




PRACTICA 8 GIT - HUNDIR LA FLOTA

IES GALILEO
DAW II

MAZILU, COSMIN NICOLAE Y PUERTA GARCÍA, JORGE
DWN
2º DAW



Índice

1. Realización y seguimiento	pág. 2
2. Testing	pág. 4
3. Conclusiones	pág. 7

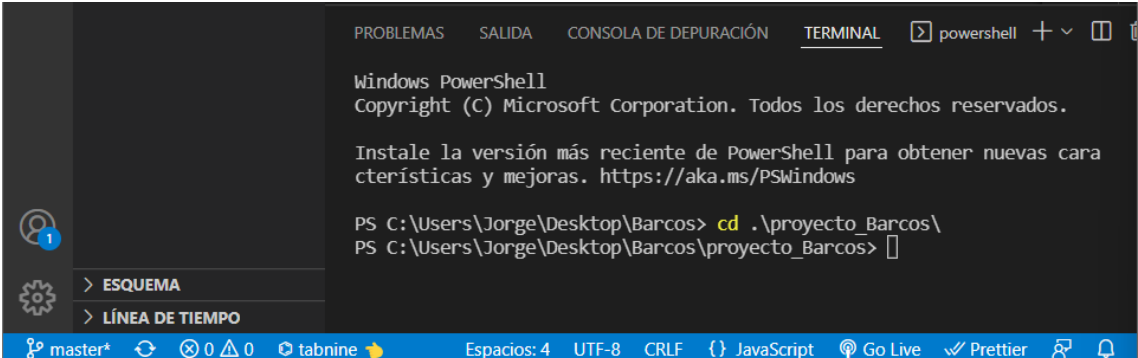
1. Realización y seguimiento

En el ejercicio, se ha dividido el trabajo en distintas partes:

- **Jorge:** testing, documentación, html y css
- **Cosmin:** JavaScript

A pesar de la limitación de las áreas de trabajo, ambas partes han colaborado en la lógica de programación como en el diseño. Es decir, ambos hemos intervenido en todos los procesos.

El seguimiento de la actividad puede visualizarse a través del proyecto alojado en *github*, así como desde la documentación¹² generada con *jsDoc*. Su instalación se realiza a través de la terminal del entorno de desarrollo (*VisualStudioCode*) y nos posicionamos en el directorio del proyecto:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\Jorge\Desktop\Barcos> cd .\proyecto_Barcos\
PS C:\Users\Jorge\Desktop\Barcos\proyecto_Barcos>
```

Se deberá instalar anteriormente el *npm* y el *node.js*. Posteriormente, le añadimos el comando *–save-dev jsdoc* para crear un repositorio

¹ <https://www.klloid.com/blog/jsdoc-series-i-instalacion-y-configuracion/>

² <https://www.npmjs.com/package/jsdoc>

```
PROBLEMAS  SALIDA  TERMINAL  ...  powershell  +  -  [ ]  [ ]  ^  x

PS C:\Users\Jorge\Desktop\Barcos> npm install --save-dev jsdoc
npm WARN idealTree Removing dependencies.jsdoc in favor of devDependencies.jsdoc

up to date, audited 26 packages in 2s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Jorge\Desktop\Barcos> [ ]
```

Tendremos que cambiar la configuración del archivo *package.json* e insertar la siguiente línea donde le decimos en que archivo crearemos el *jsdoc.json*

```
{ "scripts": {
  "docs": "jsdoc -c jsdoc.json"
},
```

Dentro del archivo creado de *jsdoc.json*, le introducimos los siguientes parámetros:

```
{
  "source": { ← fuente donde va a tomar la información
    "include": ["../Proyecto_Barcos/script.js"], ← fichero donde queremos sacar la
información
    "includePattern": ".+\\.js(doc|x)?$", ← Patron para que recoja los comentarios
    "exclude": "node_modules/" ← Para evitar que nos coja la carpeta que se ha generado
con el node.js
  },
  "opts": {
    "destination": "JsDoc/" ← En que carpeta queremos que se guarde el jsdoc
  }
}
```

Inicializamos el archivo *docs* a través del comando *npm run docs*

