



ENTORNOS DE  
DESARROLLO

# DESARROLLO DE UN PROGRAMA JAVA

IDE ECLIPSE VS TERMINAL UBUNTU

Alberto Martínez Pérez

1º CFGS DESARROLLO DE APLICACIONES WEB (DAW)

## ÍNDICE

<b>1. DESARROLLO JAVA EN ECLIPSE.....</b>	<b>2</b>
a. ESCRITURA DEL CÓDIGO FUENTE .....	2
b. COMPILACIÓN DEL CÓDIGO FUENTE .....	2
c. EJECUCIÓN DEL PROGRAMA JAVA .....	2
<b>2. DESARROLLO JAVA EN TERMINAL UBUNTU.....</b>	<b>3</b>
a. ESCRITURA DEL CÓDIGO FUENTE .....	3
b. COMPILACIÓN DEL CÓDIGO FUENTE .....	4
c. EJECUCIÓN DEL PROGRAMA JAVA .....	4
<b>3. CONCLUSIONES.....</b>	<b>5</b>

## ÍNDICE DE ILUSTRACIONES

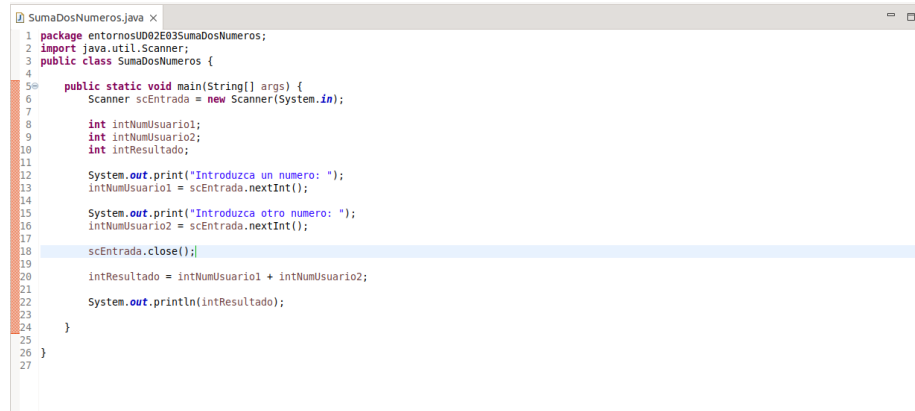
<i>Fig. 1 Código fuente en el IDE Eclipse. ....</i>	<i>2</i>
<i>Fig. 2 Botón de ejecución en el IDE Eclipse.....</i>	<i>2</i>
<i>Fig. 3 Consola del IDE Eclipse ejecutando un programa. ....</i>	<i>2</i>
<i>Fig. 4 Editor de texto nano con el código fuente del programa SumaDosNumeros.java. ....</i>	<i>3</i>
<i>Fig. 5 Editor de texto con el código fuente guardado y generado en un archivo .java (SumaDosNumeros.java).....</i>	<i>3</i>
<i>Fig. 6 Uso del comando javac nombreArchivo.java para la compilación del código fuente.....</i>	<i>4</i>
<i>Fig. 7 Utilización del comando java nombreArchivo para la ejecución del programa java.....</i>	<i>4</i>



# 1. DESARROLLO JAVA EN ECLIPSE

## a. ESCRITURA DEL CÓDIGO FUENTE

Tras crear un proyecto y una clase dentro de este pasamos a escribir el código fuente en el editor que se ha abierto. En este caso hemos realizado un programa simple que pedirá dos números por consola y que realizará la suma de ambos (fig.1).



```
1 package entornosUD02E03SumaDosNumeros;
2 import java.util.Scanner;
3 public class SumaDosNumeros {
4
5     public static void main(String[] args) {
6         Scanner scEntrada = new Scanner(System.in);
7
8         int intNumUsuario1;
9         int intNumUsuario2;
10        int intResultado;
11
12        System.out.print("Introduzca un numero: ");
13        intNumUsuario1 = scEntrada.nextInt();
14
15        System.out.print("Introduzca otro numero: ");
16        intNumUsuario2 = scEntrada.nextInt();
17
18        scEntrada.close();
19
20        intResultado = intNumUsuario1 + intNumUsuario2;
21
22        System.out.println(intResultado);
23
24    }
25
26 }
27 }
```

Fig. 1 Código fuente en el IDE Eclipse.

