

DEBUGGING

- encuentra, repara y previene errores -

ÍNDICE DE CONTENIDOS

- ¿Qué es depurar (DEBUG) ?.
- Fases de la Depuración.
- Reproducción (Reproduce).
- Diagnóstico (Diagnosis).
- Corrección (Fix).
- Reflexión (Reflect).

ÍNDICE DE CONTENIDOS

- **Depuración con IntelliJ.**
- **Referencia.**

¿Qué es depurar?

“ Depurar es **CORREGIR** los errores que existen en nuestro código **COMPRENDIENDO** la raíz del problema que los ha causado. ”

Fases de la depuración

Reproducción

Ser capaces de que vuelva a suceder el problema que ha aparecido.

Diagnóstico

Establecer y probar hipótesis sobre cuál es la causa que ha provocado el error.

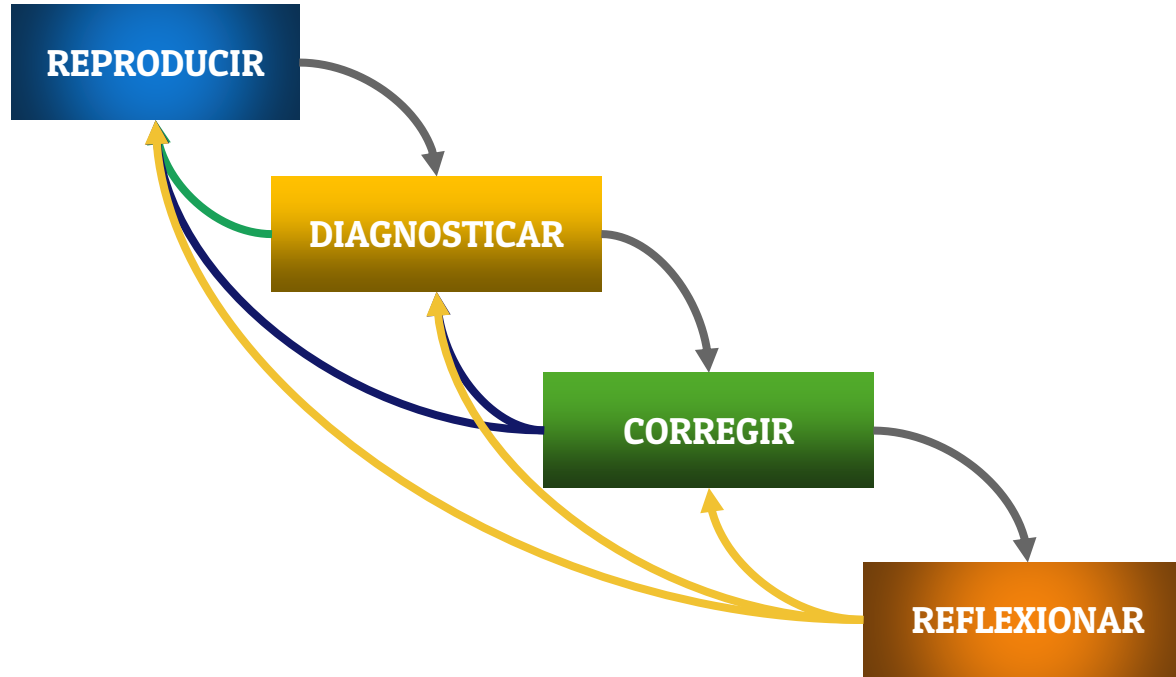
Corrección

Realizar los cambios que corrigen el error aparecido.

Reflexión

Aprender la lección y aplicar el conocimiento a problemas similares..

Fases de la depuración



Reproducción

“ Si no puedo reproducir el problema no podré solucionarlo. Pero volverá a aparecer cuando menos te lo esperas.... ”

VARIABLES A CONTROLAR

Software

- Mismo código.
- Mismo compilador (JDK???)
- Mismas librerías.

