



# eclipse

---

# DEPURADOR

Pedro J. Ramos Ruíz  
Estela Muñoz Cordon

Comandos	Step Over
	Step Into
	Step Return
Puntos de Ruptura	Break Point
	Watch Point
	Break Point Exception

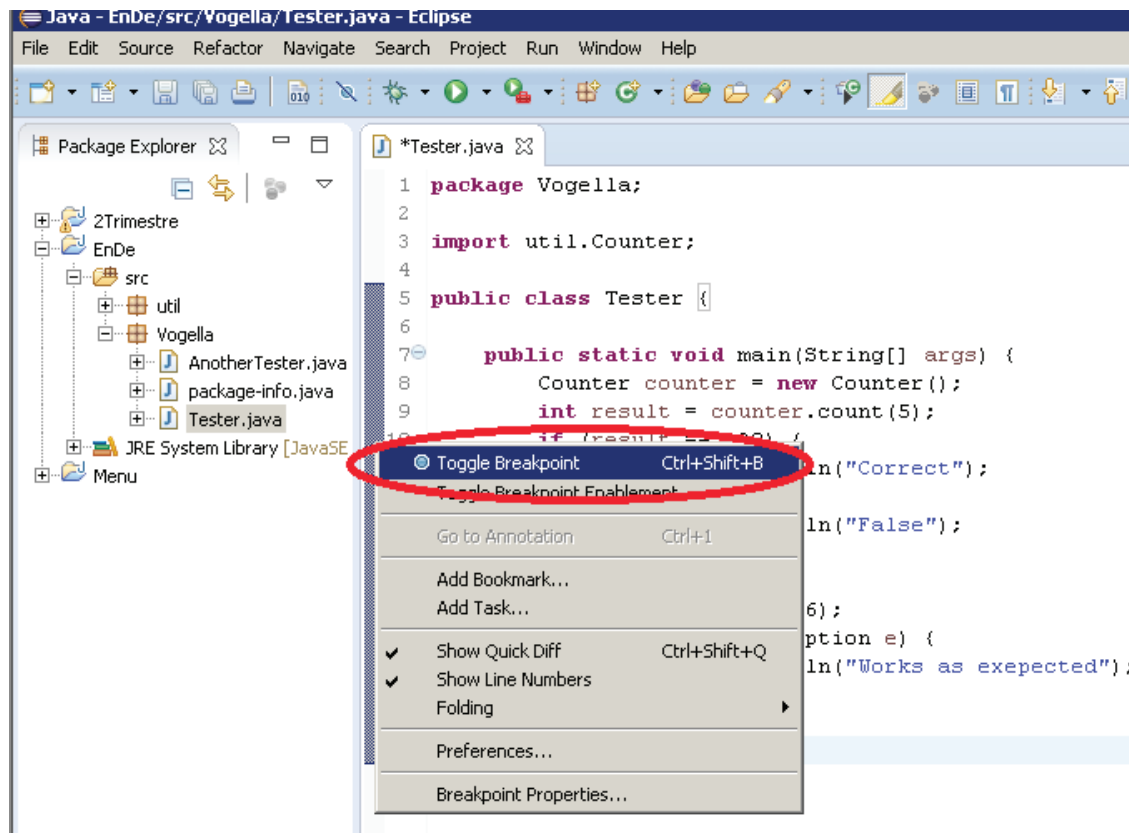
## 1. COMANDOS STEP OVER, STEP INTO, AND STEP RETURN COMMANDS. UTILIDAD, TECLAS DE ACCESO RÁPIDO Y DIFERENCIA:

- **Step Into (F5):** si entras en un bucle, hace el recorrido completo y se para cada vez que cambia el valor de la “i” (acumulador).
- **Step Out (F7) o Step Return:** se sale del método actual y vuelve al método que lo invocó; es como un retroceso entre métodos.
- **Step Over (F6):** ejecuta toda una línea entera (hasta que encuentra un “;”).



## 2. DIFERENCIA ENTRE BREAK POINT (PUNTO DE RUPTURA), WATCH POINT Y PUNTOS DE RUPTURA DE EXCEPCIÓN:

### 2.1. Break Point:



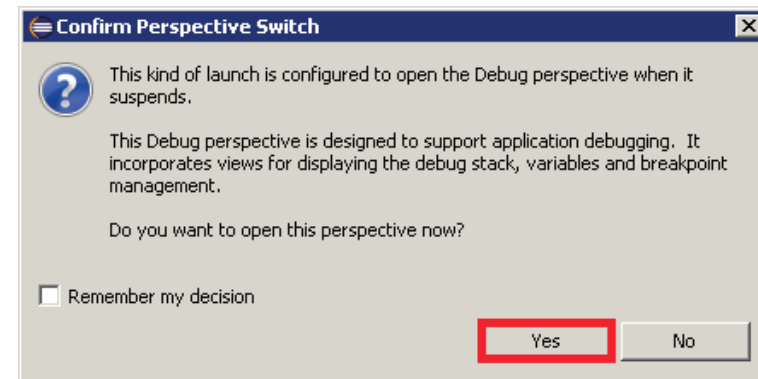
Click con el botón derecho del ratón en el **margen izquierdo** del código.

Seleccionar **“Toggle Breakpoint”** para colocar un punto de ruptura.

