

## TRABAJO DE INVESTIGACIÓN ED03

### 1.-Introducción.

En la presente unidad de trabajo se ha estudiado hacer probar un programa con el fin de comprobar que funciona perfectamente.

### 2.-Práctica.

- ▮ Busca cuales son los IDE más usados actualmente (que sirvan para programar en Java).
- ▮ Elige uno de ellos y haz una guía para manejar el depurador.
- ▮ Comprueba cuales son todas las posibilidades que ofrece.
- ▮ No puede ser Eclipse.



### NetBeans

NetBeans es un entorno de desarrollo integrado (IDE) basado en código abierto. Este IDE, gracias a su enfoque, permite **desarrollar nuestras aplicaciones como módulos** (como un componente de un software). Lo cual resulta sumamente cómodo pues podemos **reutilizar el código** para diferentes proyectos, las veces que queramos sin mayor complicación. Además es el IDE oficial de Java, pero también podemos desarrollar en otros **lenguajes como PHP, C, C++ e incluso HTML 5**.

Entre sus detractores, hay quien comenta que el desarrollo de una GUI sobre NetBeans es algo complicado y que no cuenta con la ingente cantidad de plugins y extensiones de las que sí cuenta Eclipse, además de **carecer de soporte nativo para trabajar con webapps** (.war, .jsp, y servlets).

### Depuración en Netbeans

La depuración permite **examinar las aplicaciones para buscar errores**, ya que posibilita observar las líneas que se van ejecutando, así como los valores que van tomando las variables en cada paso.

Para realizar la depuración de un programa, se debe establecer **en primer lugar un punto de interrupción** donde debe pararse la ejecución de la aplicación. Esto se consigue con alguna de las siguientes acciones sobre la línea de código en la que se desee establecer el punto de interrupción:

- Clic en el margen izquierdo
- Menú contextual > "Ocultar/Mostrar línea de punto de interrupción"
- Pulsando la combinación de teclas: Ctrl + F8
- Menú "Depurar > Ocultar/Mostrar línea de punto de interrupción"
- Al realizar alguna de esas acciones, se marca en **color rosado la línea que se ha convertido en un punto de interrupción**, y se muestra un pequeño cuadrado en el margen izquierdo.

```

package ejemplodebug;

public class Main {

    public static void main(String[] args) {
        int numMax = 100;
        int numActual = 1;
        int incremento = 1;
        while (numActual <= numMax)
        {
            System.out.println(numActual);
            numActual+=incremento;
            incremento++;
        }
    }
}

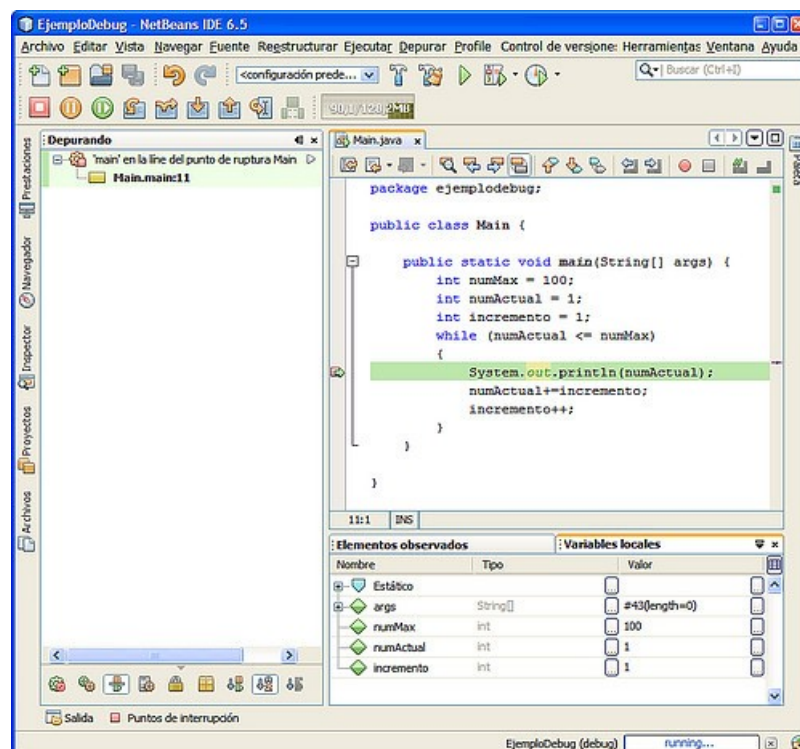
```

