ED – UD04
PRÁCTICA 2: USO DE *JAVADOC* EN ECLIPSE

Javadoc es la utilidad de Java para extraer y generar documentación directamente del código en formato HTML. Para que la documentación sea en verdad útil debemos escribir los comentarios del código de acuerdo a las recomendaciones de *Javadoc*. La documentación y el código se van a incluir dentro del mismo fichero. Veamos a continuación las recomendaciones sobre los comentarios y la documentación del código fuente.

Los tipos de comentarios para generar la documentación son:

- Comentarios de línea: comienzan con los caracteres "//" y terminan con la línea.
- Comentarios de bloque: comienzan con los caracteres "/*" y terminan con los caracteres "*/". Pueden agrupar varias líneas.
- Comentarios de documentación *Javadoc*: estos comentarios se colocan entre los delimitadores /** … */, agrupan varias líneas y cada línea irá precedida por un *. Lo más importante es que estos deben colocarse antes de la declaración de una clase, un campo, un método o un constructor. Dentro de estos delimitadores se podrán escribir etiquetas HTML. Los comentarios *Javadoc* están formados por dos partes: una descripción y seguidamente un bloque de "*tags*".

Un ejemplo de documentación puede ser el siguiente:

```
/**
* <h2>Esto es un ejemplo de documentación</h2>
* Puedo añadir etiquetas HTML, para <strong>mejorar</strong> la presentación.
* Por ejemplo, añado unas viñetas.
* <ul>
*   <li>Inserción de registros</li>
*   <li>Borrado y modificación de registros</li>
* </ul>
*
* @autor MCV 2018
* @version v1.2018
*
*/
class prueba {
. . . . .
```

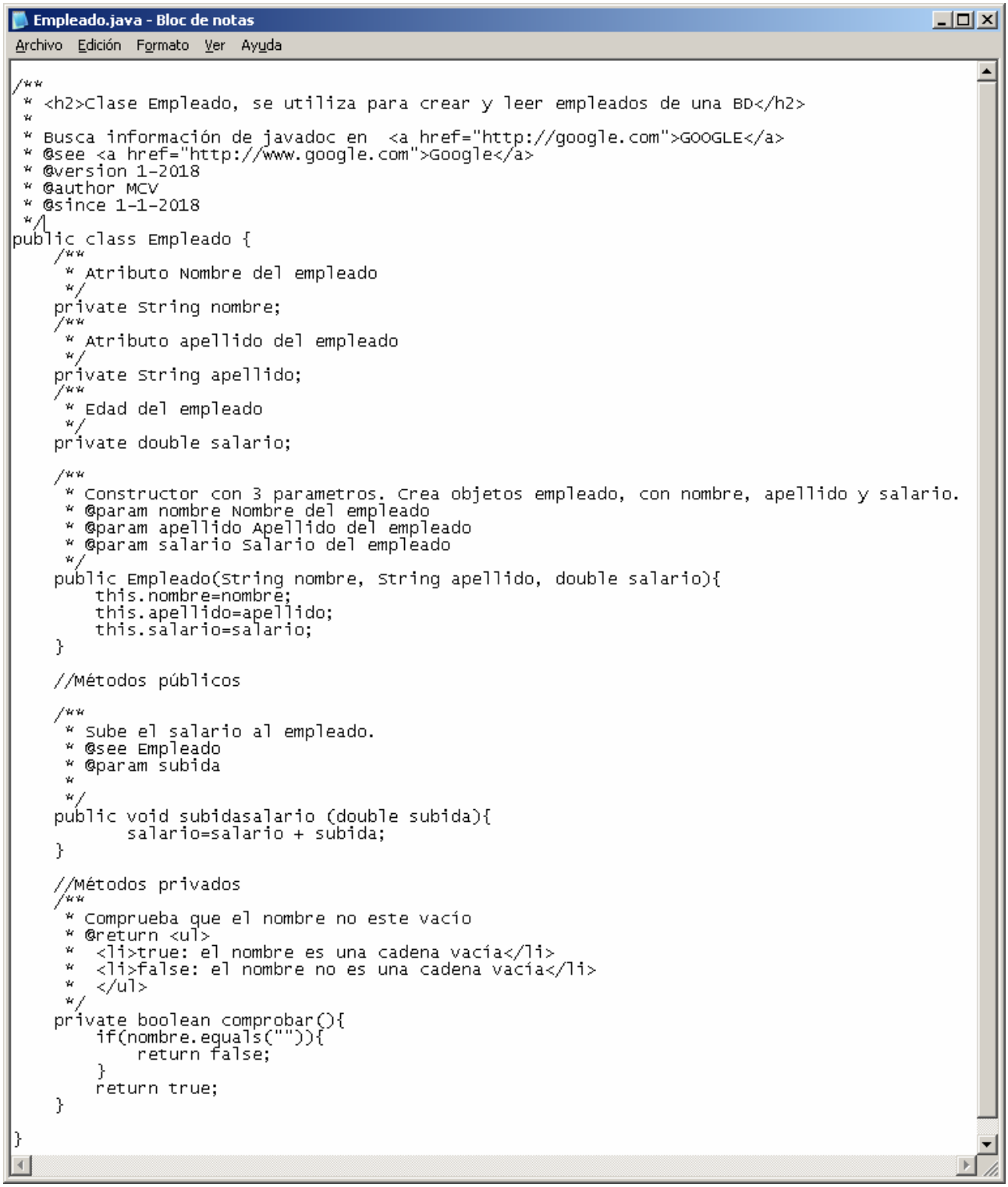
Se pueden usar *tags* para documentar ciertos aspectos concretos como la versión de la clase, el autor, los parámetros utilizados o los valores devueltos. Las etiquetas de *Javadoc* van precedidas por @. Éstas son las más utilizadas:

ETIQUETA	DESCRIPCIÓN
`@autor`	Autor de la clase.
`@version`	Versión de la clase.
`@see`	Referencia a otra clase, ya sea del API, del mismo proyecto o de otro.
`@param`	Descripción de parámetro.
`@return`	Descripción del dato que devuelve.

`@throws`	Descripción de la excepción que puede propagar.
`@deprecated`	Marca el método como obsoleto.
`@since`	Indica el número de versión desde la que existe el método.

Los comentarios de documentación *Javadoc* deben colocarse ANTES de las declaraciones de las clases, de los métodos o de los atributos.

Observa este ejemplo de documentación de una clase utilizando etiquetas:




```
/**
 * <h2>Clase Empleado, se utiliza para crear y leer empleados de una BD</h2>
 *
 * Busca información de javadoc en  <a href="http://google.com">GOOGLE</a>
 * @see <a href="http://www.google.com">Google</a>
 * @version 1-2018
 * @author MCV
 * @since 1-1-2018
 */
public class Empleado {
    /**
     * Atributo Nombre del empleado
     */
    private String nombre;
    /**
     * Atributo apellido del empleado
     */
    private String apellido;
    /**
     * Edad del empleado
     */
    private double salario;

    /**
     * Constructor con 3 parametros. Crea objetos empleado, con nombre, apellido y salario.
     * @param nombre Nombre del empleado
     * @param apellido Apellido del empleado
     * @param salario Salario del empleado
     */
    public Empleado(String nombre, String apellido, double salario){
        this.nombre=nombre;
        this.apellido=apellido;
        this.salario=salario;
    }

    //Métodos públicos

    /**
     * Sube el salario al empleado.
     * @see Empleado
     * @param subida
     *
     */
    public void subidasalario (double subida){
            salario=salario + subida;
    }

    //Métodos privados
    /**
     * Comprueba que el nombre no este vacío
     * @return <ul>
     *   <li>true: el nombre es una cadena vacía</li>
     *   <li>false: el nombre no es una cadena vacía</li>
     *   </ul>
     */
    private boolean comprobar(){
        if(nombre.equals("")){
            return false;
        }
        return true;
    }

}
```