```
PS C:\Users\Jorge\Desktop\Barcos\JsDoc> npm run docs
> docs
> jsdoc -c jsdoc.json
PS C:\Users\Jorge\Desktop\Barcos\JsDoc> 
```

2. Testing³

En el proceso de testing, hemos optado por **Mocha** y la librería **Chai**, ya que tiene una gran versatilidad y está mejor documentada en internet. Al no estar realizado con clases y sí con funciones, la comprobación dentro de las funciones **void** la realizamos a través de booleans para pautar la función y comprobar su funcionamiento.

Instalamos⁴ Mocha como en el proceso anterior, a través del **npm install mocha** y **Chai** que es su librería y la creación de repositorio con **npm install --save-dev mocha**:

```
PROBLEMAS  SALIDA  TERMINAL  ...
> powershell + v [ ] [ ] ^ x

PS C:\Users\Jorge\Desktop\Barcos\JsDoc> npm install --save-dev mocha
npm WARN idealTree Removing dependencies.mocha in favor of devDependencies.mocha

up to date, audited 103 packages in 845ms

21 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Jorge\Desktop\Barcos\JsDoc> 
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL
+ v ^ x

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\Jorge\Desktop\Barcos> npm install chai

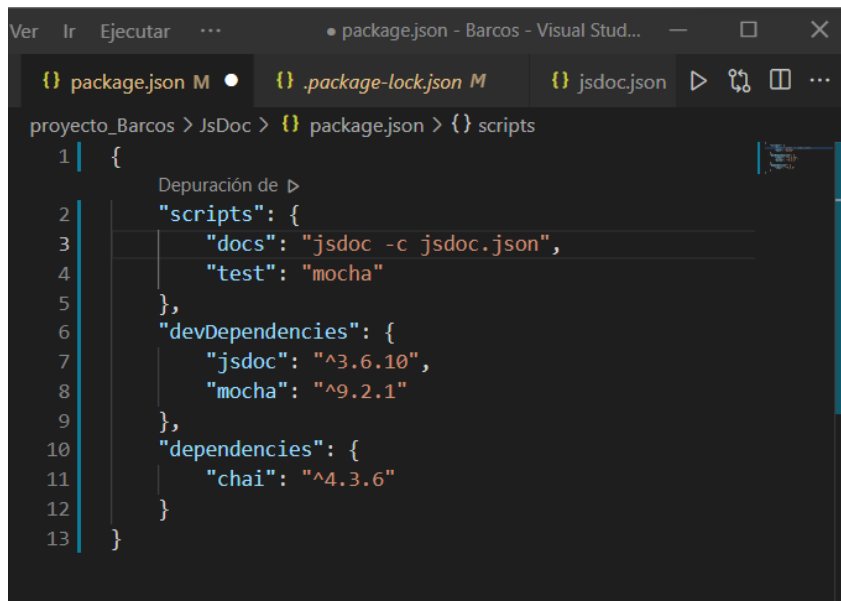
added 8 packages, removed 25 packages, and audited 9 packages in 4s

