

MANEJO DE ECLIPSE

Para poder importar cualquier proyecto que tu desees utilizar dentro del entorno de desarrollo Eclipse, puedes hacerlo por cualquiera de los siguientes métodos:

MÉTODO 1:

- 1.- Haz click derecho sobre la ventana donde se muestran los proyectos creados, del lado izquierdo de la ventana de código fuente.
- 2.- Seleccionar la opción de **import**.
- 3.- Dar click en **General --> Existing projects into workspace**.
- 4.- Se mostrará la ventana **Import Projects**, seguido del cuadro **Select root directory**, dar click en el botón browse
- 5.- Seleccionar el espacio de trabajo (workspace) en donde se encuentra el proyecto a importar.
- 6.- Una vez seleccionado el proyecto, se mostrará la ruta específica donde se encuentra el proyecto.
- 7.- Haz click en finalizar para que se muestre el proyecto importado.

MÉTODO 2:

- 1.- Haz click sobre la pestaña de **File**, en la parte superior del entorno de desarrollo Eclipse.
- 2.- Seleccionar la opción de **import**.
- 3.- Dar click en **General --> Existing projects into workspace**.
- 4.- Se mostrará la ventana **Import Projects**, seguido del cuadro **Select root directory**, dar click en el botón browse
- 5.- Seleccionar el espacio de trabajo (workspace) en donde se encuentra el proyecto a importar.
- 6.- Una vez seleccionado el proyecto, se mostrará la ruta específica donde se encuentra el proyecto.
- 7.- Haz click en finalizar para que se muestre el proyecto importado.

VER EDITOR DE CÓDIGO

Menú superior --> Ventana ---> Abrir Perspectiva ---> Perspectiva de Java

VER LA LISTA DE PROYECTOS ACCESIBLES

Menú superior --> Ventana ---> Mostrar vista ---> Explorador de Proyectos

Debuggeando

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

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

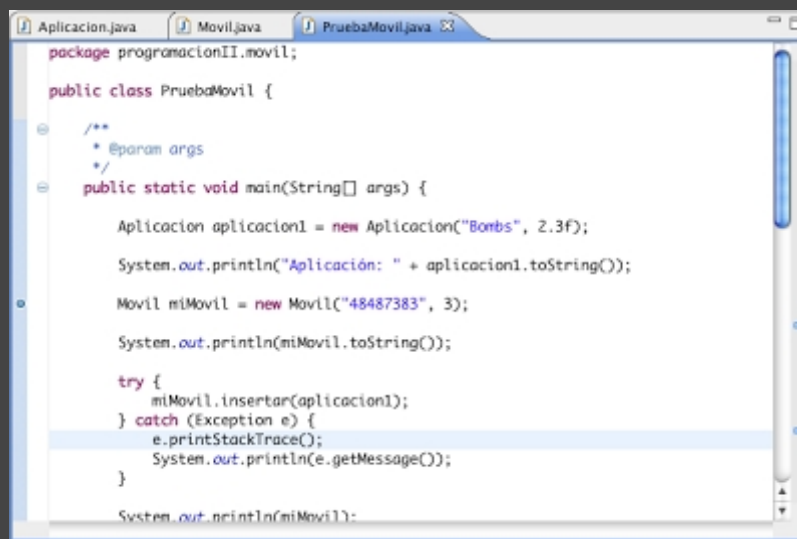
- ☐ **Step Into** (o pulsar **F5**): "paso en". Significa que se ejecutará instrucción por instrucción pero si el debugger encuentra una función (y/o una subrutina-procedimiento-método), al pulsar Step Into se irá a la primera instrucción de dicha función, entrará en ella.
- ☐ **Step Over** (o pulsar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

Depuración de Java con Eclipse

En esta entrada voy a intentar dar unas primeras nociones de cómo facilita Eclipse la depuración de código. En su modo de depuración, Eclipse nos permite ejecutar paso a paso el código y analizar el contenido de las variables.

Lo primero que hay que hacer es definir en qué punto o puntos del código queremos que la ejecución del programa se pause. Para ello, definiremos puntos de ruptura (*breakpoints*) en las líneas de código donde queramos que la ejecución pause.

Para definir un punto de ruptura (o para eliminar uno existente), nos situamos en la línea que queramos y seleccionamos "*Run -> Toggle breakpoint*" o hacemos doble-click en la barra a la izquierda de la ventana del código. Al definir un punto de ruptura aparecerá en dicha barra un punto, tal y como se puede ver en la figura.

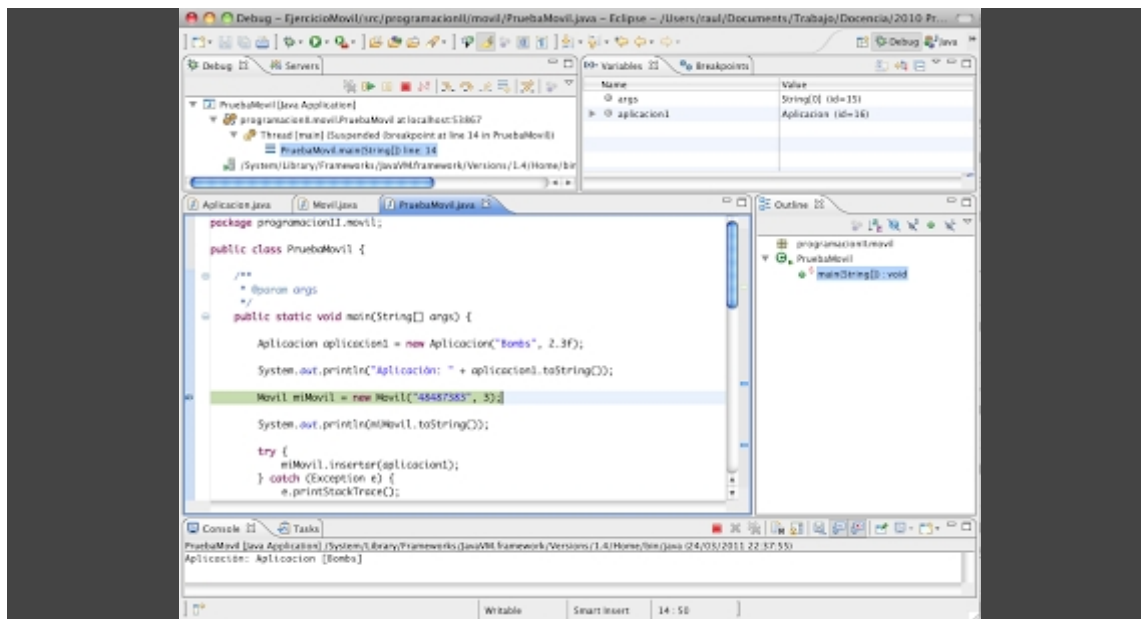


Una vez tenemos definidos los puntos de ruptura que queramos, iniciamos el depurador seleccionando desde una clase ejecutable "*Run -> Debug*" o, alternativamente, dando al botón con el bicho:



Lo primero que vemos es que el Eclipse cambia de la perspectiva *Java* a la Perspectiva *Debug*. En esta perspectiva, además de el código fuente, el esquema o la consola, podemos ver la información de las variables y una ventana donde se muestran los hilos de ejecución.

En la siguiente imagen podemos ver cómo la ejecución se ha detenido en la línea 14 del *main* (donde habíamos definido el punto de ruptura) y cómo dicha línea aparece marcada en el código fuente. En la figura podemos ver el contenido de las variables y objetos que existen en el momento en que se ha pausado la ejecución, lo cual nos va a ayudar mucho en la depuración.



Además, podemos indicar que la ejecución siga o se detenga a nuestro gusto utilizando los botones de la vista *Debug* (o las opciones del menú "*Run*"). A continuación explicamos las básicas.



Resume. Continúa la ejecución del programa hasta el próximo punto de ruptura o hasta que finaliza la ejecución.



Terminate. Finaliza la ejecución del programa.



Step into. Se ejecuta la línea actual y, en caso de ser una llamada a un método, la ejecución continúa dentro del método.



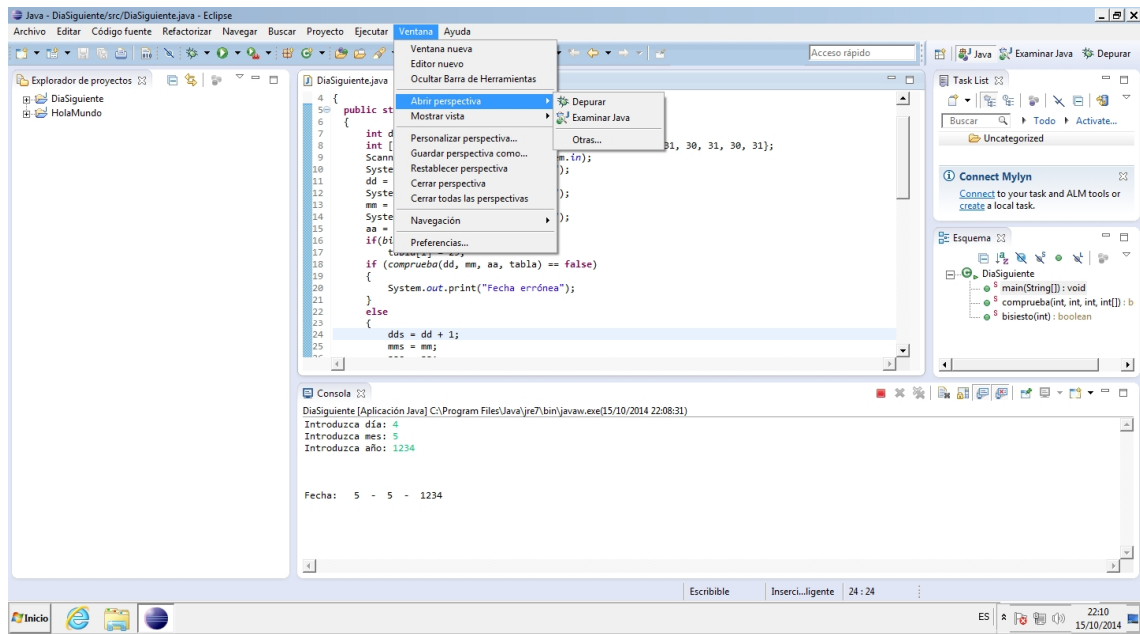
Step over. Se ejecuta la línea actual y se pasa a la línea siguiente sin entrar en los métodos.



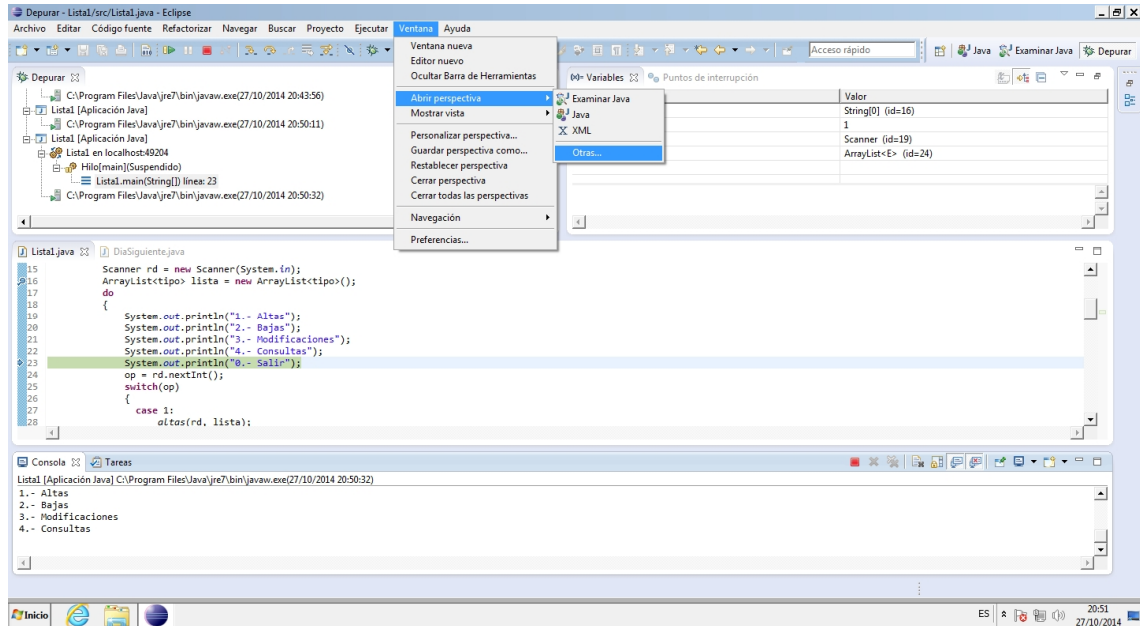
Step return. Se sigue ejecutando hasta que se ejecute un *return*. Es decir, se ejecuta por completo el método en el que estemos y se pasa a la línea siguiente a la invocación del método.

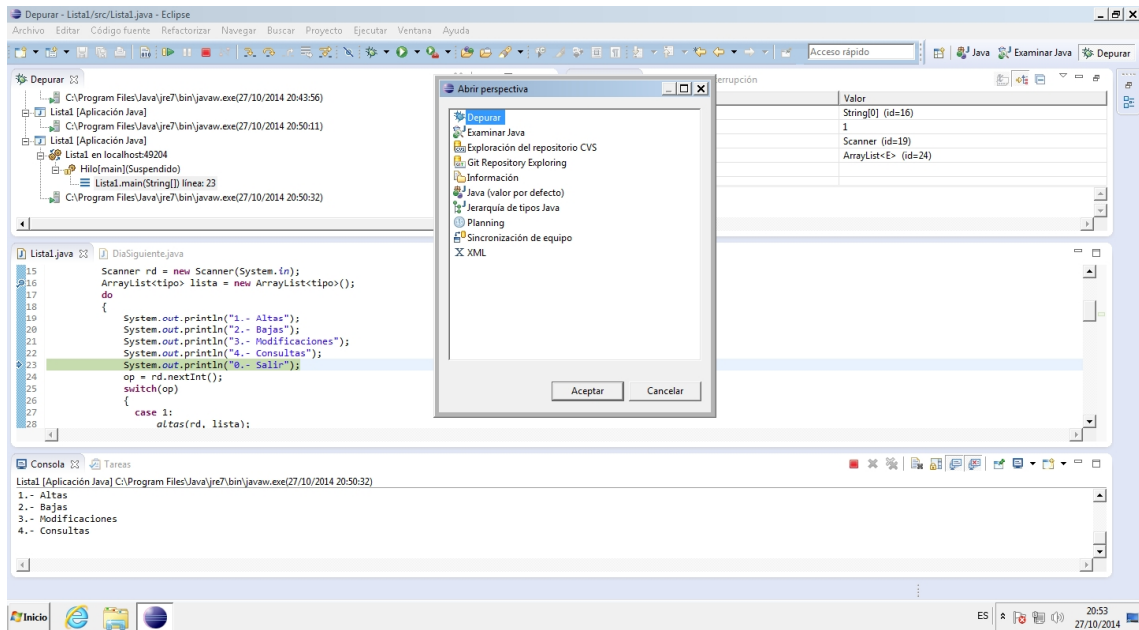
Perspectivas

En Ventana --> Abrir Perspectiva --> Depurar



O en Ventana --> Abrir Perspectiva --> Otros --> Depurar

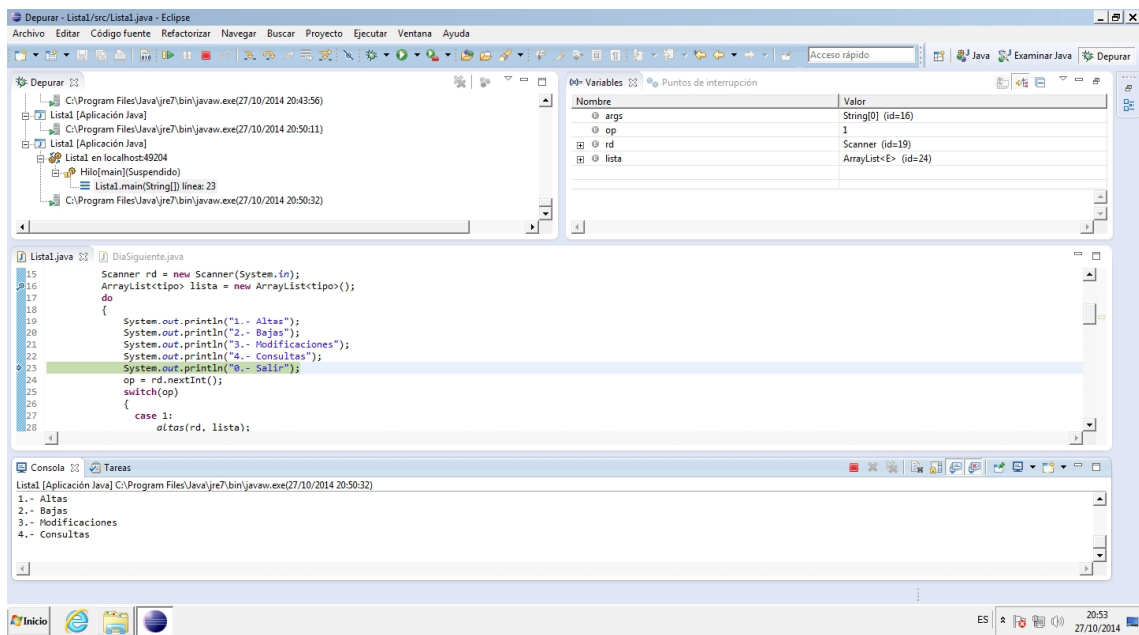




Abrimos la perspectiva Debug (Depurar)

Entramos en la perspectiva debug después de haberla añadido.

Pantalla Debug

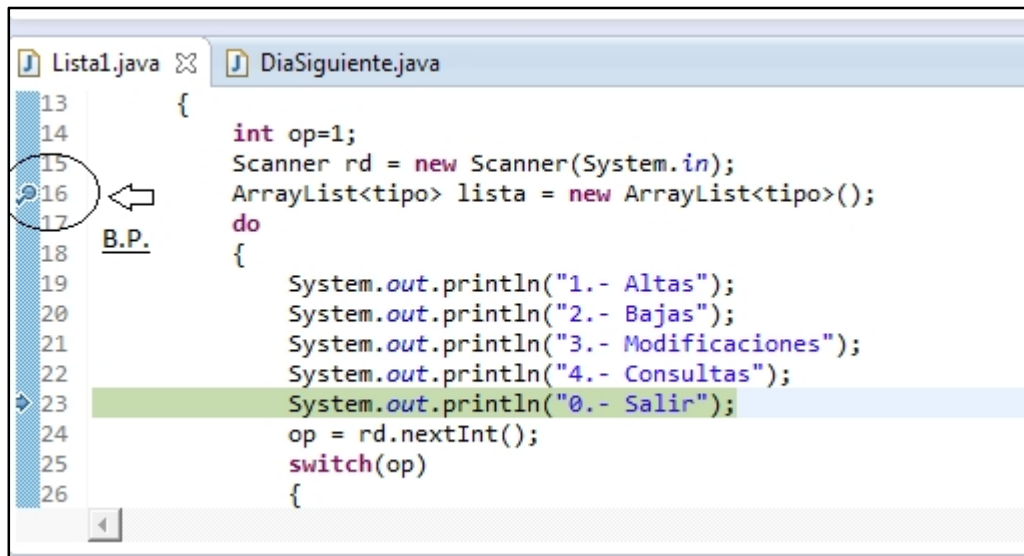


Ponemos un punto de Interrupción

Pinchamos dos clicks en la parte izquierda de una línea de código, en la columna de color azul, y se añade un punto de interrupción. Necesitamos obligatoriamente hacer una parada en una línea para poder desde ahí ejecutar paso a paso.

Vamos a depurar F11

Luego F5 / F6 para depurar.



☐ **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.

En la ventana de variables locales, podemos como en C# modificar el contenido de una variable en caliente, y seguir con la ejecución paso a paso con el nuevo valor.

Probar el programa del Día de Mañana hecho con la tabla de meses.

Poner un punto de interrupción al inicio, en la declaración de variables.

Depurar, y paso a paso con F5 / F6

Cambiar en ejecución Febrero a 30 días, en la tabla. Aceptar como fecha 29/2/2014, y ver que no hay errores y que pone como día de mañana: 30/2/2014

Capturar las pantallas y explicarlas.