

Taller 6.2 React (Punto 5)

Luisa Fernanda Triviño Gil

Estudiante Sena ADSO

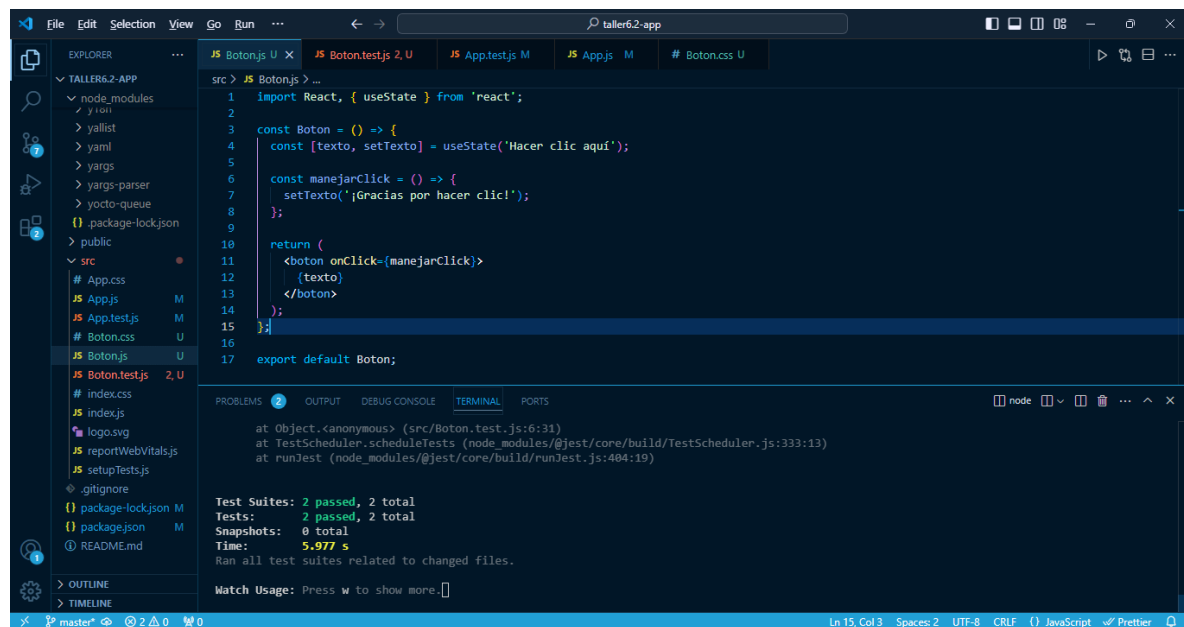
5. Pruebas con Jest y React Testing Library

Objetivo: Realizar una prueba unitaria básica.

- Tarea: Escribe una prueba que verifique si un componente llamado Boton muestra el texto "Hacer clic aquí" y que, al hacer clic en el botón, cambie el texto a "¡Gracias por hacer clic!". o Simula el clic en el botón y verifica que el texto cambie correctamente.

- Tip: Usa las funciones render, fireEvent, y getByText de React Testing Library para realizar la prueba.

