**Desenvolvimento Com React JS**

Nome: Gabriel Neves

**COMPONENTES E EVENTOS**

**Criando o Projeto React**

**Inicialização do Projeto**:

1. Abri o Prompt de Comando e executei o seguinte comando para criar um novo projeto React:

*npx create-react-app formulario-react*

1. Acesso ao Diretório do Projeto:

*cd formulario-react*

1. Instalação de Dependências:

Para gerenciar o formulário e validar os dados, instalei as bibliotecas necessárias:

*npm install formik yup*

1. Criação do Formulário:

Após a instalação, abri o Visual Studio Code e criei um novo arquivo chamado FormularioEmail.js na pasta src, contendo o seguinte código:

*import React from 'react';*

*import { useFormik } from 'formik';*

*import \* as Yup from 'yup';*

*const FormularioEmail = () => {*

*const formik = useFormik({*

*initialValues: {*

*email: '',*

*},*

*validationSchema: Yup.object({*

*email: Yup.string()*

*.email('Digite um email válido')*

*.required('O email é obrigatório'),*

*}),*

*onSubmit: values => {*

*alert(JSON.stringify(values, null, 2));*

*},*

*});*

*return (*

*<form onSubmit={formik.handleSubmit}>*

*<label htmlFor="email">Email</label>*

*<input*

*id="email"*

*name="email"*

*type="email"*

*onChange={formik.handleChange}*

*onBlur={formik.handleBlur}*

*value={formik.values.email}*

*/>*

*{formik.touched.email && formik.errors.email ? (*

*<div>{formik.errors.email}</div>*

*) : null}*

*<button type="submit">Enviar</button>*

*</form>*

*);*

*};*

*export default FormularioEmail;*

1. Integração com o App Principal:

No arquivo App.js, também na pasta src, substituí o conteúdo original pelo seguinte código para importar e utilizar o componente FormularioEmail:

*import React from 'react';*

*import FormularioEmail from './FormularioEmail';*

*function App() {*

*return (*

*<div className="App">*

*<h1>Formulário de Email</h1>*

*<FormularioEmail />*

*</div>*

*);*

*}*

*export default App;*

1. Execução do Projeto:

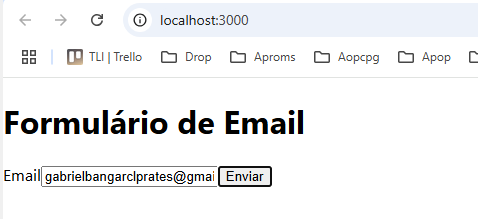
Retornei ao terminal e executei o comando:

*npm start*

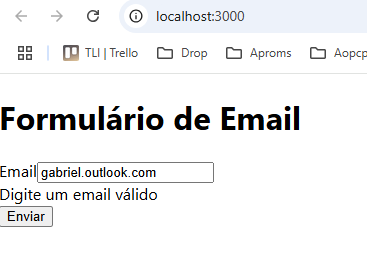
O navegador abriu automaticamente em *http://localhost:3000,* onde pude visualizar e testar o formulário, verificando a validação tanto para entradas válidas quanto inválidas.

1. Evidências de Resultado:

Valido:



Inválido:



**HOOKS, CONTEXT API E REDUX**

**Criando o Projeto Contador**

1. Inicialização do Projeto:

Abri o terminal e criei um novo projeto React para um contador com o seguinte comando:

*npx create-react-app contador*

Acessei a pasta do projeto:

*cd contador*

1. Instalação de Dependências para Testes:

Instalei a React Testing Library:

*npm install @testing-library/react*

1. Criação do Componente Contador:

No Visual Studio Code, criei um arquivo chamado Counter.js na pasta src e adicionei o seguinte código:

*import React, { useState } from 'react';*

*const Counter = () => {*

*const [count, setCount] = useState(0);*

*return (*

*<div>*

*<h1>Contador: {count}</h1>*

*<button onClick={() => setCount(count + 1)}>Incrementar</button>*

*<button onClick={() => setCount(count - 1)}>Decrementar</button>*

*</div>*

*);*

*};*

*export default Counter;*

1. Criei um arquivo de teste chamado Counter.test.js na pasta src, com os seguintes testes para verificar o comportamento do contador:

*import React from 'react';*

*import { render, fireEvent } from '@testing-library/react';*

*import Counter from './Counter';*

*test('exibe o contador com valor inicial 0', () => {*

*const { getByText } = render(<Counter />);*

*expect(getByText(/Contador: 0/i)).toBeInTheDocument();*

*});*

*test('incrementa o contador ao clicar no botão de incrementar', () => {*

*const { getByText } = render(<Counter />);*

*const incrementarButton = getByText(/Incrementar/i);*

*fireEvent.click(incrementarButton);*

*expect(getByText(/Contador: 1/i)).toBeInTheDocument();*

*});*

*test('decrementa o contador ao clicar no botão de decrementar', () => {*

*const { getByText } = render(<Counter />);*

*const decrementarButton = getByText(/Decrementar/i);*

*fireEvent.click(decrementarButton);*

*expect(getByText(/Contador: -1/i)).toBeInTheDocument();*

*});*

1. Objetivos dos Testes:

 O primeiro teste assegura que o contador inicie em 0.

 O segundo teste verifica se o contador incrementa corretamente ao clicar no botão de incrementar.

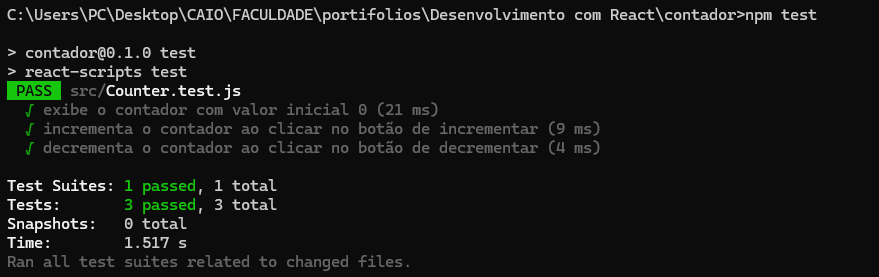
 O terceiro teste confirma que o contador decrementa corretamente ao clicar no botão de decrementar.

1. Retornei ao terminal e executei os testes com o comando:

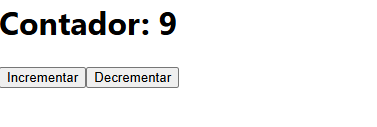
*npm test*

1. Evidências e resultados:

Testes:



Navegador:



**HOOKS, CONTEXT API E REDUX**

**Criando o Projeto Next.js**

1. Instalação do Create Next App:

No terminal, instalei o create-next-app globalmente para garantir que estava na versão mais recente:

*npm install -g create-next-app*

1. Criação do Projeto Next.js:

Utilizei o npx para criar o projeto:

*npx create-next-app meu-projeto-next*

Segui as instruções no terminal para nomear o projeto e configurar as opções padrão.

1. Acesso ao Diretório do Projeto:

*cd meu-projeto-next*

1. Criação da Página Inicial:

Dentro da pasta pages, criei um arquivo chamado index.js e adicionei o seguinte código para exibir uma mensagem de boas-vindas:

*export default function Home() {*

*return (*

*<div>*

*<h1>Bem-vindo ao meu projeto Next.js!</h1>*

*<p>Este é o componente de página inicial.</p>*

*</div>*

*);*

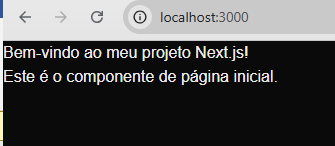
*}*

1. Execução do Projeto:

Para iniciar o servidor de desenvolvimento, executei o comando:

*npm run dev*

1. Resultado e Evidências:



Todo código está localizado no GitHub: