

# UD7: Documentación

---

1º DAW – ENTORNOS DE  
DESARROLLO

IES Macia Abela

# Índice

- Documentación
  - ¿Qué documentamos?
- Documentación en Java
- Tipos de comentarios
- Javadoc
  - Clases e Interfaces
  - Constructores y Métodos
  - Atributos
- Ejemplo
- Javadoc - NetBeans

# Documentación

- Manuales técnicos, de usuario (tutoriales, guías) y otras instrucciones que facilitan el uso y operación de un producto o servicios de software.
- A **nivel de código**, se refiere a la explicación de los recursos utilizados (**clases, atributos, métodos, etc.**) y **cómo interactúan**.
- El código se documenta mediante:
  - Uso de código limpio *naming*.
  - Comentarios no redundantes.
  - Diagramas UML.

# Documentación



*“Si un programa no merece la pena documentarlo, probablemente no merece la pena ejecutarlo”* J.Nagler. 1995 Coding Style and Good Computing Practices.

*“No documentar código malo, reescribirlo”* R. Caron. 2000 Coding Techniques and Programming Practices.

# Qué documentamos...

- Clase
  - Finalidad, autor y versión.
- Atributos
- Métodos
  - Funcionalidad.
  - Parámetros y sus tipos.
  - Valor devuelto.
  - Excepciones lanzadas.
- Fragmentos de código complejo mediante comentarios.

# Qué documentamos...

- Añadir explicaciones a todo aquello que no es evidente.
- NO repetir lo que se hace, sino explicar por qué se hace.
  - ¿De qué se encarga una clase? ¿Un paquete?
  - ¿Qué hace un método?
  - ¿Cual es el uso esperado de un método?
  - ¿Para qué se usa una variable?
  - ¿Qué se debería mejorar... si hubiera tiempo?

# Documentación en Java

- La biblioteca con librerías estándar de Java es muy extensa, está formada por:
  - Multitud de clases
  - Multitud de métodos
- Es imposible conocerlos todos con detalle suficiente como para utilizarlos correctamente.
- Por ello es muy importante disponer de una buena documentación.
- Documentación de la última versión:  
<https://docs.oracle.com/en/java/javase/22/docs/api/index.html>

# Documentación en Java

- Los diseñadores de Java definieron un método sencillo para generar la documentación de las clases y métodos:
  - La documentación se inserta en el mismo fichero que el código (en forma de comentarios).
    - De esta forma, resulta más sencillo que la documentación evolucione junto al código.
- La herramienta **Javadoc** extrae esos comentarios especiales y genera páginas html para ser vistas desde un navegador.
  - De esta manera facilitamos la navegación de una clase a otra.



# Tipos de comentarios

- Legales
- Introdutorios: encabezados de ficheros, clases.
- Explicación de la finalidad
  - Comentarios de pocas líneas.
  - Evitar:
    - Redundancia
    - Documentación de código fácil.
    - NO registro de cambios → para eso tenemos Git.
    - Comentarios de cierre de bloques o fin de métodos.
- TODO: Tareas, comunicación con equipo desarrollo.
- Documentación API: Javadoc.

# Tipos de comentarios en Java

- Comentario de una línea:
  - Empiezan por los caracteres “//” y terminan con el fin de línea. Documentan código que no necesitamos que aparezca en la documentación externa. Su puede utilizar para varias líneas seguidas.
- Comentario tipo lenguaje programación C:
  - Empiezan por los caracteres “/\*”, se pueden prolongar a lo largo de varias líneas y terminan con los caracteres “\*/”. Para comentar código obsoleto que no queremos que eliminar del todo.
- Javadoc
  - Empiezan con los caracteres “/\*\*”, se pueden prolongar a lo largo de varias líneas y terminan con los caracteres “\*/”. Para generar documentación externa la programa.

# Javadoc

- Herramienta para documentar código Java.
- Más que ayudar a comprender el código, se centra en la interfaz (API – Application Programming Interfaz) de las clases y paquetes Java.
- La estructura básica de los comentarios es la siguiente:

```
/**  
 * Parte descriptiva.  
 * Puede consistir de varias frases o párrafos.  
 * @etiqueta texto específico de la etiqueta  
 */
```

# Javadoc – Clases e Interfaces

- Herramienta para documentar código Java.
- Deben usarse al menos las etiquetas
  - `@author` y `@version`
- Etiquetas posibles:

<code>@author</code>	Nombre del autor
<code>@version</code>	Identificació de versió y fecha
<code>@see</code>	Referencia a otras clases y métodos
<code>@since</code>	Indica desde qué versión o fecha existe la clase o interfaz en el paquete
<code>@deprecated</code>	Esta clase no debería usarse pues puede desaparecer en próximas versiones

# Javadoc – Constructores y Métodos

- Deben usarse al menos las etiquetas
  - @param – Una por argumento de entrada.
  - @return – si el método no es void
  - @exception o @throws – una por tipo de excepción
- Etiquetas posibles:

@param	Nombre del parámetro	Descripción de su significado y uso
@return		Descripción de lo que se devuelve
@exception	Nombre de la excepción	Excepciones que pueden lanzarse
@since	Indica desde qué versión o fecha existe este constructor o método en la clase	
@deprecated	Este método no debería usarse pues puede desaparecer en próximas versiones	

# Javadoc – Atributos

- No es obligatorio documentarlos.
- Etiquetas posibles:

@since	Indica desde qué versión o fecha existe este atributo en la clase
@deprecated	Este atributo no debería usarse pues puede desaparecer en próximas versiones

- Pueden utilizarse etiquetas HTML para señalar u organizar ciertos elementos. Como por ejemplo, la descripción de una clase en `<strong>` o el uso de párrafos `<p>` o listas con `<ul>` y `<li>`.

# Javadoc – Ejemplo Clase

## ● Documentación Clase Circulo

```
package macia.ed;

/**
 * <strong>Representa un circulo.</strong>
 * Esta clase representa un círculo en un plano bidimensional, con un centro definido por
 * las coordenadas (centroX, centroY) y un radio.
 *
 * <p>Los atributos de esta clase son:</p>
 * <ul>
 * <li>{@code centroX}: La coordenada X del centro del círculo.</li>
 * <li>{@code centroY}: La coordenada Y del centro del círculo.</li>
 * <li>{@code radio}: El radio del círculo.</li>
 * </ul>
 *
 * <p>El centro del círculo está ubicado en el punto ({@code centroX}, {@code centroY}) del plano.
 * El radio define la distancia desde el centro hasta cualquier punto de la circunferencia.</p>
 *
 * @author Macia
 * @version 1.0
 * @since 01.04.2022
 */
public class Circulo {
```

# Javadoc – Ejemplo Clase

- Documentación Clase Circulo

```
public class Circulo {  
  
    /**  
     * Coordenada X del centro del círculo.  
     * La coordenada X define la posición horizontal del centro del círculo en el plano  
     */  
    private double centroX;  
  
    /**  
     * Coordenada Y del centro del círculo.  
     * La coordenada Y define la posición vertical del centro del círculo en el plano.  
     */  
    private double centroY;  
  
    /**  
     * Radio del círculo.  
     * El radio define la distancia desde el centro del círculo hasta cualquier punto  
     * de su circunferencia.  
     */  
    private double radio;  
}
```



