classes ES6 e podem manter e gerenciar o estado internamente. Hoje em dia, os componentes funcionais são preferidos, especialmente com o uso de *hooks*.

Construindo  
o **conceito**



Produzido pela SEDUC-SP. <https://youtu.be/iPQn54iQgh8?feature=shared>



## Construindo o **conceito**

# Props e state

## Props (Propriedades)

São argumentos que você passa para os componentes. Eles são imutáveis e permitem que você configure e personalize o comportamento e a aparência dos componentes.

**Exemplo:** um botão que recebe uma *prop label* para definir seu texto. Veja ao lado:

```
import React from 'react';
// Componente Button que recebe uma prop 'label'
function Button({ label }) {
  return (
    <button>
      {label}
    </button>
  );
}
// Componente principal da aplicação
function App() {
  return (
    <div>
      {/* Usando o componente Button e passando
      diferentes valores para a prop 'label' */}
      <Button label="Clique Aqui" />
      <Button label="Enviar" />
      <Button label="Cancelar" />
    </div>
  );
}

export default App;
```



## Construindo o conceito

# Props e state

## State

É um objeto que mantém informações dinâmicas dentro de um componente. Ao contrário das *props*, o *state* é mutável e gerenciado internamente pelo componente. É ideal para dados que mudam ao longo do tempo, como o conteúdo de um formulário ou o estado de uma animação.

**Exemplo:** ao executar o código ao lado, haverá uma mensagem na tela, "Você clicou 0 vezes". Cada vez que você clica no botão, o número aumenta.

```
Código em JavaScript:
import React, { useState } from 'react';
function Counter() {
  // Declaração de uma variável de state chamada
  'count' e uma função para atualizá-la 'setCount'
  const [count, setCount] = useState(0);

  // Função que incrementa o valor de 'count'
  const increment = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <p>Você clicou {count} vezes</p>
      { /* Botão que, ao ser clicado, chama a função
      'increment' */ }
      <button onClick={increment}>
        Clique para incrementar
      </button>
    </div>
  );
}

export default Counter;
```

## Construindo o **conceito**

# JSX (JavaScript XML)

JSX (JavaScript XML) é uma sintaxe que permite escrever elementos React utilizando uma linguagem parecida com HTML, mas diretamente em JavaScript. Ele facilita a construção de interfaces de usuário de maneira intuitiva.

### Exemplo:

```
const element = <h1>Hello,  
world!</h1>;
```



© Getty Images



Ser  
sempre +

# Situação

## Situação problema

A empresa TechStore está lançando uma nova plataforma on-line para vender seus produtos eletrônicos. Você, como parte da equipe de desenvolvimento, está responsável por colaborar diretamente com os designers e o time de marketing para garantir que o catálogo de produtos da plataforma atenda às expectativas de ambos os times e ofereça uma experiência de usuário consistente.

Durante as primeiras reuniões, você percebe que há diferentes visões sobre o design e a funcionalidade do catálogo. O time de marketing está focado em estratégias de conversão e os designers estão preocupados com a estética e a usabilidade da interface. Essa diferença de visão está gerando tensão, e o cronograma do projeto está apertado.

Como responsável pela integração das equipes, você percebe que uma má comunicação pode comprometer a qualidade e os prazos do projeto. Seu papel vai além de apenas executar a parte técnica. É necessário mediar os interesses e promover um ambiente colaborativo, evitando conflitos e garantindo que todos se sintam ouvidos e respeitados.

Situação fictícia produzida pela SEDUC-SP.



Ser  
sempre +

## Ação

Como você pode agir para ajudar a resolver essa situação desafiadora, garantindo que o projeto avance dentro do prazo e que todas as partes envolvidas tenham suas preocupações consideradas?

Considere os seguintes pontos:

- ▶ **Mediação de conflitos:** como você pode facilitar a comunicação entre as equipes, garantindo que as preocupações tanto do marketing quanto dos designers sejam compreendidas e resolvidas?
- ▶ **Gestão de expectativas:** que abordagens você pode usar para alinhar as expectativas de ambas as equipes, evitando atritos e focando o objetivo comum da empresa?
- ▶ **Colaboração eficaz:** que ações específicas você pode tomar para incentivar uma colaboração eficaz entre os times, promovendo uma cultura de respeito, empatia e foco no sucesso coletivo?

# Então ficamos assim...

- 1** Compreendemos que componentes são a base da construção de interfaces no React, encapsulando lógica e visual;
- 2** Aprendemos que *props* e *state* permitem controlar a personalização e o comportamento dinâmico dos componentes;
- 3** Aprendemos que o JSX simplifica a criação de elementos React usando uma sintaxe familiar aos desenvolvedores web.

O que nós  
**aprendemos  
hoje?**

© Getty Images



# Saiba mais

Aprofunde seu conhecimento nesse assunto!  
Explore o vídeo:

VINICIUS DACAL. ReactJS – O que são Components, Props e State. Disponível em:  
<https://youtu.be/TID49IAEPmc>. Acesso em: 8 nov. 2024.



# Referências da aula

Identidade visual: imagens © Getty Images.

ABROMOV, D.; NABORS, R. **Apresentando react.dev**. React, 16 mar. 2023. Disponível em: <https://pt-br.react.dev/blog/2023/03/16/introducing-react-dev> . Acesso em: 8 nov. 2024.

NEVES, V. **React**: o que é, como funciona e um guia dessa popular ferramenta JS. Alura, 17 jan. 2023. Disponível em: <https://www.alura.com.br/artigos/react-js> . Acesso em: 8 nov. 2024.

SILVA, J. C. B. da. **React Hooks**: o que é e como funcionam? Alura, 22 mar. 2023. Disponível em: <https://www.alura.com.br/artigos/react-hooks> . Acesso em: 8 nov. 2024.

SOUSA, R. **React.JS**: criando rotas com React Router Dom. DevMedia, 2021. Disponível em <https://www.devmedia.com.br/react-js-criando-rotas-com-react-router-dom/42901> . Acesso em: 8 nov. 2024.

VINICIUS DACAL. **ReactJS – O que são Components, Props e State**. Disponível em: <https://youtu.be/TID49IAEPmc> . Acesso em: 8 nov. 2024.

# Orientações ao professor

# Slide 6



**Orientações:** tempo sugerido para esta seção: 8 minutos.

Classes ES6, introduzidas no ECMAScript 2015 (ES6), são uma forma mais simples e sintática de escrever funções construtoras e herança em JavaScript. Elas trazem um estilo de programação orientado a objetos que é mais familiar para desenvolvedores que vêm de outras linguagens, como Java, C# ou Python.

Semana 5 – Aulas 1 e 2

# Slide 7



**Orientações:** tempo previsto para o ODA AV: 6 minutos.

Tempo total previsto para esta seção: 17 minutos.

# Slide 8



Explicação:

Componente Button:

O componente Button é uma função que recebe um objeto de *props*. Nesse caso, ele espera uma *prop* chamada *label*.

{label}: dentro do JSX do componente Button, usamos {label} para exibir o texto recebido como *prop*.

Componente App:

No componente principal App, usamos o componente Button três vezes, cada vez passando um valor diferente para a *prop label*.

O primeiro botão recebe o texto “Clique Aqui”, o segundo “Enviar” e o terceiro “Cancelar”.



# Slide 9



## Explicação:

### 1. useState:

- `useState(0)` é um *hook* do React que permite adicionar *state* a um componente funcional.
- Ele retorna um array com dois valores: o valor do estado atual (`count`) e uma função para atualizar esse estado (`setCount`).

### 2. State:

- `const [count, setCount] = useState(0);`; inicialmente, o *count* é definido como 0.
- A função `setCount` é usada para atualizar o valor de *count*.

### 3. Função Increment:

- `const increment = () => { setCount(count + 1); }`; esta função incrementa o valor de *count* em 1 sempre que é chamada.

### 4. Retorno do JSX:

- O `p` exibe o valor atual de *count*.
- O botão, ao ser clicado, chama a função *increment*, que atualiza o valor de *count*.

# Slide 10



## Características do JSX

- **Sintaxe Familiar:** o JSX se parece com HTML, o que torna a transição ao React mais fácil para desenvolvedores web.
- **Interatividade:** o JSX permite incorporar expressões JavaScript diretamente na marcação, tornando-o altamente dinâmico.
- **Transpilação:** o JSX não é entendido diretamente pelos navegadores, então ele precisa ser convertido (transpilado) em chamadas de função React, como `React.createElement`. Isso geralmente é feito por ferramentas como Babel.

# Slide 11



**Orientações:** a seção **Ser sempre +** tem como objetivo desenvolver e aprimorar as competências socioemocionais dos estudantes, focando especificamente as situações desafiadoras que podem surgir no ambiente profissional.



**Tempo:** 20 minutos.



**Gestão de sala de aula:**

- Mantenha um ambiente de diálogo aberto e respeitoso.
- Assegure a participação equitativa, promovendo uma discussão inclusiva.
- Reconheça a complexidade do tema e a diversidade de perspectivas que os estudantes podem trazer.
- Forneça um feedback construtivo e um direcionamento à medida que os estudantes exploram possíveis soluções para o cenário proposto.
- Ajude os estudantes a refinarem suas ideias e a considerarem todas as implicações de suas sugestões.



**Expectativa de Resposta:**

Espera-se que você reconheça a importância da comunicação clara e respeitosa entre as partes. Uma boa resposta inclui propor a realização de reuniões conjuntas com mediação, em que você facilita a troca de ideias e identifica pontos em comum. Também pode sugerir o uso de protótipos ou *mockups* para alinhar expectativas visuais e funcionais de forma concreta. A resposta ideal aborda o equilíbrio entre prazos e qualidade, buscando soluções em conjunto para atender às expectativas de todos.

# Slide 13



**Orientações:** professor, a seção **O que nós aprendemos hoje?** tem o objetivo de reforçar e esclarecer os conceitos principais discutidos na aula. Essa revisão pode ser uma ferramenta de avaliação informal do aprendizado dos estudantes, identificando áreas que podem precisar de atenção adicional em aulas futuras.



**Tempo:** 3 minutos.



## **Gestão de sala de aula:**

- Mantenha um tom positivo e construtivo, reforçando o aprendizado em vez de focar em correções.
- Seja direto e objetivo nas explicações para manter a atividade dentro do tempo estipulado.
- Engaje os estudantes rapidamente, pedindo confirmações ou reações breves às definições apresentadas.



## **Condução da dinâmica:**

- Explique que esta parte da seção, "Então ficamos assim...", é um momento de reflexão e esclarecimento dos conceitos abordados na aula.
- Informe que será feita uma rápida revisão para assegurar que os entendimentos dos estudantes estejam alinhados com as definições corretas dos conceitos.
- Apresente o slide com a definição sintética de cada conceito principal discutido na aula, ampliando em forma de frases completas.
- Finalize resumindo os pontos principais e reiterando a importância de cada conceito e de como ele se encaixa no contexto maior da aula.
- Reforce a ideia de que essa revisão ajuda a solidificar o entendimento dos estudantes e a prepará-los para aplicar esses conceitos em situações práticas.



## **Expectativas de respostas:**

Os estudantes devem sair da aula com um entendimento claro e preciso dos conceitos principais. A atividade serve como uma verificação rápida do entendimento dos estudantes e uma oportunidade para corrigir quaisquer mal-entendidos.



# Slide 14



**Tempo:** 2 minutos.

# Educação Profissional Paulista

Técnico em  
**Desenvolvimento  
de Sistemas**