

Guía de pruebas Aplicación Giphy Expert

Documentación oficial de jest que es una librería para hacer pruebas en react <https://jestjs.io/>

Documentación oficial enzyme <https://enzymejs.github.io/enzyme/>

Documentación oficial enzyme to json <https://www.npmjs.com/package/enzyme-to-json>

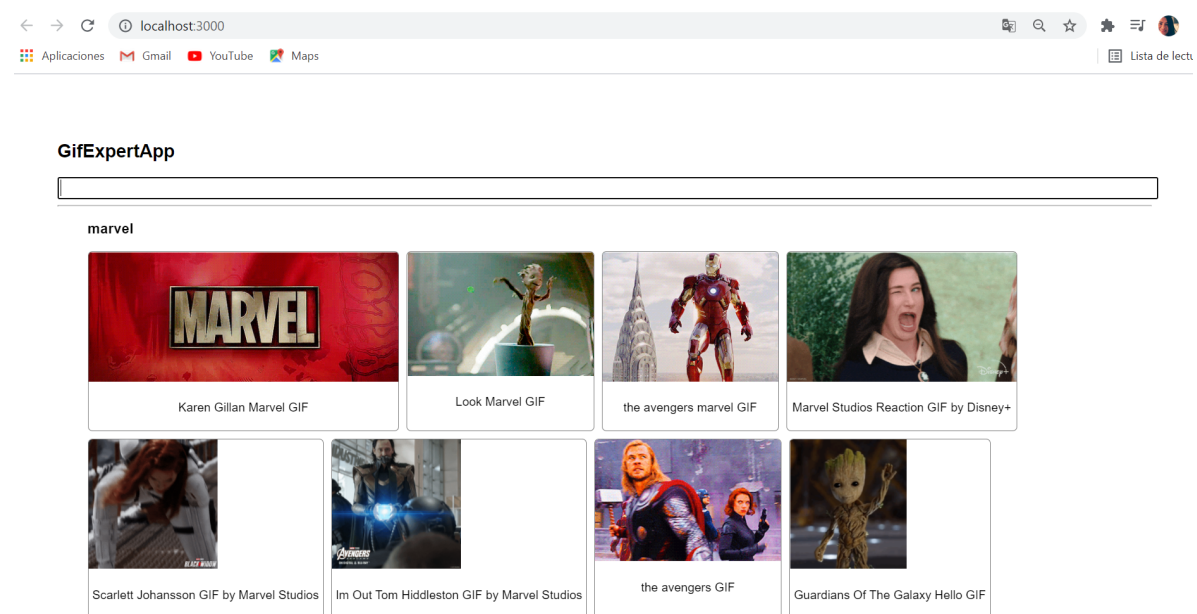
Descargamos la aplicación de Giphy Expert

<https://github.com/jennymontoya1001/react-app-giphy-testing.git>

La abrimos en visual studio code

Corremos la instrucción npm install para instalar los módulos de node

Corremos la instrucción npm start



Abrimos otra terminal

Instalamos enzyme

Docuemntación enzyme <https://enzymejs.github.io/enzyme/>

```
npm i --save-dev enzyme enzyme-adapter-react-16
```

Instalamos enzume to JSON

Documentación enzyme to json <https://www.npmjs.com/package/enzyme-to-json>

```
npm install --save-dev enzyme-to-json
```

Dentro de la carpeta src creamos el archivo que se llame setupTests.js dentro de este archivo debemos realizar la configuración de enzyme y enzyme to JSON