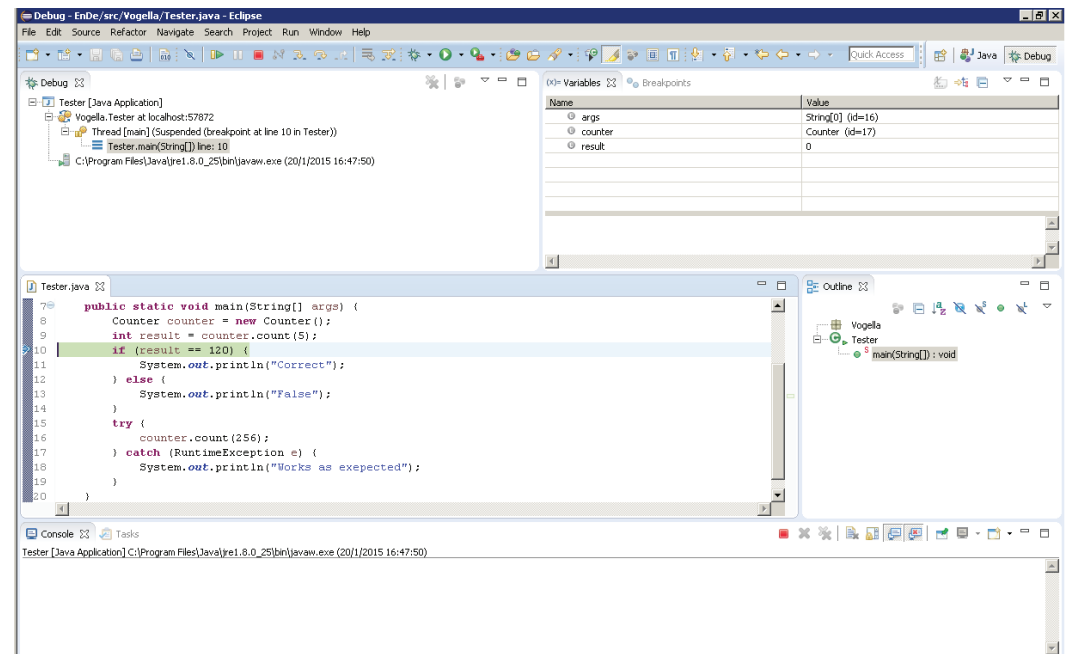
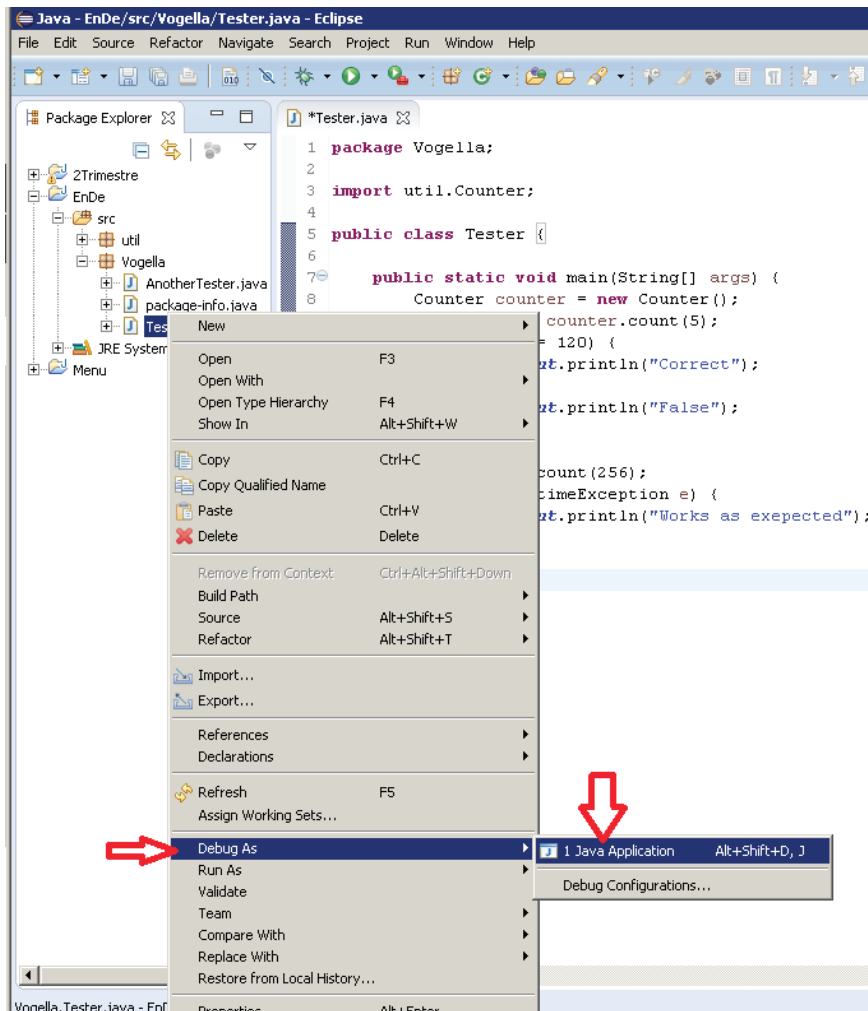
Click con el botón derecho del ratón en el archivo ".java".

Seleccionar "Debug As" y "1 Java Application (Alt+Shift+D,J)".

Confirmar el cambio de perspectiva.



Perspectiva del Depurador: Break Point de color distinto.