Entorno

Cualquier cosa que puede afectar al comportamiento:
SOs, hardware
drivers

Entradas

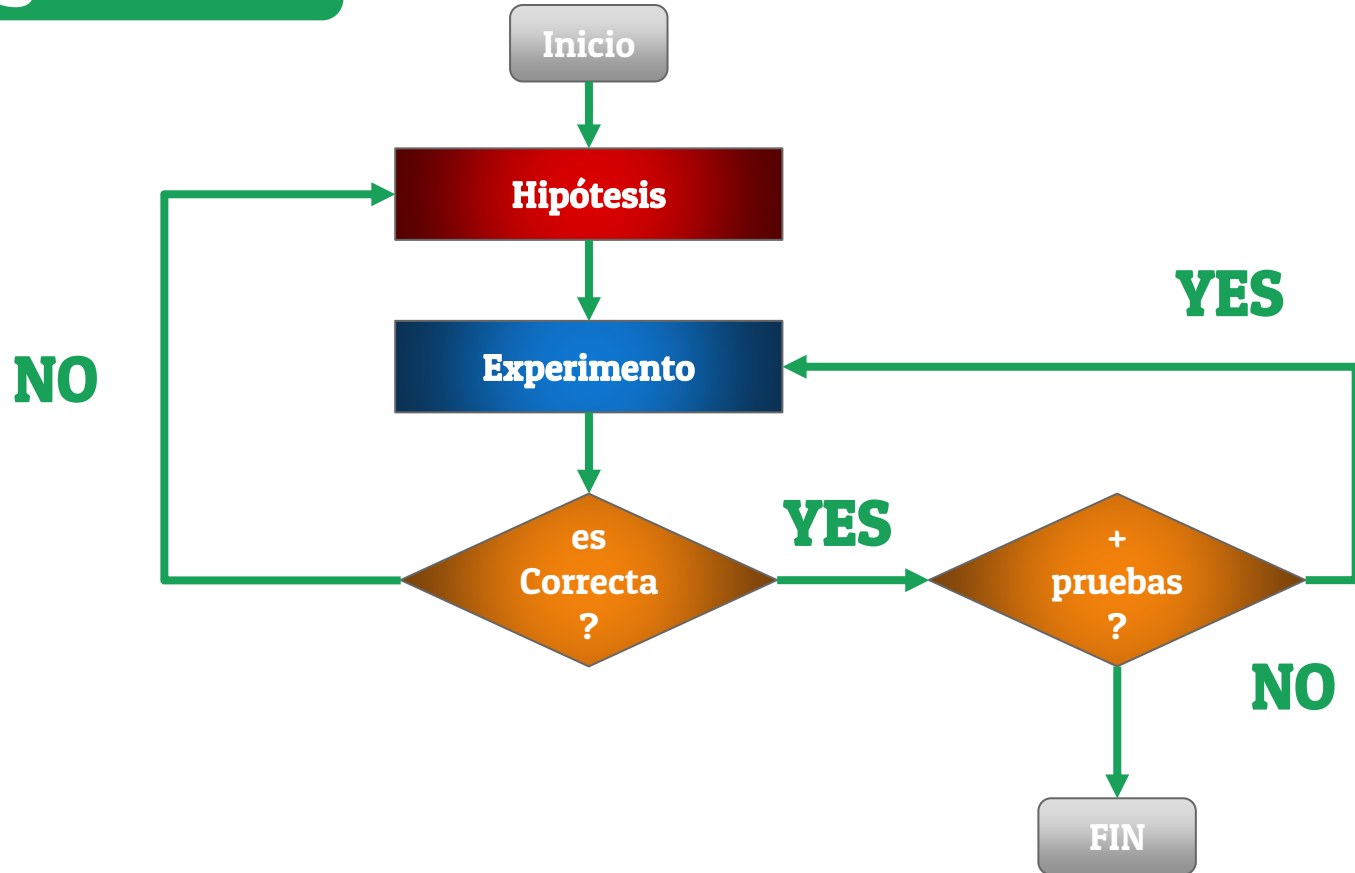
Información del usuario, del ratón etc...

