

## DOCUMENTACIÓN

1. INTRODUCCIÓN.....	1
2. OBJETIVOS PRINCIPALES DE LA DOCUMENTACIÓN:.....	1
3. ¿QUÉ DEBEMOS DOCUMENTAR ?.....	1
4. TIPOS DE COMENTARIOS:.....	1
5. JAVADOC.....	2
6. GENERAR LA DOCUMENTACIÓN.....	4
7. CASOS PRÁCTICOS.....	5

### 1. INTRODUCCIÓN

La documentación no hará que nuestro programa funcione mejor o peor, o que optimice el uso de recursos y sea más eficaz. Sin embargo, nos facilitará nuestra tarea como programador en el futuro.

En cualquier aplicación que sea utilizada por una serie de usuarios, más tarde o más pronto serán encontrados algunos errores de funcionamiento que no se habían tenido en cuenta.

Además si el programa tiene éxito, seguro que se le añadirán funcionalidades y se modificarán otras.

En un programa **bien documentado será mucho más fácil reparar errores y añadirle nuevas funcionalidades para adaptarlo a nuevos escenarios.**

[https://drive.google.com/file/d/1IvviQlucso5BQOZ3p8SqNAU0nYwYw\\_46/view?usp=sharing](https://drive.google.com/file/d/1IvviQlucso5BQOZ3p8SqNAU0nYwYw_46/view?usp=sharing)

### 2. OBJETIVOS PRINCIPALES DE LA DOCUMENTACIÓN:

- Facilidad para resolver errores de la aplicación.
- Facilidad para añadir nuevas funcionalidades futuras.

### 3. ¿QUÉ DEBEMOS DOCUMENTAR ?

- Qué hace una clase, un método o una variable y para qué se utiliza.
- Cuál es el uso correcto para un método o una variable.
- En los métodos, documentar los parámetros de la función así como el valor de retorno.
- Qué algoritmo se utiliza para resolver un problema.
- Qué deberíamos refactorizar en el futuro.

### 4. TIPOS DE COMENTARIOS:

– **Comentarios de una línea:**

//Esto es un comentario de una línea

– **Comentarios de varias líneas:**

/\*

\* Línea 1 de un comentario de varias líneas

\* Línea 2 de un comentario de varias líneas

\* ...

\* Línea N de un comentario de varias líneas

\*

\*/

Este tipo de comentarios se utilizan (y debes continuar utilizándolos) para aquellos proyectos que posteriormente no van a requerir generar la documentación asociada.

## 5. JAVADOC

- Es una herramienta para documentar proyectos Java
- Está integrada en Java (no necesita instalación extra)
- Puede utilizarse en cualquier entorno de desarrollo integrado, como, en Eclipse
- Obtiene una **página web** idéntica a la que utiliza Oracle para la documentación de todas las clases Java. <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/ArrayList.html>
- Permite extraer y generar documentación directamente desde el código fuente y trasformarlo en formato HTML.

**Debemos escribir los comentarios del código según las recomendaciones de Javadoc:**

– Los comentarios deberán incluirse entre “/\*\*” y “\*/”

/\*\*

\* **Función que determina si un número es par o no**

\*/

– Generalmente, ocuparán varias líneas.

– Deben ser insertados antes de la declaración de una clase, un atributo o un método.

– Dentro de los limitadores de los comentarios se puede escribir código HTML.

– Los comentarios Javadoc están formados por dos partes, una documentación seguida de un bloque de tags (etiquetas) con diferente información.

– Las **etiquetas de Javadoc** van seguidas del carácter @.

– Se pueden usar tags para documentar la versión de la clase, el autor, los parámetros utilizados o los valores devueltos, entre otros.