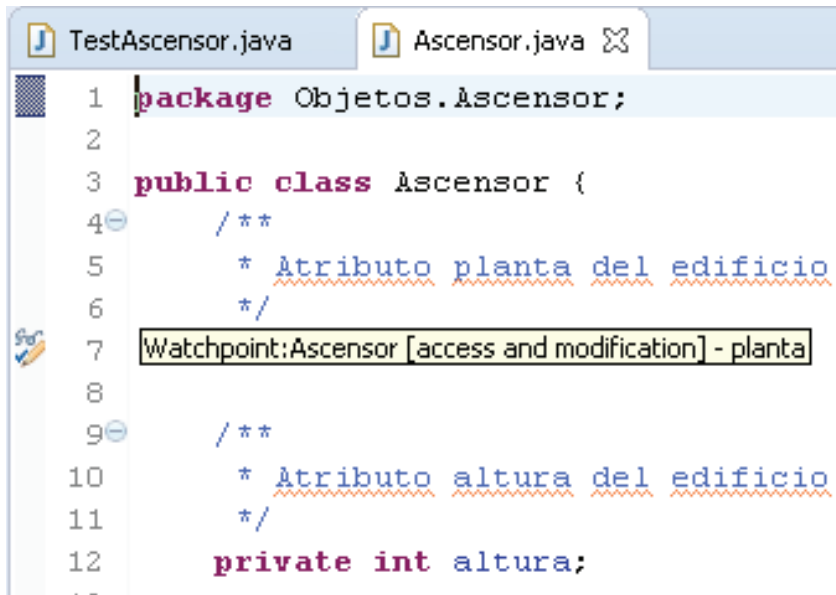


## 2.2. Watch Point:

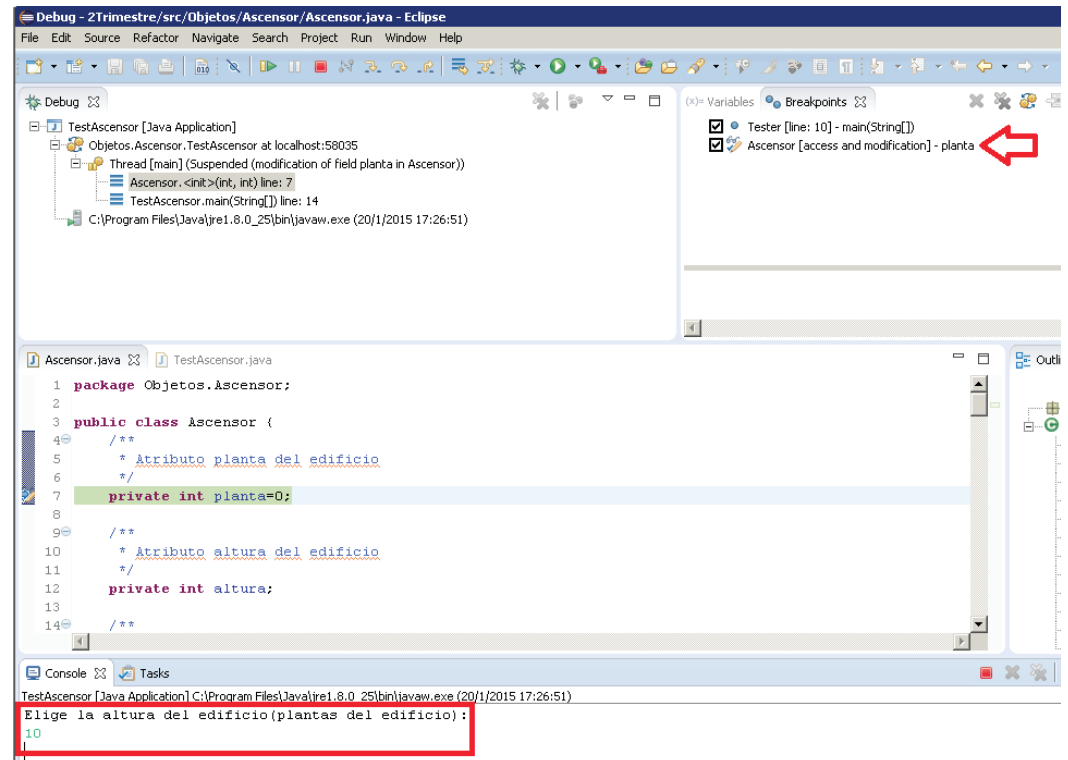
Clic en margen izquierdo.

Ejecutar el depurador  
(Run-Debug o F11).

Se detendrá en el punto  
implicado.



```
1 package Objetos.Ascensor;
2
3 public class Ascensor {
4     /**
5      * Atributo planta del edificio
6      */
7     private int planta;
8
9     /**
10    * Atributo altura del edificio
11    */
12    private int altura;
```



```
Debug - 2Trimestre/src/Objetos/Ascensor/Ascensor.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help


TestAscensor [Java Application]
  Objetos.Ascensor.TestAscensor at localhost:58035
    Thread [main] (Suspended (modification of field planta in Ascensor))
      Ascensor.<init>:(int, int) line: 7
      TestAscensor.main(String[]) line: 14
      C:\Program Files\Java\jre1.8.0_25\bin\javaw.exe (20/1/2015 17:26:51)

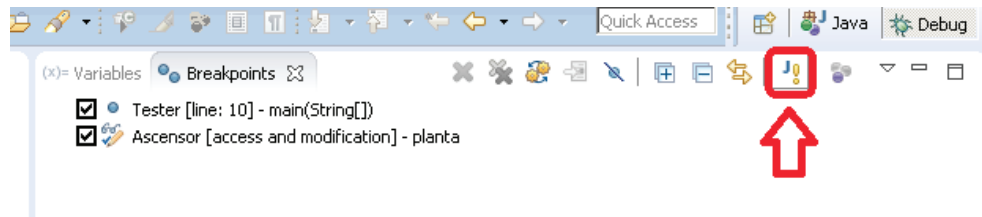
Breakpoints
  [x] Variables
  [x] Breakpoints
  [x] TestAscensor [line: 10] - main(String[])
  [x] Ascensor [access and modification] - planta

Ascensor.java
1 package Objetos.Ascensor;
2
3 public class Ascensor {
4     /**
5      * Atributo planta del edificio
6      */
7     private int planta=0;
8
9     /**
10    * Atributo altura del edificio
11    */
12    private int altura;
13
14    /**
```