Una vez establecido **al menos un punto de interrupción**, se debe **ejecutar la aplicación en modo depuración**. Esto se puede llevar a cabo **sobre el proyecto o sólo sobre el archivo actual**:

- Depurar archivo actual:
- Menú contextual > "Debug nombreArchivo"
- Menú "Depurar > Debug nombreArchivo"
- Pulsando la combinación de teclas: Ctrl + Mayúsculas + F5
- Depurar proyecto:
- Menú "Depurar > Debug Main Project"
- Icono "Debug Main Project"



Al llegar la **ejecución de la aplicación al punto de interrupción establecido**, se **destaca en color verde** la línea de código en la que se encuentre la ejecución.



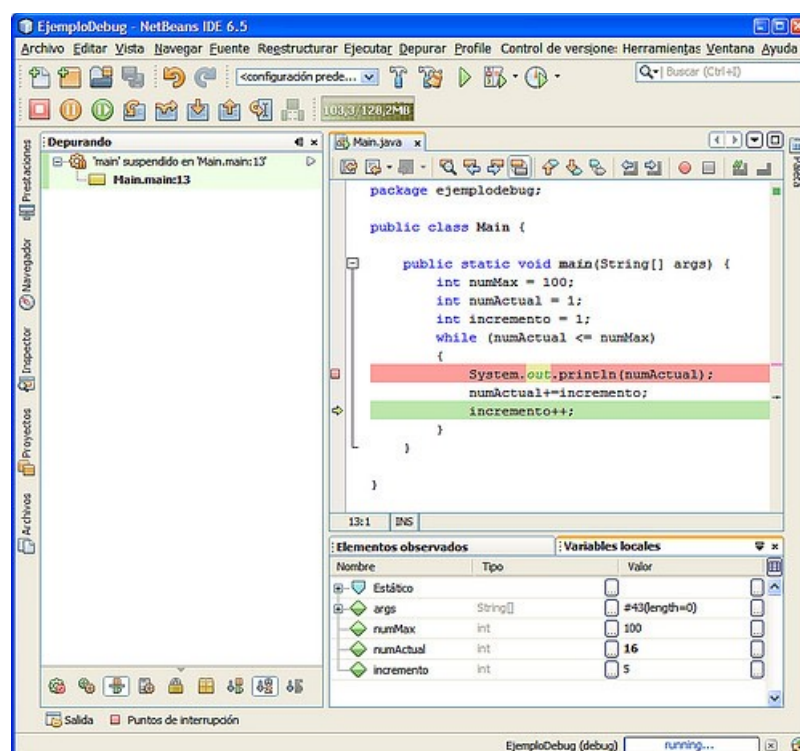
En la parte inferior se muestra la **ventana de observación de las variables locales**, en la que se puede comprobar el valor que va tomando cada variable.

A partir del momento en que se para la ejecución del programa **se puede continuar con la ejecución línea a línea** utilizando la opción "**Continuar ejecución**":

- Pulsando la tecla F8
- Menú "Depurar > Continuar ejecución"
- Icono "Continuar ejecución"



Así **se va mostrando en verde la línea que se va ejecutando en cada momento**, y se van actualizando en la ventana inferior los valores que van tomando las variables.



En caso de que se desee que **el programa continúe la ejecución sin más paradas**, se dispone de la opción "**Continuar**":

- Pulsando la tecla F5
- Menú "Depurar > Continuar"
- Icono "Continuar"