ETIQUETA	DESCRIPCIÓN
@autor	Autor de la clase. Solo para las clases.
@versión	Versión de la clase. Solo para las clases.
@see	Referencia a otra clase, ya sea de API, del mismo proyecto o de otro. Por ejemplo: @see cadena, @see paquete.clase#miembro, @see enlace.
@param	Descripción del parámetro. Una etiqueta por cada parámetro.
@return	Descripción de lo que devuelve. Solo si no es void. Podrá describir valores de retorno especiales según las condiciones que se den, dependiendo del tipo de dato.
@throws	Descripción de la excepción que puede propagar. Habrá una etiqueta throws por cada tipo de excepción.
@deprecated	Marca el método como obsoleto. Solo se mantiene por compatibilidad.
@since	Indica el número de versión desde la que existe el método.

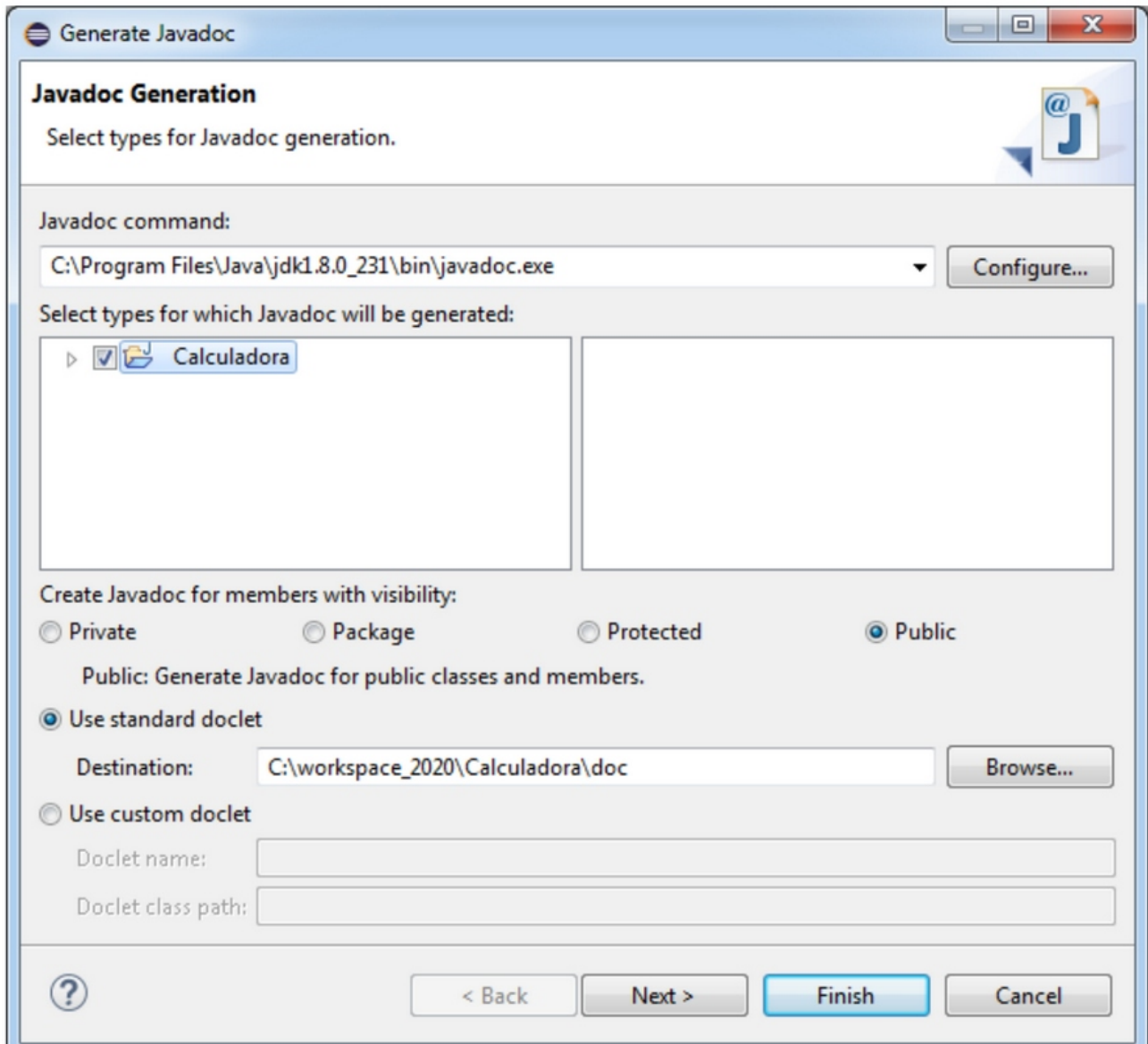
**EJEMPLO** de documentación Javadoc:

Carta.java

```
1 package Barajas;
2 import Utilidades.Idioma;
3
4 /**
5  * <h2>Clase Carta</h2>
6  * Se utilizará para representar objetos de tipo <b>Carta</b>
7  * Los atributos de la clase serán:
8  * <ul>
9  *   <li>Numero</li>
10 *   <li>Palo</li>
11 * </ul>
12 *
13 * Para información de JavaDoc...
14 * @see <a href="http://www.google.es">Google</a>
15 *
16 * @autor Jose Chamorro
17 * @version 1.0 - 2017
18 *
19 * */
20
21 public class Carta {
22
23     //Atributos
24     private int numero;
25     private int palo;
26
27     private Idioma idioma;
28
29     /**
30      * Constructor con 3 parámetros
31      * Crea una carta con el número y palo indicado
32      */
33 }
```

## 6. GENERAR LA DOCUMENTACIÓN

Para ejecutarlo desde Eclipse: **menú Project > Generate Javadoc**



- **Javadoc command:**
  - Indica dónde se encuentra el **archivo ejecutable de Javadoc** (javadoc.exe en la imagen del ejemplo con un SO Windows).
  - Desde el botón Configure puedes buscarlo en la carpeta donde está instalado el JDK.
- **Elegir el proyecto y las clases a documentar**
- **Visibilidad:**
  - Selecciona la visibilidad de los elementos a documentar.
  - Si escoges Public, solo se documentarán los elementos públicos.
  - Si escoges Private, se documentarán los elementos públicos, privados y protegidos.
- **Carpeta destino:** Donde se almacenará el código HTML.
- **Título:** Título del documento HTML que se genera (lo podrás configurar en la ventana siguiente al pulsar el botón Next).

Para abrir la documentación generada debes abrir el archivo **index.html** que encontrarás en la carpeta **doc** dentro de tu proyecto (si no has indicado lo contrario).

## 7. CASOS PRÁCTICOS

### 1: CALCULADORA

```
public class Calculadora {  
    public boolean esPar(int a) {  
        return a % 2 == 0;  
    }  
    public int suma(int a, int b){  
        return a+ b;  
    }  
}
```

Documenta la clase y los métodos con **Javadoc**. Introduce **/\*\*** y pulsa enter antes de la clase y los métodos

Genera la documentación **Javadoc** en una web en formato HTML: genera la página web de forma automática desde Eclipse.

```
/**  
 *  
 * @author usuario  
 * @version 1.0  
 * @since marzo 2023  
 * <h1>clase calculadora</h1>  
 */  
public class Calaculadora {  
    /**  
 * <h2>metodo para para comprobar si un numero es par</h2>  
 * @param a  
 * @return  
 */  
    public boolean esPar(int a) {  
        return a % 2 == 0;  
    }  
    /**  
 *  
 * @param a  
 * @param b  
 * @return  
 */  
    public int suma(int a, int b){  
        return a+ b;  
    }  
}
```

### 2: CAJA REGISTRADORA

<https://drive.google.com/file/d/1GFVF5Rw2sbhQX9lIT00ujH8itp96TlRI/view?usp=sharing>

A partir de la siguiente clase Java, que simula el uso de una caja registradora, **realiza la documentación Javadoc y genere la página web con dicha documentación.**