UA02. PRIMER ACCESO A DATOS

3. Pruebas y documentación

Una de las partes más importantes de la programación y en la que seguramente muchas veces nadie piensa es la **documentación**. Dejar el código **bien documentado y comentado** es una tarea igualmente importante y que en nuestra trayectoria laboral nos será de gran utilidad.

La manera más conocida de documentar el lenguaje Java es **JavaDoc**. Se trata de una utilidad de Oracle que permite documentar clases de Java.

En primer lugar, lo más común es usar **comentarios**. Los comentarios se pueden escribir de tres maneras:

```
1 //Comentario en una línea simple
2 /*
3 *Comentario en más de una línea
4 */
5 /**
6 *Comentario con JavaDoc
7 */
```

La estructura utilizada en **JavaDoc** es muy similar a la que se utiliza normalmente para comentarios, pero en la API se añade un asterisco extra al empezar, que es el característico para definirlo.

La gran peculiaridad de esta API es que permite añadir tags **HTML** dentro de los comentarios. Eso es debido a que JavaDoc, a través de los comentarios, genera un documento que servirá como documentación del proyecto.

Los comentarios para documentar, normalmente, se pueden poner en el código, encima de cualquier clase, en los métodos o atributos que queramos documentar.

Cuando se añadan comentarios, es necesario tener en cuenta que se debe describir cuál es la función de lo que se comenta. Por ejemplo, si comentamos una clase, haremos una breve explicación de lo que se encarga.

```
1  /**
2  * Aquí pondremos un resumen de la finalidad de esta clase
3  * Ej: Clase para documentar el funcionamiento de JavaDoc
4  *
5  * @author David Bermúdez
6  *
7  */
8  public class JavaDoc {
9      /**
10     * Atributo nombre de la clase javaDoc
11     */
12     private String nombre;
13     /**
14     * <p>Ejemplo de Javadoc con tags html.
15     * <a href="http://ciclo.iesnervion.es">IES Nervión</a>
16     * </p>
17     * @param ejemplo String que pasamos por parametro
18     * @return String con el resultado del metodo
19     * @see <a href="http://iesdavid.github.io">Teoría y ejemplos
20     * @since 1.0
21     */
22     public String ejemploMetodoJavadoc(String ejemplo) {
23         // Comentario con explicación de lo que se realiza en el
24         return “OK”;
25     }
26 }
```

Tags más importantes

TAG	DESCRIPCIÓN
@author	Sirve para poner el autor del desarrollo.
@deprecated	Indica que el método o clase es obsoleto (propio de versiones anteriores) y que no se recomienda su uso.
@param	Se usara para definir un parámetro de un método, es requerido para todos los parámetros del método.
@return	Se usa para indicar qué es lo que devuelve el método, no se usa para los métodos void.
@see	Asocia con otro método o clase.
@version	Se usa para definir la versión del método o la clase

Para generar la documentación JavaDoc, podemos hacerlo a través de nuestro IDE accediendo a **Project > Generate JavaDoc**, nos aparecerá una ventana que nos permitirá elegir dónde queremos guardar nuestra documentación. Y ya lo tendremos creado.