

Especialização em Desenvolvimento Full Stack

React

Prof. Esp. Marco Antônio Gusmão Carvalho

Quem Sou Eu?

Prof. Esp. Marco Antônio Gusmão Carvalho



Graduado e Licenciado em **Análise de Sistemas e Tecnologias da Informação** (FATEC), Especialista em **Tecnologias para Aplicações Web** (UNOPAR), professor de tecnologia a mais 10 anos, Coordenador dos Cursos da área de Informação e Comunicação da ETEC Palmital, atuando também, como desenvolvedor autônomo de aplicação corporativas utilizando, principalmente, DotNet e ReactJS.

Agenda do dia

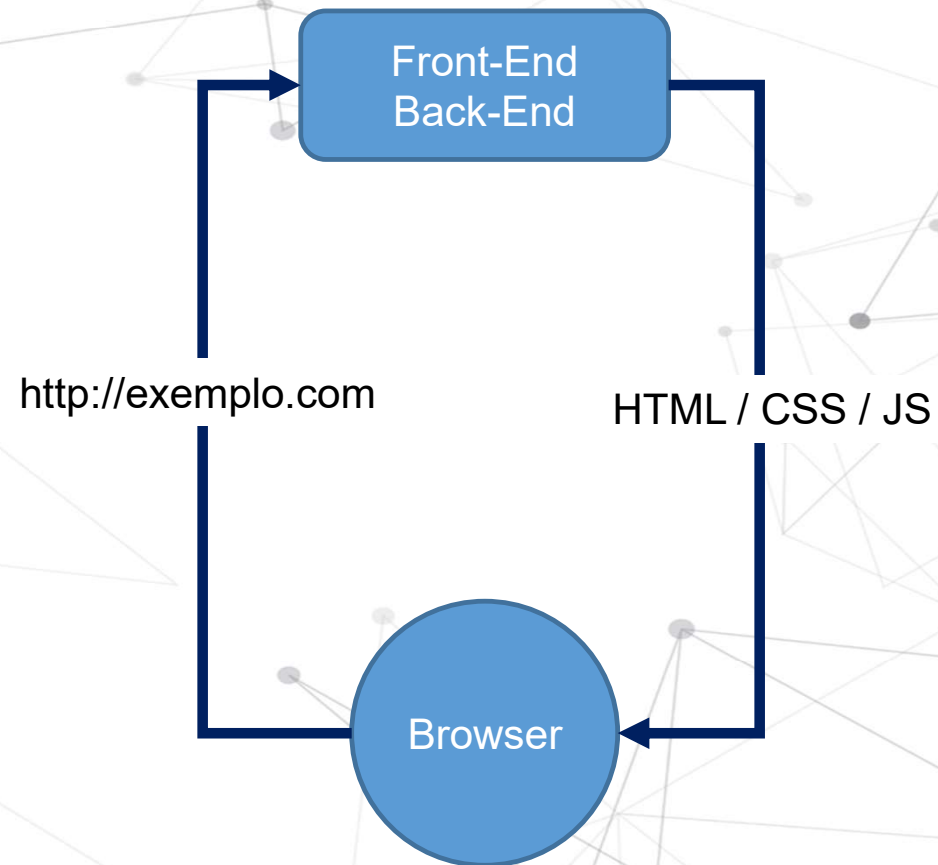
Manhã

- O que é o ReactJS
- Analisando as diferenças entre aplicações web SSR e SPAs
- Ambiente de Desenvolvimento
- Mãos no Código