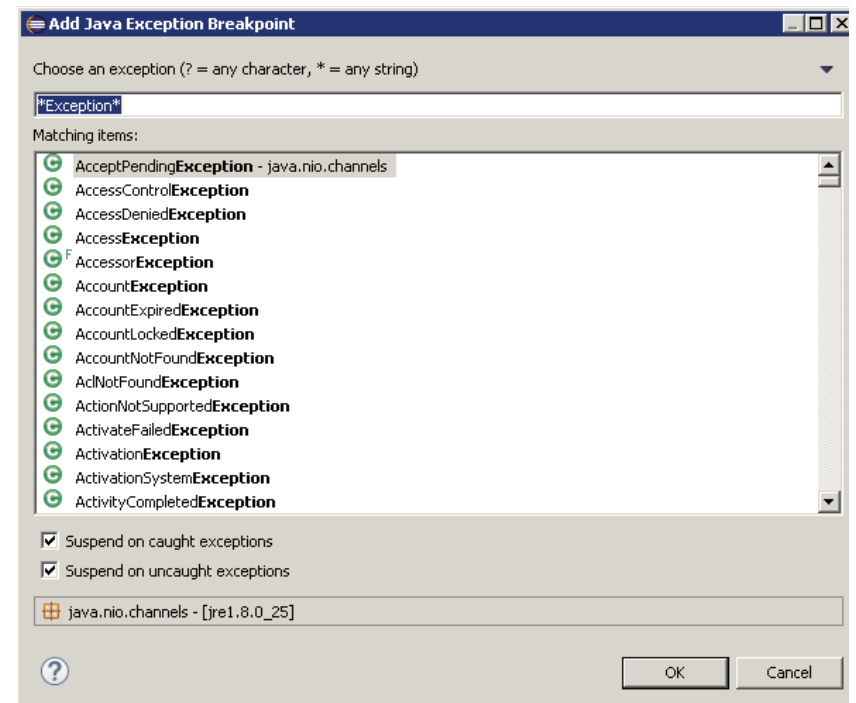
```
TestAscensor [Java Application] C:\Program Files\Java\jre1.8.0_25\bin\javaw.exe (20/1/2015 17:26:51)
Elige la altura del edificio(plantas del edificio):
10
```

## 2.3. Puntos de Ruptura de Excepción:

Pulsar el botón: 



Sale un menú con una lista de excepciones de java.

