



UT 4.B Documentación : Javadoc

ENTORNO DE DESARROLLO

I.E.S. Luis Vives - Diseño y Aplicaciones Multiplataforma

Introducción

Documentación de código

JAVADOC

La documentación de un software es fundamental para el mantenimiento y la mejora continua.

Existen muchos métodos para realizar documentación de todas las fases del proyecto y muchas herramientas que nos pueden ayudar en su realización.

<https://userguiding.com/es/blog/herramientas-documentacion-software/>

Java nos da la posibilidad de documentar directamente en el propio código utilizando javadoc.

Javadoc nos permite generar documentación de APIs en formato HTML a partir de código fuente Java. Javadoc es el estándar de la industria para documentar clases de Java. La mayoría de los IDEs los generan automáticamente.

Actividad 1

Mira en la web <https://userguiding.com/es/blog/herramientas-documentacion-software/> y elige una herramienta para ver qué utilidad tiene en la documentación de código.

Prepara una presentación de 10 minutos que ilustre el funcionamiento básico de la herramienta elegida para mostrarla en clase.

La presentación se realizará en grupos de 2 estudiantes.

Para generar API con Javadoc han de usarse etiquetas (tags)/palabras reservadas precedidas por el carácter "@". Estas etiquetas se escriben al principio de cada clase, miembro o método, dependiendo de qué objeto se desee describir, mediante un comentario iniciado con "/*" y acabado con "*/".

A continuación, se indican algunas de las palabras reservadas.

| Tag | Descripción | Uso | Versión |
|-------------|--|--|---------|
| @author | Nombre del desarrollador. | nombre_autor | 1.0 |
| @version | Versión del método o clase. | versión | 1.0 |
| @param | Definición de un parámetro de un método, es requerido para todos los parámetros del método. | nombre_parametro descripción | 1.0 |
| @return | Informa de lo que devuelve el método, no se puede usar en constructores o métodos "void". | descripción | 1.0 |
| @throws | Excepción lanzada por el método, posee un sinónimo de nombre @exception | nombre_clase descripción | 1.2 |
| @see | Asocia con otro método o clase. | referencia (#método()); clase#método(); paquete.clase; paquete.clase#método()). | 1.0 |
| @since | Especifica la versión del producto | indicativo numerico | 1.2 |
| @serial | Describe el significado del campo y sus valores aceptables. Otras formas válidas son @serialField y @serialData | campo_descripcion | 1.2 |
| @deprecated | Indica que el método o clase es antigua y que no se recomienda su uso porque posiblemente desaparecerá en versiones posteriores. | descripción | 1.0 |

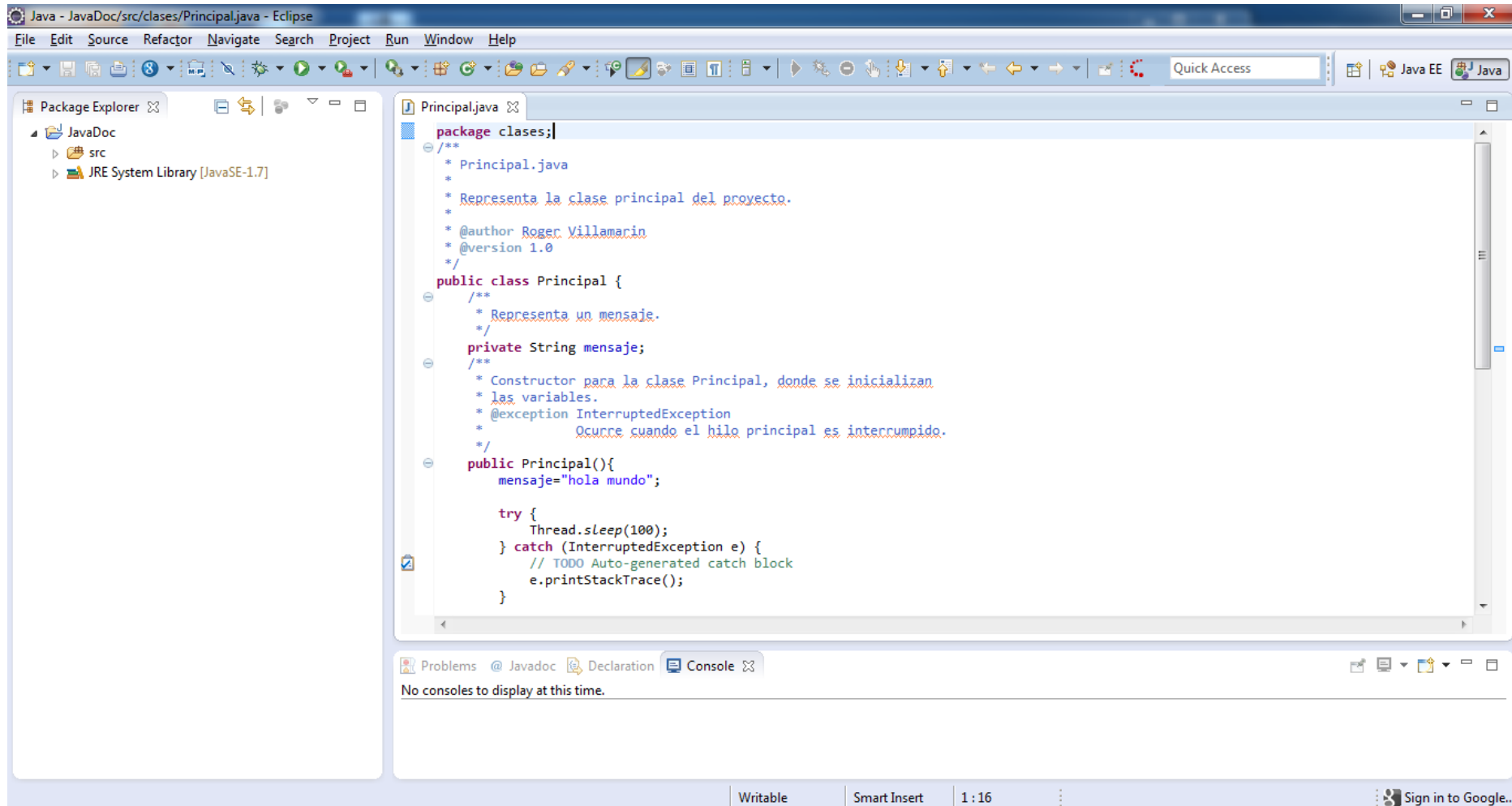
[Javadoc | Marco de Desarrollo de la Junta de Andalucía \(juntadeandalucia.es\)](http://juntadeandalucia.es)

```
/**
 * Inserta un título en la clase descripción.
 * AL ser el título obligatorio, si es nulo o vacío se lanzará
 * una excepción.
 *
 * @param titulo EL nuevo título de la descripción.
 * @throws IllegalArgumentException Si titulo es null, está vacío o contiene sólo espacios.
 */
public void setTitulo (String titulo) throws IllegalArgumentException
{
    if (titulo == null || titulo.trim().equals(""))
    {
        throw new IllegalArgumentException("El título no puede ser nulo o vacío");
    }
    else
    {
        this.titulo = titulo;
    }
}
```

