

## Documentació del codi font amb JavaDoc

JavaDoc és l'utilitat de Java per extreure i generar documentació directament del codi en format HTML. Perquè la documentació sigui útil hem d'escriure els comentaris del codi d'acord amb les recomanacions de JavaDoc.

La documentació i el codi estaran inclosos al mateix fitxer.

Recomanacions sobre els comentaris i la documentació del codi font:

- **Comentaris de línia:** Comencen amb els caràcters `//` i terminen amb la línia.
- **Comentaris de diverses línies:** Comencen amb els caràcters `/*` i terminen amb els caràcters `*/`.
- **Comentaris de documentació JavaDoc:** Comentaris que comencen amb `/**` i acaben amb `*/`. Contenen diverses línies i cada línia comença amb el caràcter `*`. Aquests comentaris cal col·locar-los abans de la declaració d'una classe, un camp, un mètode o un constructor.. Dins d'aquests delimitadors es poden escriure etiquetes HTML. Els comentaris Javadoc estan formats per dues parts: descripció + bloc de tags.



Més documentació a

[http://www.javahispano.org/antiguo\\_javahispano\\_org/2001/11/1/documentar-con-javadoc.html](http://www.javahispano.org/antiguo_javahispano_org/2001/11/1/documentar-con-javadoc.html)

<https://www.discoduroderoer.es/como-utilizar-javadoc/>

Exemple de comentari JavaDoc

```
/**
 * <h2>clase Empleado, se utiliza para crear y leer empleados de una BD </h2>
 *
 * Busca información de Javadoc en <a href="http://www.google.com">GOOGLE</a>
 * @see <a href="http://www.google.com"b >Google</a>
 * @version 1-2017
 * @author profe
 * @since 1-1-2017
 */
```

 Institut Sabadell	<b>CFGS DAM M05-UF2</b> Optimització del programari Documentació amb JavaDoc	
--	--	---

## Etiquetes (tags) de documentació JavaDoc:

Les etiquetes en JavaDocs comencen pel caràcter @. Les que més es fan servir són:

**@author:** Autor de la classe. Únicament per documentar la classe.

**@version:** Versió de la classe. Únicament per a classes.

**@see:** Referència a altra classe, ja sigui de l'API, del mateix projecte o d'altre projecte.

**@param:** Descripció del paràmetre. Una etiqueta per a cada paràmetre.

**@throws:** Descripció de l'excepció que pot propagar. Hi haurà una etiqueta per a cada tipus d'excepció

**@deprecated:** Marca el mètode com a obsolet. Únicament es manté per compatibilitat.

**@since:** Indica el número de versió des de la qual existeix el mètode

Exemple:

```
package javadoc;
```

```
/**
 * <h2>clase Empleado, se utiliza para crear y leer empleados de una BD </h2>
 *
 * Busca información de Javadoc en <a href="http://www.google.com">GOOGLE</a>
 * @see <a href="http://www.google.com">Google</a>
 * @version 1-2017
 * @author epozo
 * @since 1-1-2017
 */
```

```
public class Empleado {
    /**
     * Atributo Nombre del empleado
     */
    private String nombre;
    /**
     * Atributo Apellido del empleado
     */
    private String apellido;
    /**
     * Atributo Salario del empleado
     */
    private double salario;
```

```
/**
 * Constructor con tres parámetros
 * Crea objetos empleado con nombre, apellido y salario
 * @param nombre Nombre del empleado
 * @param apellido Apellido del empleado
 * @param salario Salario del empleado
 */
public Empleado(String nombre, String apellido, double salario) {
    this.nombre = nombre;
    this.apellido = apellido;
    this.salario = salario;
}

// Métodos públicos

/**
 * Sube el salario al empleado
 * @see Empleado
 * @param subida Valor de la subida del salario
 */
public void subidaSalario (double subida) {
    salario = salario + subida;
}

//Métodos privados

/**
 * Comprueba que el nombre no esté vacío
 * @return <ul>
 *     <li>true: el nombre es una cadena vacía</li>
 *     <li>false: el nombre No es una cadena vacía</li>
 * </ul>
 */
private boolean comprobar() {
    if(nombre.equals("")) {
        return false;
    }
    else
```

```

        return true;
    }
}

```

### Generar la documentació:

Una vegada hem posat els comentaris Javadoc al nostre codi, podem activar la generació de la documentació amb Javadoc i que la mostri en forma de pàgina web.

Realitza les següents tasques:

1. Crea un nou projecte en Eclipse i dins, crea una classe amb el codi anterior. Fixa't com es diu la classe i el package on està ficada. Pots canviar els noms si vols. Aquesta classe està preparada per generar la documentació amb Javadocs.
2. Executa Javadocs:  
Menú: Project - Generar Javadoc

*(llegeix el punt següent abans de donar al Finish perquè sàpigues què significa cada opció de la pantalla que s'obre!!)*

3. S'obre una finestra on es demana:
  - a. Javadoc command: s'indica on es troba l'arxiu executable de Javadoc (javadoc.exe). En configurar selecciona la carpeta dins del JDK en bin (està a C/ Archivos de programa/ Java/ jdk...../ bin/javadoc.exe).(a les noves versions ja el posa l'Eclipse si no dius el contrari) (en algunes màquines d'aula el jdk està en la carpeta d'Android , glup!!)
  - b. Als quadres inferiors s'escull el projecte i les classes a documentar.
  - c. Se selecciona la visibilitat dels elements que es volen documentar. Si posem Private es documenten tots els membres públics, privats i protegits.
  - d. S'indica la carpeta destí on es guarda el codi HTML. Es recomana canviar el nom que posa per defecte doncs amb el nom de doc a vegades no carrega correctament els estils.
4. Es fa click a Finish i a continuació permet canviar la ruta on es guardarà. Pots deixar la que proposa.
5. (versions anteriors) Ens demana les opcions que volem per la generació del document HTML: marquem barra de navegació i índex.  
(versió actual) Es veu com es genera el javadoc que es guarda en una nova carpeta dins del teu projecte que es dirà amb el nom que tu hakis posat (o "doc" si no l'has canviat tu).

Ara obre la carpeta que s'ha generat i vés visualitzant els diversos html que s'han creat: els diversos índex, Empleado.html.

No preocupis si encara no entens tot (hi ha coses de POO que encara no hem vist)

Verifica si s'han interpretat correctament les etiquetes html (h2 i ul) que havíem posat als comentaris del font.

Verifica que al document "Empleado.html" a la part de "Methods summary" i "Methods details" estiguin descrits els dos mètodes que teníem al font. Si no es veuen tots dos, quin és el problema?

### **Ara fes-ho tu:**

Documenta el projecte del Memory amb JavaDoc (o qualsevol altre dels que ja tinguis fet de M3-UF2) . Afegeix a la documentació com a mínim:

- Autor
- Versió
- Descripció de la classe
- Descripció de cada mètode i els seus paràmetres fent servir etiquetes HTML

Genera la documentació escollint les opcions del document HTML que vegis que són més útils.