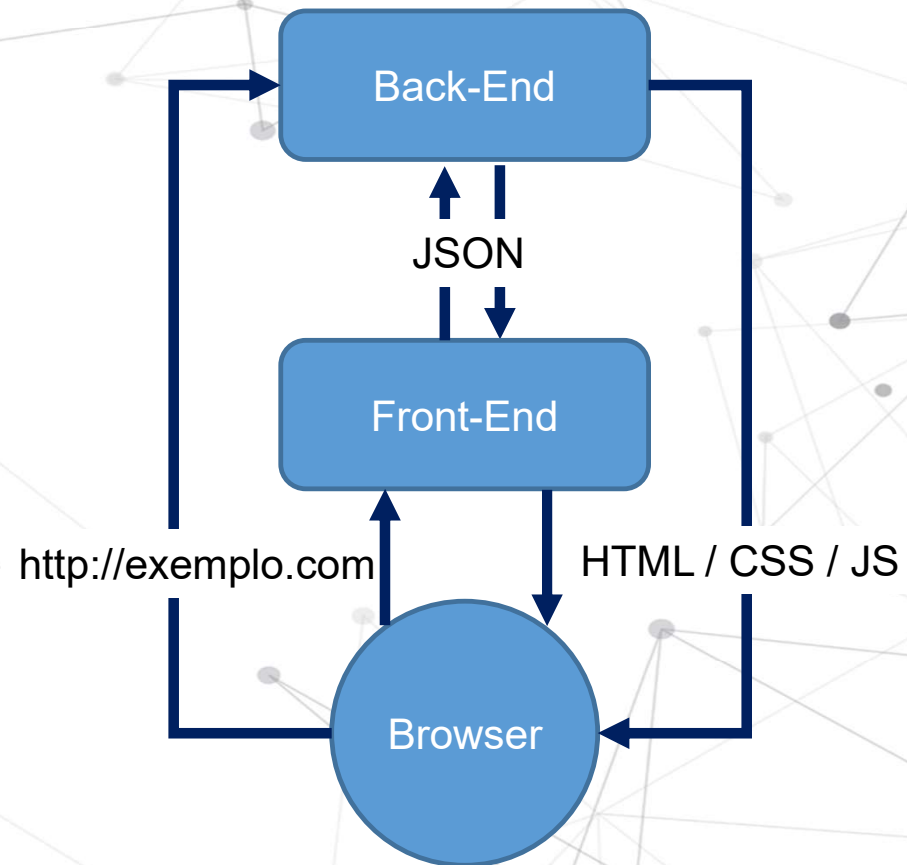
Tarde

- Variações Sintáticas
- Gerenciando Estados
- Propriedades
- React Hooks

**Server Side
Rendering**



Single Page Application



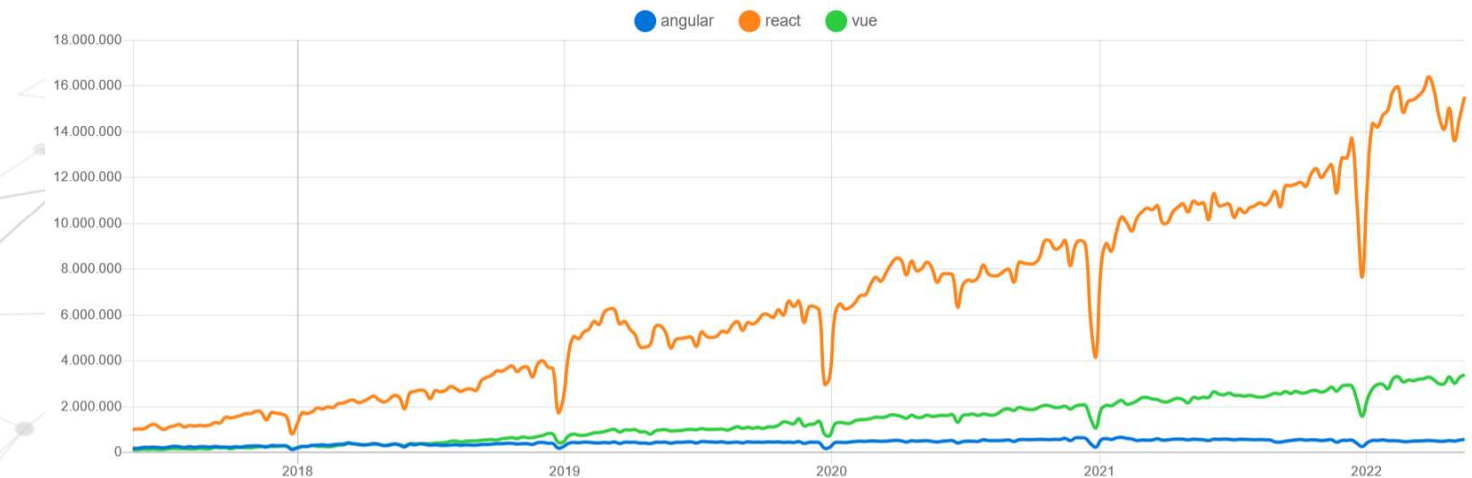


