

¿Que es debuggear? ¿Que es un BreakPoint? ¿Como agregar un BreakPoint en eclipse? ¿Como debuggear en eclipse?

¿Que es debuggear?

El debugger o mejor dicho en español el depurador, es una Herramienta o Aplicación que permite la ejecución controlada de un programa o código para seguir cada instrucción ejecutada y localizar así el Bugs o errores, códigos de protección, etc. Este proceso es el proceso de depuración.

¿Que es un BreakPoint?

Un **BreakPoint** es un punto de corte, que en programación es una línea/s específica/s en el cual queremos que se detenga el flujo habitual del programa. Los fines pueden ser varios, verificar el flujo del programa o valores de los datos, etc.

¿Como agregar un Break Point en eclipse?

Para agregar un **BreakPoint** es muy fácil, nos posicionamos hacia la izquierda en la línea que quieras que se detenga el debugger y hacemos doble click, luego aparecerá un circulo en dicha línea, este circulo es un **BreakPoint**.

```
1 package ar.com.java.white.box.main;
2
3 import ar.com.java.white.box.entidades.Persona;
4
5 /**
6  * <p>Main clase de ejecucion.
7  * Pruebas para el lado oscuro de java, ejemplo ejecucion en debug DISPLAY.
8  * http://java-white-box.blogspot.com.ar/</p>
9  *
10 * @author nconde
11 * @since 28/05/2012
12 * @version 1.0
13 */
14 public class Main {
15
16     /**
17      * @param args
18      */
19     public static void main(String[] args) {
20         Persona persona = new Persona();
21         persona.setApellido("Gomez");
22         persona.setNombre("Pepe");
23
24         persona.imprimirPersona();
25
26         System.out.println("Fin");
27
28     }
29
30 }
```

BreakPoint marcado en rojo

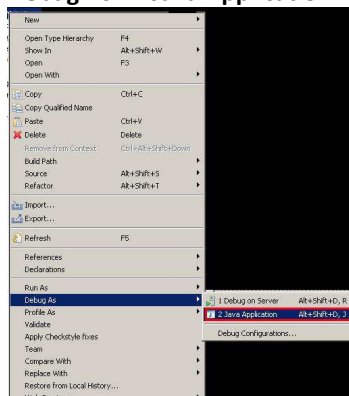
¿Como debuggear en eclipse?

Entendido que es el debugger, podemos proceder a debuggear con eclipse.

Primero antes que nada, tenemos claro que para debuggear algo, ese algo se ejecuta, es decir un método main, montado en algún servidor, etc. PERO ALGO QUE SE EJECUTA.

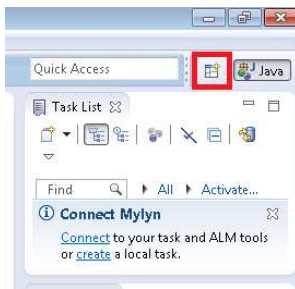
Hacemos click derecho sobre la clase que lanza el programa, vamos:

Debug As -> Java Application

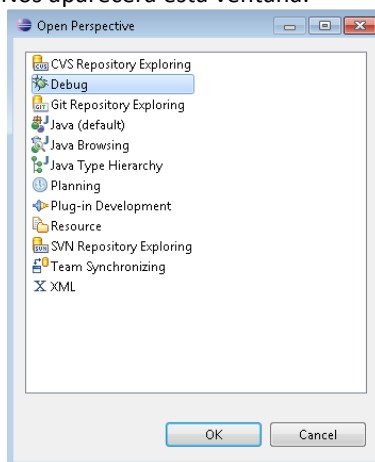


Si ejecutamos directamente un programa sin poner un **Punto de Corte (BreakPoint)**, el programa se ejecutara de golpe, sin parar en ningún lado, ya que no existe un punto de corte.

Otra forma de mostrar el depurador de código (debug):

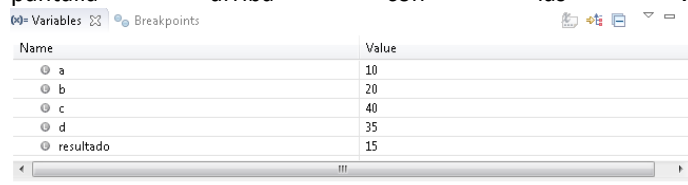


Nos aparecerá esta ventana.



Elegimos **Debug** y después **OK**. Nos aparecerá una nueva pestaña al lado.

Para probar el modo **Debug**, ahora pinchamos en **Run->Debug** nos llevara al modo **Debug**, veremos una pantalla arriba con las variables del programa.



Y en la parte de abajo vemos el seguimiento que le vamos haciendo.



Cuando el fondo esta en verde, es que no se ha ejecutado la línea, también veremos que afecta a la ventana de variables antes vista.

Una vez llegado el punto de **Punto de Corte (BreakPoint)** nos movemos con las siguientes teclas:

- **Step Into** (o pulsar **F5**): "paso en". Significa que se ejecutará instrucción por instrucción pero si el debugger encuentra una función (y/o una subrutina-procedimiento-método), al pulsar Step Into se irá a la primera instrucción de dicha **función, entrará** en ella.

- **Step Over** (o pulsar **F6**): "paso sobre". Significa que se ejecutará instrucción por instrucción pero si el debugger encuentra una función (y/o subrutina-procedimiento-método), al pulsar Step Over se irá a la siguiente instrucción del código **sin entrar en la subrutina**.
- **Step Return** (o pulsar **F7**): "paso retorno". Imagina que **nos encontramos dentro de una función** y quieres salir de ella. Una forma sería encontrar la última instrucción, poner ahí un BP, pulsar Resume y finalmente Step Into, sin embargo, tenemos la opción de pulsar Step Return y el debugger automáticamente encontrará el final de la función y **nos situará fuera de ella**.
- **Resume** (o pulsar **F8**): Reanudar. **Se utiliza para depurar el código. El debugger parará en el siguiente Breakpoint(BP)** que hayas puesto. Esto es muy útil cuando no quieres analizar instrucción por instrucción y quieres que el debugger se pare directamente en una línea donde tienes puesto un BP.
- **Run to Line** (o pulsar **Ctrl+R**). Ejecutar hasta la línea. Sencillo, se reanuda la ejecución del código hasta la línea que has seleccionado.