Diagnóstico

“ El proceso de diagnóstico se realiza en nuestro cerebro, estableciendo hipótesis sobre cuáles deben ser los problemas. ”

**“ Si queremos ser efectivos
debemos ser metódicos...” ”**

Diagnóstico



DIRECTRICES

- Todos los experimentos deben probar algo.
- Probar una cosa cada vez.
- Lleva contabilidad de los experimentos.
- No ignores nada, sobre todo si pasa algo inesperado.

Corrección

“ Una vez he completado el diagnóstico y he comprendido el problema. ”

OBJETIVO

- Arreglar el problema.
- Evitar añadir nuevos problemas.
- Mantener la legibilidad, la documentación, la corrección de los test.
- ...

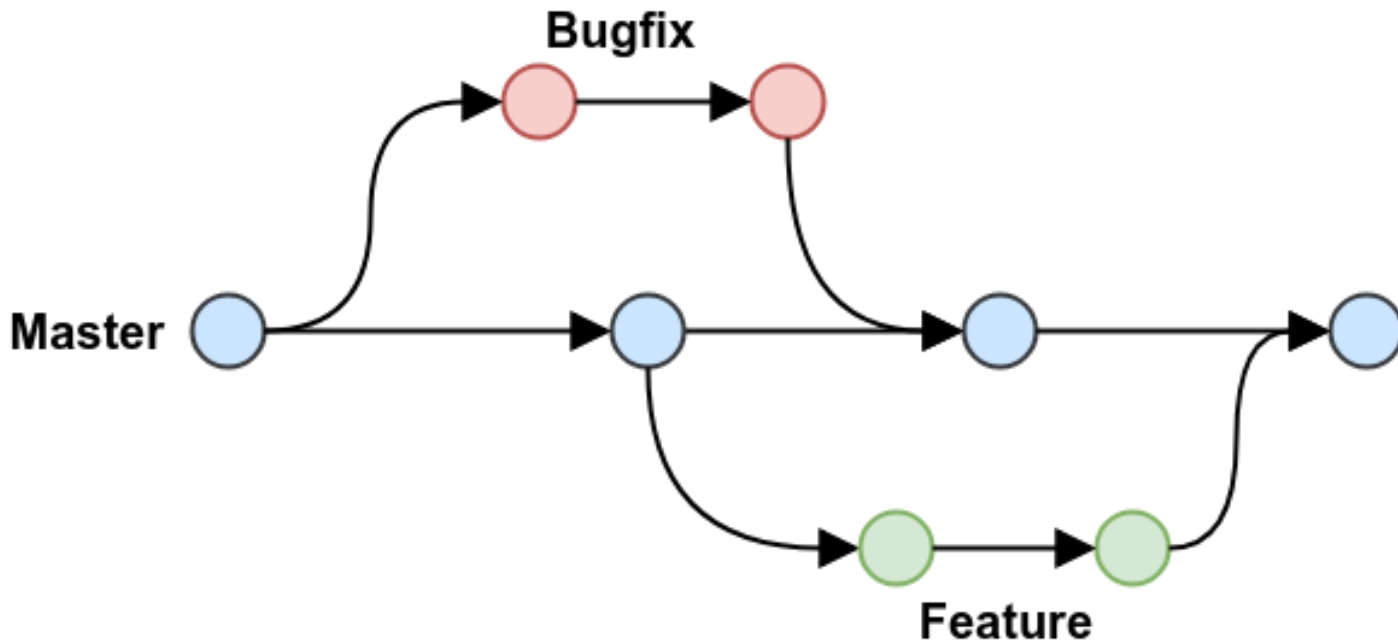
ESTRATEGIAS

- **Crear una nueva rama (GIT) para la corrección.**
- **Realizar las mismas pruebas que nos llevaron al diagnóstico.**
- **“Refactoring” tras realizar la corrección.**

ESTRATEGIAS

- **Un commit-push para cada corrección antes de hacer el merge de nuevo con la rama master.**

Corrección



Reflexión

“ Tras realizar la corrección del error debemos reflexionar sobre todo el proceso desarrollado. ”

6 STAGES OF DEBUGGING

1

THAT CAN'T
HAPPEN.