## b. COMPILACIÓN DEL CÓDIGO FUENTE

En el caso de los IDEs la compilación de código se produce al mismo tiempo de la ejecución por lo que para compilar el código y crear el archivo .class simplemente deberemos ejecutar el programa. Para ello pulsamos en el botón “Run” de la barra de botones del IDE (fig. 2).

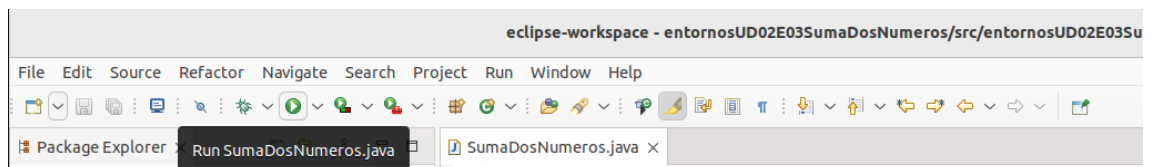


Fig. 2 Botón de ejecución en el IDE Eclipse.

## c. EJECUCIÓN DEL PROGRAMA JAVA

La ejecución se produce al mismo momento de la compilación por lo que en la consola del IDE se nos ejecutará el programa y nos solicitará los diferentes datos que necesite para ejecutar los procesos (fig. 3).

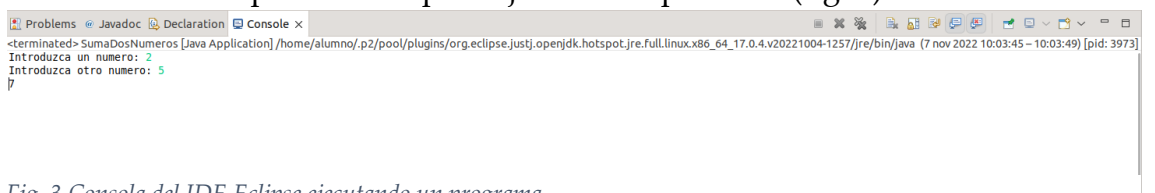


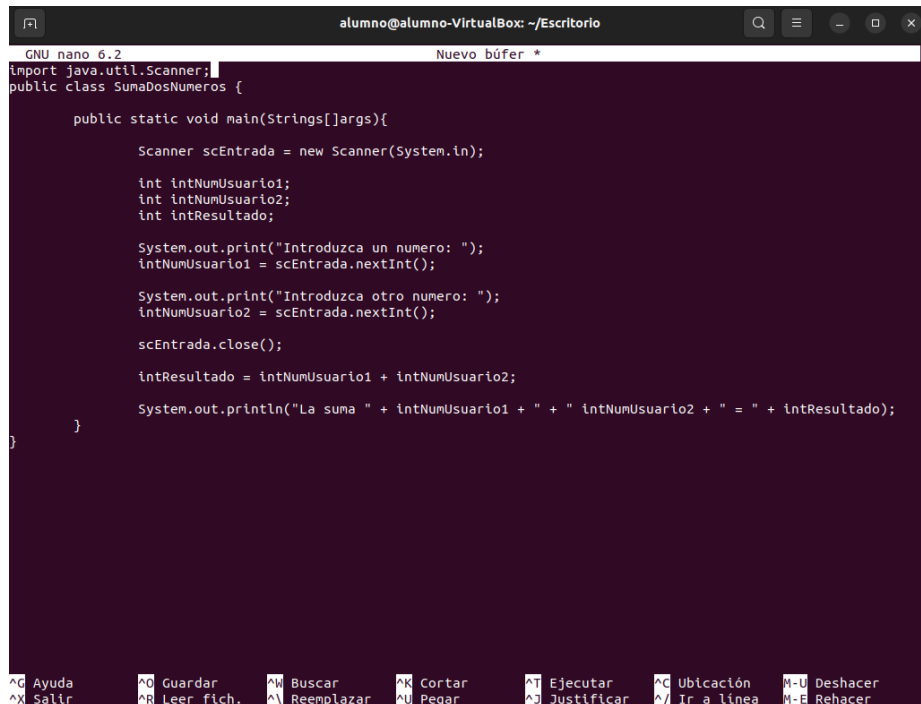
Fig. 3 Consola del IDE Eclipse ejecutando un programa.