Creación de Javadoc

ECLIPSE

Una vez tengamos nuestro proyecto documentado como se mostró en la entrada anterior, podemos continuar con la generación de dicha documentación.



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left shows a project named 'JavaDoc' with a source folder 'src' and the 'JRE System Library [JavaSE-1.7]'. The main editor window displays the source code of 'Principal.java'. The code includes package declarations, Javadoc comments for the class and its constructor, and the implementation of the 'Principal' class. The Javadoc comments are in Spanish and describe the class and its constructor. The constructor 'Principal()' initializes a 'mensaje' variable to 'hola mundo' and includes a try-catch block for 'InterruptedException'.

```
package clases;

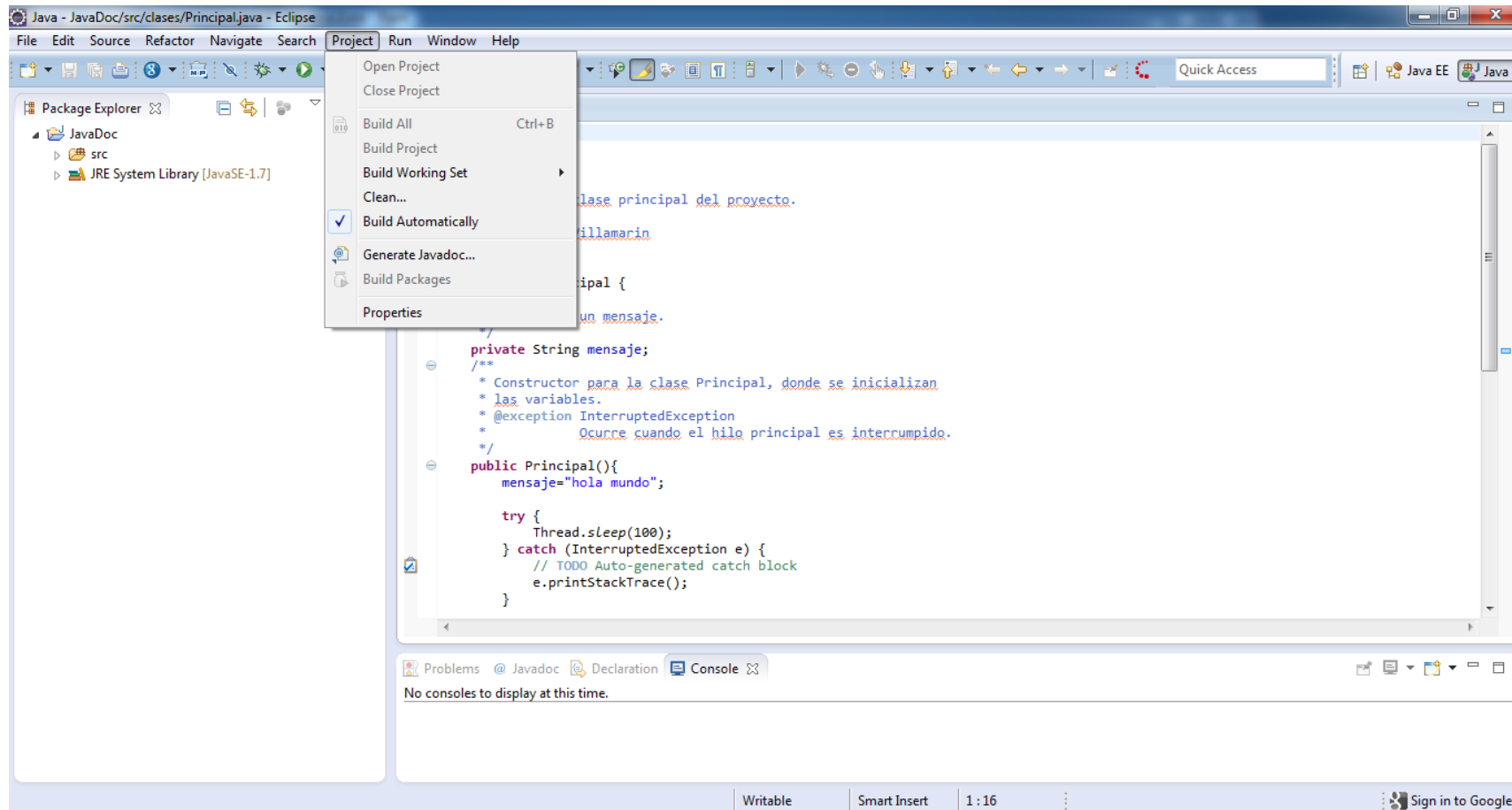
/**
 * Principal.java
 *
