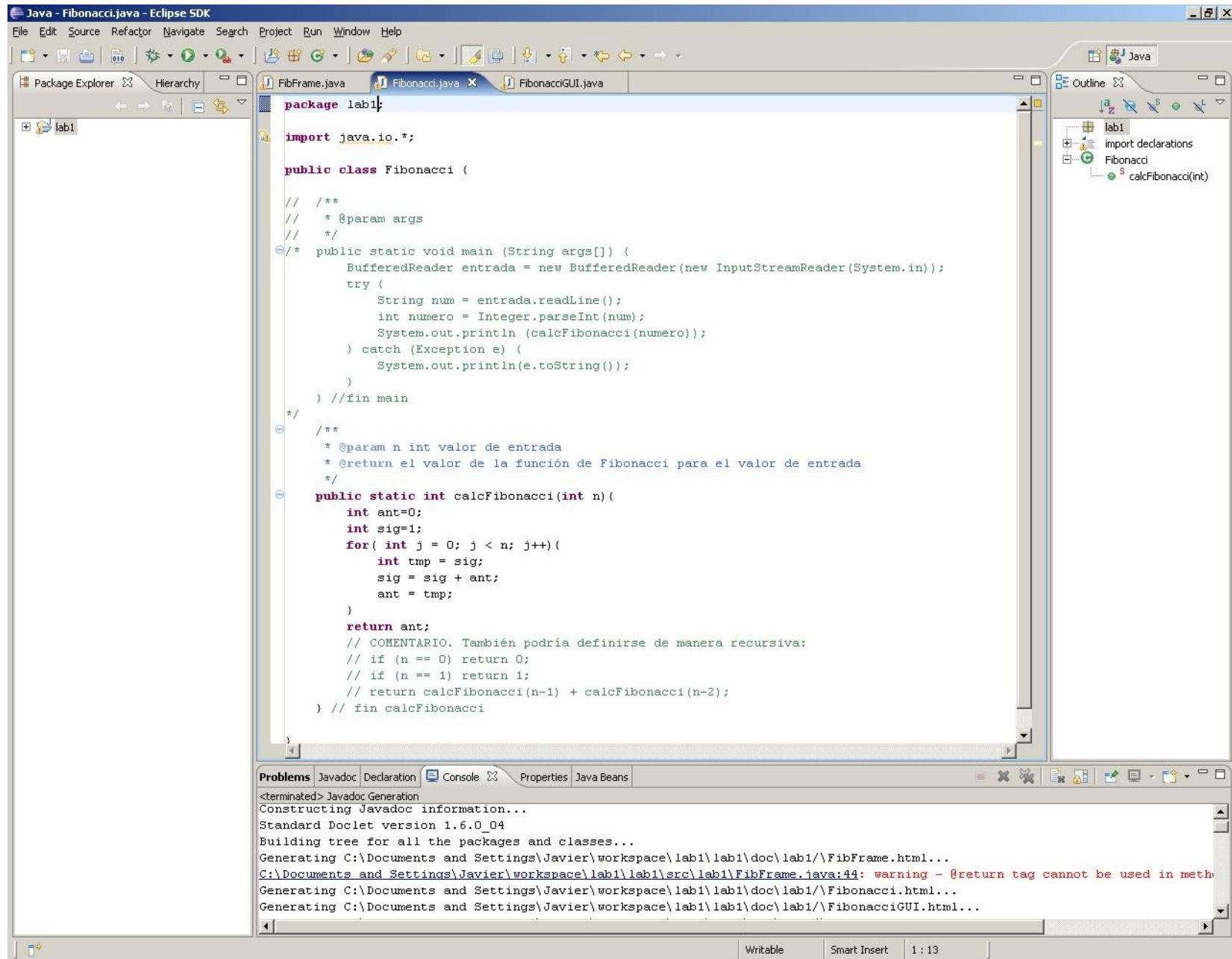


Auto-Documentación en Java: Javadoc

- La herramienta Javadoc genera de manera automática documentación a partir de los programas fuente Java. Para ello, estos programas deben tener comentarios y etiquetas entendidas por Javadoc `/** */`

Etiqueta	Quién lo utiliza	Objetivo
<code>@author name</code>	Clases e interfaces	Autor del código. Se pone una etiqueta para cada autor.
<code>@deprecated</code>	Clases y métodos	Método anticuado. Mejor no utilizarlo.
<code>@exception name description</code>	Métodos	Excepciones que el método puede elevar. Se pone una etiqueta para cada excepción posible.
<code>@param name description</code>	Métodos	Para describir los parámetros, su utilización y su tipo. Se pone una etiqueta para cada parámetro.
<code>@return description</code>	Métodos	Para describir los valores devueltos por cada método y su tipo.
<code>@since</code>	Clases y métodos	Desde qué versión está. Ej: desde JDK 1.1
<code>@see ClassName</code>	Clases, interfaces, métodos y atributos	Pondrá la dirección para conectarse con esta clase en la documentación.
<code>@see ClassName#NombreMétodo</code>	Clases, interfaces, métodos y atributos.	Pondrá la dirección para conectarse con este método en la documentación.
<code>@version text</code>	Clases, interfaces	Información acerca de la versión.

Código Java



The screenshot shows the Eclipse IDE with the following components:

- Package Explorer:** Shows a package named 'lab1'.
- Code Editor:** Contains the following Java code:

```
package lab1;

import java.io.*;

public class Fibonacci {

    /**
     * @param args
     */
    /* public static void main (String args[] ) {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        try {
            String num = entrada.readLine();
            int numero = Integer.parseInt(num);
            System.out.println (calcFibonacci(numero));
        } catch (Exception e) {
            System.out.println(e.toString());
        }
    } //fin main
    */

    /**
     * @param n int valor de entrada
     * @return el valor de la función de Fibonacci para el valor de entrada
     */
    /* public static int calcFibonacci(int n){
        int ant=0;
        int sig=1;
        for( int j = 0; j < n; j++){
            int tmp = sig;
            sig = sig + ant;
            ant = tmp;
        }
        return ant;
        // COMENTARIO. También podría definirse de manera recursiva:
        // if (n == 0) return 0;
        // if (n == 1) return 1;
        // return calcFibonacci(n-1) + calcFibonacci(n-2);
    } // fin calcFibonacci
    */
}
```
- Outline:** Shows the class structure: 'lab1' containing 'import declarations', 'Fibonacci', and 'calcFibonacci(int)'. The 'calcFibonacci(int)' method is currently selected.
- Problems Console:** Shows the following output:

```
<terminated> Javadoc Generation
Constructing Javadoc information...
Standard Doclet version 1.6.0_04
Building tree for all the packages and classes...
Generating C:\Documents and Settings\Javier\workspace\lab1\lab1\doc\lab1\FibFrame.html...
C:\Documents and Settings\Javier\workspace\lab1\lab1\src\lab1\FibFrame.java:44: warning - @return tag cannot be used in meth
Generating C:\Documents and Settings\Javier\workspace\lab1\lab1\doc\lab1\Fibonacci.html...
Generating C:\Documents and Settings\Javier\workspace\lab1\lab1\doc\lab1\FibonacciGUI.html...
```
- Status Bar:** Shows 'Writable', 'Smart Insert', and '1 : 13'.

Javadoc

The screenshot shows a Mozilla Firefox browser window with the title "Fibonacci - Mozilla Firefox". The address bar shows the file path: `file:///C:/Documents%20and%20Settings/Javier/workspace/lab1/lab1/doc/index.html`. The browser's menu bar includes "Archivo", "Editar", "Ver", "Historial", "Marcadores", "Herramientas", and "Ayuda". The search bar contains "Google".

The main content area displays the Javadoc for the `lab1.Fibonacci` class. On the left, there is a sidebar titled "All Classes" with links to `FibFrame`, `Fibonacci`, and `FibonacciGUI`.

The main content area is titled "lab1 Class Fibonacci". It shows the class hierarchy: `java.lang.Object` is the superclass, and `lab1.Fibonacci` is the subclass.

The class definition is shown as follows:

```
public class Fibonacci
extends java.lang.Object
```

The "Constructor Summary" section lists the constructor:

```
Fibonacci ()
```

The "Method Summary" section lists the static method:

```
static int calcFibonacci (int n)
```

The "Methods inherited from class java.lang.Object" section lists the following methods: `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`.

The "Constructor Detail" section shows the constructor signature:

```
public Fibonacci ()
```

The "Method Detail" section shows the signature for the `calcFibonacci` method:

```
public static int calcFibonacci (int n)
```

The "Parameters" section lists the parameter `n` as "int valor de entrada".

The "Returns" section lists the return value as "el valor de la función de Fibonacci para el valor de entrada".

The status bar at the bottom shows the file path: `file:///C:/Documents and Settings/Javier/workspace/lab1/lab1/doc/lab1/Fibonacci.html`.