Boton.js



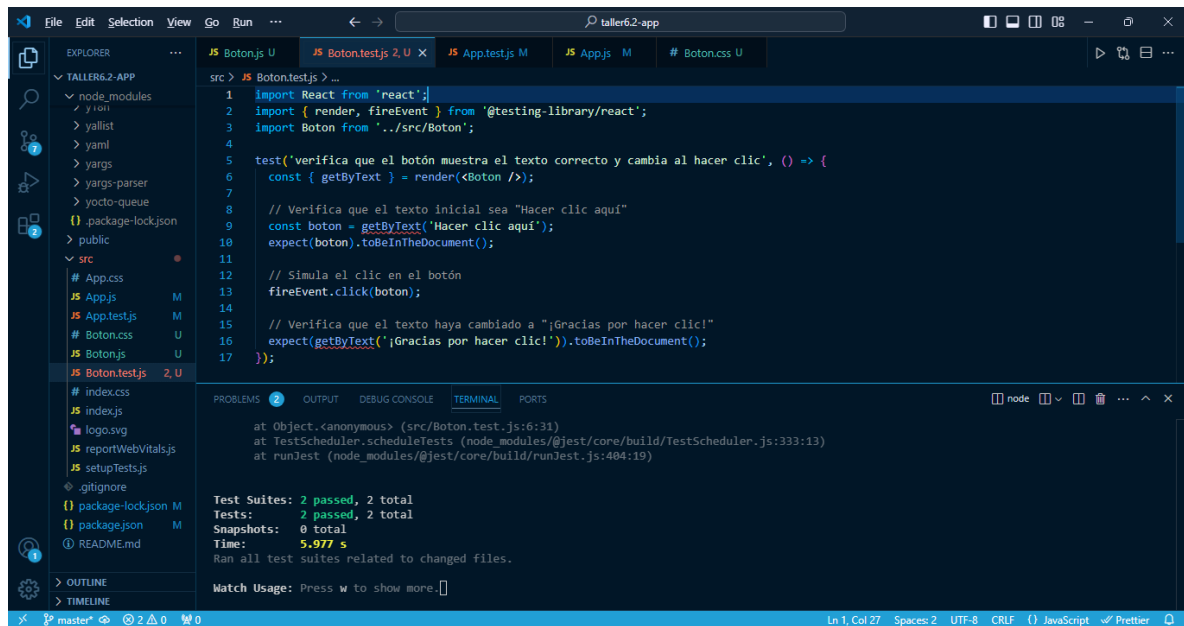
```
src > JS Boton.js > ...
1  import React, { useState } from 'react';
2
3  const Boton = () => {
4    const [texto, setTexto] = useState('Hacer clic aquí');
5
6    const manejarClick = () => {
7      setTexto('¡Gracias por hacer clic!');
8    };
9
10   return (
11     <boton onClick={manejarClick}>
12       {texto}
13     </boton>
14   );
15 }
16
17 export default Boton;
```

```
at Object.<anonymous> (src/Boton.test.js:6:31)
at TestScheduler.scheduleTests (node_modules/@jest/core/build/TestScheduler.js:333:13)
at runJest (node_modules/@jest/core/build/runJest.js:404:19)

Test Suites: 2 passed, 2 total
Tests:       2 passed, 2 total
Snapshots:  0 total
Time:        5.977 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

Boton.test.js



The screenshot shows the VS Code editor with the file `Boton.test.js` open. The file contains the following code:

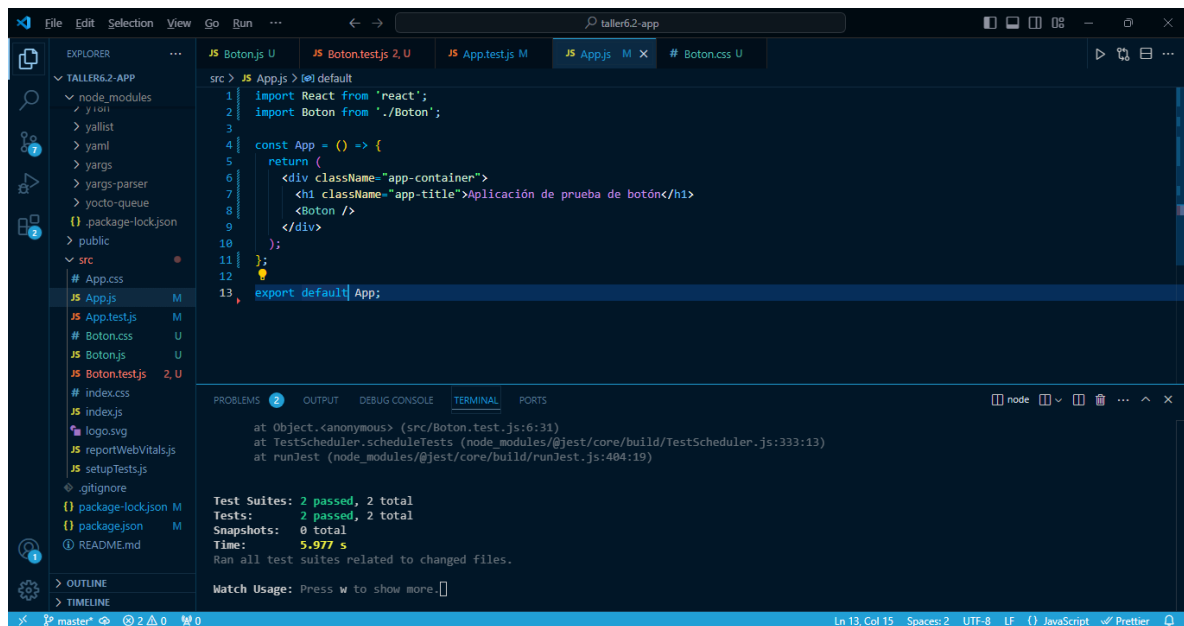
```
1 import React from 'react';
2 import { render, fireEvent } from '@testing-library/react';
3 import Boton from '../src/Boton';
4
5 test('verifica que el botón muestra el texto correcto y cambia al hacer clic', () => {
6   const { getByText } = render(<Boton />);
7
8   // Verifica que el texto inicial sea "Hacer clic aquí"
9   const boton = getByText('Hacer clic aquí');
10  expect(boton).toBeInTheDocument();
11
12  // Simula el clic en el botón
13  fireEvent.click(boton);
14
15  // Verifica que el texto haya cambiado a "¡Gracias por hacer clic!"
16  expect(getByText('¡Gracias por hacer clic!')).toBeInTheDocument();
17 });
```

The terminal output shows the test results:

```
Test Suites: 2 passed, 2 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 5.977 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

App.js



The screenshot shows the VS Code editor with the file `App.js` open. The file contains the following code:

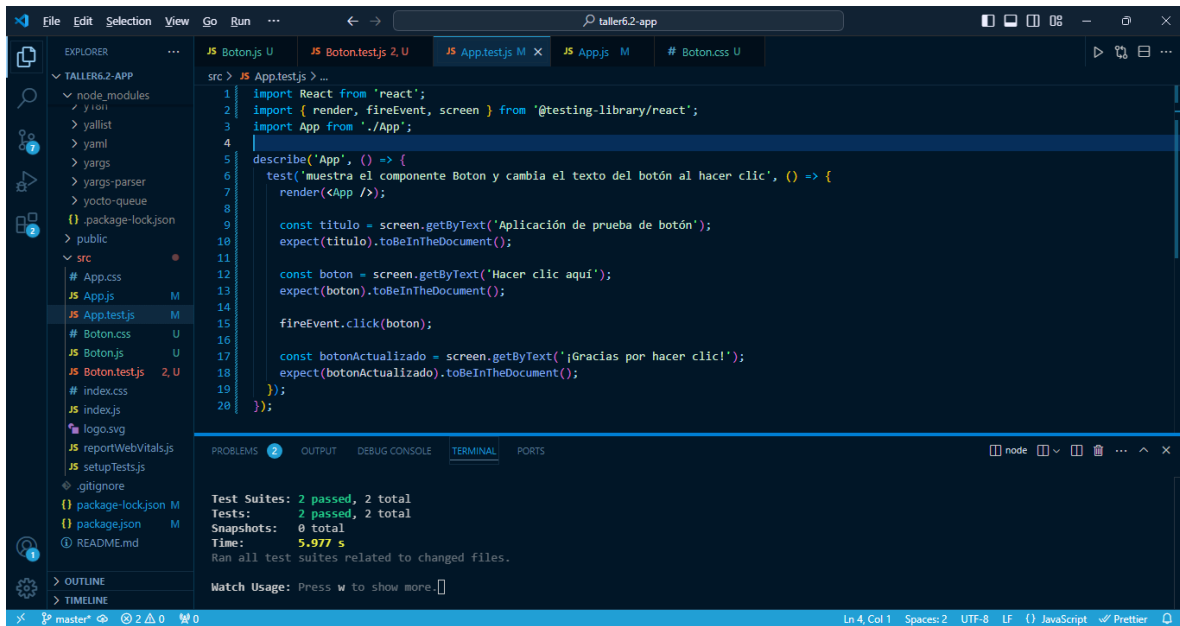
```
1 import React from 'react';
2 import Boton from './Boton';
3
4 const App = () => {
5   return (
6     <div className="app-container">
7       <h1 className="app-title">Aplicación de prueba de botón</h1>
8       <Boton />
9     </div>
10   );
11 }
12
13 export default App;
```

The terminal output shows the test results:

```
Test Suites: 2 passed, 2 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 5.977 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

App.test.js



The image shows a Visual Studio Code editor window with the file `App.test.js` open. The editor has a dark theme. On the left, the Explorer sidebar shows the project structure with folders like `node_modules`, `public`, and `src`. The `src` folder is expanded, showing files like `App.css`, `App.js`, `App.test.js` (selected), `Boton.css`, `Boton.js`, `Boton.test.js`, `index.css`, `index.js`, `logo.svg`, `reportWebVitals.js`, `setupTests.js`, and `.gitignore`. The main editor area shows the content of `App.test.js`, which is a Jest test file. The code imports `React`, `render`, `fireEvent`, and `screen` from `@testing-library/react`, and `App` from `./App`. It defines a `describe` block for `App` with a `test` case that renders the app and checks for the presence of a button and its text. The terminal at the bottom shows the output of running the tests, indicating that 2 test suites and 2 tests passed.

```
src > JS App.test.js > ...
1 import React from 'react';
2 import { render, fireEvent, screen } from '@testing-library/react';
3 import App from './App';
4
5 describe('App', () => {
6   test('muestra el componente Boton y cambia el texto del botón al hacer clic', () => {
7     render(<App />);
8
9     const titulo = screen.getByText('Aplicación de prueba de botón');
10    expect(titulo).toBeInTheDocument();
11
12    const boton = screen.getByText('Hacer clic aqui');
13    expect(boton).toBeInTheDocument();
14
15    fireEvent.click(boton);
16
17    const botonActualizado = screen.getByText('Gracias por hacer clic!');
18    expect(botonActualizado).toBeInTheDocument();
19  });
20 });
```

Test Suites: 2 passed, 2 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 5.977 s
Ran all test suites related to changed files.
Watch Usage: Press w to show more.