PUC Minas

DIRETORIA DE
EDUCAÇÃO CONTINUADA

Pós Graduação *Lato Sensu*

Pós Graduação

Desenvolvimento Web Full Stack

Disciplina

Frameworks front end: React (FFR)

Professor

Samuel Martins

samuelmartins.sw@gmail.com

No capítulo anterior...

- Introdução a disciplina;
- Introdução ao React;
  - Componentes;
  - Create react app;
  - Criação de uma aplicação componentizada.

No capítulo de hoje...

- Ciclos de vida de um componente;
- Comunicação entre componentes;
- Escrevendo rotas *client-side* com React Router.

Ciclos de vida

- Descrevem ações que podem ser tomadas em momentos diferentes da existência de um componente;
  - Exemplo: quero alterar a cor de background *sempre que o componente sofrer qualquer atualização de propriedades*;
  - Exemplo 2: quero alterar o título da aba *sempre que o componente sumir da tela*;

- Possíveis de serem manipulados em class components ou utilizando *hooks*.

Ciclos de vida

Fonte: Vecteezy.com

Function components (múltiplas propriedades)

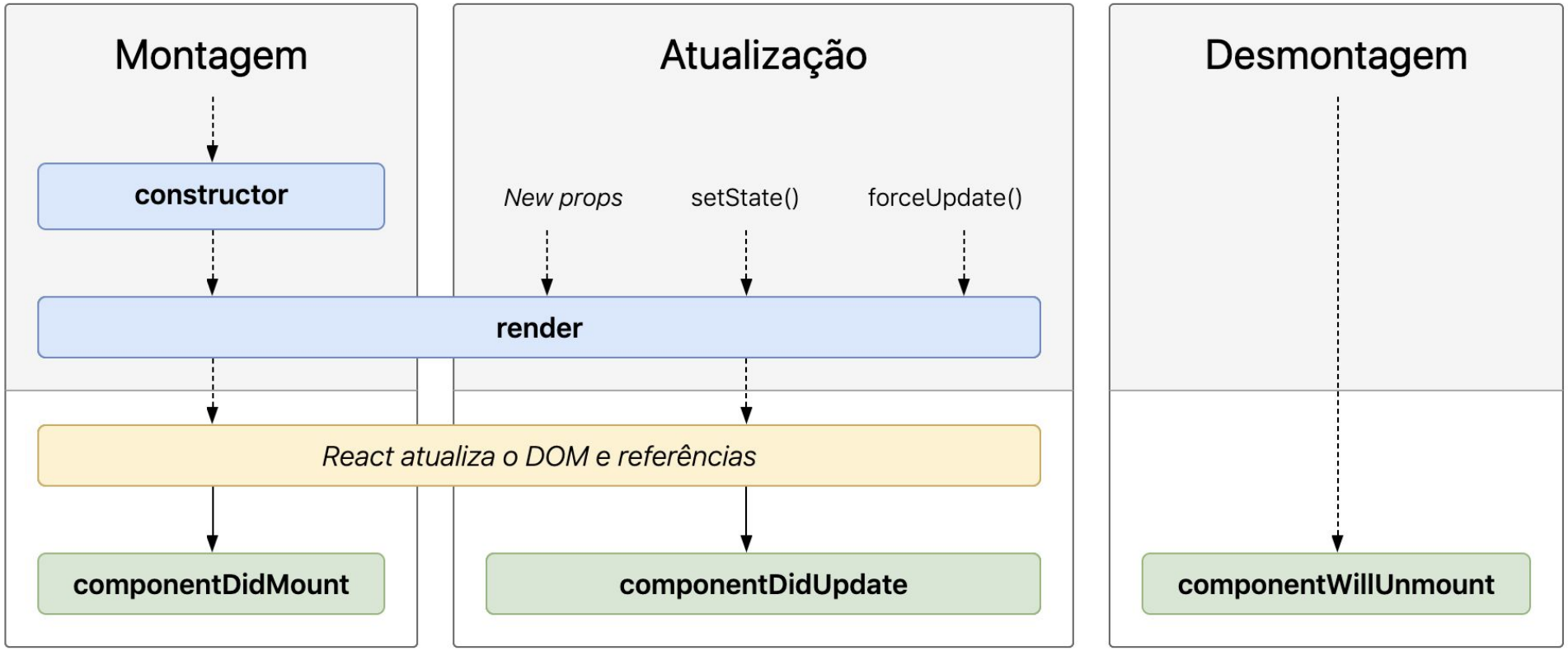
```
1 export const SidebarComponent = (props) => {
2   <aside>
3     <h3>{props.title}</h3>
4
5     <ul>
6       {props.items.map(item => <li>{item}</li>}
7     </ul>
8   </aside>
9 }
```

Function component -> Class component

```
import React, { Component } from "react";

export class SidebarComponent extends Component {
  render() {
    return (
      <aside>
        <h3>{this.props.title}</h3>

        <ul>
          {this.props.items.map(item => (
            <li>{item}</li>
          ))}
        </ul>
      </aside>
    );
  }
}
```



React lifecyle methods diagram: http://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/

componentDidMount()

- Executado sempre que o componente for montado (inserido na árvore de elementos DOM);

```
1 class MyComponent extends Component {
2   componentDidMount() {
3     console.log('Mounted!')
4   }
5   // Ciclo do render
6   render() {
7     return <h1>Hello world</h1>
8   }
9 }
```

componentDidUpdate()

- Chamado imediatamente após ocorrer alguma atualização (propriedades ou estados);

- Útil para requisições externas ou outros efeitos colaterais.

```
class MyComponent extends Component {
  componentDidUpdate(prevProps, prevState) {
    // Typical usage (don't forget to compare props):
    if (this.props.userID !== prevProps.userID) {
      this.fetchData(this.props.userID);
      window.title = 'New user received!'
    }
  }

  render() {
    return <h1>Hello world</h1>
  }
}
```

componentWillUnmount()

- Executado imediatamente após o componente ser destruído (removido da árvore DOM)

```
1 class MyComponent extends Component {
2   handleResize = () => console.log('resized!');
3
4   componentDidMount() {
5     window.addEventListener('resize', this.handleResize)
6   }
7
8   componentWillUnmount() {
9     window.removeEventListener('resize', this.handleResize)
10  }
11 }
```

Comunicação entre componentes

Comunicação entre componentes

- Possibilidade de compartilhar dados entre componentes;
- Possibilidade de alterar dados de outros componentes;
- Componentes que compartilham alterações tendem a ser mais reutilizáveis;

- ***Exemplo***: quero chamar um callback sempre que um componente for salvo. Esse callback nem sempre deve ter o mesmo comportamento dependendo do contexto em que estiver inserido.

Caso de uso

- Aplicativo de TODO-List (lista de tarefas);
  - Componente <TodoListItems /> contém um array com todos os itens da listagem;
  - Componente <TodoItem /> recebe as informações específicas de cada componente.

