

PORTFÓLIO DE DESENVOLVIMENTO COM REACT

5ºSEMESTRE

DATA: 11/10/2024

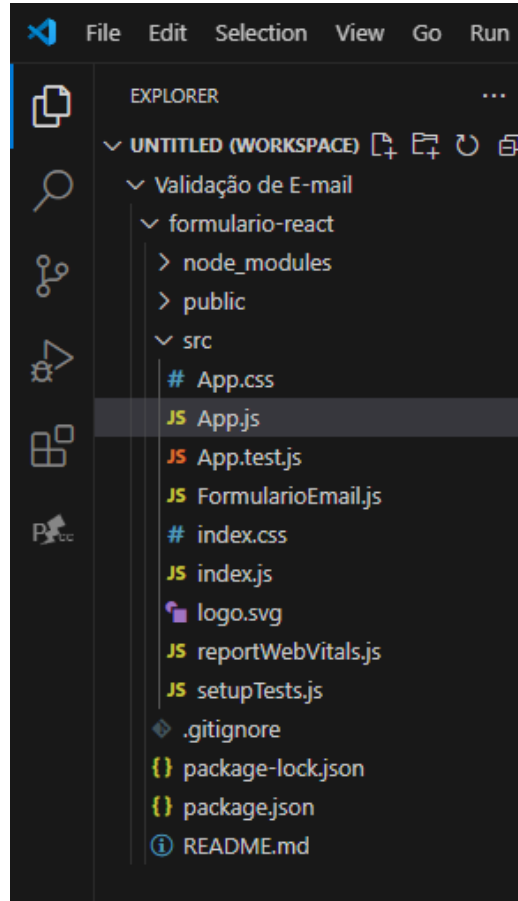
PROFESSORES: Gian Carlo Decarli

NOME DO ALUNO: Fernando Henrique Panini

ATIVIDADE - I

Crie um projeto com formulário onde conste o campo de entrada E-mail, fazendo a validação de esquemas utilizando a biblioteca yup e formik para gerenciar estados do formulário.

Arquivos Criados:



Códigos do Arquivo FormularioEmail.js:

```
JS FormularioEmail.js X JS App.js
Validação de E-mail > formulario-react > src > JS FormularioEmail.js > ...
1  import React from 'react';
2  import { useFormik } from 'formik';
3  import * as Yup from 'yup';
4
5  const FormularioEmail = () => {
6    const formik = useFormik({
7      initialValues: {
8        email: '',
9      },
10     validationSchema: Yup.object({
11       email: Yup.string()
12         .email('Email inválido')
13         .required('Campo obrigatório'),
14     }),
15     onSubmit: (values) => {
16       alert(`Email enviado: ${values.email}`);
17     },
18   });
19
20   return (
21     <form onSubmit={formik.handleSubmit}>
22       <div>
23         <label htmlFor="email">Email:</label>
24         <input
25           id="email"
26           name="email"
27           type="email"
28           onChange={formik.handleChange}
29           onBlur={formik.handleBlur}
30           value={formik.values.email}
31         />
32         {formik.touched.email && formik.errors.email ? (
33           <div style={{ color: 'red' }}>{formik.errors.email}</div>
34         ) : null}
35       </div>
36       <button type="submit">Enviar</button>
37     </form>
38   );
39 };
40
41 export default FormularioEmail;
```

Códigos do Arquivo App.js:

```
JS FormularioEmail.js JS App.js X
Validação de E-mail > formulario-react > src > JS App.js > ...
1  import React from 'react';
2  import FormularioEmail from './FormularioEmail';
3
4  function App() {
5    return (
6      <div className="App">
7        <h1>Formulário de Email</h1>
8        <FormularioEmail />
9      </div>
10   );
11 }
12
13 export default App;
14
```

Servidor Funcionando:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK

Compiled successfully!

You can now view formulario-react in the browser.

  Local:            http://localhost:3001
  On Your Network:  http://192.168.1.25:3001

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Resultado:

E-mail Válido:

localhost:3001

Formulário de Email

Email:

Enviar

localhost:3001 diz

Email enviado: ferpanini@outlook.com

OK

E-mail Inválido:

localhost:3001

Formulário de Email

Email:

