

lab-tdd

Introducción

En este laboratorio vamos a realizar una calculadora con la metodología TDD, en lenguaje javascript y con jest como framework para realizar las pruebas.

Lo primero que hay que hacer es instalar las siguientes dependencias:

- npm: Instalar node siguiendo lo descrito en esta página: <https://nodejs.org/es/download/>
- package.json: npm init -f; Si hay alguna duda consultar aquí: <https://www.npmjs.com/get-npm>
- jest: npm install --save-dev jest; Documentación oficial: <https://jestjs.io/es-ES/docs/getting-started>

Tras esto lanzar el comando npm test para ejecutar las pruebas.

Suma

Función utilizada para resolver la suma:

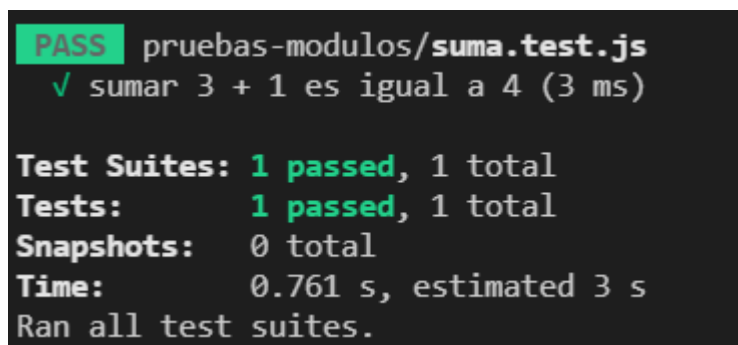
```
//Función suma que recibe dos números y retorna la suma de dichos números
function suma(a, b) {
  return a + b;
}
module.exports = suma;
```

Test utilizado para la suma:

```
//Requerimos la funcion que tenemos almacenada en modulo suma y hacemos el test dandole los valores correctos para la prueba
const suma = require('../calc-modulos/suma');

test('sumar 3 + 1 es igual a 4', () => {
  expect(suma(3, 1)).toBe(4);
});
```

Resultado del test suma:



```
PASS pruebas-modulos/suma.test.js
  ✓ sumar 3 + 1 es igual a 4 (3 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.761 s, estimated 3 s
Ran all test suites.
```

Resta

Función utilizada para resolver la resta:

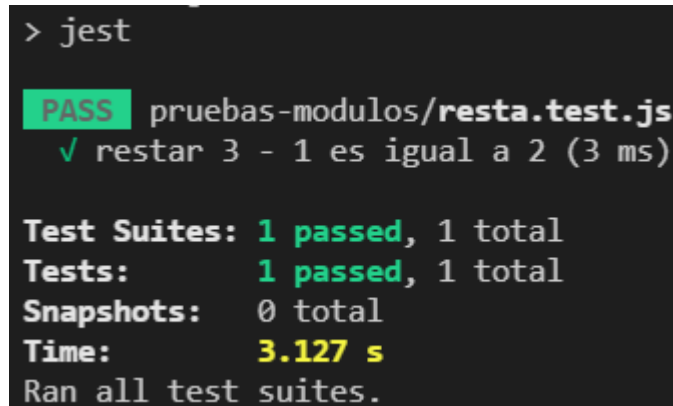
```
//Función resta que recibe dos números y retorna la resta de dichos números

function resta(a, b) {
  return a - b;
}
//Exporta el módulo resta
module.exports = resta;
```

Test utilizado para la resta:

```
//Requerimos la funcion que tenemos almacenada en modulo resta y hacemos el test dandole los valores correctos para la prueba
const resta = require('../calc-modulos/resta');
test('restar 3 - 1 es igual a 2', () => {
  expect(resta(3, 1)).toBe(2);
});
```

Resultado del test resta:



```
> jest

PASS pruebas-modulos/resta.test.js
  ✓ restar 3 - 1 es igual a 2 (3 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        3.127 s
Ran all test suites.
```

Multiplicación

Función utilizada para resolver la multiplicación:

```
//Función de la multiplicacion que recibe dos números y retorna la mult de dichos números

function mult(a, b) {
  return a * b;
}
//Exporta el módulo mult
module.exports = mult;
```

Test utilizado para la multiplicación:

```
//Requerimos la funcion que tenemos almacenada en modulo mult y hacemos el test dandole los valores correctos para la prueba
const mult = require('../calc-modulos/mult');

test('multiplicar 7 * 8 es igual a 56', () => {
  expect(mult(7, 8)).toBe(56);
});
```

Resultado del test multiplicación:

```
PASS lab-tdd/pruebas-modulos/mult.test.js
✓ multiplicar 7 * 8 (2 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.095 s
Ran all test suites matching /mult.test.js/i.
```

División

Función utilizada para resolver la división:

```
//Función division que recibe dos números y retorna la division de dichos números
function division(a, b) {
  return a / b;
}
module.exports = division;
```

Test utilizado para la división:

```
//Requerimos la funcion que tenemos almacenada en modulo division y hacemos el test dándole
los valores correctos para la prueba
const division = require('../calc-modulos/division');

test('dividir 6 entre 2 es igual a 3', () => {
  expect(division(6, 2)).toBe(3);
});
```

Resultado del test división:

```
> jest

PASS pruebas-modulos/division.test.js
✓ dividir 6 entre 2 es igual a 3 (2 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.856 s, estimated 3 s
Ran all test suites.
```

Raiz

Utilizamos esta página como referencia en la forma de realizar la raíz cuadrada: <https://wizardprogrammer.blogspot.com/2018/12/calcular-la-raiz-caudrada-sin-la-clase.html> que está en otros lenguajes pero lo transformamos en javascript y esta fórmula para resolverlo

$$x = \frac{1}{2} \left(x + \frac{a}{x} \right)$$

Función utilizada para resolver la raíz

```
//Función raíz que recibe dos números y retorna la raíz de dichos números
function raiz(a) {
  x= 1.0;
  for(k = 1; k < 10; k++){
    x = (x + a/x) / 2;
  }
  a = x;
  return a;
}
//Exporta el módulo raíz
module.exports = raiz;
```

Test utilizado para la raíz:

```
//Requerimos la funcion que tenemos almacenada en modulo raiz y hacemos el test dandole los valores correctos para la prueba
const raiz = require('../calc-modulos/raiz');

test('la raíz de 121 es igual a 11.0', () => {
  expect(raiz(121)).toBe(11.0);
});

test('la raíz de 8 es igual a 2.828', () => {
  expect(raiz(8)).toBe(2.82842712474619);
});
```

Resultado del test raíz:

```
> jest

PASS pruebas-modulos/raiz.test.js
  ✓ la raíz de 121 es igual a 11.0 (3 ms)
  ✓ la raíz de 8 es igual a 2.828 (1 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.088 s, estimated 3 s
Ran all test suites.
```

Resultado del test con todas las operaciones en el main:

```
> jest

PASS pruebas-modulos/raiz.test.js
PASS pruebas-modulos/mult.test.js
PASS pruebas-modulos/division.test.js
PASS pruebas-modulos/suma.test.js
PASS pruebas-modulos/resta.test.js

Test Suites: 5 passed, 5 total
Tests:       6 passed, 6 total
Snapshots:   0 total
Time:        2.325 s
Ran all test suites.
```