2

THAT DOESN'T
HAPPEN ON MY
MACHINE

3

THAT SHOULDN'T
HAPPEN.

4

WHY DOES THAT
HAPPEN?

5

OH I SEE.

6

HOW DID THAT
EVER WORK?

Depuración con IntelliJ

Para depurar los programas debo:

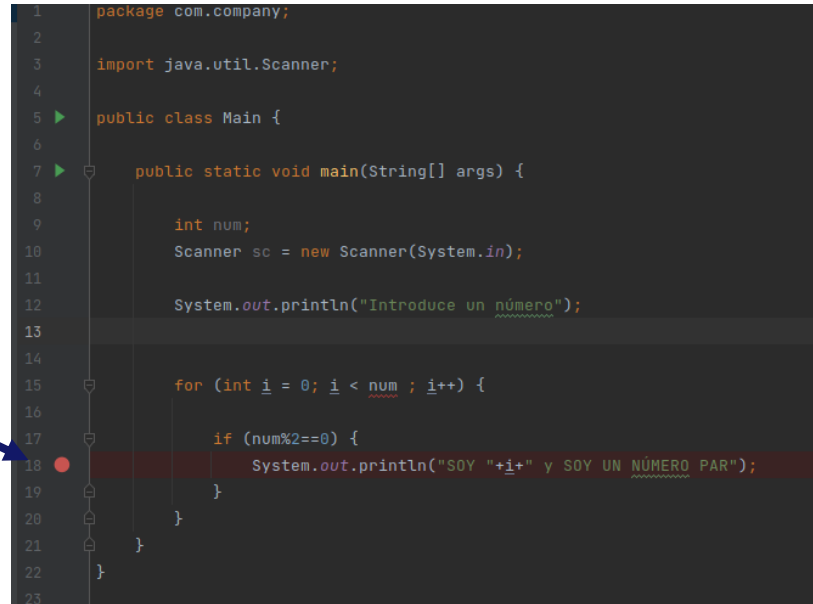
1. Establecer los puntos de parada, **BREAKPOINTS**, en los puntos del programa adecuados.
1. **ARRANCAR** la aplicación en **MODO DEBUG**.
1. Ejecutar la aplicación **PASO a PASO** para detectar los errores.