## 2. DESARROLLO JAVA EN TERMINAL UBUNTU

### a. ESCRITURA DEL CÓDIGO FUENTE

En el caso de querer usar la terminal de Ubuntu para el desarrollo Java deberemos usar la terminal de Ubuntu y usar un editor de texto, por ejemplo, nano.

En nano escribimos el código fuente del programa que queremos crear (fig. 4) y guardamos el archivo con extensión .java (fig. 5).



```
GNU nano 6.2                               Nuevo búfer *
import java.util.Scanner;
public class SumaDosNumeros {

    public static void main(Strings[]args){

        Scanner scEntrada = new Scanner(System.in);

        int intNumUsuario1;
        int intNumUsuario2;
        int intResultado;

        System.out.print("Introduzca un numero: ");
        intNumUsuario1 = scEntrada.nextInt();

        System.out.print("Introduzca otro numero: ");
        intNumUsuario2 = scEntrada.nextInt();

        scEntrada.close();

        intResultado = intNumUsuario1 + intNumUsuario2;

        System.out.println("La suma " + intNumUsuario1 + " + " + intNumUsuario2 + " = " + intResultado);

    }
}
```

Fig. 4 Editor de texto nano con el código fuente del programa SumaDosNumeros.java.

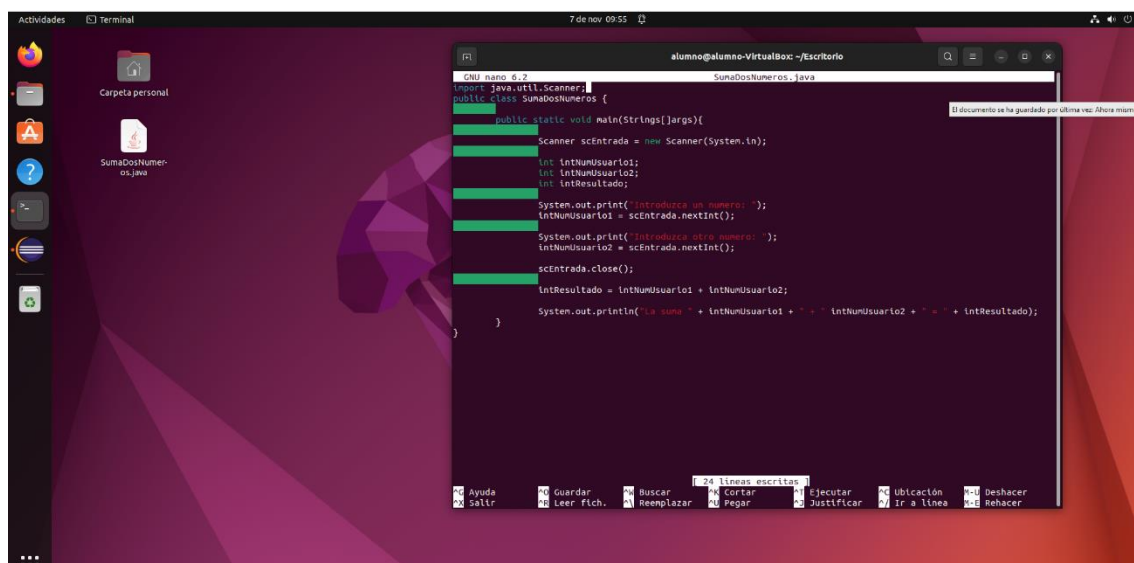


Fig. 5 Editor de texto con el código fuente guardado y generado en un archivo .java (SumaDosNumeros.java).

