

## PRACTICA 3.1: DEPURADOR DE NETBEANS

Abrimos Eclipse y creamos un proyecto Java. Añadimos una clase llamada LlenarNumeros y escribimos en su código fuente el siguiente texto:

```
package llenarnumeros;
public class LlenarNumeros {

    public static void main(String[] args) {
        int n=5;
        int tabla[] = new int [n];
        tabla = llenar(n);
        int suma = sumar(tabla);

        System.out.println("La suma es "+suma);
    }

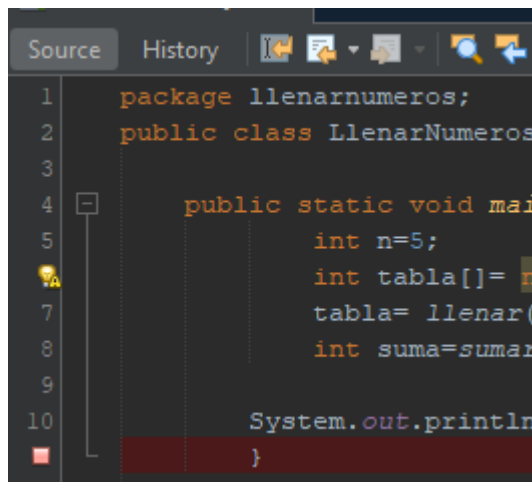
    private static int [] llenar (int n) {
        int tabla[] = new int [n];
        for (int i=0; i<n; i++)
            tabla[i] = i*10;
        return tabla;
    }

    private static int sumar (int [] tabla) {
        int suma = 0;
        int n = tabla.length;
        for(int i=0; i<n; i++)
            suma = suma + tabla[i];
        return suma;
    }
}
```

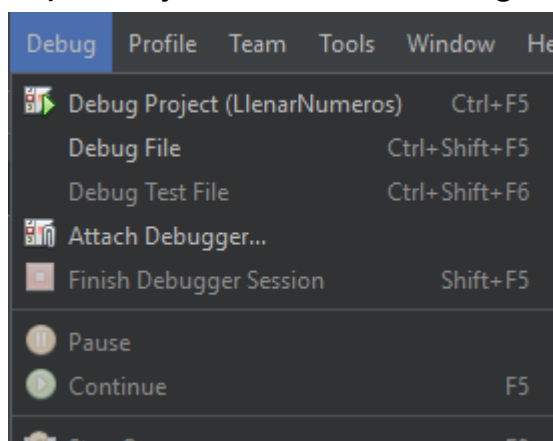
En ese programa se definen dos métodos además del método `main()` (que es el método principal a partir del cual se desarrollan las operaciones descritas en el programa). El primer método, llamado `llenar()`, recibe un número entero `n` y devuelve un vector con `n` números enteros. El segundo método, llamado `sumar()`,

recibe un vector de números enteros, suma sus elementos y devuelve esa suma.