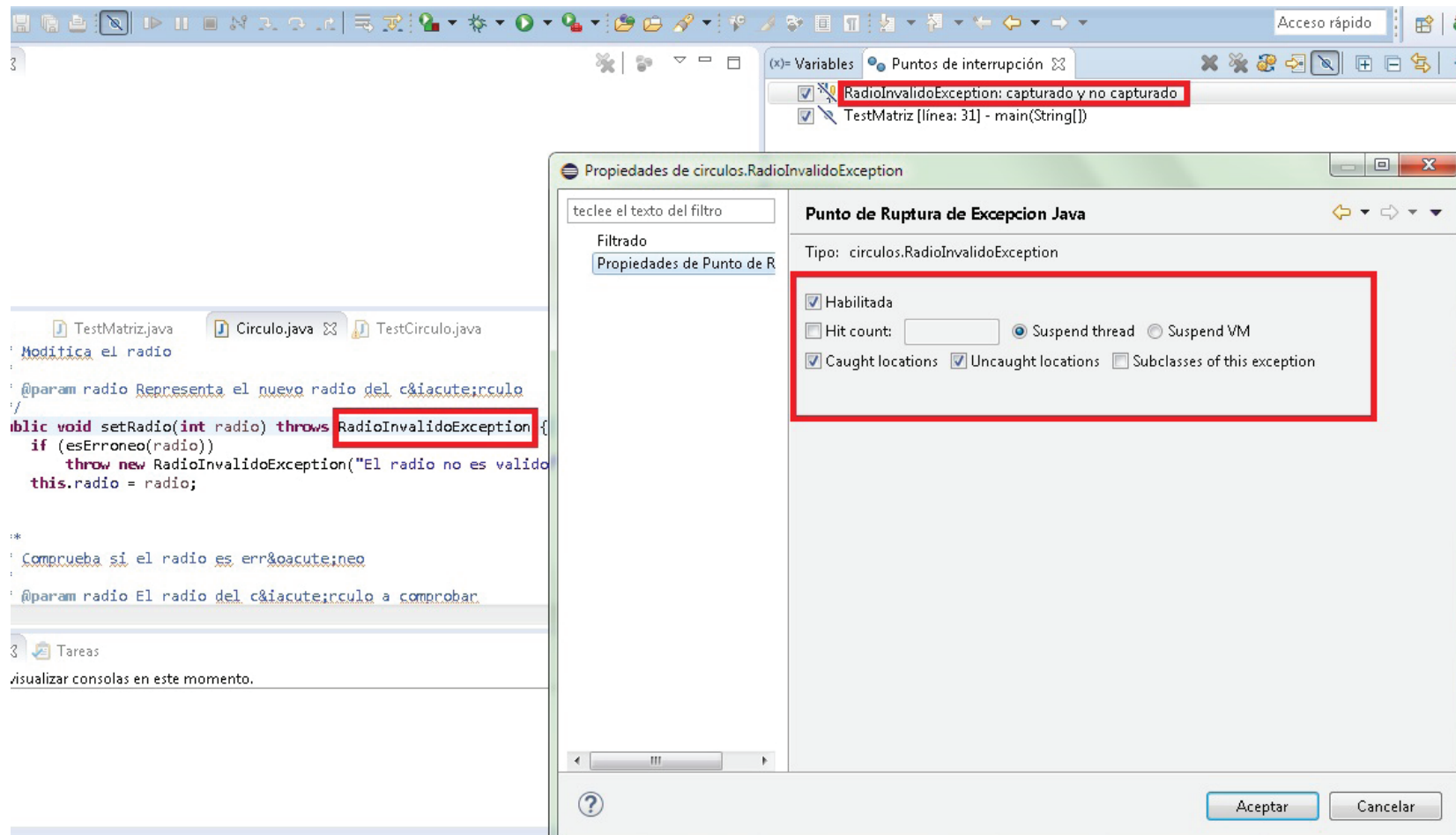




Hay dos tipos de excepciones: try-catch o (Capturar) y throw (no Capturar) y se detiene según la opción elegida. Si existe la excepción.

Hit count: salta después de las veces indicadas.

Se puede elegir que hacer después de encontrar la excepción: Suspender la máquina virtual o la ejecución del código.

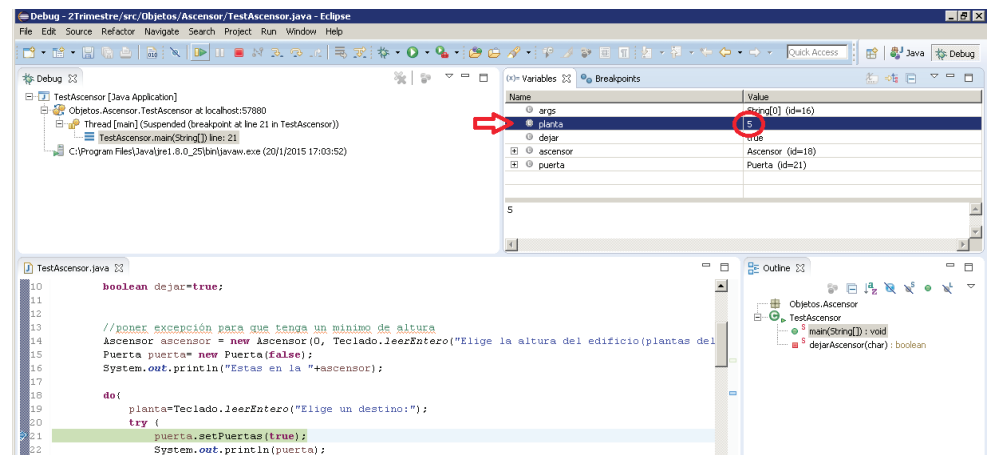
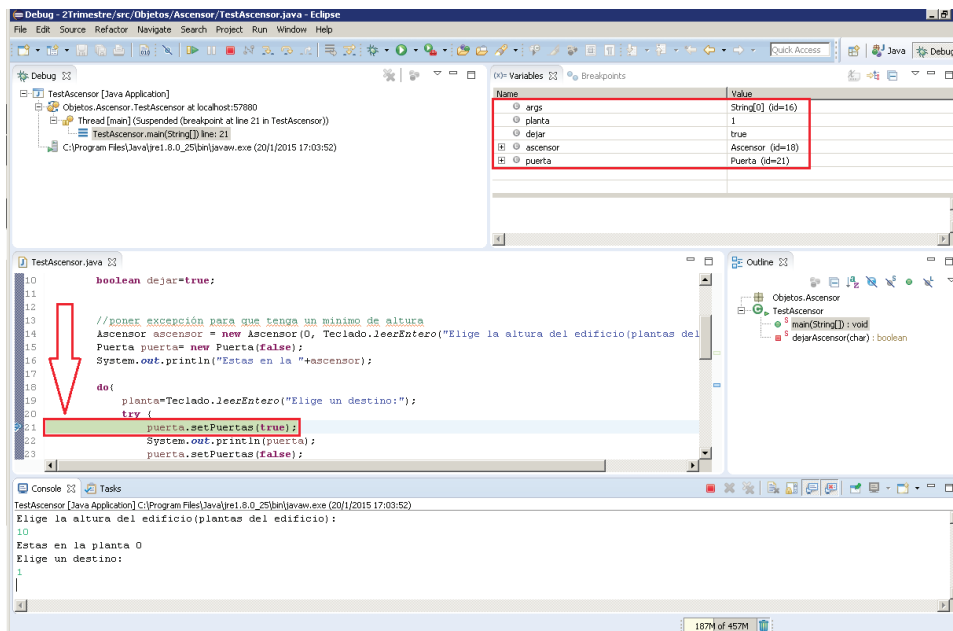


### 3. CAMBIO DEL VALOR DE VARIABLES EN UNA SESIÓN DE DEPURACIÓN:

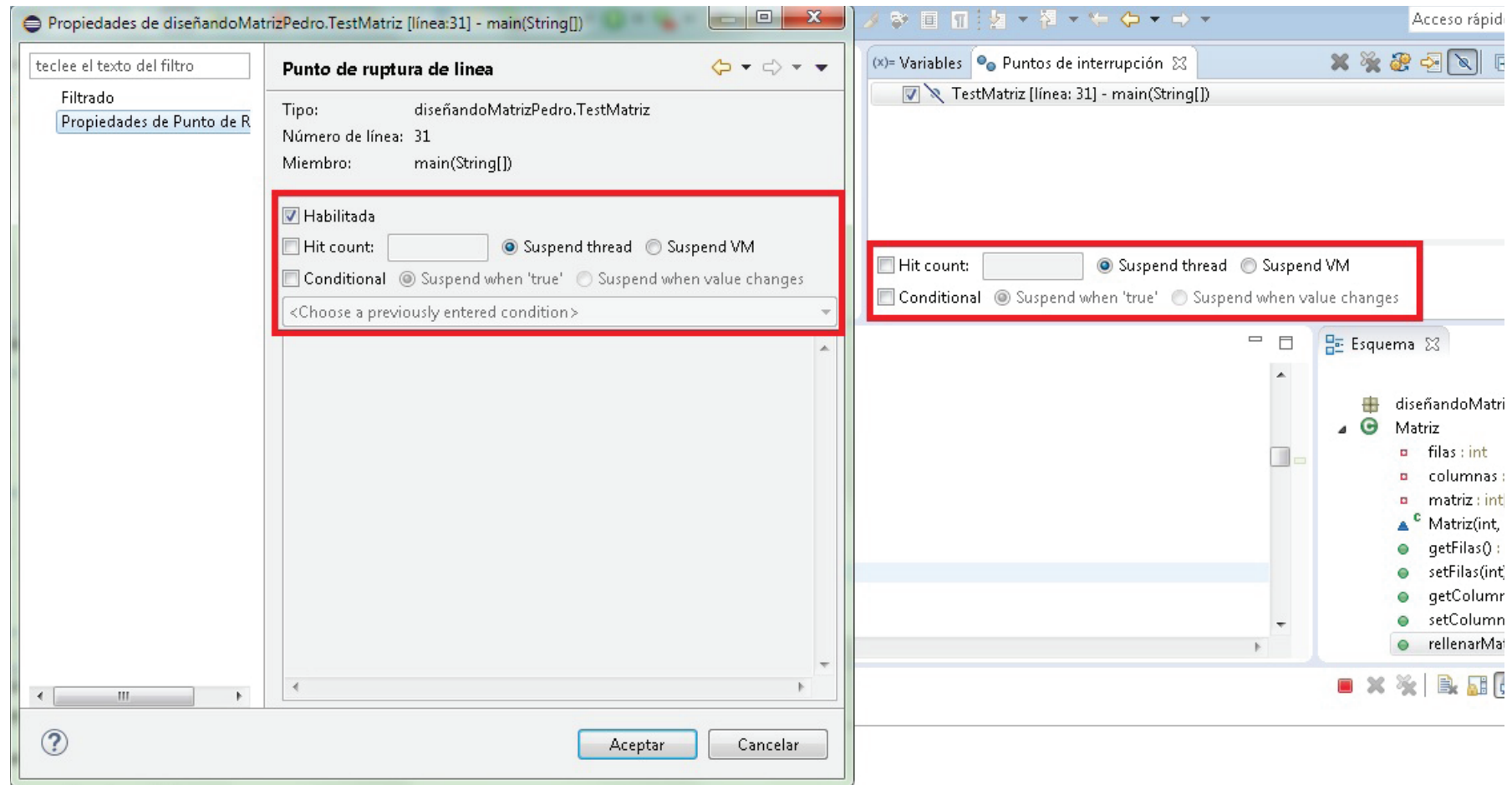
Ejecutar el Depurador: Botón “Resume” (F8) hasta el Break Point.

Muestra las variables y podemos cambiar su valor (1 de la variable “planta” por un 5).

Ejecutar de nuevo o “Resume” y seguirá hasta el siguiente Punto de ruptura o hasta la finalización del programa.