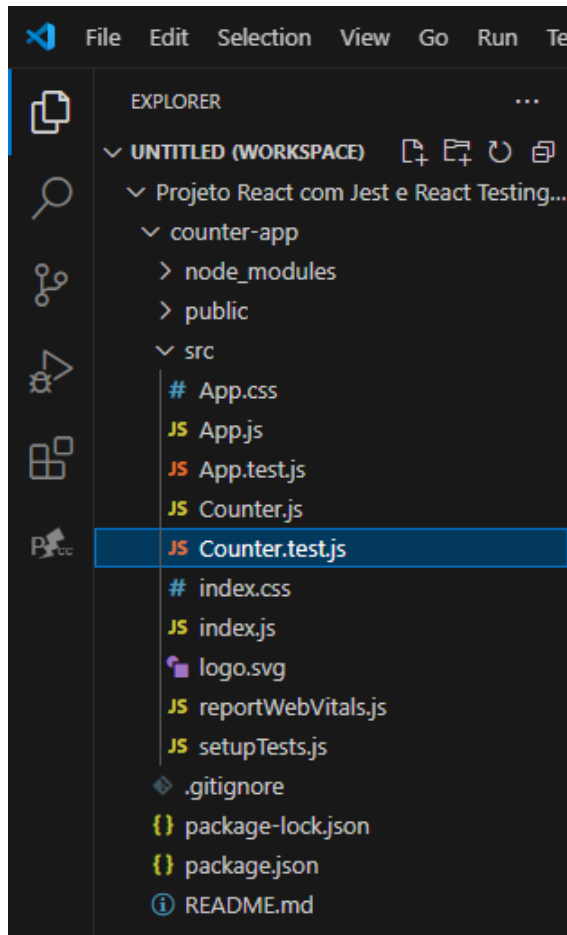
Email inválido

Enviar

ATIVIDADE – II

Crie um componente React simples e escrever testes para ele usando a biblioteca de testes Jest e React Testing Library com o objetivo de garantir que o componente funcione corretamente e que o feedback visual aos usuários seja fornecido de forma adequada.

Arquivos Criados:



Códigos do Arquivo Counter.js:

```
JS Counter.js X JS Counter.test.js JS App.test.js
Projeto React com Jest e React Testing Library > counter-app > src > JS Counter.js > ...
1 // src/Counter.js
2 import React, { useState } from 'react';
3
4 const Counter = () => {
5   const [count, setCount] = useState(0);
6
7   const increment = () => setCount(count + 1);
8   const decrement = () => setCount(count - 1);
9
10  return (
11    <div>
12      <h1>Contador: {count}</h1>
13      <button onClick={increment}>Incrementar</button>
14      <button onClick={decrement}>Decrementar</button>
15    </div>
16  );
17 };
18
19 export default Counter;
20
```

Códigos do Arquivo Counter.test.js:

```
JS Counter.js JS Counter.test.js X
Projeto React com Jest e React Testing Library > counter-app > src > JS Counter.test.js > describe('Count
1 // src/Counter.test.js
2 import React from 'react';
3 import { render, screen, fireEvent } from '@testing-library/react';
4 import Counter from './Counter';
5
6 describe('Counter Component', () => {
7   test('renders with initial count', () => {
8     render(<Counter />);
9     const countElement = screen.getByText(/Contador: 0/i);
10    expect(countElement).toBeInTheDocument();
11  });
12
13   test('increments the counter', () => {
14     render(<Counter />);
15     const incrementButton = screen.getByText(/Incrementar/i);
16     fireEvent.click(incrementButton);
17     const countElement = screen.getByText(/Contador: 1/i);
18     expect(countElement).toBeInTheDocument();
19   });
20
21   test('decrements the counter', () => {
22     render(<Counter />);
23     const decrementButton = screen.getByText(/Decrementar/i);
24     fireEvent.click(decrementButton);
25     const countElement = screen.getByText(/Contador: -1/i);
26     expect(countElement).toBeInTheDocument();
27   });
28 });
29
```

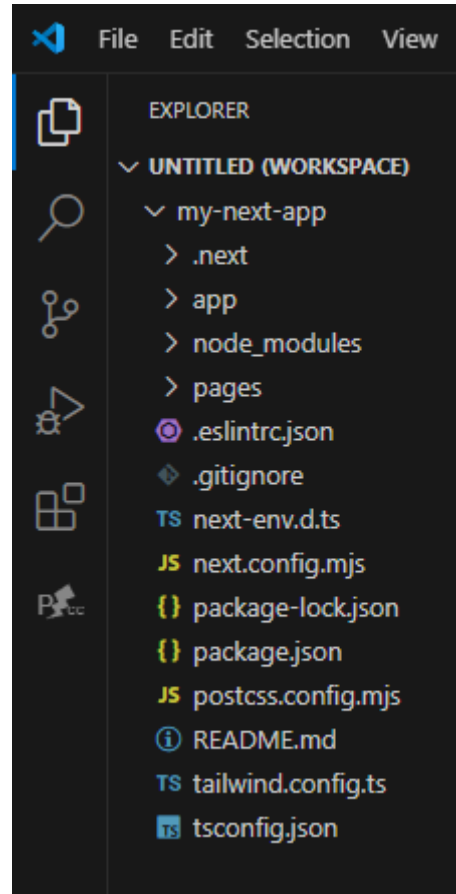
Resultado dos Testes:

```
Test Suites: 1 failed, 1 passed, 2 total
Tests:       1 failed, 3 passed, 4 total
Snapshots:   0 total
Time:        2.915s, estimated 3s
Ran all test suites.
```


