

# JAVA para No Programadores

Módulo 5

# Debugging

# Debugging

A la hora de encontrar y resolver problemas en nuestros programas es crucial que seamos hábiles usando el **debugger**.

Es una herramienta simple integrada en cualquier IDE que nos permite definir un *breakpoint* (punto de parada) y ejecutar nuestro programa en modo debug (depuración). Al hacer esto, cuando el programa se detenga podremos ir a paso a paso por nuestro programa y si llegamos a una línea que sea una función/procedimiento, ejecutarla línea por línea para ir viendo cómo se modifican el valor de las variables y las acciones que realiza nuestro programa.

## Características

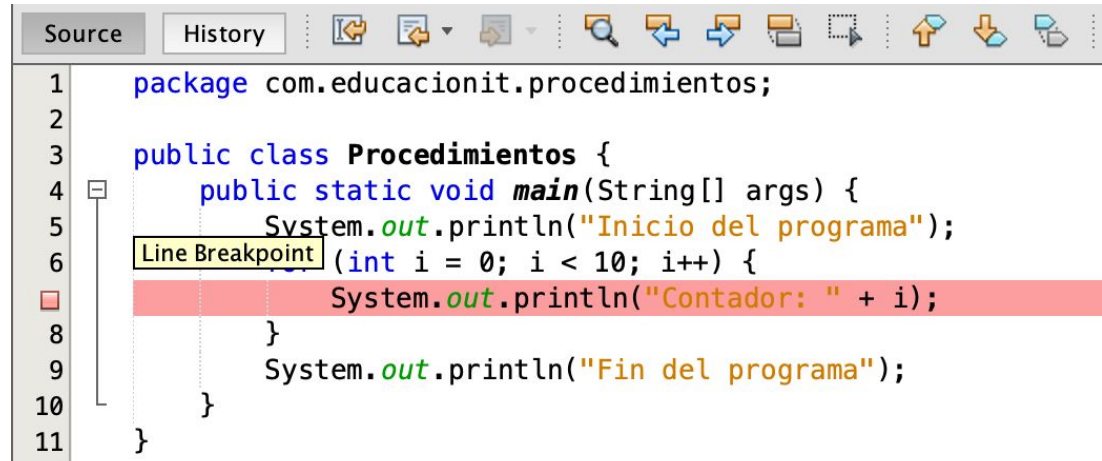
- Deben retornar un valor.
- Pueden recibir parámetros, aunque no es obligatorio es recomendado.

## Pasos

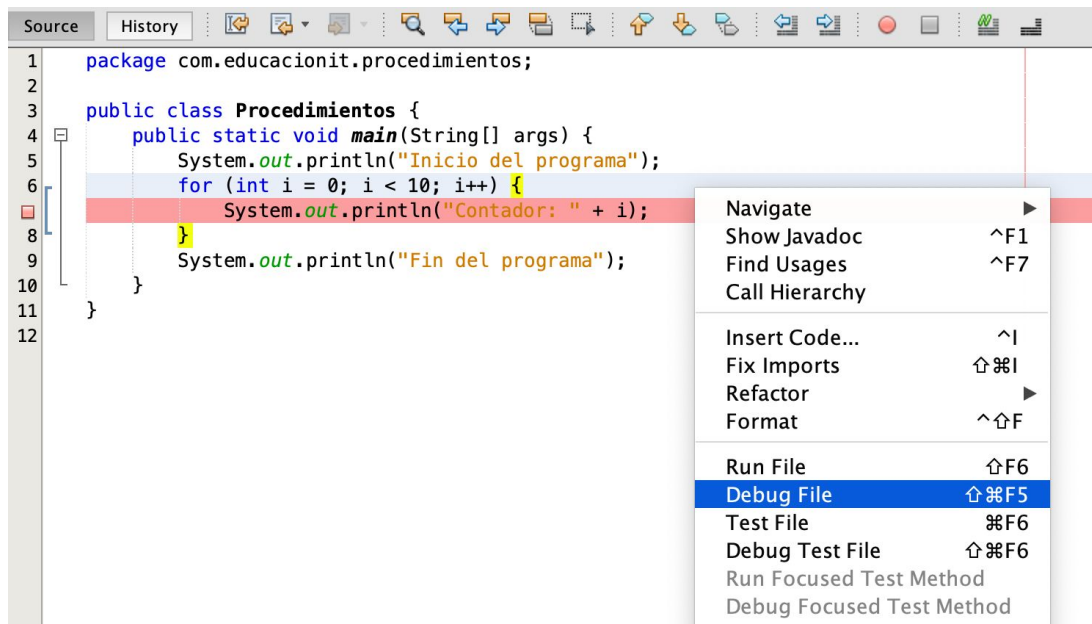
- Agregar uno o más breakpoints.
- Iniciar el programa en modo debug.
- Ejecutar nuestro programa línea por línea.