Depuración con IntelliJ

ESTABLECIMIENTO DE LOS BREAKPOINTS

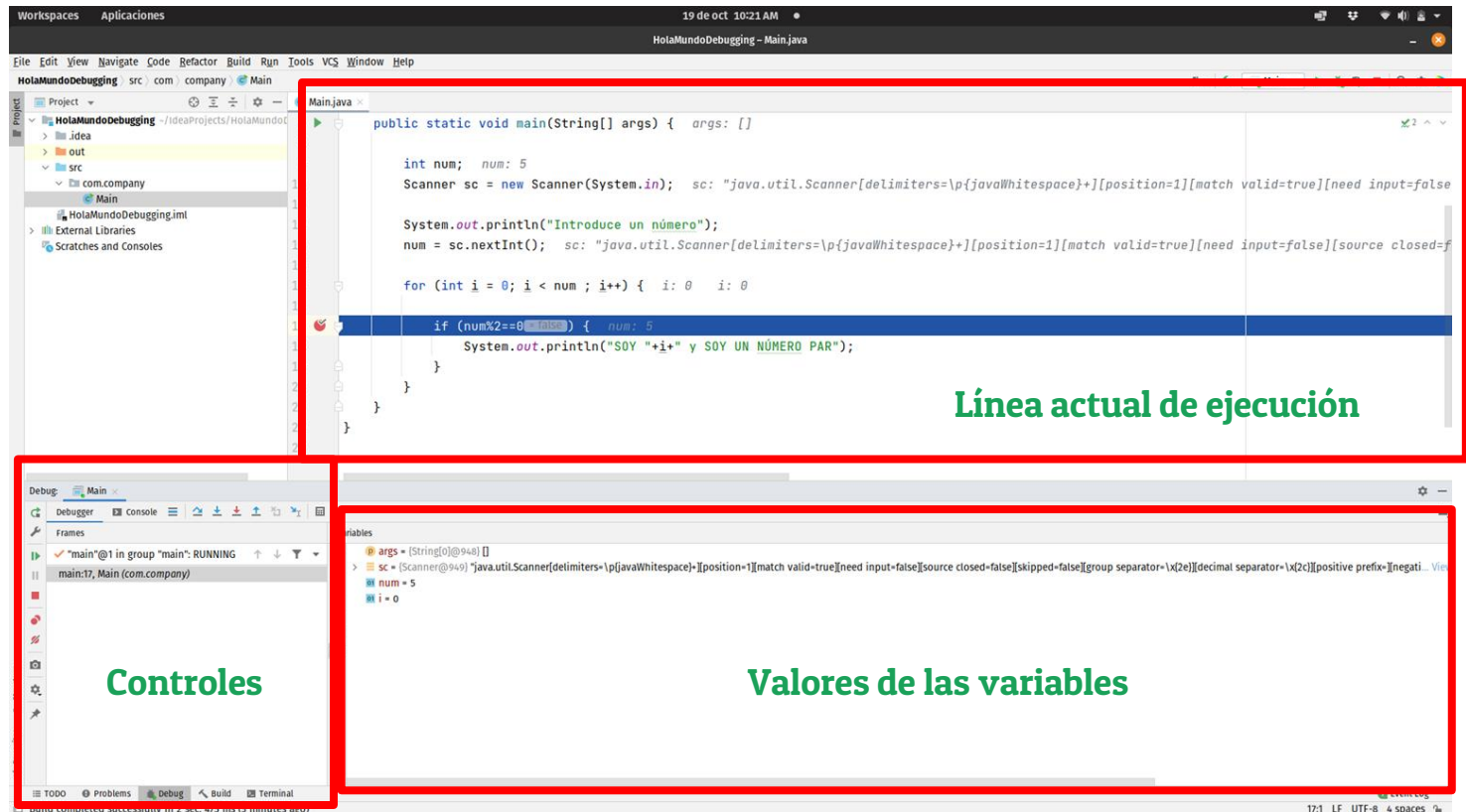
Hago click sobre la línea en la que quiero pasar (añadir el breakpoint).

