

# JEST

TESTING

COVERAGE

MOCKS

# TESTING

## ¿POR QUÉ ES IMPORTANTE EL TESTING?

- Ya lo hablamos en el modulo de buenas practices
- No se dará por Buena ninguna solución que no tenga el 80% del coverage de test
- Framework de test: **JEST**

# TESTING

## EJEMPLO PARA HACER

- ¿Qué podemos testear aquí? Debate

```
function area_circle(radius)
{
    return Math.PI*(radius**2);
}
```

# TESTING

## EJEMPLO PARA HACER

- Fichero con el mismo nombre + .test.js
- Importamos modulo/s

```
const area_circle = require("../5_0_examples");
```

Run | Debug

```
describe("area_circle()", () => {
```

Run | Debug

```
  test("calculates area correctly", () => {  
    expect(area_circle(1)).toBe(Math.PI);  
  });  
});
```

# TESTING

## EJEMPLO PARA HACER

- *npm run test*
- *jest (en el propio directorio)*

```
PS C:\Users\David\Dropbox\albaniles_digitales\core_javascript> npm test -- './6 JEST/scripts/5_0_examples.test.js'
Debugger attached.

> test
> jest ".\\6 JEST\\scripts\\5_0_examples.test.js"

Debugger attached.
PASS 6 JEST/scripts/5_0_examples.test.js
  area_circle()
    ✓ calculates area correctly (1 ms)

Test Suites: 1 passed, 1 total
Tests:      1 passed, 1 total
Snapshots:  0 total
Time:       0.546 s, estimated 1 s
Ran all test suites matching /.\\6 JEST\\scripts\\5_0_examples.test.js/i.
Waiting for the debugger to disconnect...
Waiting for the debugger to disconnect...
```

# TESTING

## NOTA DE REPASO

- Titula el test case adecuadamente para saber qué estás testeando
- Comprueba y piensa las precondiciones (cómo llega el Código a ese punto)
- Mantén el test sencillo, si es complicado, señal de refactorización
- Incluye el resultado esperado
- Incluye casos extremos

# TESTING

## REPASO EN NUESTRO EJEMPLO

- ¿qué pasa si llega un radio que no es un número?
- Radio 0. Radio infinito
- Ejemplos normales

Run | Debug

```
test("calculates area correctly", () => {  
  expect(area_circle(1)).toBe(Math.PI);  
  expect(area_circle(2)).toBe(4*Math.PI);  
  expect(area_circle(0)).toBe(0);  
  expect(area_circle(Infinity)).toBe(Infinity);  
});
```

# COVERAGE

## HERRAMIENTA

- Sirve para ver todas las líneas de Código que ejecutan tus tests
- ¡OJO! No implica que se testeen unitariamente los códigos.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	93.33	100	87.5	92.85	
scripts	66.66	100	50	66.66	
5_0_examples.js	66.66	100	50	66.66	4
scripts/testing/testing/car	100	100	100	100	
car.js	100	100	100	100	



# MOCKS

## SUSTITUCIONES

- No queremos testear siempre las funciones internas
- Se alargaría en el tiempo

# MOCKS

## EJEMPLO

```
function super_area_circle(radius)
{
  let area = 0;
  if (examples_utils.is_radio(radius))
  {
    area = Math.PI*(radius**2);
  }
  else
  {
    area = 21;
  }
  return area;
}
```

```
function is_radio(radius)
{
  console.log("inside radio");
  let output = true;
  if (radius == 0)
  {
    output = false;
  }
  console.log("output is " + output);
  return output;
}

module.exports = {
  ...
  is_radio,
};
```

# MOCKS

## EJEMPLO

Run | Debug

```
describe("super_area_circle()", () => {
```

Run | Debug

```
test("calculates area finger way", () => {
```

```
    expect(examples.super_area_circle(1)).toBe(Math.PI);
```

```
    jest.spyOn(examples_utils, "is_radio").mockReturnValue(false);
```

```
    expect(examples.super_area_circle(1)).toBe(21);
```

```
});
```

```
});
```

# MOCKS

## GUÍA Y DIRECTRICES

- <https://jestjs.io/docs/mock-function-api#mockfnmockreturnvaluevalue>

# BIBLIOGRAFÍA

1. JavaScript: The Definitive Guide, 7<sup>th</sup> Edition
2. Eloquent JavaScript, 3th edition, Marijn Haverbeke
3. Clean Code: A Handbook of Agile Software Craftsmanship (Robert C. Martin Series)
4. Modern c++ Programming with Test-Driven Development, Jeff Langr
5. Refactoring: Improve the design of existing Code, Martin Fowler
6. Game programming patterns, Robert Nystrom