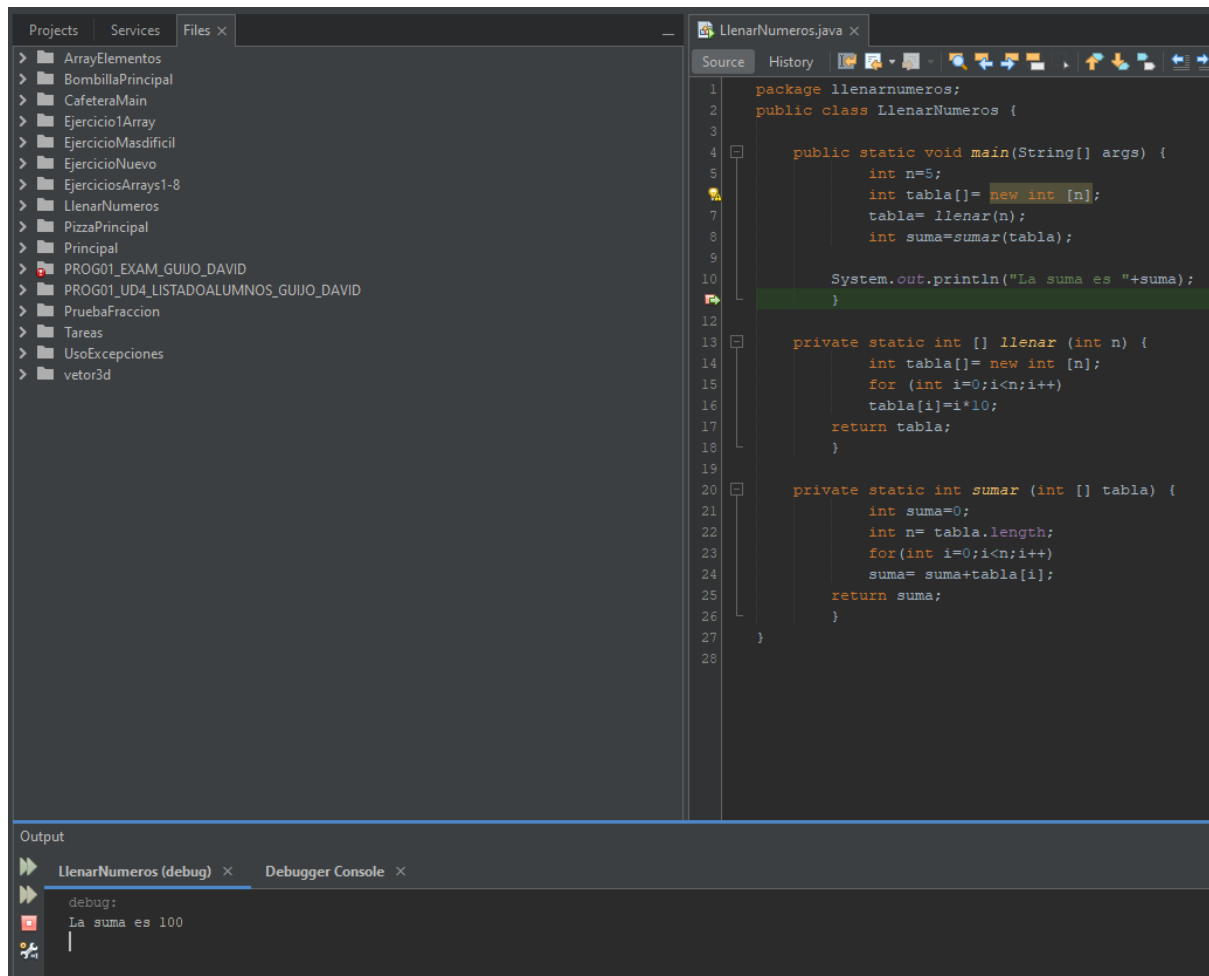
Para comenzar con el proceso de depuración debemos de hacer clic izquierdo en la parte izquierda, donde se muestran los números de cada línea, esta será el breakpoint, es decir que el programa se depurará hasta esa línea que hemos establecido.



Después de esto clicamos en la pestaña Debug ubicado en la parte superior y le damos a Debug File



El programa se ejecutará hasta el breakpoint (punto de ruptura) que hayamos establecido. Al llegar al punto de ruptura el programa se detendrá



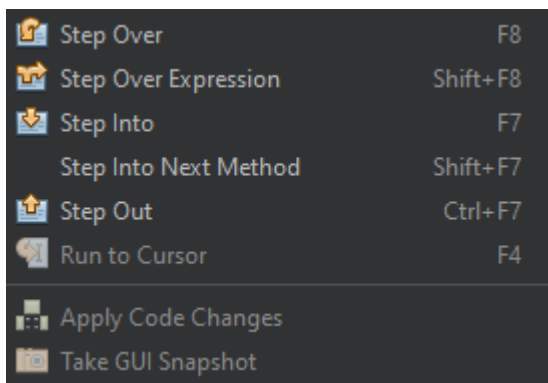
A continuación podremos usar los siguientes botones:

-Step Over: Para ejecutar paso a paso cada instrucción pero “saltándose” las llamadas a métodos (es decir, no entrando en las instrucciones de estos)

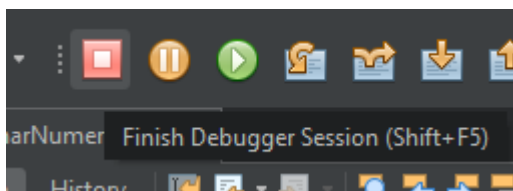
-Step Into: Para ejecutar el programa paso a paso, instrucción por instrucción.

-Step Out: Ejecuta una línea de código. Si la línea de código actual se encuentra dentro de un método, se ejecutarán todas las instrucciones que queden del método y se vuelve a la instrucción desde la que se llamó al método.

-Return to Cursor: Se ejecuta el programa hasta la instrucción donde se encuentra el cursor.