Especialização em Desenvolvimento Full Stack

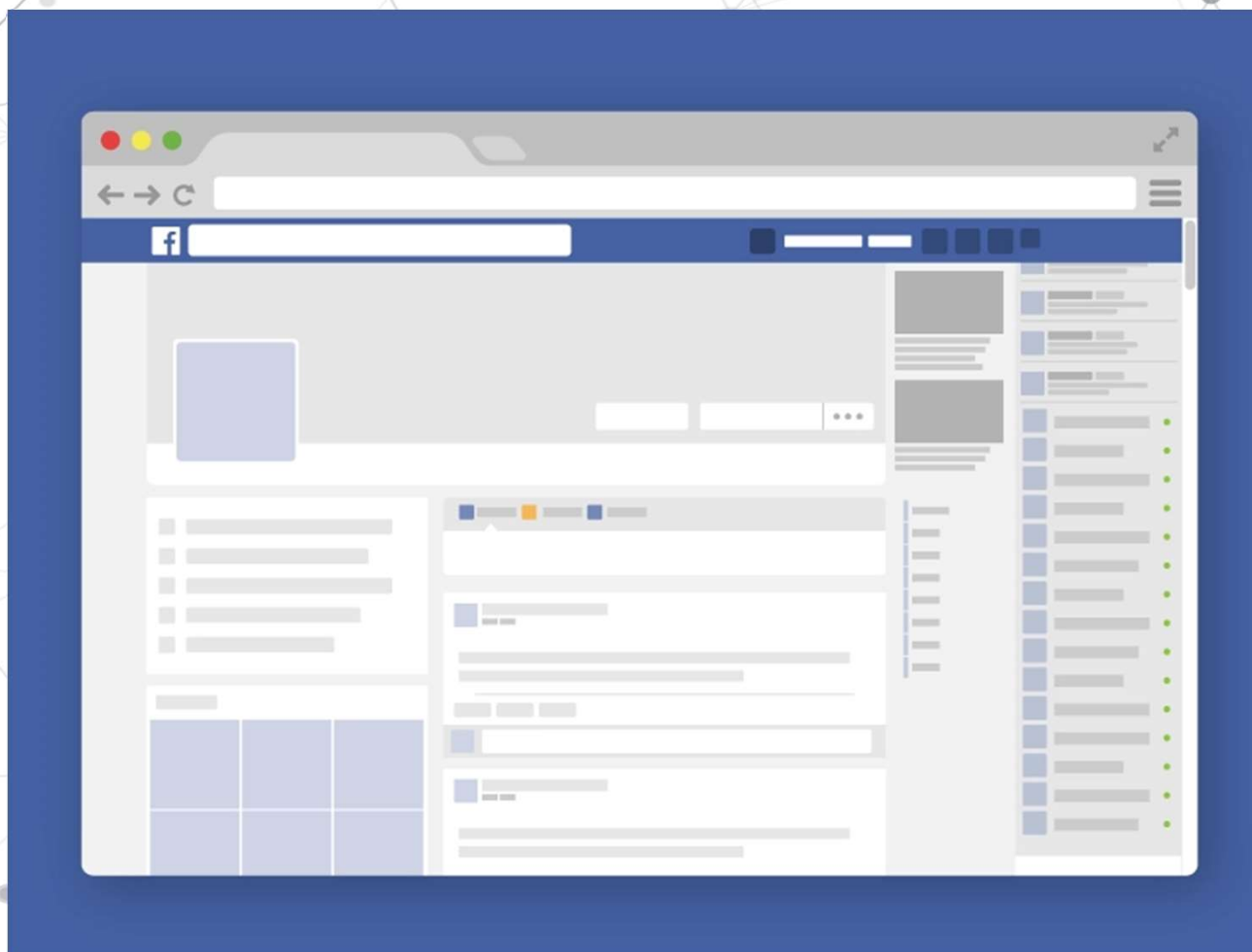
- Criado em 2011 para o Facebook
- Biblioteca Baseada em Componentes para UIs
- Disponibilizado em 2013 para a comunidade
- Largamente utilizado no desenvolvimento de SPAs

O que é ReactJS

Downloads in past 5 Years ▾



O que é ReactJS

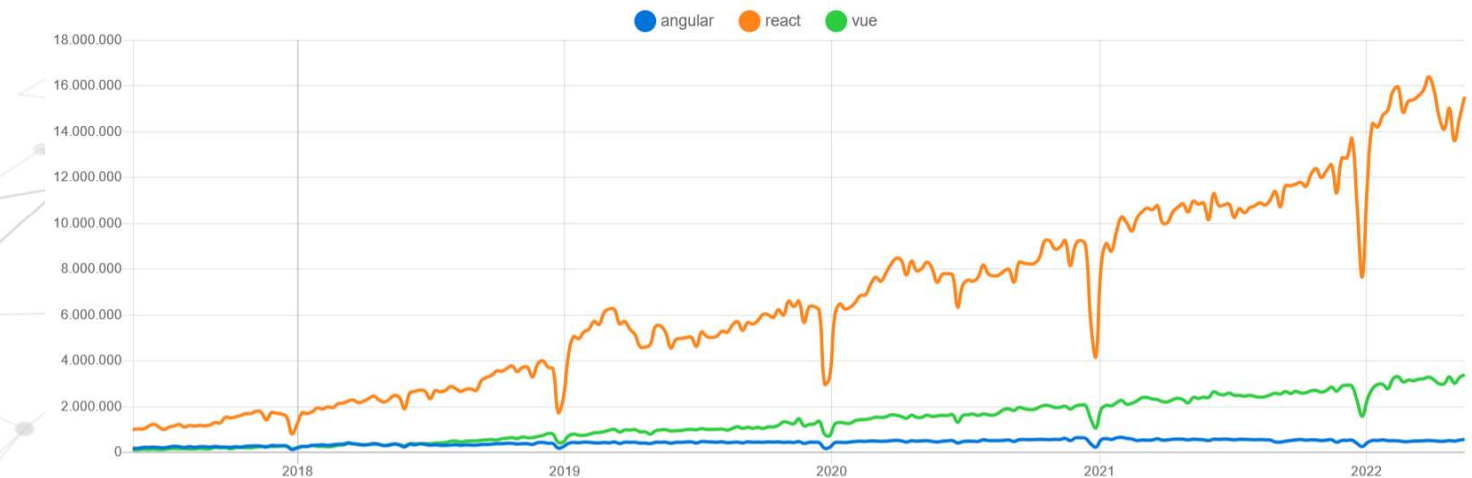


Especialização em Desenvolvimento Full Stack

- Criado em 2011 para o Facebook
- Biblioteca Baseada em Componentes para UIs
- Disponibilizado em 2013 para a comunidade
- Largamente utilizado no desenvolvimento de SPAs

O que é ReactJS

Downloads in past 5 Years ▾



Ferramentas

HTML



CSS



JS



Mãos no Código

Iniciando o desenvolvimento de uma
aplicação em ReactJS

Estilizando Componentes

Variações Sintáticas

 TS

TypeScript

```
1 import React from 'react'
2
3 class Exemplo extends React.Component {
4   render() {
5     return (
6       <h1>Exemplo</h1>
7     )
8   }
9 }
10
11 export default Exemplo
```

```
1 import React from 'react'
2
3 function Exemplo() {
4   return (
5     <h1>Exemplo</h1>
6   )
7 }
8
9 export default Exemplo
```

```
1 import React from 'react'
2
3 const Exemplo = () => {
4   return (
5     <h1>Exemplo</h1>
6   )
7 }
8
9 export default Exemplo
```


Gerenciando Estados

useState

```
const [state, setState] = useState(initialState);
```

Retorna um valor e uma função para atualizar o valor.

Durante a renderização inicial, o estado retornado é o mesmo que o valor passado como argumento inicial (`initialState`).