found 0 vulnerabilities
PS C:\Users\Jorge\Desktop\Barcos> 
```

³ <https://programmerclick.com/article/4123194090/>

⁴ Importante que la terminal sea en el documento donde tengamos el **package.json**

Dentro de nuestro *package.json*, añadimos en scripts a Mocha como programa para realizar los test:



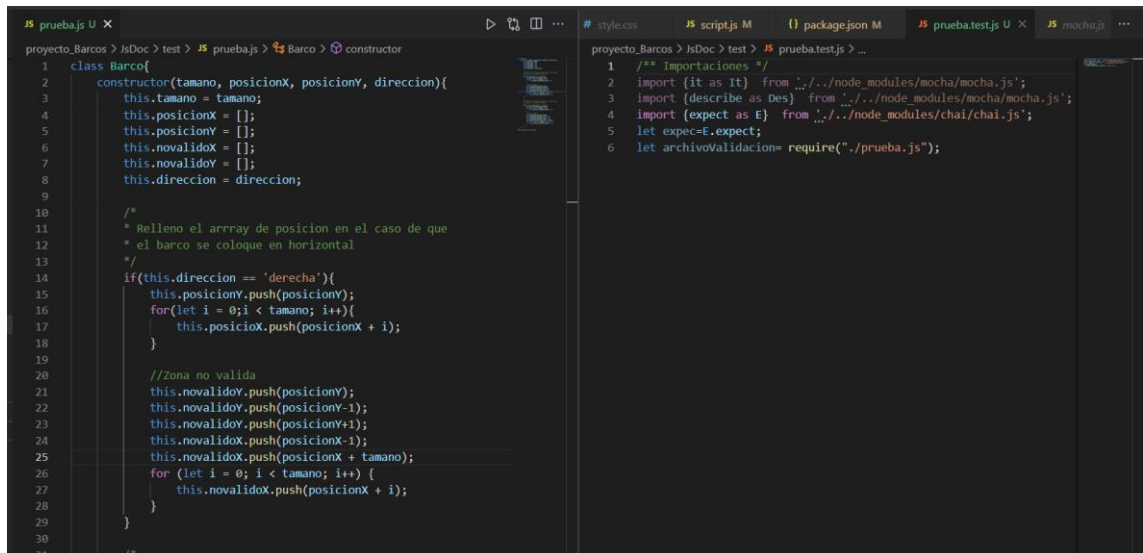
```
1 {
2   Depuración de >
3   "scripts": {
4     "docs": "jsdoc -c jsdoc.json",
5     "test": "mocha"
6   },
7   "devDependencies": {
8     "jsdoc": "^3.6.10",
9     "mocha": "^9.2.1"
10  },
11  "dependencies": {
12    "chai": "^4.3.6"
13  }
14 }
```

Crearemos la carpeta⁵ “test” donde alojaremos el código a testear. Como nuestro proyecto está alojado en un solo fichero JavaScript, importaremos ese fichero a una clase que la llamaremos *prueba.js*. En esta clase, clonaremos lo que tenemos y queremos que vaya a ser exportado para a otro archivo JavaScript de prueba. Para hacer un ejemplo, utilizaremos la clase barco creada que nos resultará más intuitivo⁶.

Para hacer el test, necesitamos ahora crear el archivo prueba.test.js para que corra sobre el mocha. En la nueva clase, importaremos de mocha.js y de chai.js las funciones *it*, *describe* y *expect* así como la clase del fichero test.js y para más funcionalidades el *script.js* general.

⁵ Importante: por tema de haber insertado el mocha en dependencias del fichero JsDoc, tendrá que estar la carpeta test dentro de ella

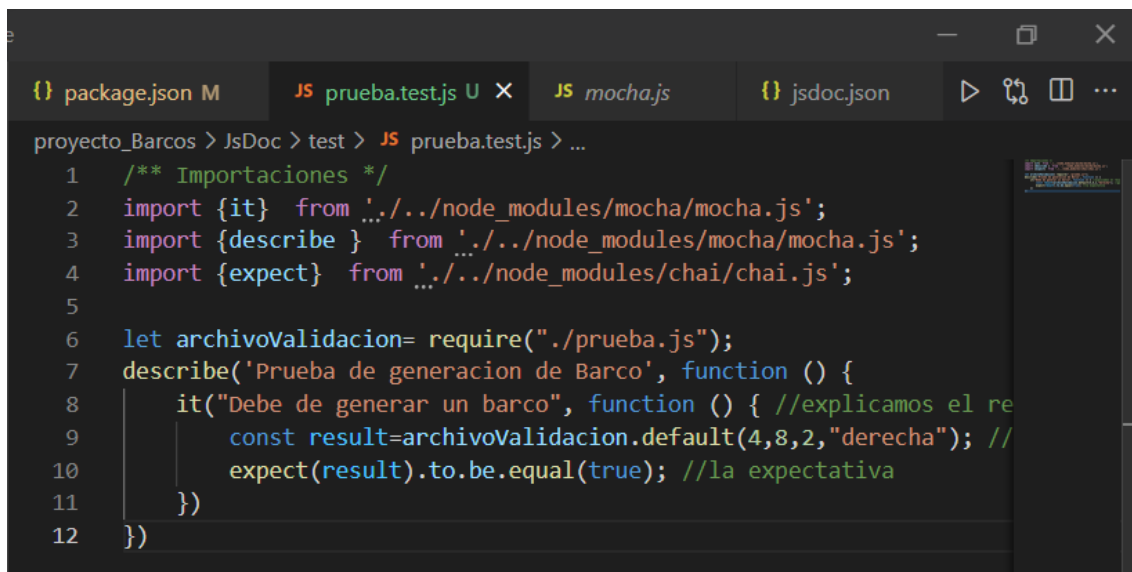
⁶ No se aportan pruebas gráficas porque se ha copiado el código y se ha puesto la clase de forma que pueda ser importada, salvo que se ha añadido un return true para comprobar su funcionamiento