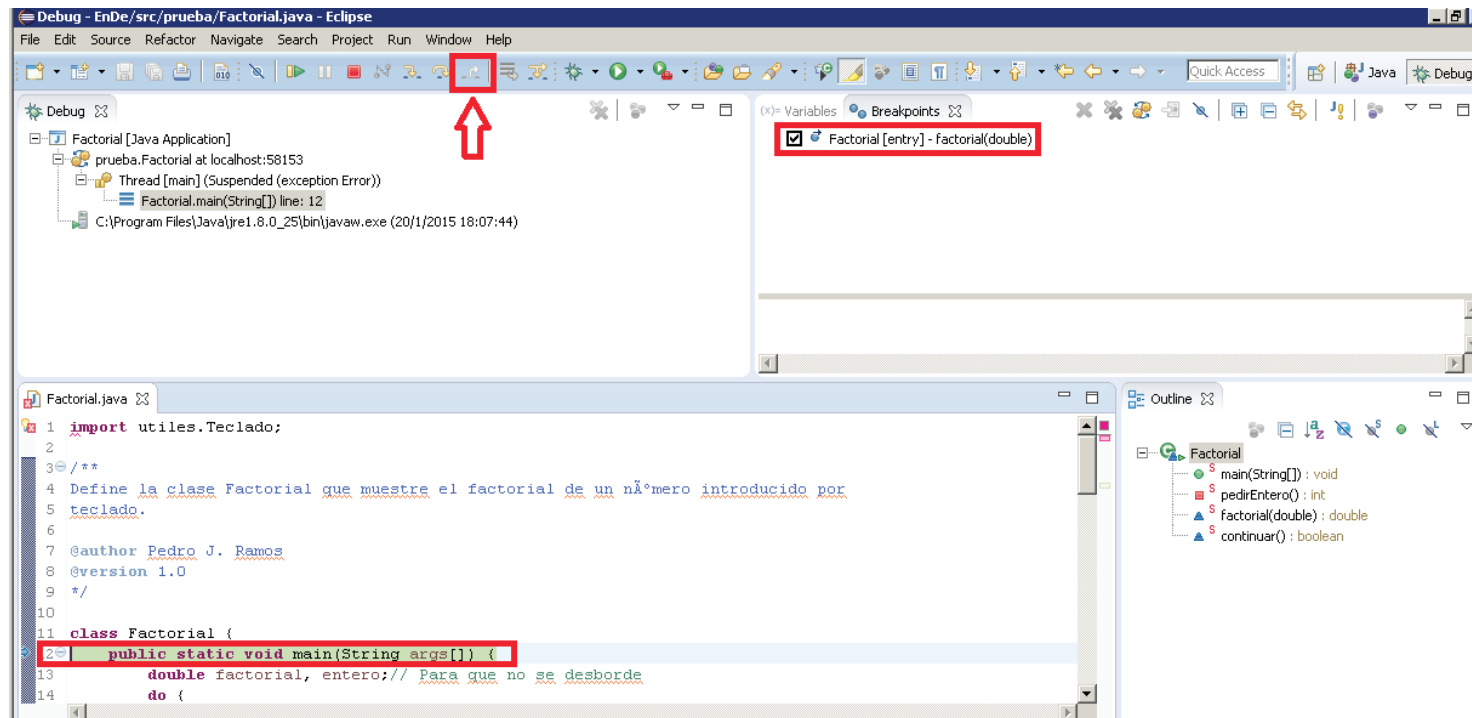






## 4. UTILIZA EL DEPURADOR PARA ENTENDER UN MÉTODO RECURSIVO

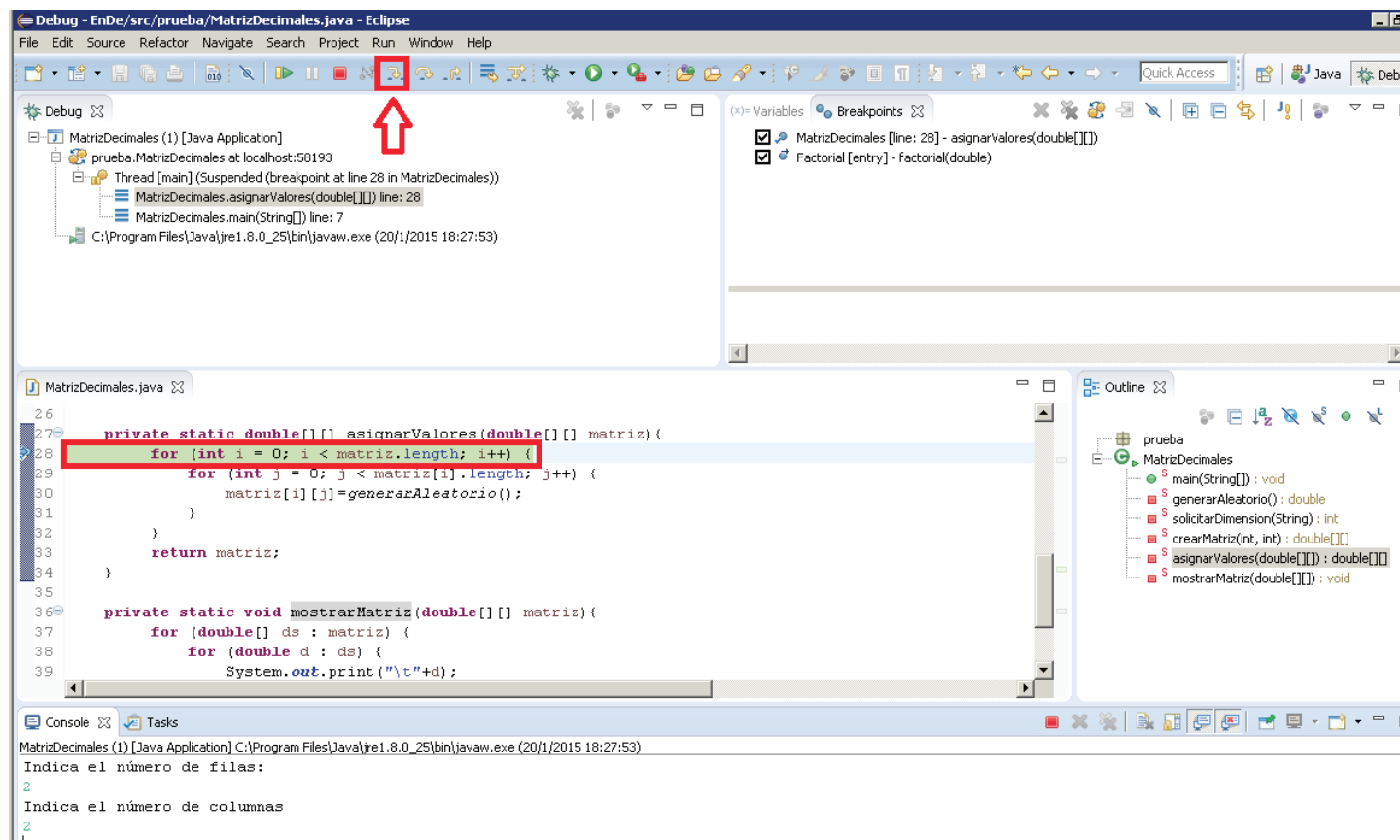
El método se llama así mismo y cada vez que llega al Punto de Ruptura de nuevo, se vuelve a parar.