A função `setState` é usada para atualizar o estado. Ela aceita um novo valor de estado e coloca na fila de re-renderização do componente.

```
setState(newState);
```

Durante as próximas re-renderizações, o primeiro valor retornado por `useState` sempre será o estado mais recente após a aplicação das atualizações.

**Gerenciando
Estados**

Usando Funções

```
1  import React, { useState } from 'react'
2
3  const Counter = () => {
4    const [count, setCount] = useState(0)
5
6    return (
7      <div>
8        Valor: {count}
9        <button type='button' onClick={() => setCount(count + 1)}>Contar</button>
10      </div>
11    )
12  }
13
14  export default Counter
```

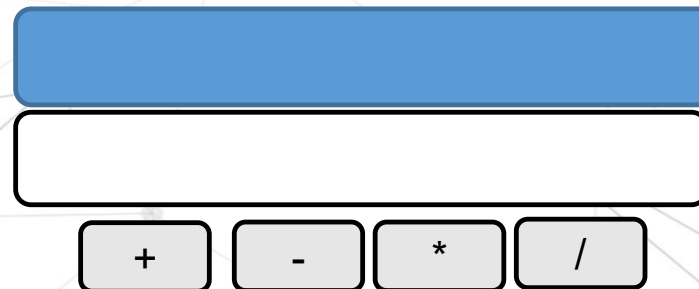
**Gerenciando
Estados**

Usando Classes

```
1  import React from "react";
2
3  class Counter extends React.Component {
4    constructor() {
5      super();
6      this.state = {
7        count: 0
8      };
9    }
10
11    render() {
12      return (
13        <div>
14          Valor: {this.state.count}
15          <button onClick={() => this.setState({ count: this.state.count + 1 })}>Contar</button>
16        </div>
17      );
18    }
19  }
20
21  export default Counter
```

Primeiro Projeto Solo

Crie um componente React que funcione como uma calculadora simples



Propriedades Usando Classes

```
1  import React from "react";
2
3  class Counter extends React.Component {
4    constructor(props) {
5      super(props);
6      this.state = {
7        count: parseInt(props.inicio)
8      };
9    }
10
11    render() {
12      return (
13        <div>
14          Valor: {this.state.count}
15          <button onClick={() => this.setState({ count: this.state.count + 1 })}>Contar</button>
16        </div>
17      );
18    }
19  }
20
21  export default Counter
```

```
4  function App() {
5    return (
6      <div>
7        <Counter inicio='10' />
8      </div>
9    );
10 }
```

Propriedades

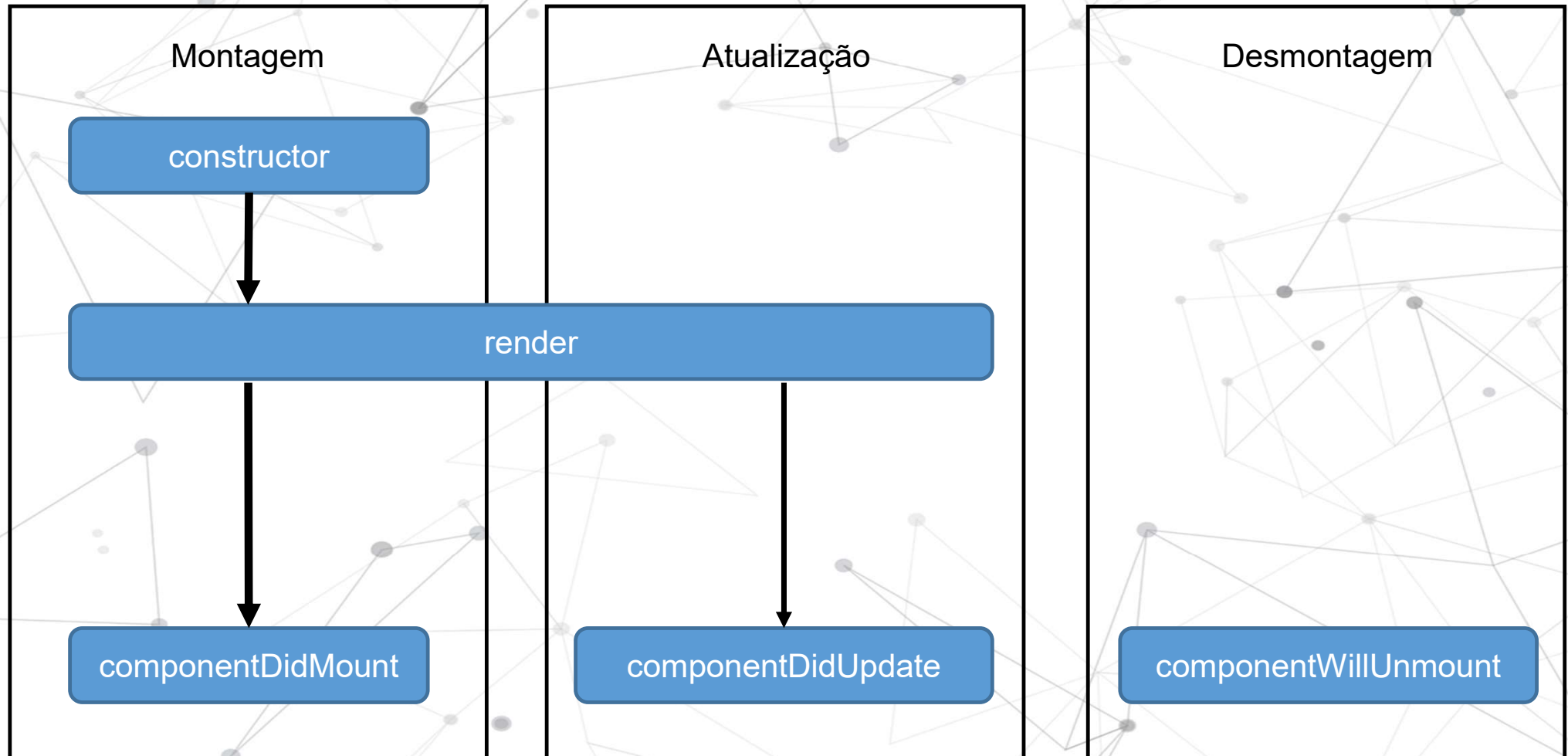
```
1 import React, { useState } from "react";
2
3 const Counter = (props) => {
4   const [count, setCount] = useState(parseInt(props.inicio))
5   return (
6     <div>
7       Valor: {count}
8       <button onClick={() => setCount(count + 1 )}>Contar</button>
9     </div>
10  );
11 }
12
13 export default Counter
```

Usando Funções

```
4 function App() {
5   return (
6     <div>
7       <Counter inicio='10' />
8     </div>
9   );
10 }
```


React Hooks

Além do `useState` temos outros Hooks que permitem o gerenciamento do ciclo de vida dos componentes.





useEffect

```
// componentDidMount
useEffect(() => {
  // Todo
}, [])

// componentDidUpdate
useEffect(() => {
  // Todo
})

// componentDidUpdate
useEffect(() => {
  // Todo
}, [dependencies])

// componentWillUnmount
useEffect(() => {
  return () => {
    // Todo
  }
}, [])
```

useLayoutEffect

```
useLayoutEffect(() => {  
  // Todo  
}, [])
```

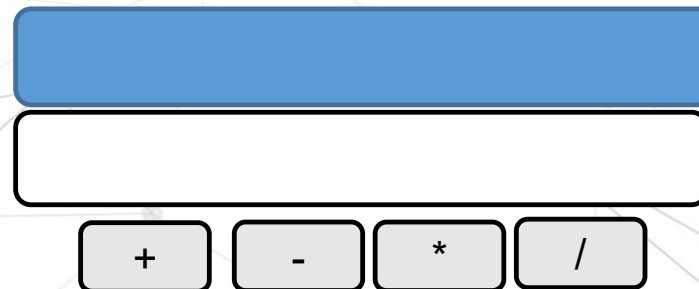
useLayoutEffect

render

useEffect

Segundo Projeto Solo

Construa uma aplicação React que monte a calculadora em componente individuais, sendo: Display, Input e Botão



Desenvolvendo Componente Visual (Aluraflix)

Especialização em Desenvolvimento Full Stack

React

Prof. Esp. Marco Antônio Gusmão Carvalho