 * Representa la clase principal del proyecto.
 *
 * @author Roger Villamarin
 * @version 1.0
 */
public class Principal {
    /**
     * Representa un mensaje.
     */
    private String mensaje;

    /**
     * Constructor para la clase Principal, donde se inicializan
     * las variables.
     * @exception InterruptedException
     *         Ocurre cuando el hilo principal es interrumpido.
     */
    public Principal(){
        mensaje="hola mundo";

        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

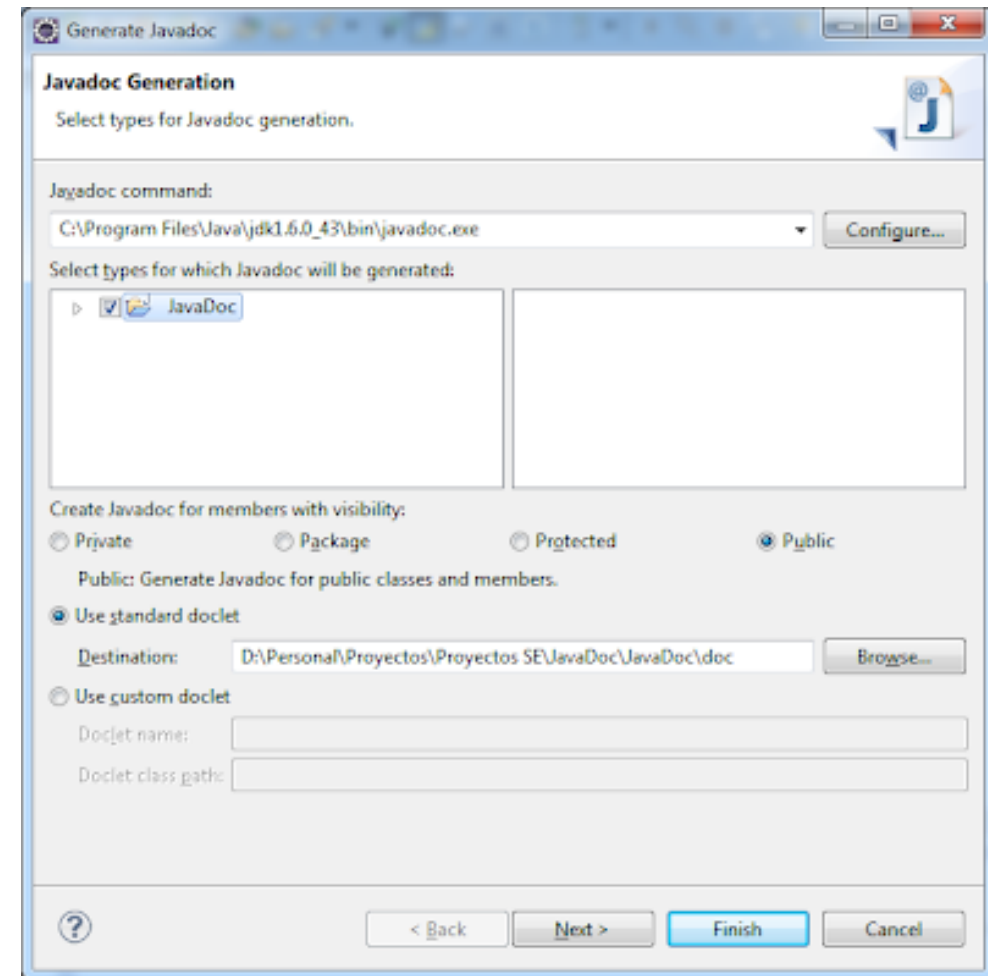
The bottom of the IDE shows the 'Problems' and 'Javadoc' tabs, and a 'Console' tab with the message 'No consoles to display at this time.'.

Luego nos dirigimos al menú Project y seleccionamos la opción Generar Javadoc.

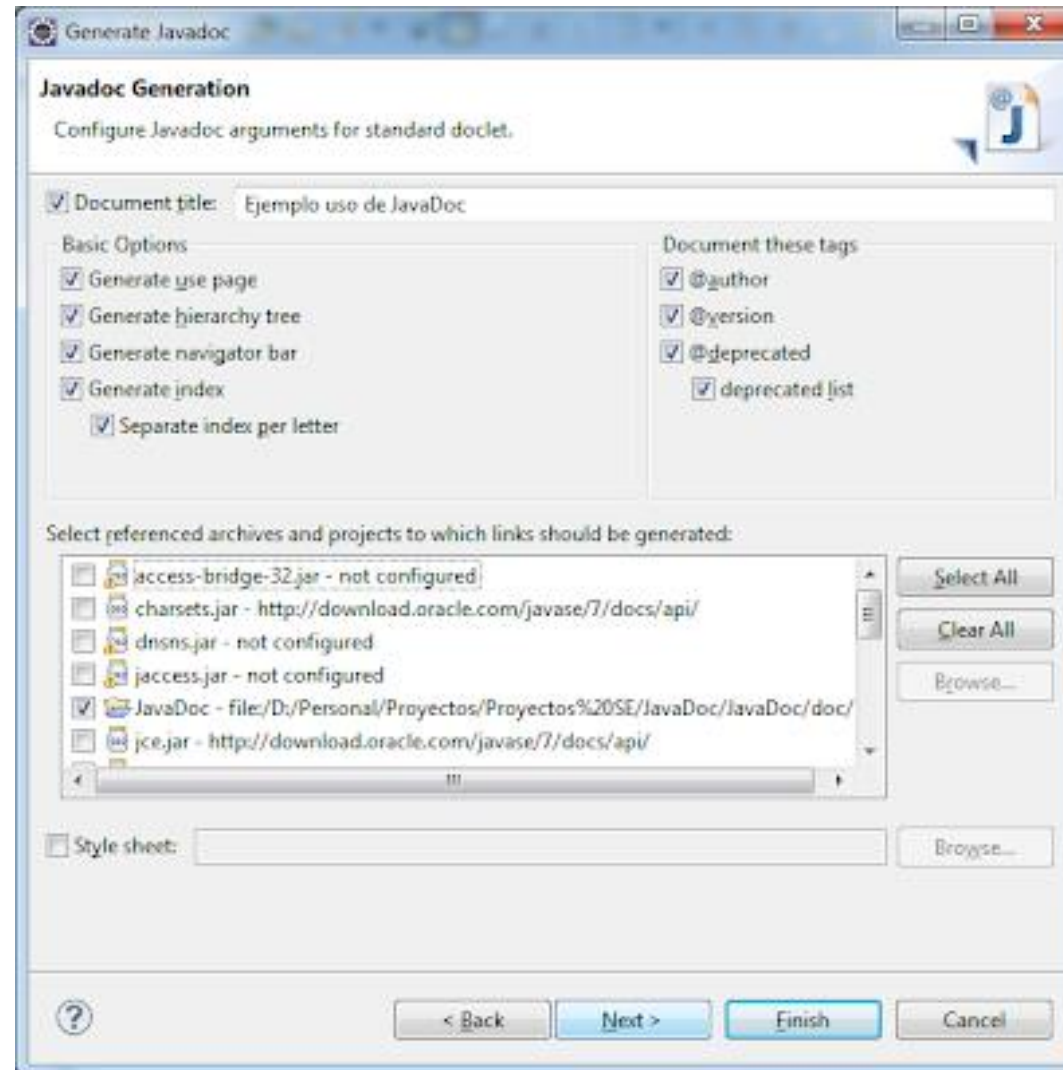