## Agregar breakpoint

En el lado izquierdo del editor de código, vemos el número de cada línea. Al hacer click sobre el número podemos indicar que deseamos agregar un breakpoint en la línea correspondiente.

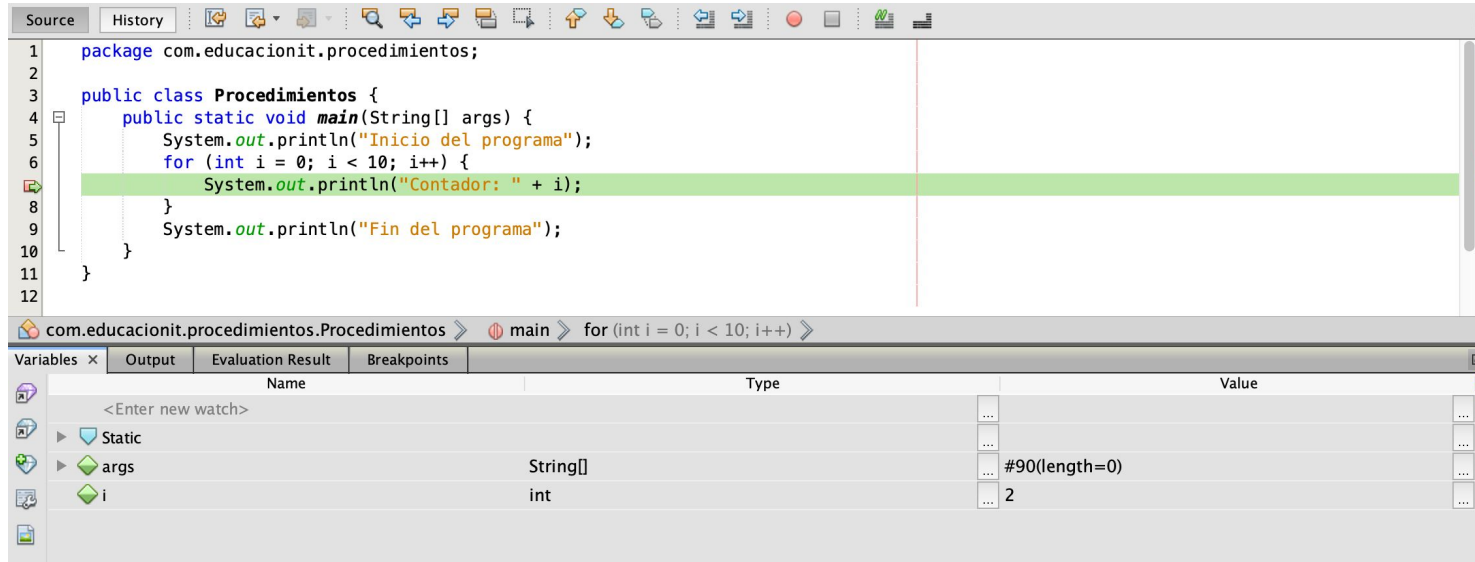


# Modo debug



Al apretar el botón derecho para desplegar el menú de opciones, vemos una que se llama **“Debug File”**, la cual sirve para ejecutar el programa en modo debug (o sea, frenando el programa cuando se llega a una línea que tenga un breakpoint).