# Javadoc – Ejemplo Clase

## ● Documentación Clase Circulo

```
/**
 * <strong>Constructor.</strong>
 * @param cx centro: coordenada X.
 * @param cy centro: coordenada Y.
 * @param r radio.
 */
public Circulo(double cx, double cy, double r) {
    centroX = cx;
    centroY = cy;
    radio = r;
}

/**
 * <strong>Recupera el valor de X.</strong>
 * @return centroX: coordenada X.
 */
public double getCentroX() {
    return centroX;
}

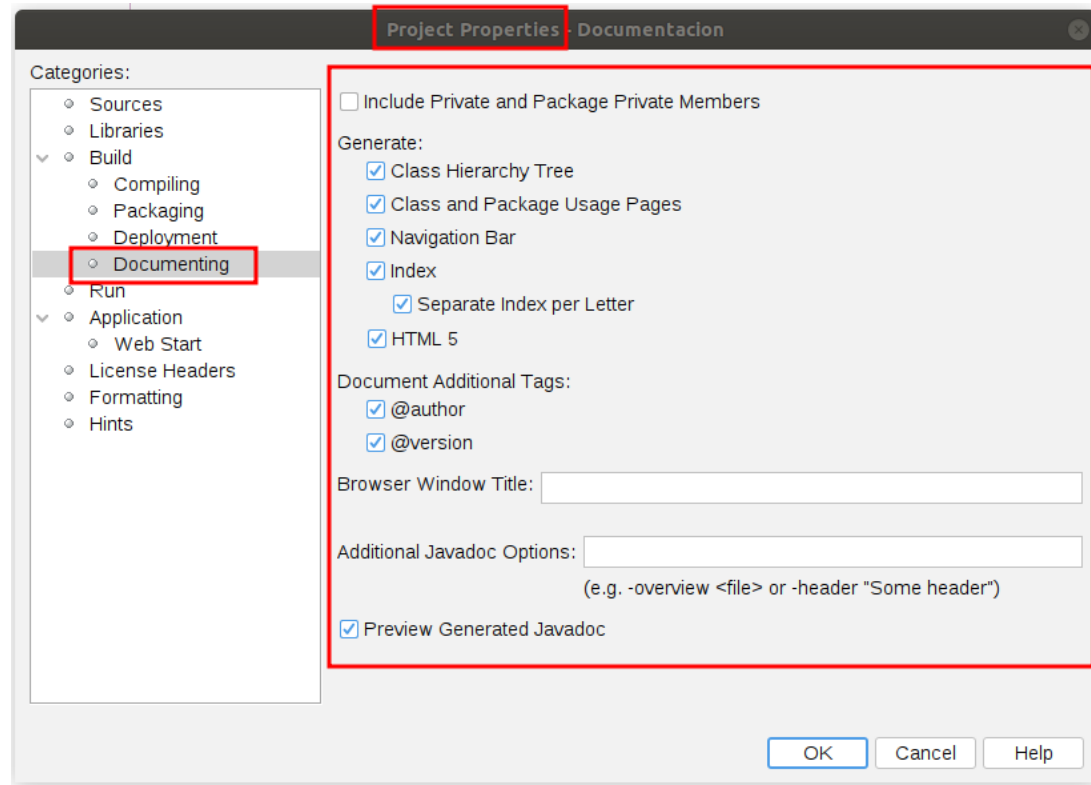
/**
 * <strong>Calcula la longitud de la circunferencia.</strong>
 * @return circunferencia.
 */
public double getCircunferencia() {
    return 2 * Math.PI * radio;
}
```

```
/**
 * <strong>Desplaza el círculo a otro lugar.</strong>
 * @param deltaX movimiento en el eje X.
 * @param deltaY movimiento en el eje Y.
 */
public void mueve(double deltaX, double deltaY) {
    centroX = centroX + deltaX;
    centroY = centroY + deltaY;
}

/**
 * <strong>Escala el círculo (cambia su radio).</strong>
 * @param s factor de escala.
 */
public void escala(double s) {
    radio = radio * s;
}
```

# Javadoc – NetBeans

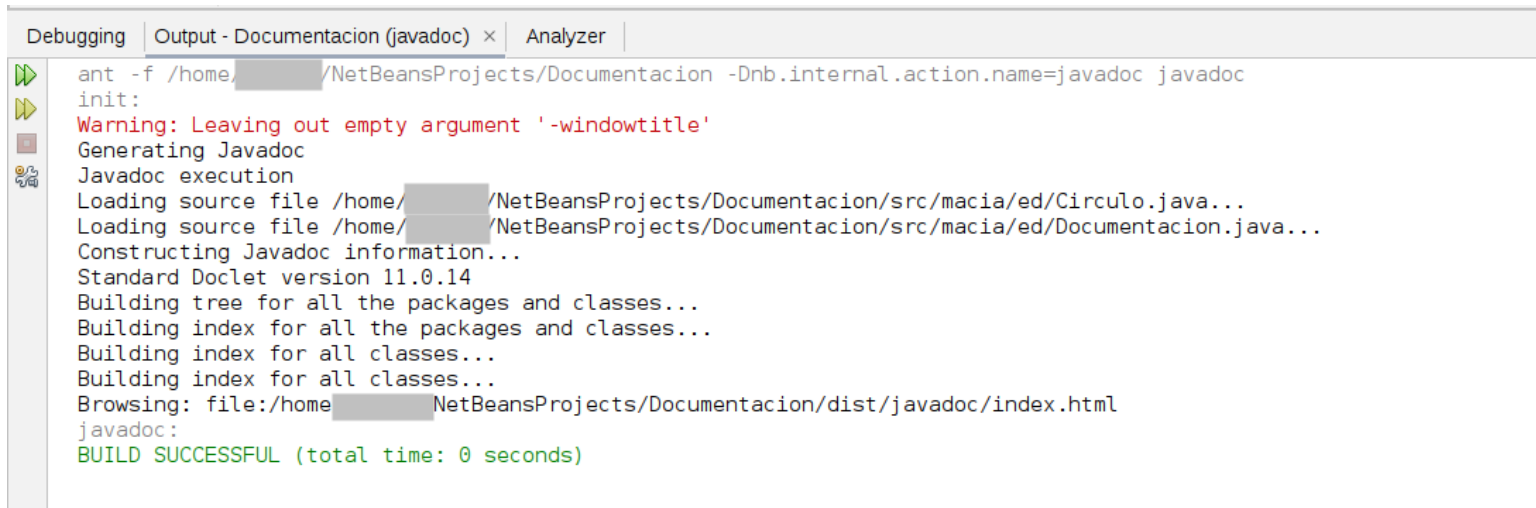
- Ajustar nuestras preferencias en las propiedades del proyecto:



- **Tools → Analyze Javadoc** (revisa posibles errores)

# Javadoc – NetBeans

- Generar Documentación:
  - **Run → Generate Javadoc** (desde el menú)
    - Revisar la salida por consola con el resultado de la ejecución:



```
ant -f /home/[redacted]/NetBeansProjects/Documentacion -Dnb.internal.action.name=javadoc javadoc
init:
Warning: Leaving out empty argument '-windowtitle'
Generating Javadoc
Javadoc execution
Loading source file /home/[redacted]/NetBeansProjects/Documentacion/src/macía/ed/Circulo.java...
Loading source file /home/[redacted]/NetBeansProjects/Documentacion/src/macía/ed/Documentacion.java...
Constructing Javadoc information...
Standard Doclet version 11.0.14
Building tree for all the packages and classes...
Building index for all the packages and classes...
Building index for all classes...
Building index for all classes...
Browsing: file:/home/[redacted]NetBeansProjects/Documentacion/dist/javadoc/index.html
javadoc:
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Los ficheros generados, se encuentran en el subdirectorio dist/javadoc dentro del proyecto.

# Javadoc – NetBeans



- Se abre automáticamente, la documentación en formato HTML:

PACKAGECLASSUSE TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHODDETAIL: FIELD | CONSTR | METHODSEARCH

**Package** macia.ed

**Class Circulo**

java.lang.Object<sup>Ⓔ</sup>  
macia.ed.Circulo

---

```
public class Circulo
extends ObjectⒺ
```

**Representa un círculo.** Esta clase representa un círculo en un plano bidimensional, con un centro definido por las coordenadas (centroX, centroY) y un radio.

Los atributos de esta clase son:

- centroX: La coordenada X del centro del círculo.
- centroY: La coordenada Y del centro del círculo.
- radio: El radio del círculo.

El centro del círculo está ubicado en el punto (centroX, centroY) del plano. El radio define la distancia desde el centro hasta cualquier punto de la circunferencia.

**Since:**  
01.04.2022

**Version:**  
1.0

**Author:**  
Macia

**Constructor Summary**

Constructors

Constructor	Description
<code>Circulo(double cx, double cy, double r)</code>	<b>Constructor.</b>

# ¿Preguntas?