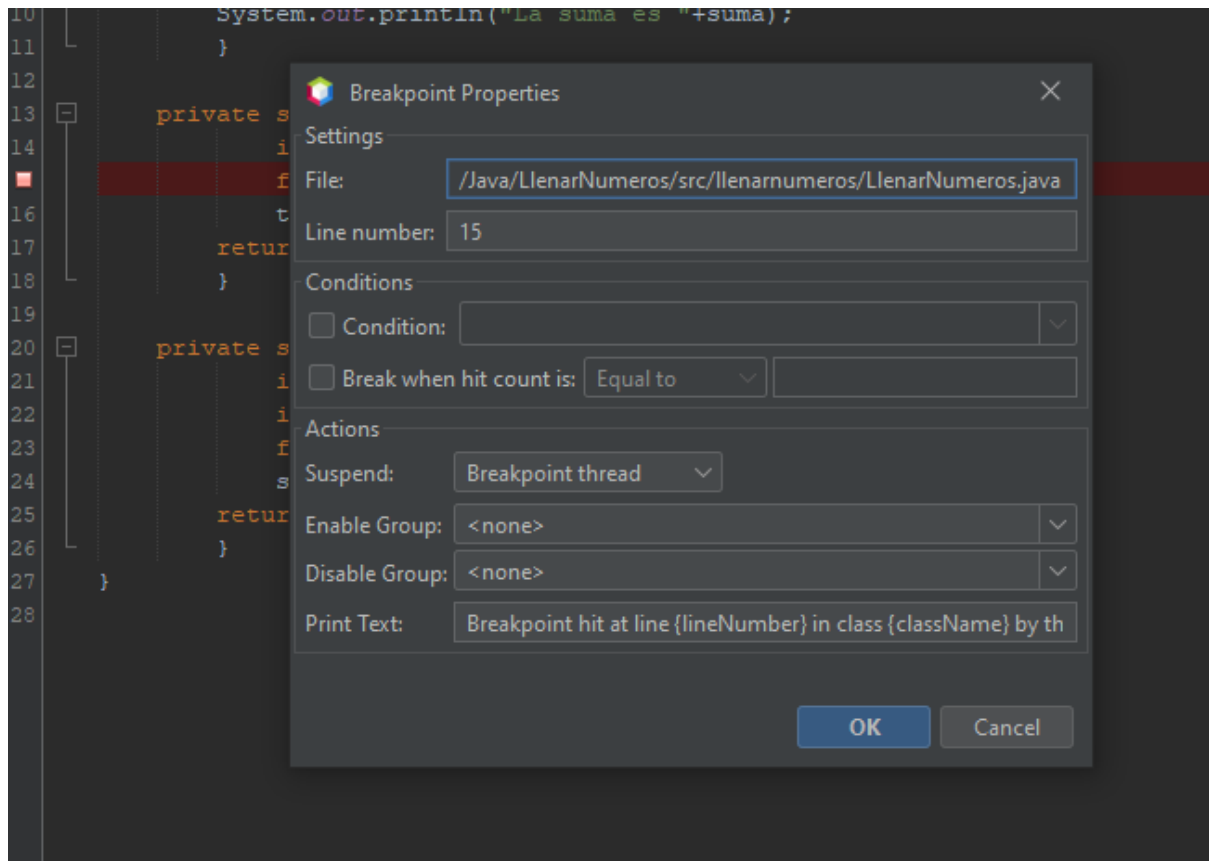


En cualquier momento podemos finalizar la depuración clicando en Finish Debugger Session

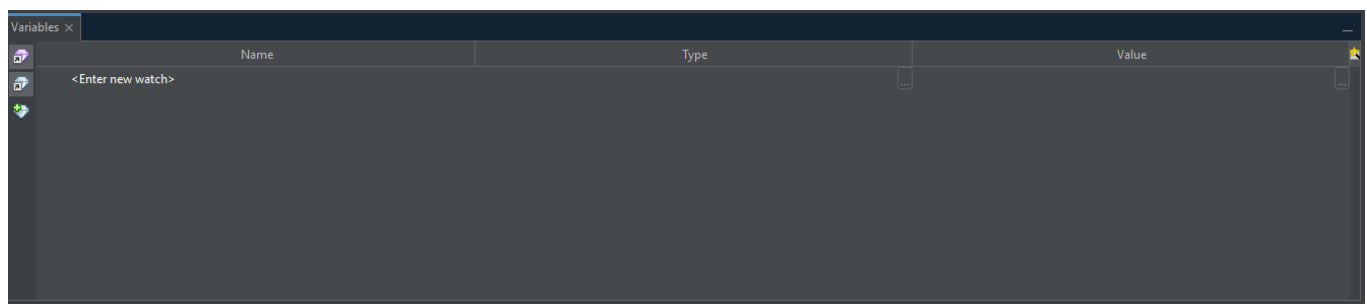


Podemos eliminar el breakpoint dando clic izquierdo sobre el mismo punto rojo que hemos establecido

Para establecer breakpoints condicionales debemos de hacer clic derecho en el punto de ruptura establecido y le damos a Breakpoint → Properties



Si clicamos en Window situado en la parte superior y le damos a Debugging y clicamos en variables podemos modificar los valores de las variables



En definitiva, el debugger es una herramienta muy útil de Netbeans el cual nos proporciona funciones como la de ejecutar el programa paso a paso permitiéndonos recuperar el control tras cada breakpoint, deteniendo el programa tras esta línea de ruptura. Además, podemos conocer el valor de las variables a cada paso