

React Básico

Conteúdo de apoio

Node e npm

A versão do [Node](#) utilizada durante o curso é a 18.16.1, caso queira utilizar versões mais recentes, fique atento com possíveis incompatibilidades.

Ao instalar o Node, também será instalado o [npm](#), que é o Gerenciador de Pacotes do Node (**N**ode **P**ackage **M**anager), ele é utilizado para adicionar e remover dependências, bem como instalá-las.

Instalando dependências:

```
npm i
```

Adicionando nova dependência:

```
npm i --save nome_dependencia
```

Removendo dependência:

```
npm uninstall nome_dependencia
```

Rodando scripts:

```
npm run nome_script
```

Vite

A versão do [Vite](#) utilizada no curso é a 4.4.5, caso queira utilizar versões mais recentes, fique atento com possíveis incompatibilidades.

Inicializando o projeto:

```
npm create vite@latest
```

Estrutura

A versão do [React](#) utilizada no curso é a 18.2.15, caso queira utilizar versões mais recentes, fique atento com possíveis incompatibilidades.

Arquivos/pastas importantes

package.json: Metadados sobre o projeto como nome, versão, dependências e scripts

public: Pasta onde ficam os arquivos estáticos do projeto, como imagens, favicon e outros recursos que não precisam ser processados pela ferramenta de build

index.html: Ponto de entrada para o projeto, é aqui que todo o código do projeto será carregado para o usuário

main.jsx: Script responsável por injetar o código js do projeto no index.html

App.jsx: Componente inicial do projeto

Componentes

Import/Export

Default export (Exportação padrão):

```
// Export
export default App;

// Import
import algumNome from './App';
```

Named export (Exportação nomeada):

```
// Export
export { App };

// Import
import { App } from './App';
```

Componente

```
const NomeComponente = (props) => {  
  return (  
    <div>  
      Algum conteúdo  
    </div>  
  );  
};
```

Fragment (Fragmento)

```
const NomeComponente = (props) => {  
  return (  
    <></>  
    // OU  
    <React.Fragment></React.Fragment>  
  );  
};
```

Propriedades

Passando propriedades para o componente:

```
<NomeComponente nomePropriedade="Joana" />
```

Acessando propriedades no componente:

```
const NomeComponente = (props) => {  
  const { nomePropriedade } = props;  
  // OU  
  props.nomePropriedade;  
};
```

Children (Filhos):

```
// Passando children para o componente  
<NomeComponente>  
  <h1>Algum texto</h1>  
</NomeComponente>  
  
// Recebendo children no componente  
const NomeComponente = (props) => {  
  const { children } = props;  
};
```

CSS

Importando CSS globalmente:

```
import './global.css';
```

Importando CSS de forma modular:

É necessário criar o arquivo CSS com a extensão .module.css e então importá-lo dessa forma:

```
import style from './MeuEstilo.module.css';
```

Para utilizar, as classes são acessadas utilizando ponto:

```
<div className={style.container}>
```

Roteamento

A versão do [react-router-dom](#) utilizada no curso é a 4.4.5, caso queira utilizar versões mais recentes, fique atento com possíveis incompatibilidades.

Instalação

```
npm i react-router-dom --save
```

Criando rotas

```
import { Routes, Route } from 'react-router-dom';

const Router = () => {
  return (
    <Routes>
      <Route path="/" element={<Inicial />} />
      <Route path="/sobre-nos" element={<SobreNos />} />
    </Routes>
  );
};

export { Router };
```

Configurando o projeto para usar as rotas

```
import { BrowserRouter } from 'react-router-dom';

import { Router } from './Router';

const App = () => {
  return (
    <BrowserRouter>
      <Router />
    </BrowserRouter>
  );
};
```

Adicionando links no projeto

```
import { Link } from 'react-router-dom';

const AlgumComponente = () => {
  return (
    <div>
      <Link to="/">Texto do link</Link>
    </div>
  );
};
```

Layout

```
import { Outlet } from 'react-router-dom';

const LayoutPadrao = () => {
  return (
    <div>
      <Outlet />
    </div>
  );
};
```

Adicionando o Layout nas rotas

```
const Router = () => {
  return (
    <Routes>
      <Route path="/" element={<LayoutPadrao />}>
        <Route path="/" element={<Inicial />} />
        <Route path="/sobre-nos" element={<SobreNos />} />
      </Route>
    </Routes>
  );
};
```


Context API

Estrutura base

```
import { createContext } from 'react';
export const AppContext = createContext({});

export const AppContextProvider = (props) => {
  const { children } = props;

  return (
    <AppContext.Provider value={contexto}>
      {children}
    </AppContext.Provider>
  );
};
```

Adicionando contexto no App

```
<AppContextProvider>
  <BrowserRouter>
    <Router />
  </BrowserRouter>
</AppContextProvider>
```

Hooks

[useState](#): Adiciona estado variável ao seu componente

```
const [estado, setEstado] = useState(estadoinicial);
```

Alterando o estado

```
setEstado(10);
```

Alterando o estado utilizando o estado atual

```
setEstado(estadotual => estadotual + 1);
```

[useContext](#): Permite que o componente acesse e se inscreva em um contexto

```
const contexto = useContext(Contexto);
```

[useEffect](#): Permite sincronizar o componente com um sistema externo

```
useEffect(() => {  
  // O que será executado quando o valor das dependências mudar  
  // Caso o array de dependências esteja vazio, executa quando o componente é montado  
}, []); // Array de dependências
```

Explore mais!

Para se aprofundar no React, leia a documentação da biblioteca e das demais bibliotecas utilizadas durante o curso.

Também fique atento aos anúncios e lançamentos relacionados ao React e a comunidade para entender o que está ou não sendo utilizado para direcionar seus estudos.



Ilustração feita por upklyak,
disponível no Freepik.
Adaptada pelo autor.

Referências bibliográficas

Documentação Node.js, 2023. Disponível em: <https://nodejs.org/en/docs>

npm Docs, 2023. Disponível em: <https://docs.npmjs.com/>

Vite Guide, 2023. Disponível em: <https://vitejs.dev/guide/>

Learn React, 2023. Disponível em: <https://react.dev/learn>

React Router, 2023. Disponível em: <https://reactrouter.com/en/main>