

Documentación del código Java

Adolfo Sanz De Diego

Junio de 2011

1 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 `*/`.

2 Etiquetas Javadoc

Tag	Descripción	Uso
@author	Nombre del desarrollador.	nombre_autor
@deprecated	Indica que no se recomienda su uso porque puede desaparecer en versiones futuras.	descripción

Tag	Descripción	Uso
@param	Definición de un parámetro de un método.	nombre_parámetro descripción_parámetro
@return	Informa de lo que devuelve el método. No usar en constructores o métodos "void".	descripción_retorno

Tag	Descripción	Uso
@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

3 Ejemplo

```
/**
 * Retorna un "Hola + nombre".
 * Al ser el nombre obligatorio, si es nulo o vacío se lanzará una excep
 *
 * @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 cont
 */
public String retornaHolaNombre(final String nombre) throws IllegalArgum
    if (nombre == null || nombre.trim().equals("")) {
        throw new Exception ("El nombre no puede ser nulo o vacío");
    } else {
        this.titulo = titulo;
    }
}
```

4 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**.