

# Aula 03 - Props e Comunicação entre Componentes

Prof. Me. José Olinda

# O que são props?

- Props (propriedades) são os dados que você passa de um componente pai para um componente filho em React.
- Elas são imutáveis dentro do componente que as recebe – o filho não deve modificar diretamente as props.
- Props podem ser qualquer valor JavaScript: strings, números, arrays, objetos ou funções (callbacks).

# Como passar props (JSX)

- Em JSX, você passa props como atributos de um componente, parecido com HTML:

```
<MeuComponente nome="José" idade={30} ativo={true} />
```

- Valores JS (variáveis, expressões) devem vir entre chaves {} ; strings podem ser escritas diretamente.

# Recebendo props no componente

- O componente recebe um objeto `props`. Você pode acessar `props.nome` ou usar destructuring:

```
// sem destructuring
function Saudacao(props) {
  return <h1>Olá, {props.nome}!</h1>;
}
```

```
// com destructuring (mais comum e legível)
function Saudacao({ nome }) {
  return <h1>Olá, {nome}!</h1>;
}
```

# children – conteúdo aninhado

- Componentes React podem receber `children`, conteúdo que fica entre as tags:

```
function Card({ children }) {  
  return <div className="card">{children}</div>;  
}  
  
// uso  
<Card>  
  <h2>Título</h2>  
  <p>Descrição</p>  
</Card>
```

# Valores padrão e props opcionais

- Você pode atribuir valores padrão via parâmetros da função:

```
function Botao({ texto = 'Clique', cor = 'blue', onClick }) {  
  return <button style={{ backgroundColor: cor }} onClick={onClick}>{texto}</button>;  
}
```

# Spread props

- Para passar muitas props de um objeto, você pode usar o spread:

```
const video = { titulo: 'React', autor: 'Ana' };
<Video {...video} />
```

# Passando funções: comunicação filho → pai

- Para o filho enviar dados ao pai, passe uma função do pai como prop. O filho chama essa função quando necessário.

```
function Pai() {  
  const receber = (msg) => console.log(msg);  
  return <Filho enviar={receber} />;  
}  
  
function Filho({ enviar }) {  
  return <button onClick={() => enviar('oi do filho')}>Enviar</button>;  
}
```

# Arquivo: Video.jsx

```
// Exemplo 1: Video com estado local de likes
import { useState } from 'react';

function Video({ titulo, autor, children }) {
  const [likes, setLikes] = useState(0);

  return (
    <div className="video">
      <h2>{titulo}</h2>
      <p>Publicado por: {autor}</p>
      {children && <div className="descricao">{children}</div>}
      <button onClick={() => setLikes(l => l + 1)}>Curtir ({likes})</button>
    </div>
  );
}
```

# Arquivo: VideoLifted.jsx

```
// VideoLifted – pai controla likes (lifting state)
function VideoLifted({ id, titulo, autor, likes, onLike, children }) {
  return (
    <div className="video">
      <h2>{titulo}</h2>
      <p>Publicado por: {autor}</p>
      {children}
      <button onClick={() => onLike(id)}>Curtir ({likes})</button>
    </div>
  );
}

export { VideoLifted };
```

# Arquivo: App.jsx

```
import Video from './Video';
import { VideoLifted } from './Video';
import './App.css';
import { useState } from 'react';

function App() {
  const [videos, setVideos] = useState([
    { id: 1, titulo: 'Aprendendo React', autor: 'João Silva', likes: 0 },
    { id: 2, titulo: 'Introdução ao JavaScript', autor: 'Maria Souza', likes: 0 },
    { id: 3, titulo: 'CSS para Iniciantes', autor: 'Carlos Pereira', likes: 0 },
  ]);

  const handleLike = (id) => {
    setVideos(prev => prev.map(v => v.id === id ? { ...v, likes: v.likes + 1 } : v));
  };

  return (
    <div>
      <h1>Vídeos:</h1>
      <table border="1">
        <thead>
          <tr>
            <th>ID</th>
            <th>Título</th>
            <th>Autor</th>
            <th>Likes</th>
          </tr>
        </thead>
        <tbody>
          {videos.map((video) => (
            <tr key={video.id}>
              <td>{video.id}</td>
              <td>{video.titulo}</td>
              <td>{video.autor}</td>
              <td>{video.likes}</td>
            </tr>
          ))}
        </tbody>
      </table>
      <button onClick={handleLike}>Clique para curtir o vídeo 1</button>
    </div>
  );
}

export default App;
```

- Arquivo: App.jsx (continuação)

```
return (
  <div className="App">
    <h1>Minha Galeria de Vídeos</h1>

    <h2>Exemplo: estado local em cada Video</h2>
    <Video titulo="Aprendendo React" autor="João Silva">Descrição do vídeo</Video>

    <h2>Exemplo: lifting (pai controla likes)</h2>
    {videos.map(v => (
      <VideoLifted
        key={v.id} id={v.id}
        titulo={v.titulo} autor={v.autor}
        likes={v.likes} onLike={handleLike}
      />
    )))
  </div>
```

# Arquivo: main.jsx

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

# Exercício Prático - Vídeo Like App

- Refatore o código do Video Like App para utilizar props e comunicação entre componentes.