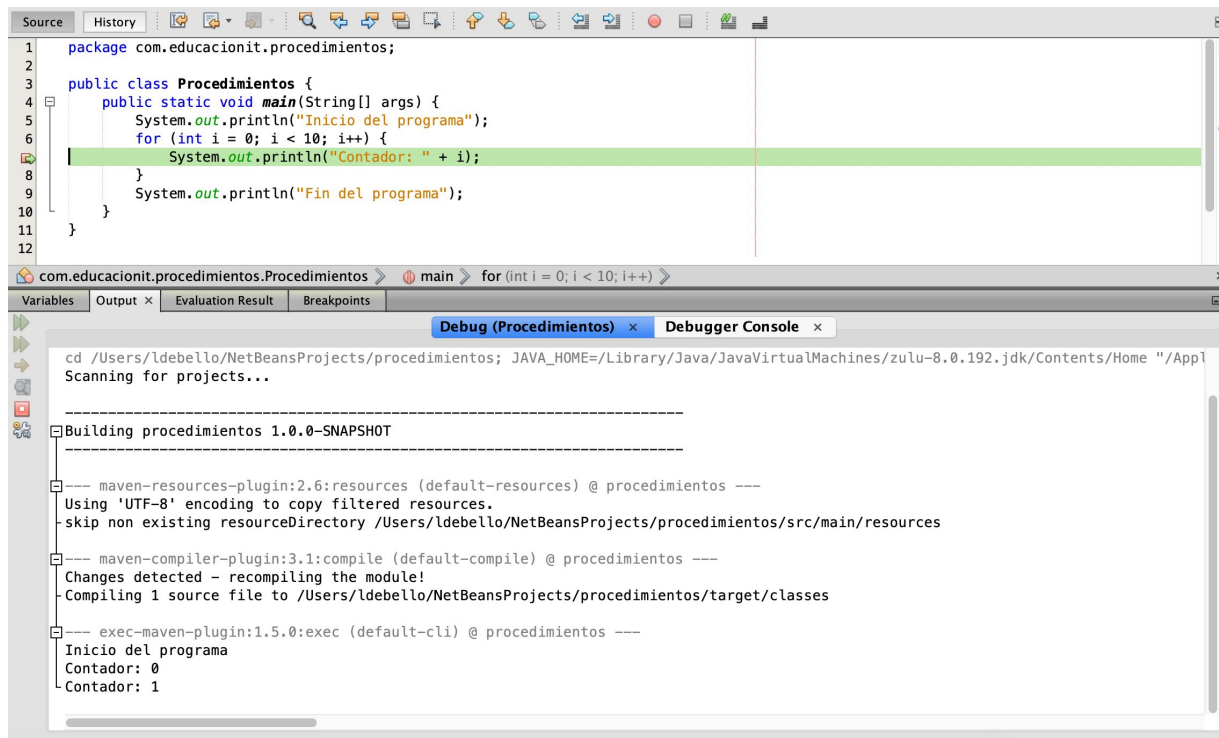
# Variables



Al ejecutar el programa en modo debug, veremos algunos tabs, por ej. “Variables”, en el cual podremos ver el valor actual de cada variable.

Si este tab no estuviera presente podemos ir a “Window → Debugging → Variables” (Esto varía entre los distintos IDE).

# Output



The screenshot displays an IDE window with two main panes. The top pane shows the source code of a Java class named `Procedimientos` in the package `com.educacionit.procedimientos`. The code includes a `main` method that prints "Inicio del programa", loops from `i = 0` to `i = 9` printing "Contador: " followed by `i`, and finally prints "Fin del programa". The line `System.out.println("Contador: " + i);` is highlighted in green. The bottom pane shows the "Debugger Console" tab, which contains the execution output. The output starts with the command prompt path, followed by build logs for Maven plugins, and then the program's execution output: "Inicio del programa", "Contador: 0", and "Contador: 1".

```
1 package com.educacionit.procedimientos;
2
3 public class Procedimientos {
4     public static void main(String[] args) {
5         System.out.println("Inicio del programa");
6         for (int i = 0; i < 10; i++) {
7             System.out.println("Contador: " + i);
8         }
9         System.out.println("Fin del programa");
10    }
11 }
12
```

com.educacionit.procedimientos.Procedimientos > main > for (int i = 0; i < 10; i++) >

Variables × Output × Evaluation Result × Breakpoints ×

Debug (Procedimientos) × Debugger Console ×

```
cd /Users/ldebello/NetBeansProjects/procedimientos; JAVA_HOME=/Library/Java/JavaVirtualMachines/zulu-8.0.192.jdk/Contents/Home "/AppI
Scanning for projects...

-----
[×] Building procedimientos 1.0.0-SNAPSHOT
-----

[×] --- maven-resources-plugin:2.6:resources (default-resources) @ procedimientos ---
Using 'UTF-8' encoding to copy filtered resources.
-skip non existing resourceDirectory /Users/ldebello/NetBeansProjects/procedimientos/src/main/resources

[×] --- maven-compiler-plugin:3.1:compile (default-compile) @ procedimientos ---
Changes detected - recompiling the module!
Compiling 1 source file to /Users/ldebello/NetBeansProjects/procedimientos/target/classes

[×] --- exec-maven-plugin:1.5.0:exec (default-cli) @ procedimientos ---
Inicio del programa
Contador: 0
Contador: 1
```

Además, vemos un tab llamado *“Output”*, que nos va mostrando las cosas que fue escribiendo nuestro programa a la fuente de salida, en este caso la consola. Si este tab no estuviera presente podemos ir a *“Window → Output”* (Esto varía entre los distintos IDE).

# Línea a línea

Depurar nos permite ejecutar nuestro programa **línea a línea**. Para ello, cada IDE usa algún botón o combinación de teclas para avanzar a la línea siguiente. En NetBeans, esa tecla es F8. Podemos consultar esta tecla en “*Debug* → *Step Over*”.



# ¡Muchas gracias!

¡Sigamos trabajando!