## b. COMPILACIÓN DEL CÓDIGO FUENTE

Para compilar un código fuente Java debemos utilizar el comando `javac` `nombreArchivo.java` (en este caso `javac SumaDosNumeros.java`). Esto dará lugar a un archivo `.class` que es el archivo bytecode que será interpretado por la máquina virtual de Java.

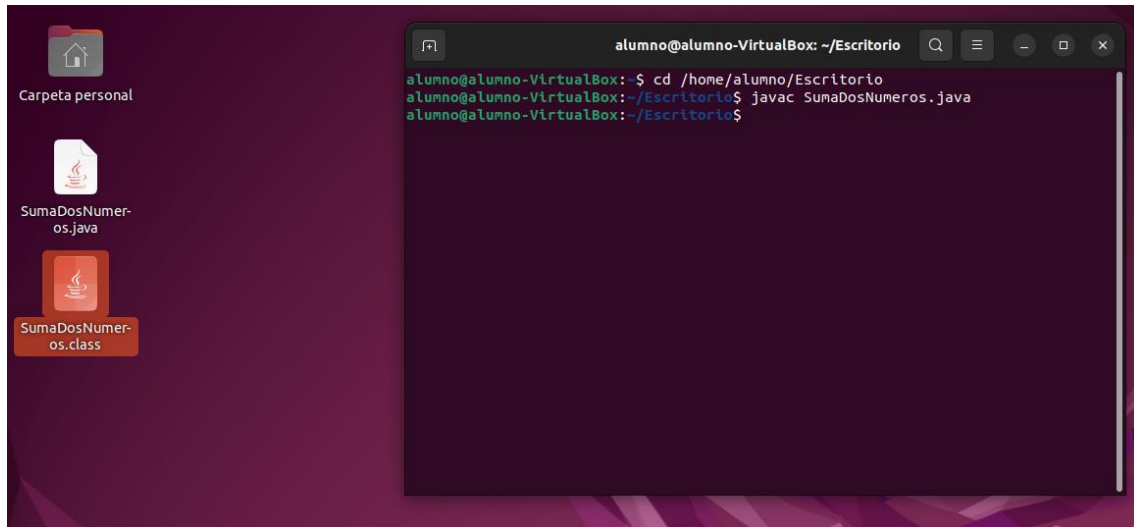


Fig. 6 Uso del comando `javac` `nombreArchivo.java` para la compilación del código fuente.

## c. EJECUCIÓN DEL PROGRAMA JAVA

Para utilizar un programa Java (es decir, para que la JVM pueda interpretar el *bytecode* generado en la compilación) a través de la terminal de comandos debemos usar el comando `java` `nombreArchivo` (en este caso `java SumaDosNumeros`) y con ello se iniciará la interpretación.

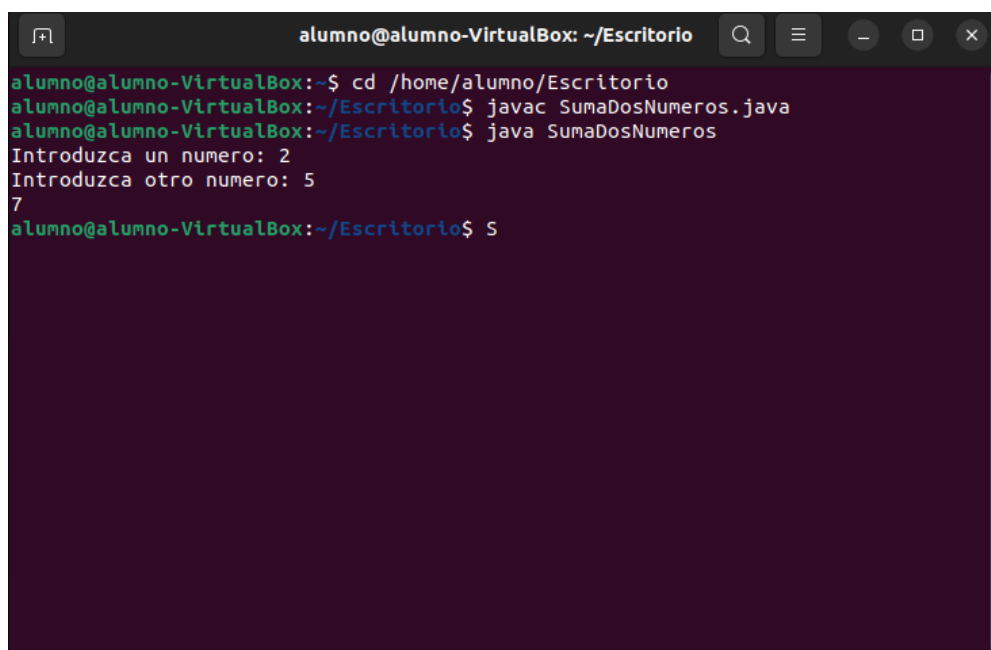


Fig. 7 Utilización del comando `java` `nombreArchivo` para la ejecución del programa `java`.

### 3. CONCLUSIONES

Como se ha podido ver es posible escribir, compilar y ejecutar un programa Java tanto a través de un IDE de Java (en este caso Eclipse) como a través de la terminal de comandos de Ubuntu usando para ello un procesador o editor de texto (en este caso nano), pero la manera de hacerlo difiere en algunos puntos.

Por ejemplo, la utilización de un IDE para la escritura del código tiene muchas ventajas ya que estos programas nos facilitan la escritura del código gracias a mecánicas como:

- El autocompletado (por ejemplo, al abrir una llave ({} se crea automáticamente una llave de cierre (}) al hacer un salto de línea).
- La predicción de opciones de métodos (por ejemplo, al escribir "*System.out.*" el IDE nos devolverá los diferentes tipos de métodos que podemos usar en ese caso lo cual reduce en gran medida los errores de sintaxis).
- El aviso de errores y *warnings* en el panel lateral y en consola (por ejemplo, si dejamos un objeto scanner sin cerrar, el IDE nos avisará de ello o si hemos cometido un error sintáctico como escribir "*Sistem*" en lugar de "*System*" el IDE también nos comunicará el error), etc.

Estas mecánicas no existen si utilizamos un editor de texto desde la terminal de Ubuntu para la escritura del código y tendremos que escribir el código sin ningún tipo de ayuda algo que en un programa simple como el de este caso puede no ser un gran problema, pero en un programa con varios módulos y/o clases sí que podría resultar en un problema importante.

Además, a la hora de compilar y ejecutar el programa Java también habrá diferencias en función de la opción que elijamos.

Si utilizamos un IDE:

- Se compilará y ejecutará el programa al mismo tiempo simplemente pulsando el botón de ejecución.
- El IDE ordenará automáticamente los archivos .java y .class en carpetas correspondientes sin necesidad de que el programador se preocupe de ello.
- No será necesaria la instalación del JDK en el sistema ya que será el propio IDE el que funcionará como compilador (Javac) y como máquina virtual (JVM).

Si utilizamos la terminal:

- Será necesario compilar y ejecutar en dos pasos diferentes. Primero habrá que compilar el archivo .java para generar el archivo .class correspondiente y, a continuación, ejecutar ese .class.
- Habrá que tener cuidado con la localización de los archivos.
- Para la compilación y ejecución del programa será necesario el haber instalado previamente el JDK ya que necesitaremos tener tanto el compilador de Java como la máquina virtual de Java en el sistema.