También hago click para quitar el breakpoint

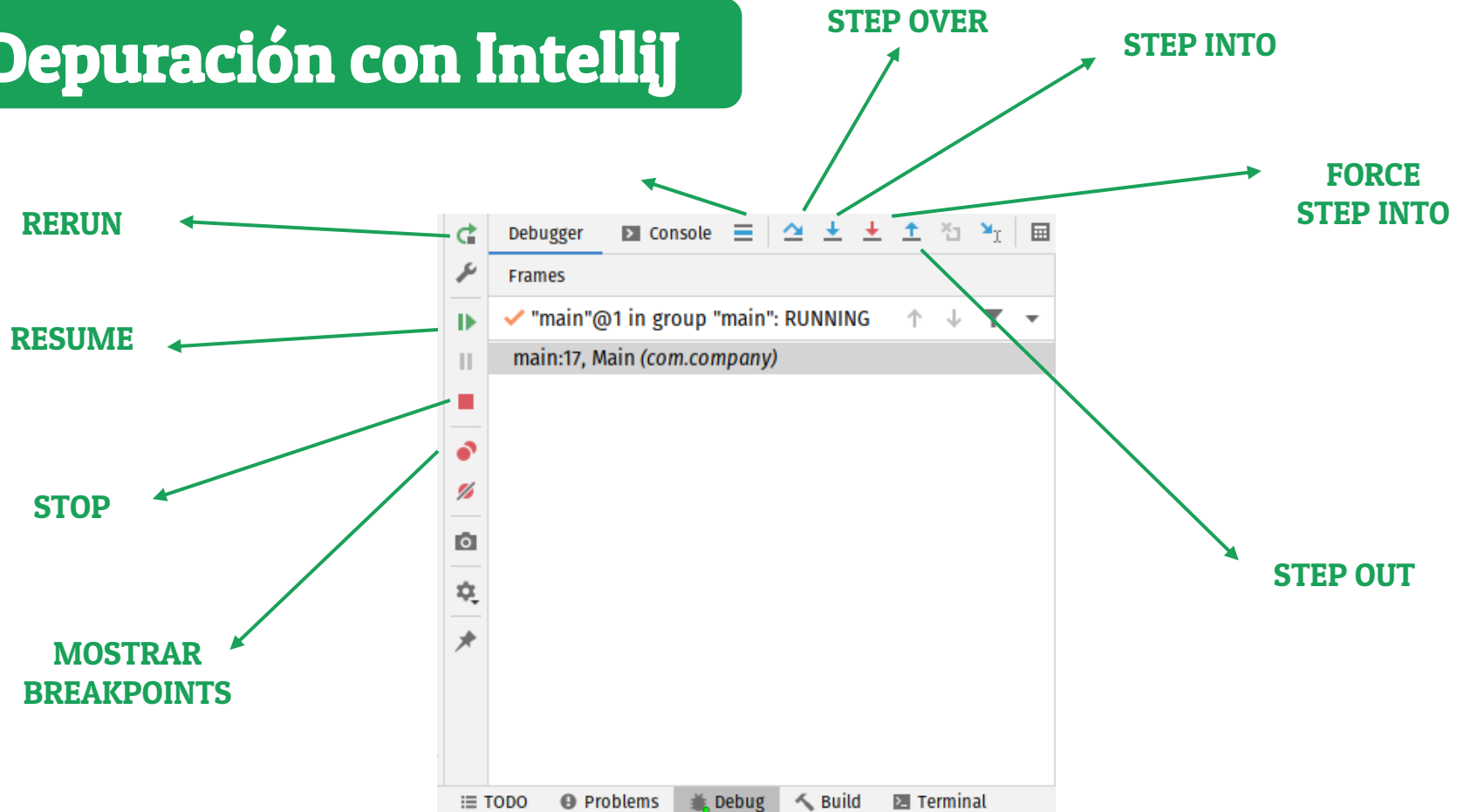


```
1 package com.company;
2
3 import java.util.Scanner;
4
5 public class Main {
6
7     public static void main(String[] args) {
8
9         int num;
10        Scanner sc = new Scanner(System.in);
11
12        System.out.println("Introduce un número");
13
14
15        for (int i = 0; i < num ; i++) {
16
17            if (num%2==0) {
18                System.out.println("SOY "+i+" y SOY UN NÚMERO PAR");
19            }
20        }
21    }
22 }
23
```

