Documentación del código Java

Adolfo Sanz De Diego

Junio de 2011

# El Javadoc

Es una utilidad de para la generación de documentación de APIs en formato HTML.

Es el estándar para documentar clases de Java.

La mayoría de los IDEs los generan automáticamente.

<http://download.oracle.com/javase/6/docs/api/>

Han de usarse etiquetas HTML o ciertas palabras reservadas precedidas por "@".

Estas etiquetas se escriben al principio de cada clase, propiedad o método.

Dentro de un comentario iniciado con /\*\* y acabado con \*/.

# Etiquetas Javadoc

|  |  |  |  |
| --- | --- | --- | --- |
| Ta | g | Descripción | Uso |

**@author** Nombre del desarrollador. nombre\_autor

**@deprecated** Indica que no se recomienda su uso porque descripción puede desaparecer en versiones futuras.

**@param** Definición de un parámetro de un método. nombre\_parámetro descripción\_parámetro

**@return** Informa de lo que devuelve el método. descripción\_retorno No usar en constructores o métodos "void".

**@see** Enlace a otro método o clase. referencia (#método(); clase#método(); paquete.clase; paquete.clase#método())

**@throws** Excepción lanzada por el método. nombre\_clase\_excepción descripción\_excepción

# Ejemplo

/\*\*  
 \* Retorna un "Hola + nombre".  
 \* Al ser el nombre obligatorio, si es nulo o vacío se lanzará una excepción.  
 \*  
 \* @author Adolfo Sanz De Diego (asanzdiego@gmail.com)  
 \* @param nombre El nombre que queremos retornar con un "Hola + nombre".  
 \* @return Retorna "Hola + nombre".  
 \* @throws IllegalArgumentException Si nombre es null, está vacío o contiene solo espacios.  
 \*/  
public String retornaHolaNombre(final String nombre) throws IllegalArgumentException {  
 if (nombre == null || nombre.trim().equals("")) {  
 throw new Exception ("El nombre no puede ser nulo o vacío");  
 } else {  
 this.titulo = titulo;  
 }  
}

# Diagramas UML automáticos

**GraphViz**

* Librería de software libre y multiplataforma para creación de **gráficos** a partir de textos simples.

**UMLGraph**

* Permite la definición declarativa de **diagramas UML de clases y de secuencia**.
* Se puede integrar con Javadoc.
* Dos proyectos interesantes basados en UMLGraph:
  + **DotUML** que integra UMLGraph en **Maven**.
  + **LightUML** que integra UMLGraph en **Eclipse**.