De la documentación de enzyme

```
import Enzyme from 'enzyme';
import Adapter from 'enzyme-adapter-react-16';

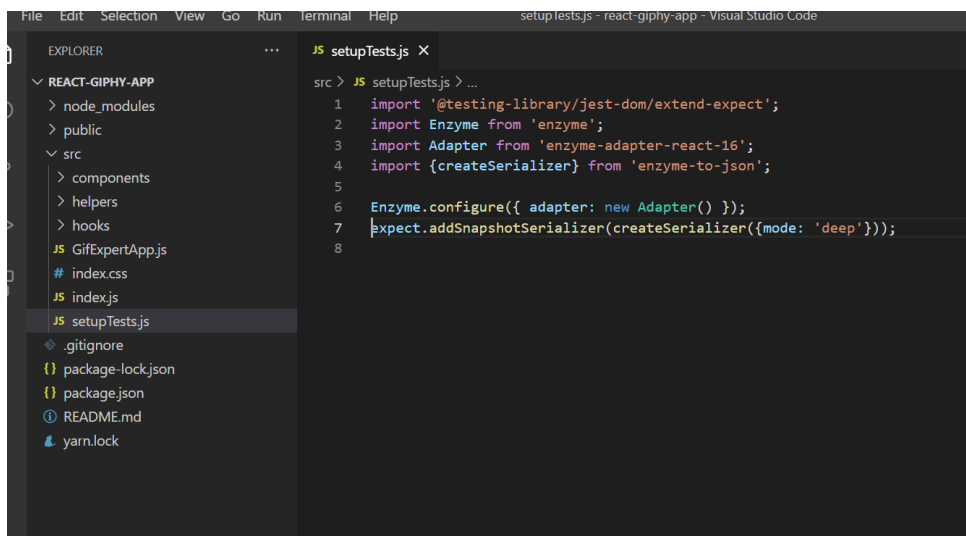
Enzyme.configure({ adapter: new Adapter() });
```

De la documentación de enzyme JSON

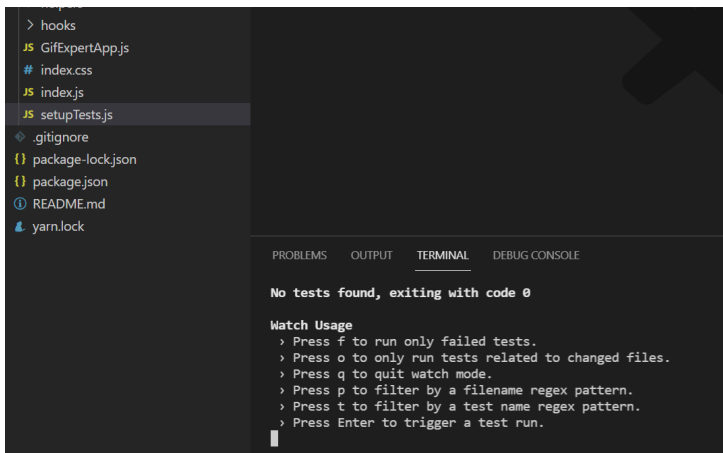
```
import {createSerializer} from 'enzyme-to-json';

expect.addSnapshotSerializer(createSerializer({mode:
'deep'}));
```

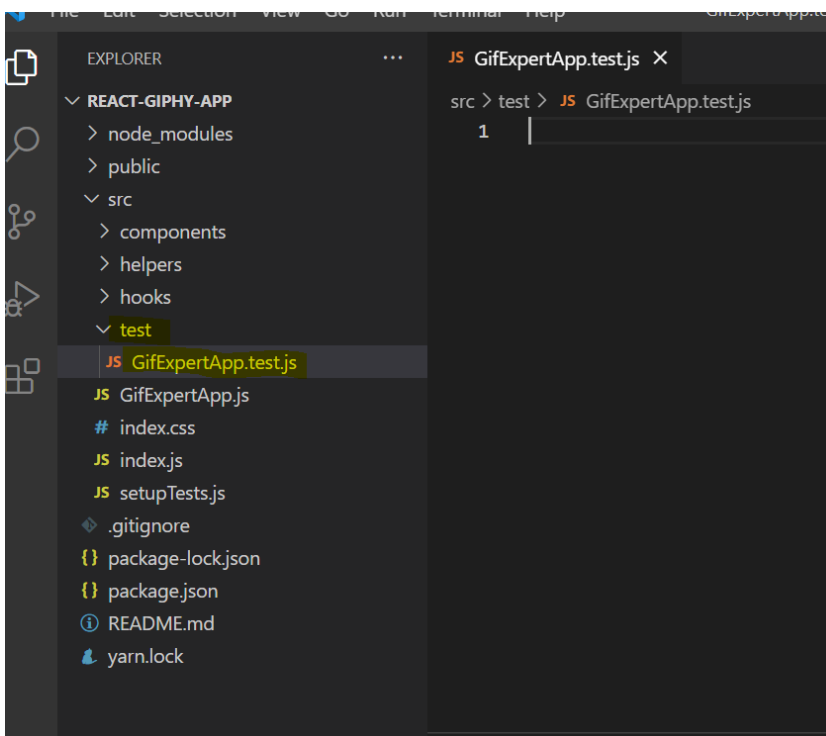
El archivo debe quedar de la siguiente manera