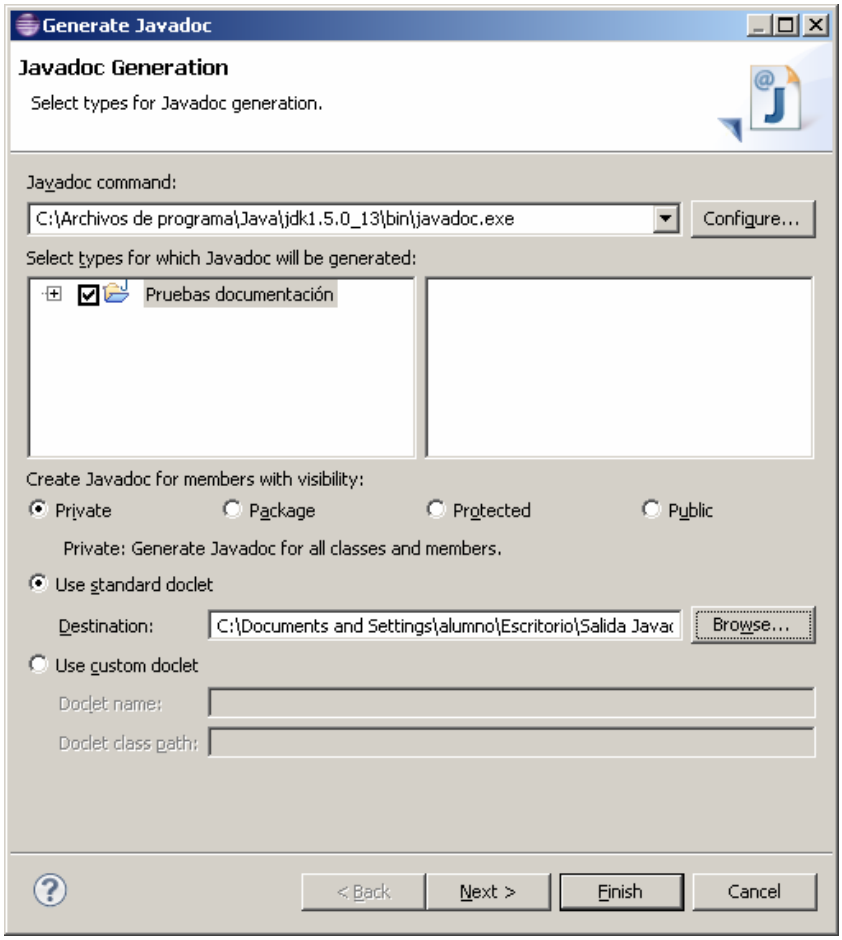
La mayor parte de los *IDEs* incluyen un botón o un enlace para configurar y ejecutar *Javadoc*. Para ejecutar *Javadoc* desde *Eclipse*, se abre el menú *Project* y se elige *Generate Javadoc*. En la ventana que se muestra se pedirá la siguiente configuración:

- En *Javadoc command* se tiene que indicar dónde se encuentra el archivo ejecutable de *Javadoc* (el archivo *javadoc.exe*). Se pulsa el botón *Configure* para buscarlo dentro de la carpeta donde se encuentra instalado el *JDK*. Dentro de esa carpeta se elige la carpeta *bin*.