```
1 class Barco {
2   constructor(tamano, posicionX, posicionY, direccion) {
3     this.tamano = tamano;
4     this.posicionX = [];
5     this.posicionY = [];
6     this.novalidoX = [];
7     this.novalidoY = [];
8     this.direccion = direccion;
9
10    /*
11     * Relleno el array de posicion en el caso de que
12     * el barco se coloque en horizontal
13     */
14    if(this.direccion == 'derecha'){
15      this.posicionY.push(posicionY);
16      for(let i = 0; i < tamano; i++){
17        this.posicionX.push(posicionX + i);
18      }
19
20      //Zona no valida
21      this.novalidoY.push(posicionY);
22      this.novalidoY.push(posicionY-1);
23      this.novalidoY.push(posicionY+1);
24      this.novalidoX.push(posicionX-1);
25      this.novalidoX.push(posicionX + tamano);
26      for (let i = 0; i < tamano; i++) {
27        this.novalidoX.push(posicionX + i);
28      }
29    }
30  }
31}
```

```
1 /** Importaciones */
2 import {it as It} from '../node_modules/mocha/mocha.js';
3 import {describe as Des} from '../node_modules/mocha/mocha.js';
4 import {expect as E} from '../node_modules/chai/chai.js';
5 let expect=E.expect;
6 let archivoValidacion= require("../prueba.js");
```

Dentro del describe, se encabeza el titulo de la prueba y a la función donde se realiza, en el it se explica cual es la opción correcta y declaramos el constante resultado con la prueba y en el **expect** se especifica el resultado correcto.



```
1 /** Importaciones */
2 import {it} from '../node_modules/mocha/mocha.js';
3 import {describe} from '../node_modules/mocha/mocha.js';
4 import {expect} from '../node_modules/chai/chai.js';
5
6 let archivoValidacion= require("../prueba.js");
7 describe('Prueba de generacion de Barco', function () {
8   it("Debe de generar un barco", function () { //explicamos el re
9     const result=archivoValidacion.default(4,8,2,"derecha"); //
10     expect(result).to.be.equal(true); //la expectativa
11   })
12 })
```

Ahora abrimos la terminal e introducimos el comando *npm test*:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

> test
> mocha

Prueba de generacion de Barco
  1) Debe de generar un barco

0 passing (6ms)
1 failing

1) Prueba de generacion de Barco
  Debe de generar un barco:
    TypeError: archivoValidacion.default is not a function
    at Context.<anonymous> (test\prueba.test.js:8:47)
    at processImmediate (node:internal/timers:466:21)

PS C:\Users\Jorge\Desktop\Barcos\proyecto Barcos\JsDoc> █
```

3. Conclusiones

La mayor dificultad ha sido el planteamiento de la mecánica del juego en el lenguaje de JavaScript y los plazos de entrega. Aparte de ello, Cosmin experimentó diversos problemas al tener múltiples usuarios de git, por lo que sus *commits* se retrasaron en la línea temporal del trabajo.

Como se ha comentado en el apartado 1, la división del trabajo ha sido exitosa ya que nos hemos intercambiado dudas y solucionados problemas tanto dentro de nuestra delimitación como la ajena, en resumen, una relación simbiótica.

En cuanto a la implantación de herramientas de Git y GitHub, hemos comprobado su gran funcionalidad al compatibilizar nuestro trabajo en remoto. Tanto los plugins JsDoc, Mocha y Chai, son bastante intuitivos y prácticos.

En conclusión, se han trabajado las herramientas solicitadas y se han implementado de forma exitosa a pesar de que no se haya finalizado el trabajo en su totalidad.