## 5. UTILIZA EL DEPURADOR PARA ENTENDER EL RECORRIDO DE UN ARRAY BIDIMENSIONAL CON FOR CLÁSICO:

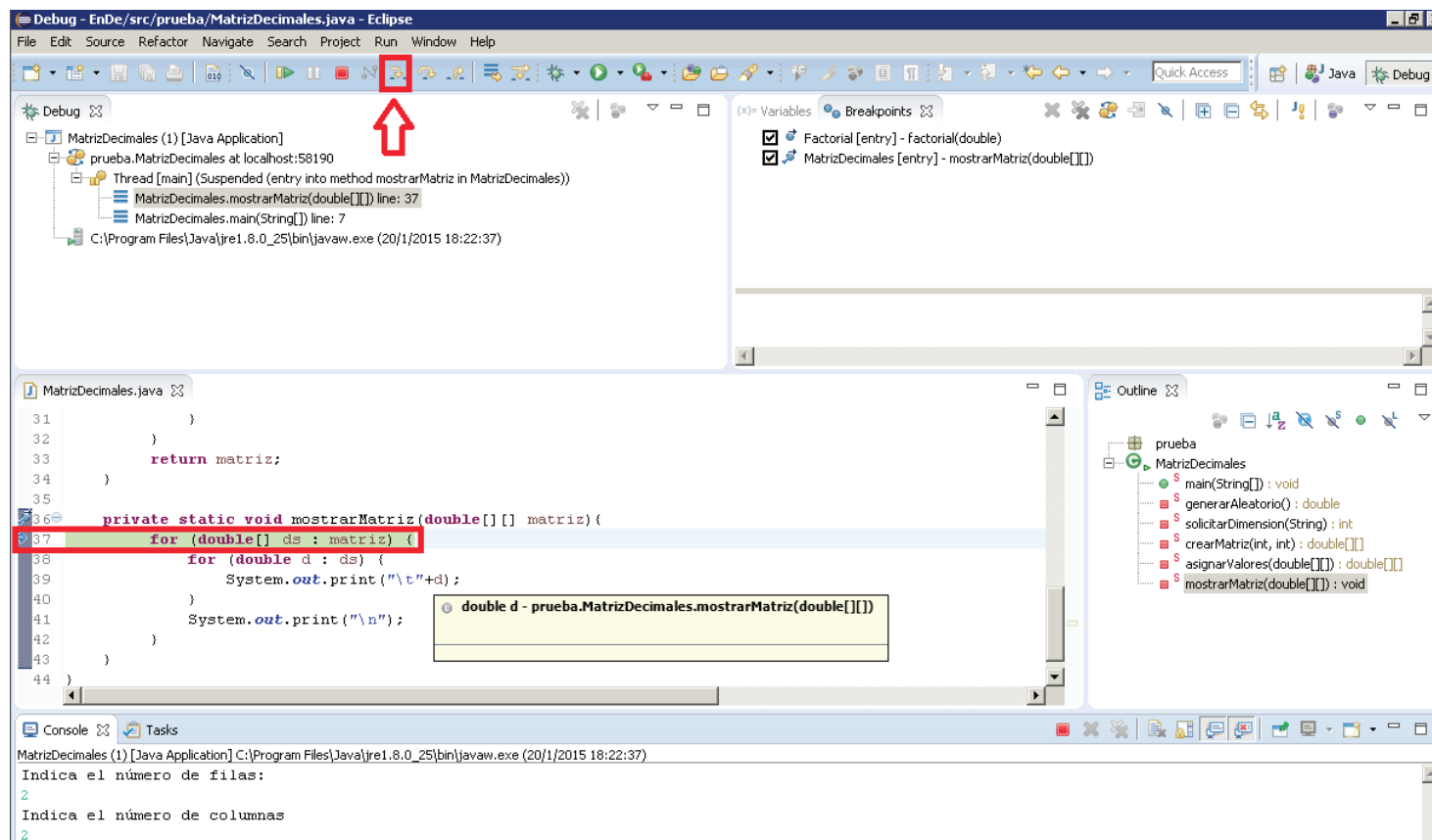
**F5:** Recorre el for entero y vuelve al principio, justo al momento en el que la “i” se incrementa, deteniéndose hasta que le demos a continuar.

**F6:** lo hace línea a línea



## 6. UTILIZA EL DEPURADOR PARA ENTENDER EL RECORRIDO DE UN ARRAY BIDIMENSIONAL CON FOR MEJORADO:

Recorre todos los valores de la “i” del bucle for, finalizándolo por completo (se use F5 o F6).



## 7. DENTRO DE UN BUCLE. INDICA QUÉ SUCECE EN UN BUCLE FOR AL AVANZAR MEDIANTE F7 ¿SE EJECUTA DE UNA PASADA TODO SU BLOQUE? ¿SE SALE DEL FOR?:

Finaliza el método que contiene el for en el que está el “Break Point”, sin ejecutarse, y pasa al método que lo invocó.