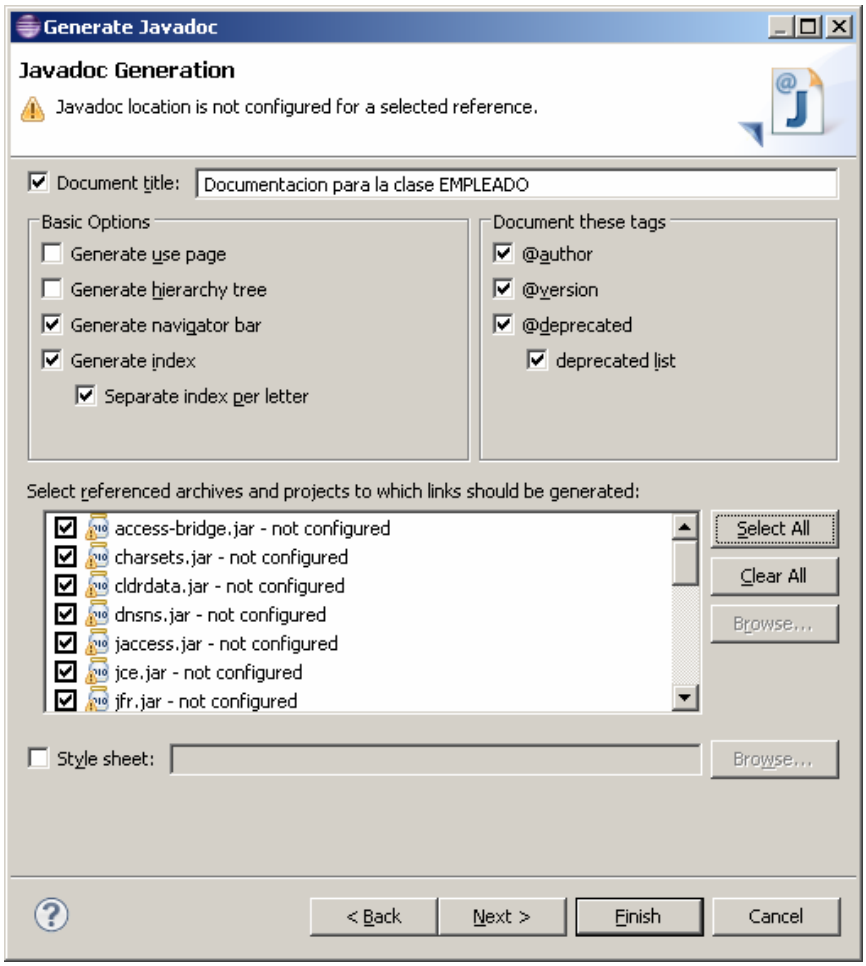
- En los dos cuadros inferiores se elegirá el proyecto y las clases a documentar.

- Se selecciona la visibilidad de los elementos que se van a documentar. Con *Private* se documentarán todos los miembros públicos, privados y protegidos.

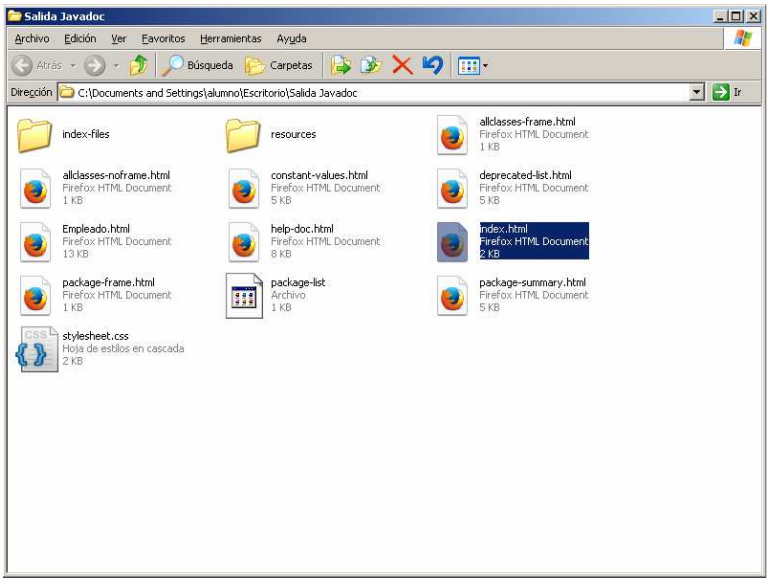
- Por último, se indica la carpeta de destino donde se almacenará el código *HTML* (conviene poner un nombre diferente al que viene por defecto pues con hay veces que no se cargan los estilos).



Se pulsa *Next* y, en la siguiente ventana, se indica el título del documento *HTML* que se genera y se eligen las opciones para la generación de las páginas *HTML*. Como mínimo se seleccionará la barra de navegación y el índice.



Al pulsar *Finish* nos vuelve a aparecer la interfaz de Eclipse. En la carpeta que hayamos indicado como destino de la salida de *Javadoc* tendremos una estructura web de documentos.



Al abrir *index.html* con un navegador accederemos a la página principal de la documentación generada: