

React

Uma biblioteca Javascript para criar interfaces

Allan Ramos

Desenvolvedor na Pagar.me

Github: allangrds \ **Twitter:** allangrds \ **Medium:** allangrds

Revisar é
preciso



```
<p>
```

```
  Número:
```

```
    <span id="number"></span>
```

```
</p>
```

```
<button id="aumenta">Aumentar</button>
```

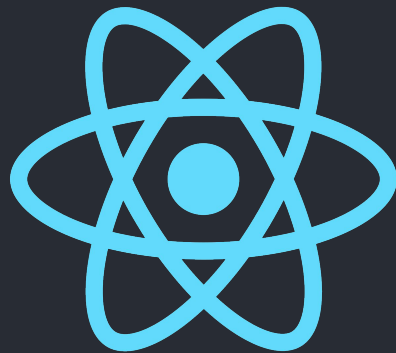
```
<button id="diminui">Diminuir</button>
```



```
(function() {  
    const numero = document.getElementById('number');  
  
    numero.textContent = '0';  
})();  
  
document.getElementById("aumenta").addEventListener("click", function() {  
    const numero = document.getElementById('number');  
  
    numero.textContent = parseInt(numero.textContent) + 1;  
});  
  
document.getElementById("diminui").addEventListener("click", function() {  
    const numero = document.getElementById('number');  
  
    numero.textContent = parseInt(numero.textContent) - 1;  
});
```



```
1 class Tela extends React.Component {
2   constructor() {
3     super();
4     this.state = { numero: 0 };
5
6     this.aumentaNumero = this.aumentaNumero.bind(this);
7     this.diminuiNumero = this.diminuiNumero.bind(this);
8   }
9
10  aumentaNumero() {
11    this.setState({
12      numero: ++this.state.numero
13    });
14  }
15
16  diminuiNumero() {
17    this.setState({
18      numero: --this.state.numero
19    });
20  }
21
22  render() {
23    return (
24      <div>
25        <p>Número: { this.state.numero }</p>
26        <button onClick={this.aumentaNumero}>Aumenta</button>
27        <button onClick={this.diminuiNumero}>Diminui</button>
28      </div>
29    );
30  }
31 }
32
33 ReactDOM.render(<Tela />, document.getElementById('root'));
```

Props

Propriedades externas de um componente



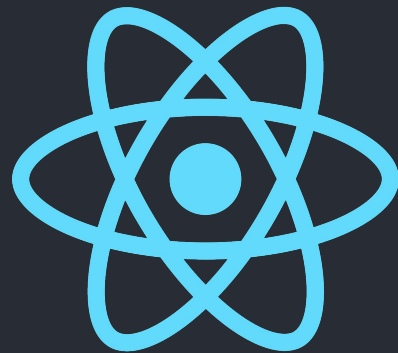
```
1 <Botao title="Clique aqui" />
```

Botão com props “title”



```
1 class Botao extends React.Component {  
2   render() {  
3     return (  
4       <button>  
5         { this.props.title }  
6       </button>  
7     )  
8   }  
9 }
```

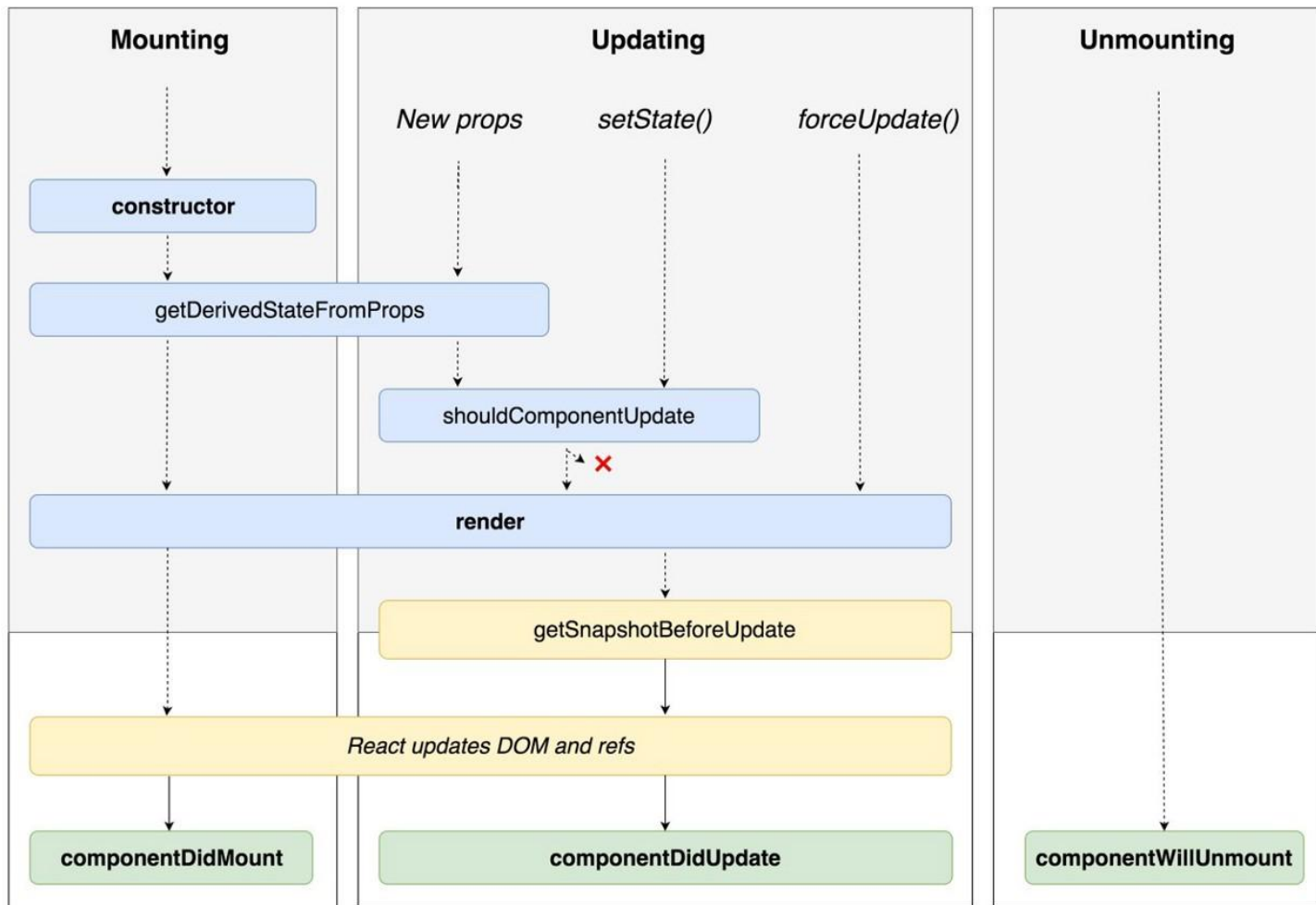
Botão com props “title”



Life cycle

Assim como o ciclo de vida de uma pessoa, tal é o componente. E para cada momento da vida(life cycle), uma função.

```
1 class Tela extends React.Component {
2   constructor() {
3     super();
4     this.state = { loading: true };
5   }
6
7   componentDidMount() {
8     setTimeout(() => {
9       this.setState({
10         loading: false
11       });
12     }, 4000);
13   }
14
15
16   render() {
17     const { loading } = this.state;
18     const { name } = this.props;
19
20     return (
21       <p>
22         {
23           loading
24             ? 'Carregando...'
25             : `Olá ${name}`
26         }
27       </p>
28     );
29   }
30 }
31
32 ReactDOM.render(<Tela name="allan" />, document.getElementById('root'));
```

“Render Phase”

Pure and has no side effects.
May be paused, aborted or
restarted by React.

“Pre-Commit Phase”

Can read the DOM.

“Commit Phase”

Can work with DOM,
run side effects,
schedule updates.

Vamos fazer algo
juntos?

Live Coding

1. Deverá existir um componente Paragrafo;
2. Deverá existir um componente Button;
3. Deverá existir o componente mestre App;
4. Deverá existir um botão “aumenta” e outro “diminui”;
5. Deverá mostrar um loading por 4 seg após a aplicação ser iniciada;
6. Ao clicar no “aumenta”, deverá aumentar o número;
7. Ao clicar no “diminui”, deverá diminuir o número.



```
class Paragrafo extends React.Component {  
  render() {  
    return <p>{this.props.texto}</p>  
  }  
}  
  
class Botao extends React.Component {  
  render() {  
    return (  
      <button onClick={this.props.onClick}>  
        {this.props.titulo}  
      </button>  
    )  
  }  
}
```

```
class App extends React.Component {
  constructor() {
    super();

    this.state = {
      numero: 0
    };

    this.aumentar = this.aumentar.bind(this);
    this.diminuir = this.diminuir.bind(this);
  }

  aumentar() {
    const { numero } = this.state;

    this.setState({
      numero: numero + 1
    });
  }

  diminuir() {
    const { numero } = this.state;

    this.setState({
      numero: numero - 1
    });
  }

  render() {
    console.log(this.state);
    return (
      <div>
        <Paragrafo texto={`Número: ${this.state.numero}`} />
        <Botao onClick={this.aumentar} titulo="Aumentar" />
        <Botao onClick={this.diminuir} titulo="Diminuir" />
      </div>
    )
  }
}
```


#partiuExercício

TODO

- Tarefa #1
- Tarefa #2

What needs to be done?

Add #3

Continuação do ToDoist

1. Criar um componente Input;
2. Criar um componente Button;
3. Criar componente mestre App;
4. No componente Input, passar a função onChange como prop;
5. No componente Button, passar a função onClick como prop.
6. Ao clicar no botão, inserir no estado do “App” o valor que está no input.