Luego nos aparecerá un asistente que nos ayudará con esta tarea. Nos aseguramos de tener seleccionado el proyecto al cual le generaremos la documentación y debemos indicarle donde está el **javadoc.exe**, la herramienta que nos ayuda a generar la documentación, que siempre está en el jdk, en la carpeta **bin**.

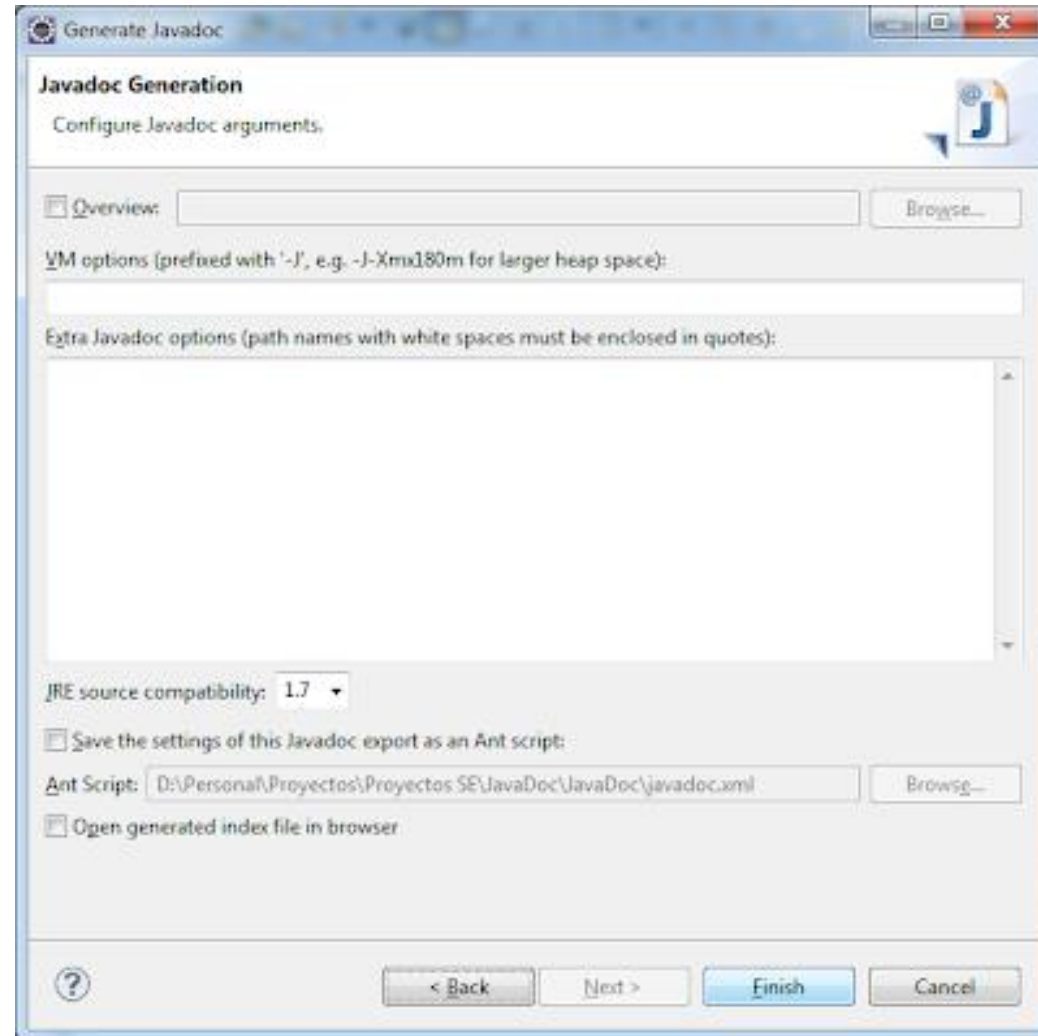
Hay que indicar en nivel de **visibilidad** que vamos a considerar.



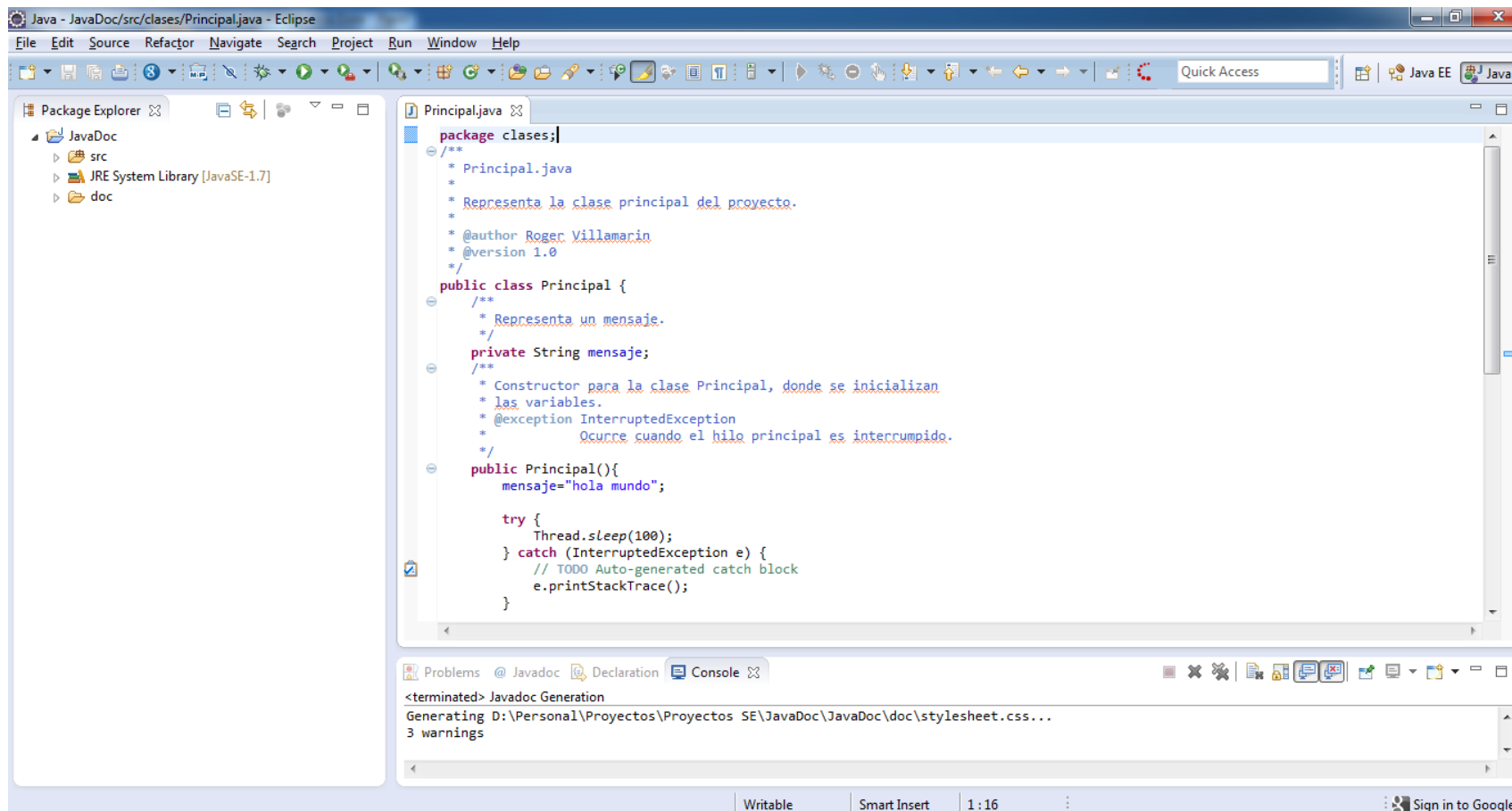
Ingresamos el título principal de dicha documentación y nos aseguramos de chequear el proyecto y presionamos siguiente.



Luego veremos la siguiente pantalla con algunas configuraciones adicionales, podemos dejarla en blanco si así lo deseamos.



Hecho esto, veremos una carpeta llamada doc, en nuestro proyecto, dentro de ella se encuentran los archivos generados, son archivos html, junto con los estilos y demás que vienen por defecto para la documentación en java.



```
Java - JavaDoc/src/clases/Principal.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer
  JavaDoc
    src
    JRE System Library [JavaSE-1.7]
    doc

Principal.java
package clases;

/**
 * Principal.java
 *
 * Representa la clase principal del proyecto.
 *
 * @author Roger Villamarin
 * @version 1.0
 */
public class Principal {
    /**
     * Representa un mensaje.
     */
    private String mensaje;

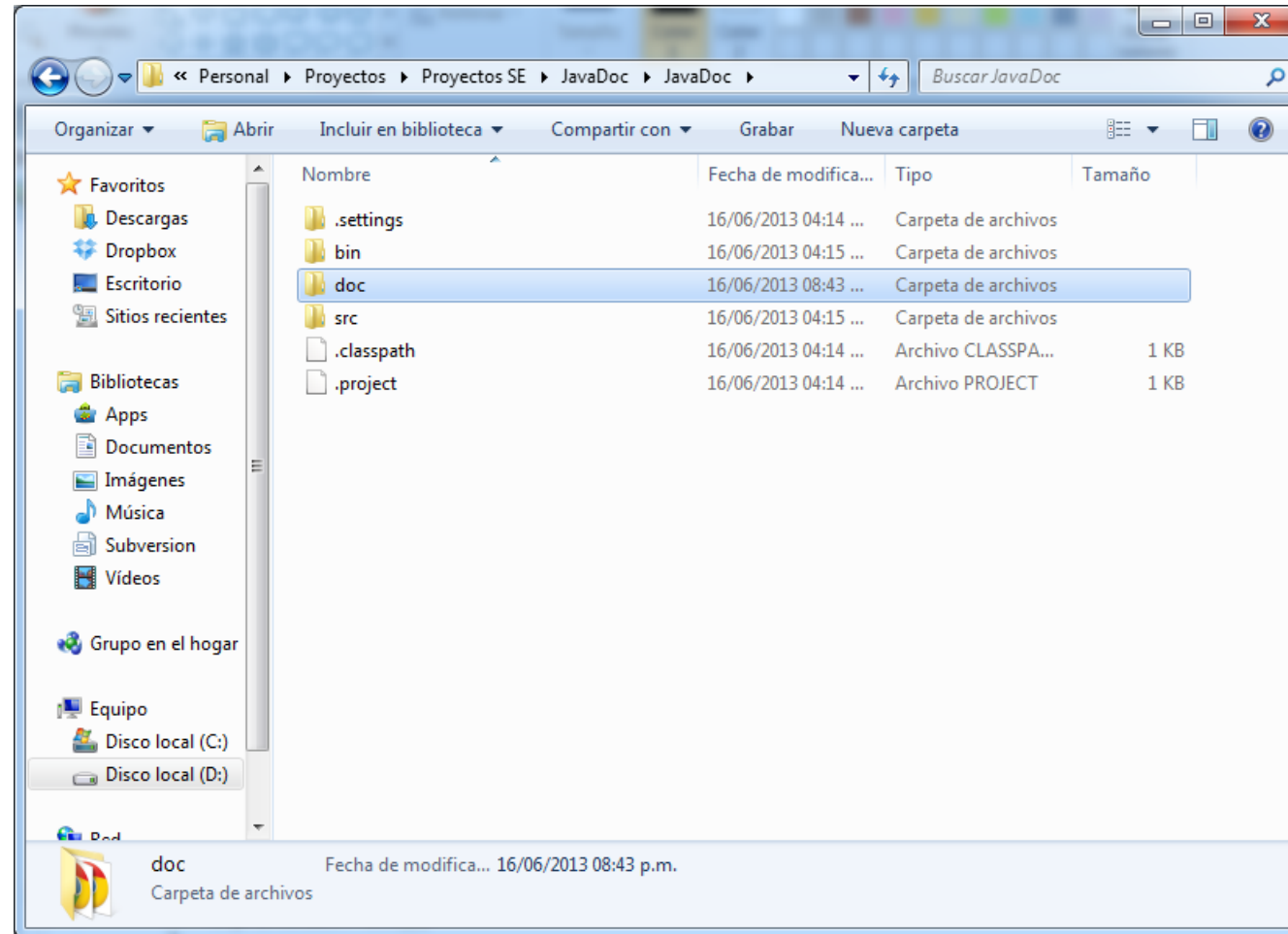
    /**
     * Constructor para la clase Principal, donde se inicializan
     * las variables.
     * @exception InterruptedException
     *         Ocurre cuando el hilo principal es interrumpido.
     */
    public Principal(){
        mensaje="hola mundo";

        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}

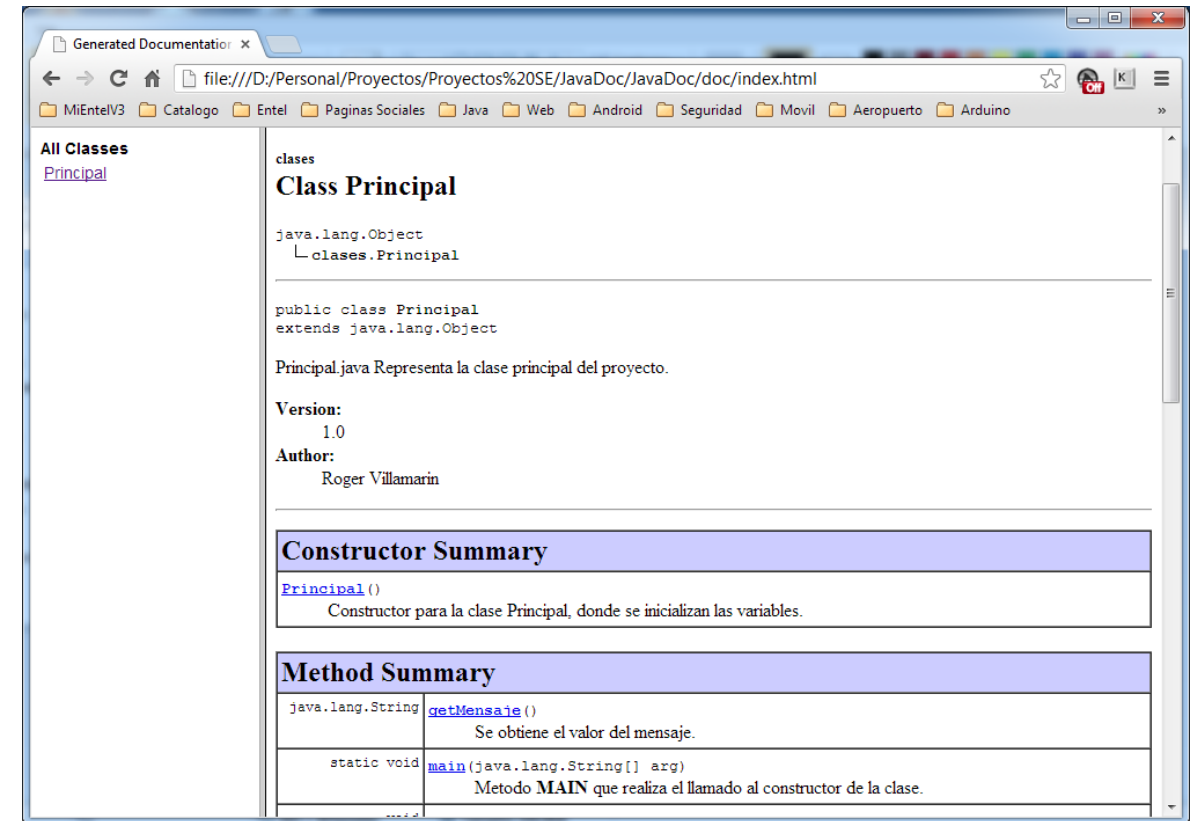
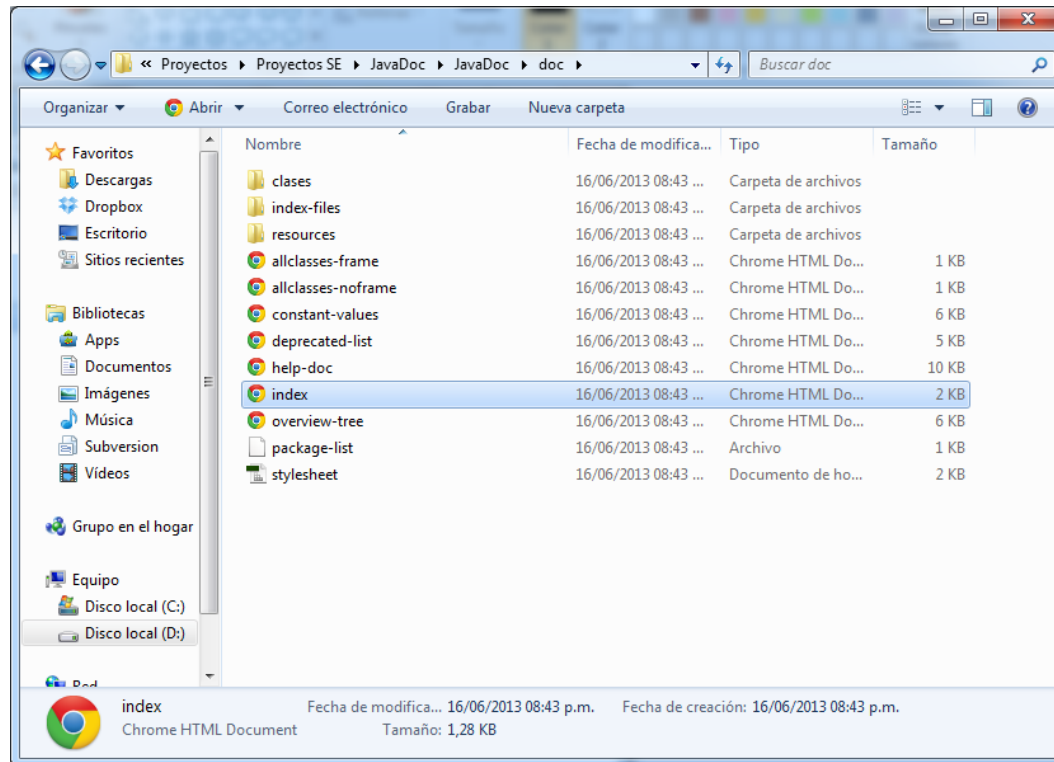
Problems @ Javadoc Declaration Console
<terminated> Javadoc Generation
Generating D:\Personal\Proyectos\Proyectos SE\JavaDoc\JavaDoc\doc\stylesheet.css...
3 warnings

Writable Smart Insert 1:16 Sign in to Google...
```

Si nos vamos a la raíz del proyecto veremos la carpeta así como en la siguiente imagen.



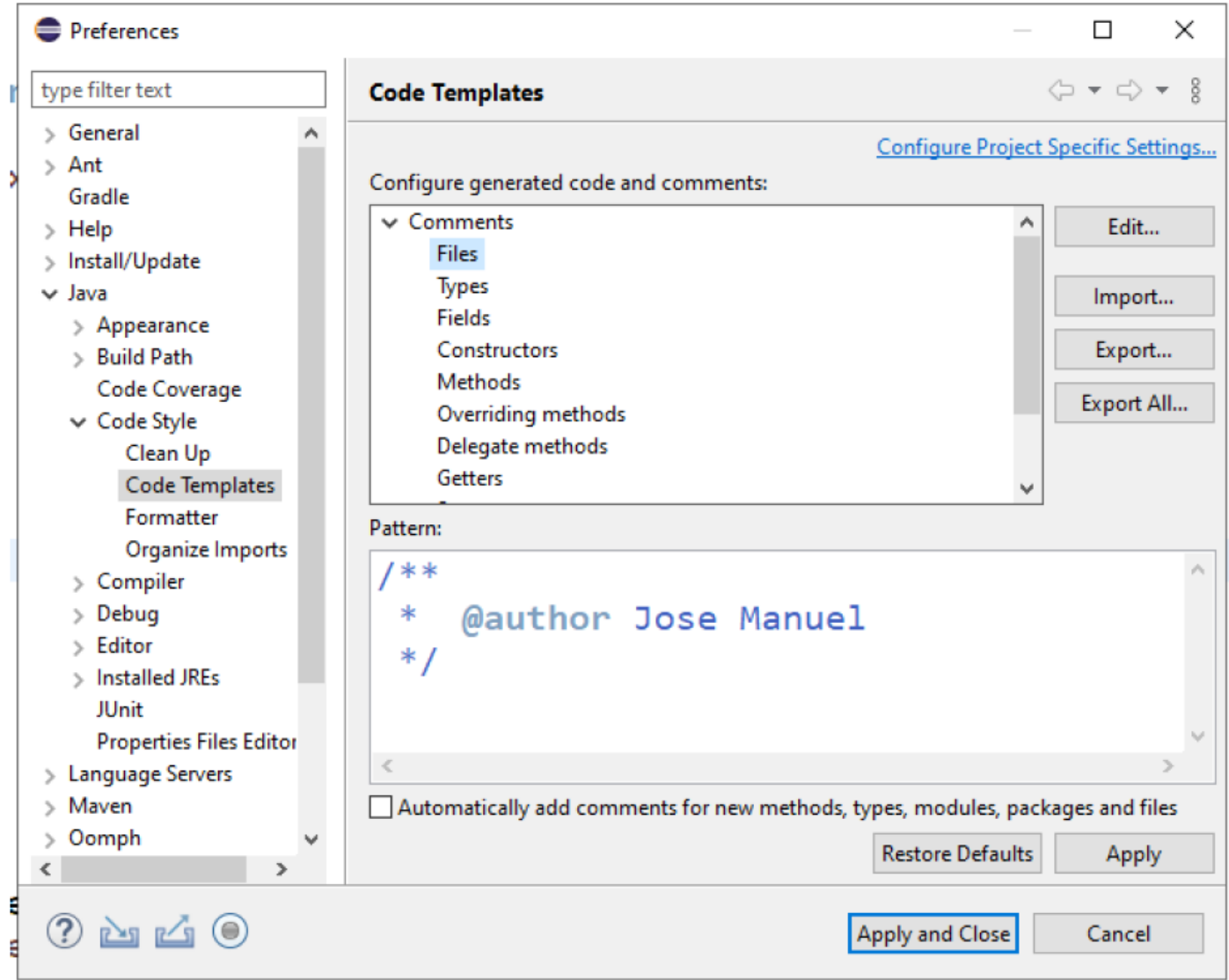
Y dentro de ella veremos el archivo index.html y al abrirlo veremos la documentación generada, además podremos navegar fácilmente entre las clases, métodos y demás.



En Window->Preferences tenéis acceso a modificar los comentarios que aparecen por defecto asociados a ficheros, atributos, métodos, etc.

Podrás añadir tags pulsando en el botón edit.

Después, cuando estés editando el código de tu programa y teclees `/**` y pulses INTRO en tu código aparecerá automáticamente el comentario predefinido.



Actividad 11

Documenta las clases que hemos creado para Fracción o Geometría utilizando JavaDoc

Ver [javadoc-The Java API Documentation Generator \(oracle.com\)](http://javadoc-the-java-api-documentation-generator.oracle.com)

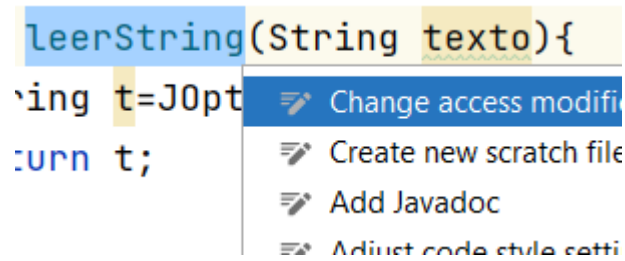
en la zona de Java Doc Tags para las etiquetas disponibles.

Documentar cada clase y cada método. Incluir nombre autor, versión, parámetros, tipo devuelto, excepciones generadas. Utiliza html en la cabecera de cada clase para resaltar la información de la misma.

Actividad 10

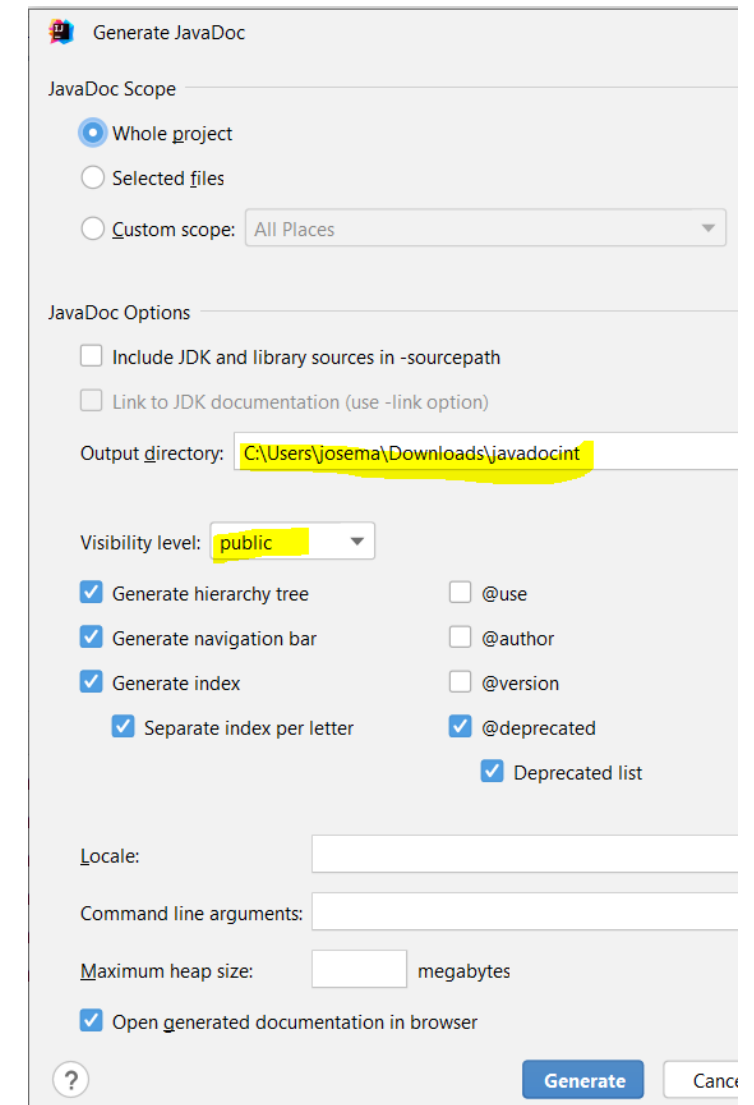
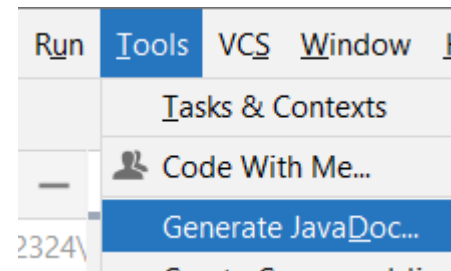
- Documenta las clases que hemos creado para el proyecto ed y genera un Javadoc

Si marcas el nombre de un método y pulsas **Alt+Enter** podrás crear comentarios básicos automáticamente



Genera la documentación con:

Tools -> Generate Javadoc



<https://www.jetbrains.com/help/idea/javadocs.html>