

# Reporte de uso de Javadoc

Fecha de entrega: 03 de Julio del 2017

## ¿Que es Javadoc?

Javadoc es una herramienta que extrae los comentarios JavaDoc contenidos en el programa Java indicado y construye con ellos ficheros .html que puede servir como documentacion de clase.

## Formato de los comentarios JavaDoc

Los comentarios JavaDoc están destinados a describir, principalmente, clases y métodos. Como están pensados para que otro programador los lea y utilice la clase (o método) correspondiente, se decidió fijar al menos parcialmente un formato común, de forma que los comentarios escritos por un programador resultaran legibles por otro. Para ello los comentarios JavaDoc deben incluir unos indicadores especiales, que comienzan siempre por '@' y se suelen colocar al comienzo de línea.

Se usan indicadores para el número de versión ([@version](#)), el autor ([@author](#)) y otros. Es importante observar que los indicadores no son obligatorios; por ejemplo en un método sin parámetros no se incluye obviamente el indicador [@param](#).

Ejemplo de comentario usando indicadores:

```
/**
 * Metodo que guarda un registro de un empleado vendido en la base de datos.
 * @param conexion El parametro conexion es objeto con la informacion de la conexion con la base de datos.
 * @param empleado EL parametro empleado contiene la informacion que sera guardada en la base de datos.
 * @throws SQLException
 */
```

Este ejemplo fue extraido de los comentarios usados en la documentacion de nuestro sistema.

## Indicadores mas usuales:

- [@author nombreDelAutor descripción](#).  
Indica quién escribió el código al que se refiere el comentario. Si son varias personas se escriben los nombres separados por comas o se repite el indicador, según se prefiera. Es normal incluir este indicador en el comentario de la clase y no repetirlo para cada método, a no ser que algún método haya sido escrito por otra persona.
- [@version númeroVersión descripción](#).  
Si se quiere indicar la versión. Normalmente se usa para clases, pero en ocasiones también para métodos.
- [@param nombreParámetro descripción](#).  
Para describir un parámetro de un método.
- [@return descripción](#).  
Describe el valor de salida de un método.
- [@see nombre descripción](#). Cuando el trozo de código comentado se encuentra relacionada con otra clase o método, cuyo nombre se indica en *nombre*.
- [@throws nombreClaseExcepción descripción](#). Cuando un método puede lanzar una excepción ("romperse" si se da alguna circunstancia) se indica así.

- `@deprecated descripción`. Indica que el método (es más raro encontrarlos para una clase) ya no se usa y se ha sustituido por otro.
- `@author nombreDelAutor descripción`.  
Indica quién escribió el código al que se refiere el comentario. Si son varias personas se escriben los nombres separados por comas o se repite el indicador, según se prefiera. Es normal incluir este indicador en el comentario de la clase y no repetirlo para cada método, a no ser que algún método haya sido escrito por otra persona.
- `@version númeroVersión descripción`.  
Si se quiere indicar la versión. Normalmente se usa para clases, pero en ocasiones también para métodos.
- `@param nombreParámetro descripción`.  
Para describir un parámetro de un método.
- `@return descripción`.  
Describe el valor de salida de un método.
- `@see nombre descripción`. Cuando el trozo de código comentado se encuentra relacionada con otra clase o método, cuyo nombre se indica en *nombre*.
- `@throws nombreClaseExcepción descripción`. Cuando un método puede lanzar una excepción ("romperse" si se da alguna circunstancia) se indica así.
- `@deprecated descripción`. Indica que el método (es más raro encontrarlos para una clase) ya no se usa y se ha sustituido por otro.

## Conclusion

Usar herramientas para la documentación del código, es de gran importancia, ya que cuando se integren nuevos programadores al equipo de desarrollo, estos podrán comprender el código de manera más rápida, también influye el detalle que se haya tenido al generar la documentación debido a que el nivel de rapidez de comprensión de nuevos integrantes dependerá de este factor.

Aunque existen más herramientas para generar documentación

Al usar Javadoc, se pudo notar que cuando genera los archivos .html genera un formato en el que es muy sencillo moverse entre las descripciones de las funciones y las clases ahí descritas, también cabe mencionar que el archivo .html generado cuenta con un panel lateral que muestra todas las clases del proyecto, lo que hace más rápido el acceso a la descripción de estas.