**React**

**Pruebas en el archivo 08-imp-exp.js.**

Documentación: <https://es.reactjs.org/>

<https://es.reactjs.org/docs/getting-started.html>

<https://jestjs.io/>

<https://jestjs.io/docs/snapshot-testing>

<https://jestjs.io/docs/expect>

Al realizar 1 prueba podemos evaluar n cantidad de **expect** como queramos, ejemplo:

**Código de archivo heroes.js**

const heroes = [

    {

        id: 1,

        name: 'Batman',

        owner: 'DC'

    },

    {

        id: 2,

        name: 'Spiderman',

        owner: 'Marvel'

    },

];

export default heroes;

**Código de archivo 08-imp-exp**

import heroes from '../data/heroes';

export const getHeroeById = (id) => heroes.find( (heroe) => heroe.id === id );

export const getHeroesByOwner = ( owner ) => heroes.filter( (heroe) => heroe.owner === owner);

**Código de programación de pruebas**

import {describe, expect, test} from '@jest/globals';

import { getHeroeById, getHeroesByOwner } from '../../base-pruebas/08-imp-exp'

import heroes from '../../data/heroes';

describe('Pruebas en 08-imp-exp', () => {

    test('getHeroeById debe de retornar un objeto por id de heroe', () => {

        const HeroeId = 1;

        const Heroe = getHeroeById( HeroeId );

        const HeroeTest = {

            id: 1,

            name: 'Batman',

            owner: 'DC'

        };

        expect( Heroe ).toStrictEqual( HeroeTest );

    })

    test('getHeroeById debe de retornar undefined si no existe el id heroe', () => {

        const HeroeId = 100;

        const Heroe = getHeroeById( HeroeId );

        expect( Heroe ).toBe( undefined );

        // toBeFalsy es una forma corta de comparar contra null, undefined o false

        expect( Heroe ).toBeFalsy();

    })

    test('getHeroesByOwner debe de retornar heroes de DC y Marvel', () => {

        const HeroesDC = getHeroesByOwner("DC");

        const HeroesMarvel = getHeroesByOwner("Marvel");

        expect( HeroesDC.length ).toEqual(3)

        expect( HeroesDC.length ).toBe(3)

        expect( HeroesDC ).toEqual( heroes.filter( (heroe) => heroe.owner === "DC" ) )

        expect( HeroesMarvel.length ).toEqual(2)

        expect( HeroesMarvel.length ).toBe(2)

        expect( HeroesMarvel ).toEqual( heroes.filter( (heroe) => heroe.owner === "Marvel" ) )

    })

})