Exemplo no code sandbox

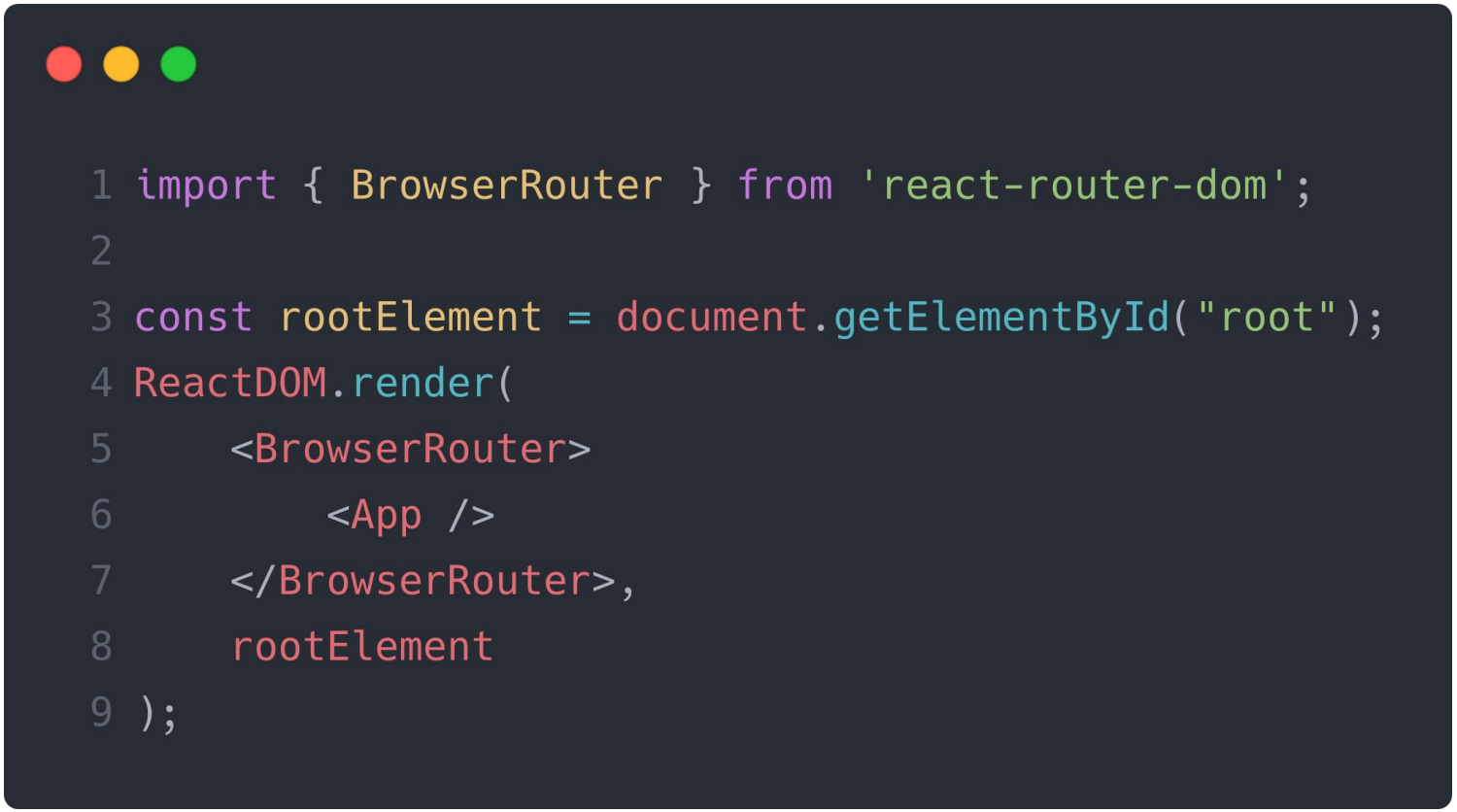
https://codesandbox.io/s/communication-between-components-i63xk

https://reacttraining.com/react-router/web/guides/quick-start

Introdução ao React Router

React Router

- Utilizado na criação de Single Page Applications;
- Conceito similar ao Express, do NodeJs, com a diferença de aceitar apenas requisições GET;
  - Parâmetros;
  - Rotas “herdadas”;
- *Component-driven*;



```
1 import { BrowserRouter } from 'react-router-dom';
2
3 const rootElement = document.getElementById("root");
4 ReactDOM.render(
5     <BrowserRouter>
6         <App />
7     </BrowserRouter>,
8     rootElement
9 );
```

Declaração de rotas precisam estar “envelopadas” com o componente BrowserRouter

```jsx
import React from "react";
import { Link } from "react-router-dom";
import { ApplicationRoutes } from "./ApplicationRoutes";

export function App() {
  return (
    <div>
      <ul>
        <li><Link to="/">Home</Link></li>
        <li><Link to="/about">About</Link></li>
      </ul>
      <ApplicationRoutes />
    </div>
  );
}
```

```jsx
import React from "react";
import { Route } from "react-router-dom";
import { Home, About } from "./components";

export const ApplicationRoutes = () => (
  <>
    <Route path="/" exact component={Home} />
    <Route path="/about/" component={About} />
  </>
);
```

```
1 const Index = () => <h1>Hello, this is the Index page!</h1>;
2 const About = () => <h1>About page</h1>;
```

Exemplo no code sandbox

https://codesandbox.io/s/react-router-example-gghyv

Passagem de parâmetros

```
1 export const ApplicationRoutes = () => (
2   <>
3     <Route path="/product/:id" component={Products} />
4   </>
5 );
```

```
1 export const Products = (props) => (
2   <h1>Product id is: {props.match.params.id}</h1>
3 )
```

Exercício 02