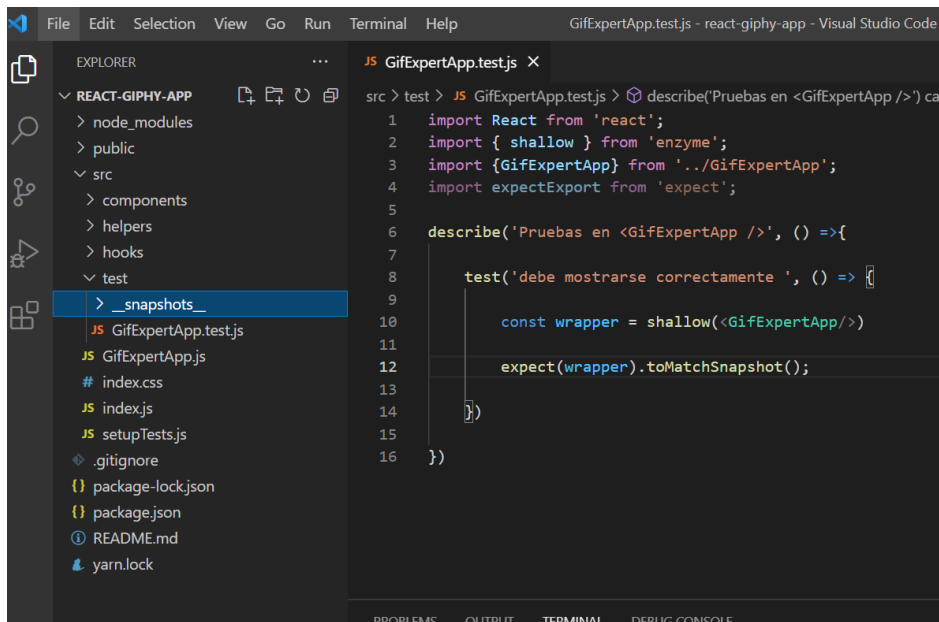
En nuestra terminal corremos la instrucción npm run test para asegurarnos que nuestro test suite esté pasando



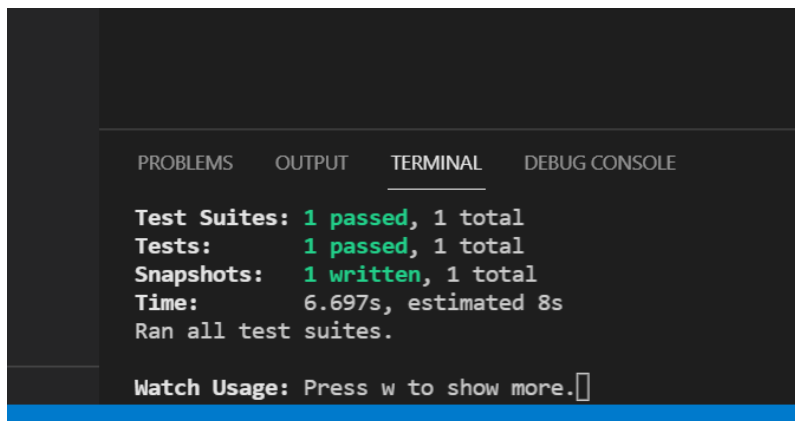
Dentro de src creamos una carpeta llamada test y dentro de ella un archivo llamado GifExpertApp.test.js



Vamos a hacer que el componente haga match con el snapshot

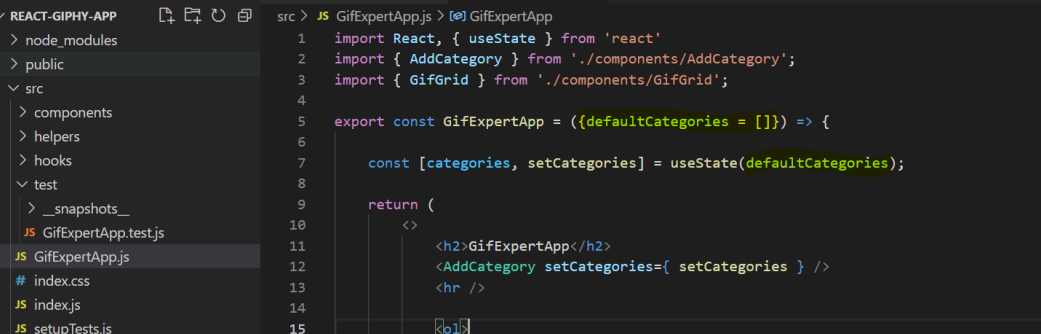


Las pruebas pasan



Vamos a hacer otra prueba para que nos muestre una lista de categorías, cómo nuestro componente no permite realizar esta acción ya que le estamos enviando una categoría, vamos a realizar el siguiente cambio. Lo pondremos a recibir un arreglo vacío de categorías

Dentro les archivo GifExpertApp.js



The image shows a Visual Studio Code editor window with a React application. The Explorer sidebar on the left displays the file structure of 'REACT-GIPHY-APP'. The main editor area shows the 'GifExpertApp.js' file with the following code:

```
import React, { useState } from 'react'
import { AddCategory } from '../components/AddCategory';
import { GifGrid } from '../components/GifGrid';

export const GifExpertApp = ({defaultCategories = []}) => {

  const [categories, setCategories] = useState(defaultCategories);

  return (

    <h2>GifExpertApp</h2>
    <AddCategory setCategories={ setCategories } />
    <hr />
    <ol>
      {
        categories.map( category => (
          <GifGrid
```

El snapshot falló porque ahora teníamos un elemento, y ya no tenemos nada

```

    <AddCategory
      setCategories={[Function]}
    />
    <hr />
    - <ol>
    -   <GifGrid
    -     category="One Punch"
    -     key="One Punch"
    -   />
    - </ol>
    + <ol />
  </Fragment>

10 |         const wrapper = shallow(<GifExpertApp/>)
11 |
> 12 |         expect(wrapper).toMatchSnapshot();
    |                             ^
13 |

```

Ahora realizaremos nuestra prueba

```
JS GifExpertApp.test.js X
src > test > JS GifExpertApp.test.js > describe('Pruebas en <GifExpertApp />') callback > test('debe mostrarse una lista de categorías ')
13
14   })
15
16   test('debe mostrarse una lista de categorías ', () => {
17     //le enviaremos dos categorías
18     const categories = ['One Punch', 'Dragon Ball'];
19     const wrapper = shallow(<GifExpertApp defaultCategories={categories}/>)
20
21     //Nos aseguramos de que todo esté igual con el toMatchSnapshot
22     expect(wrapper).toMatchSnapshot();
23     // Debemos validar que el wrapper en este caso tenga dos GifGrid correspondientes
24     //a las dos categorías
25     expect(wrapper.find('GifGrid').length).toBe(categories.length);
26   })
27
28
29 })
```

Presionamos u para actualizar el snapshot

Nuestras pruebas pasan

```
24 //a las dos categorías
25 expect(wrapper.find('GifGrid').length).toBe(categories.length);

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

✓ debe mostrarse correctamente (9ms)
✓ debe mostrarse una lista de categorías (4ms)

> 1 snapshot updated.
Snapshot Summary
> 1 snapshot updated from 1 test suite.

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:  1 updated, 1 passed, 2 total
Time:        4.1s
Ran all test suites.

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