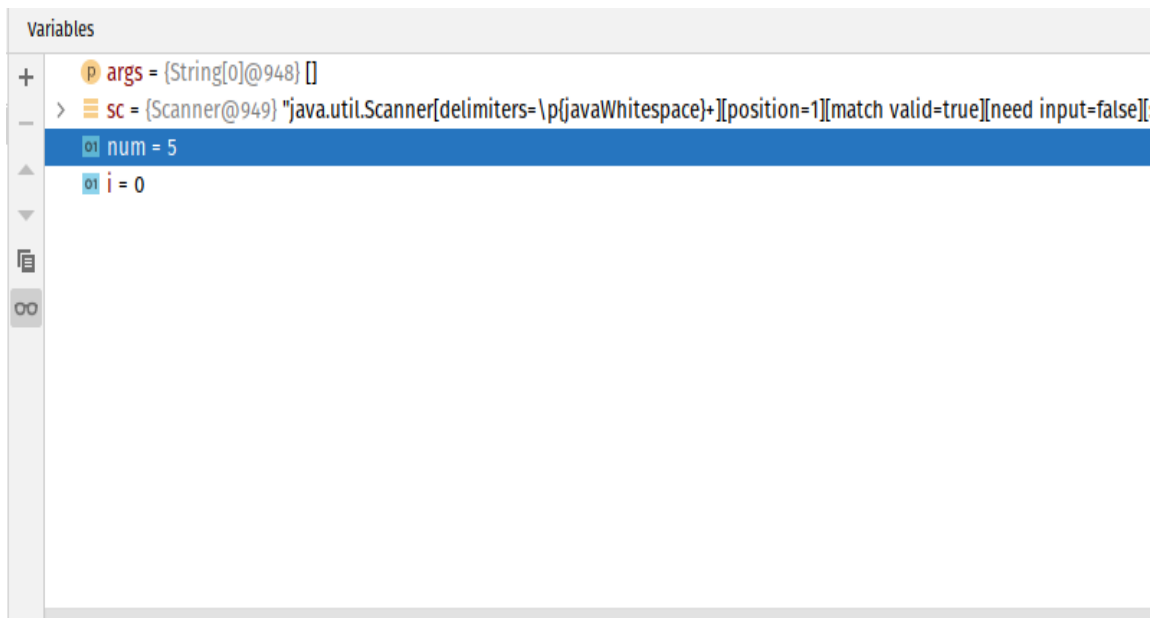
Depuración con IntelliJ



Depuración con IntelliJ



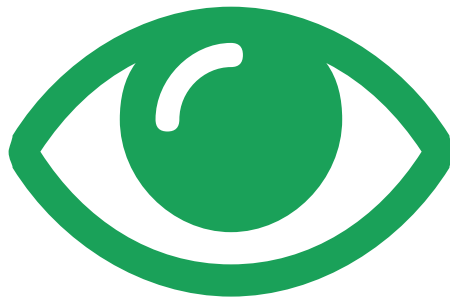
Depuración con IntelliJ



DESDE AQUÍ PUEDO..

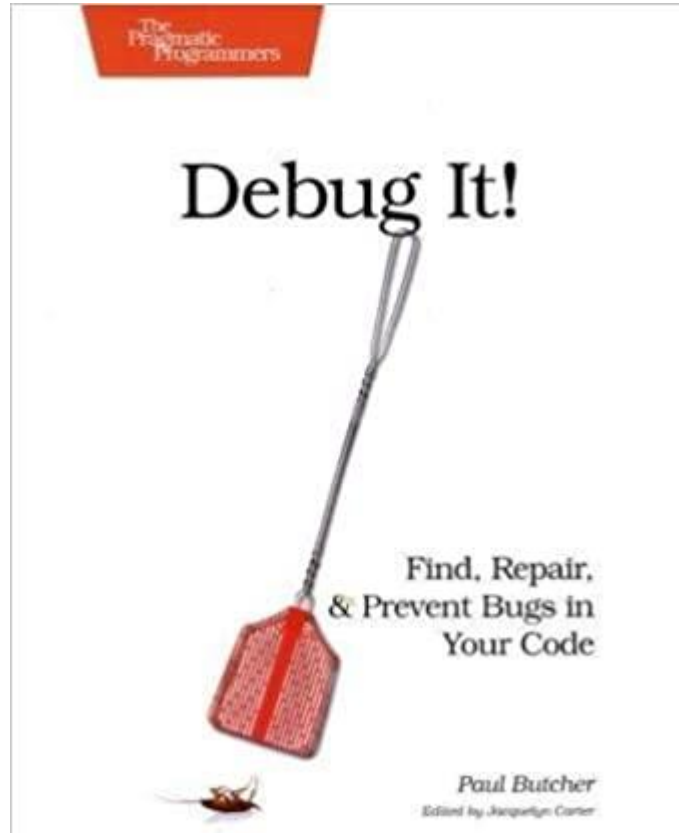
- **Inspeccionar.**
- **Establecer valor.**
- **Evaluar expresiones**
- **Añadir to “Watches”**
- **Mostrar “Referring Objects”**

Depuración con IntelliJ



EJEMPLO DE USO

Referencia



END



jgardur081@g.educaand.es