ATIVIDADE - III

Crie uma página web usando Next.js, onde o usuário poderá inserir dois números em um formulário. Ao submeter o formulário, a página deve exibir a soma desses dois números.

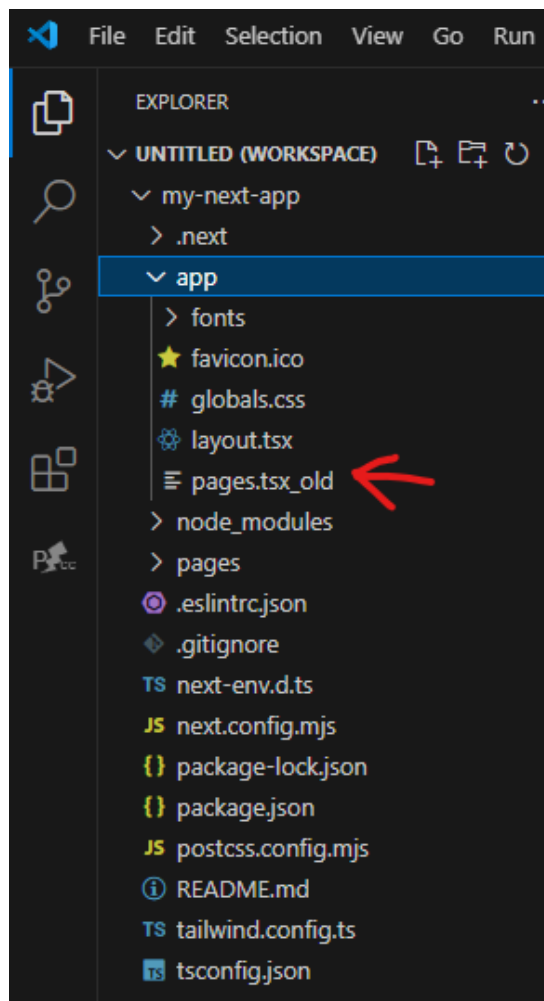
Arquivos Criados:



Códigos do Arquivo Index.js:

```
JS index.js X
my-next-app > pages > JS index.js > Home
1  import { useState } from 'react';
2
3  export default function Home() {
4    const [num1, setNum1] = useState('');
5    const [num2, setNum2] = useState('');
6    const [result, setResult] = useState(null);
7
8    const handleSubmit = (e) => {
9      e.preventDefault();
10     const sum = Number(num1) + Number(num2);
11     setResult(sum);
12   };
13
14   return (
15     <div style={{ padding: '20px' }}>
16       <h1>Calculadora de Soma</h1>
17       <form onSubmit={handleSubmit}>
18         <input
19           type="number"
20           placeholder="Digite o primeiro número"
21           value={num1}
22           onChange={(e) => setNum1(e.target.value)}
23           required
24         />
25         <input
26           type="number"
27           placeholder="Digite o segundo número"
28           value={num2}
29           onChange={(e) => setNum2(e.target.value)}
30           required
31         />
32         <button type="submit">Calcular</button>
33       </form>
34       {result !== null && <h2>Resultado: {result}</h2>}
35     </div>
36   );
37 }
38
```

Arquivo pages.tsx_old:



Servidor Funcionando:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POLYGLOT NOTEBOOK

PS C:\Users\ferpa\OneDrive\Área de Trabalho\Portfólio Desenvolvimento Com React\Página web usando Next.js\my-next-app> npm run dev
>>

> my-next-app@0.1.0 dev
> next dev

▲ Next.js 14.2.13
- Local:      http://localhost:3000

✓ Starting...
✓ Ready in 1635ms
o Compiling / ...
✓ Compiled / in 1244ms (260 modules)
GET / 200 in 1336ms
█
```

Resultado:



Calculadora de Soma

<input type="text" value="2"/>	<input type="text" value="5"/>	<input type="button" value="Calcular"/>
--------------------------------	--------------------------------	---

Resultado: 7