

**Universidad Nacional de la Patagonia Austral  
Unidad Académica Río Gallegos**

Carrera: Analista de Sistemas

Asignatura: Resolución de Problemas y Algoritmos

---

---

**INSTRUCTIVO PARA EL USO DEL  
ENTORNO DE DESARROLLO ECLIPSE**

---

---

- 2008-

## Instructivo para el uso del entorno de desarrollo Eclipse

### 1) Creación de espacio o carpeta de trabajo (workspace)

Primero, vamos a seleccionar un workspace (espacio de trabajo), en el cual se grabarán todos los proyectos que desarrollaremos en la materia. Para nuestro caso, se creará un nuevo proyecto por cada trabajo práctico, y allí se almacenarán todos los ejercicios que desarrollemos.

Entonces, una vez ejecutado el Eclipse, nos aparece una ventana que solicita abrir un *workspace* para iniciar el trabajo.

Pulsamos el botón *Browse* y seleccionamos la carpeta RESOLUCION que nos servirá como carpeta de trabajo (Fig. 1), recordemos que allí se encontraban creados los archivos JAVA y CLASS resueltos hasta el momento.

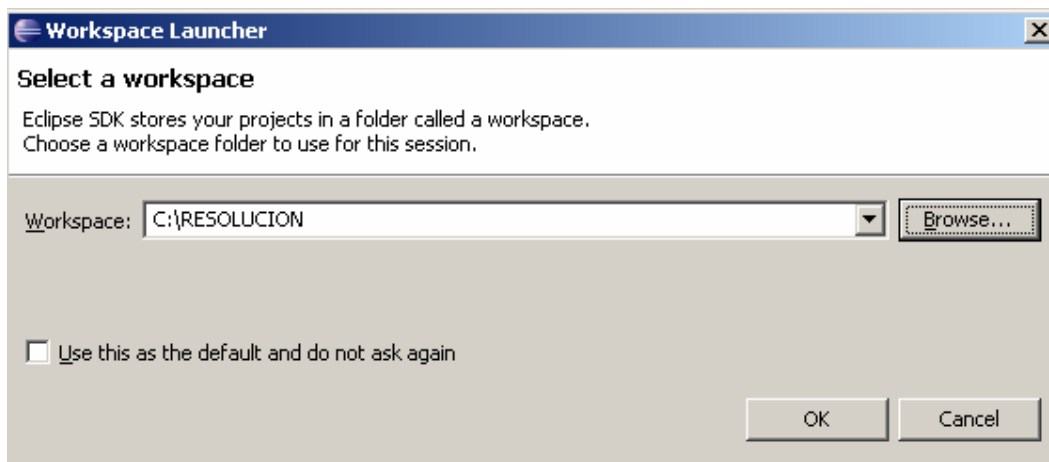


Fig. 1

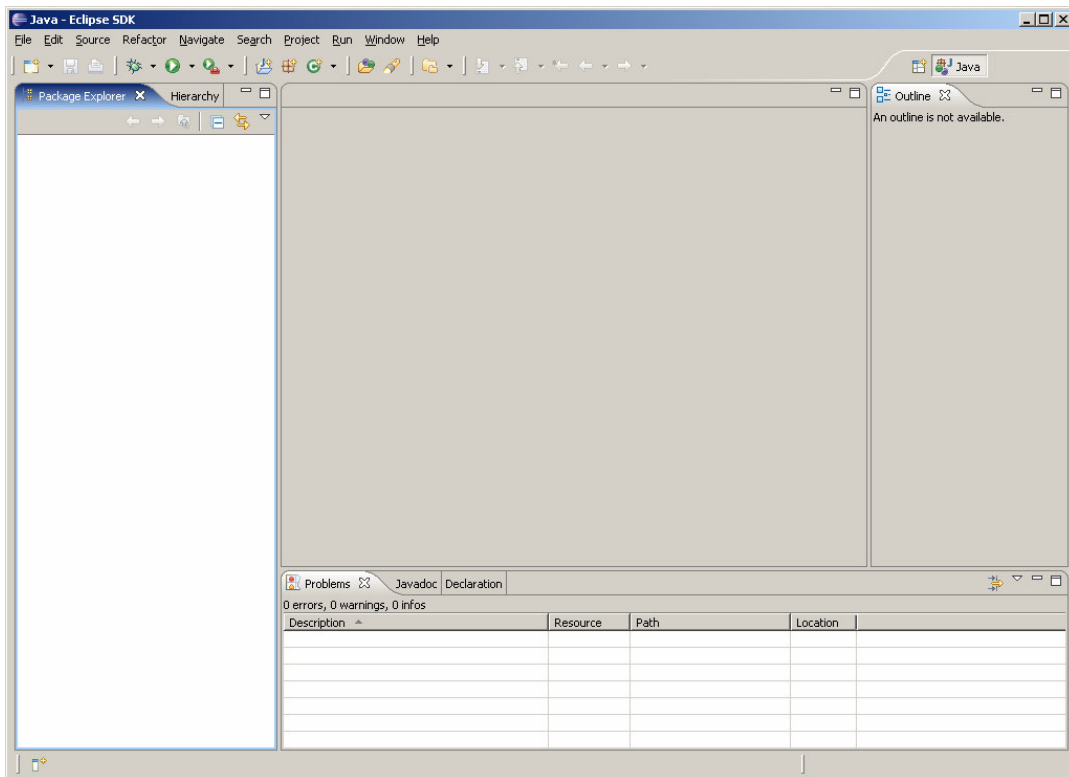
Una vez creada la carpeta de trabajo, Eclipse nos muestra la ventana de bienvenida: (Fig. 2)



Fig. 2

### Instructivo para el uso del entorno de desarrollo Eclipse

Cerramos la ventana y aparecerá la perspectiva Java, que es la que se abre por defecto: (Fig. 3)



**Fig. 3**

Una perspectiva es una distribución particular de las diferentes vistas que nos proporciona Eclipse. Las perspectivas habituales son tres:

- Java ->es la inicial y normalmente será la que se use.
- Java Browsing ->muy parecida a la Java, con diferente distribución de las vistas.
- Debug ->es la que usaremos habitualmente para realizar sesiones de depuración del código

## Instructivo para el uso del entorno de desarrollo Eclipse

### 2) Creación de Proyectos

A continuación, se creará un proyecto seleccionando en la línea de menús principal "File > New > Project..." (Fig. 4). También es posible seleccionar "New > Project..." haciendo clic derecho en cualquier parte una vista de Eclipse (como por ejemplo, el Package Explorer).

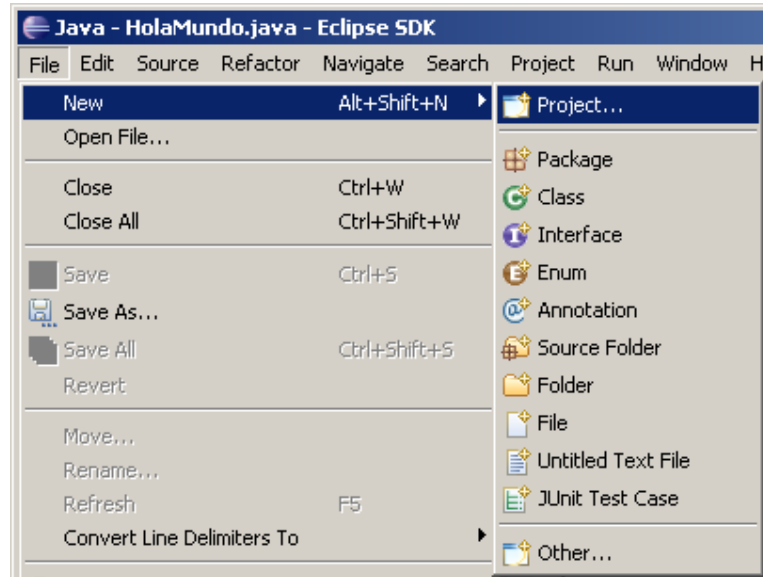


Fig. 4

De la ventana *New Project*, seleccionamos la opción *Java Project*, es decir, creamos un proyecto Java. Hacemos clic en el botón "Next >" del asistente de creación. (Fig. 5)

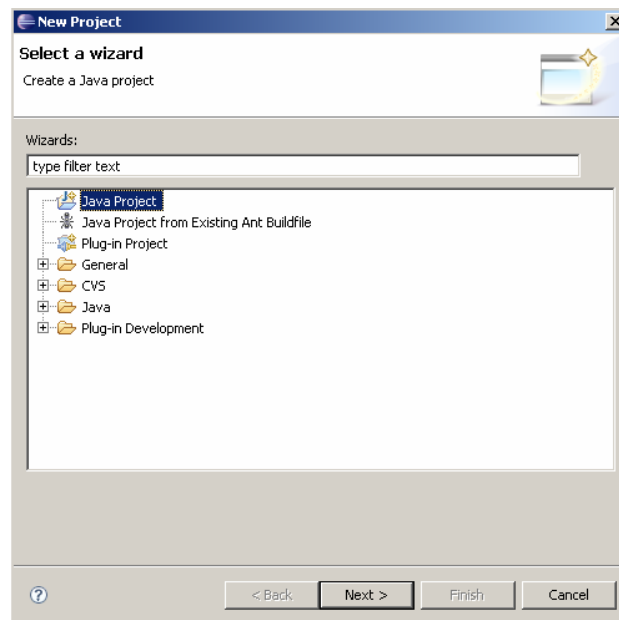
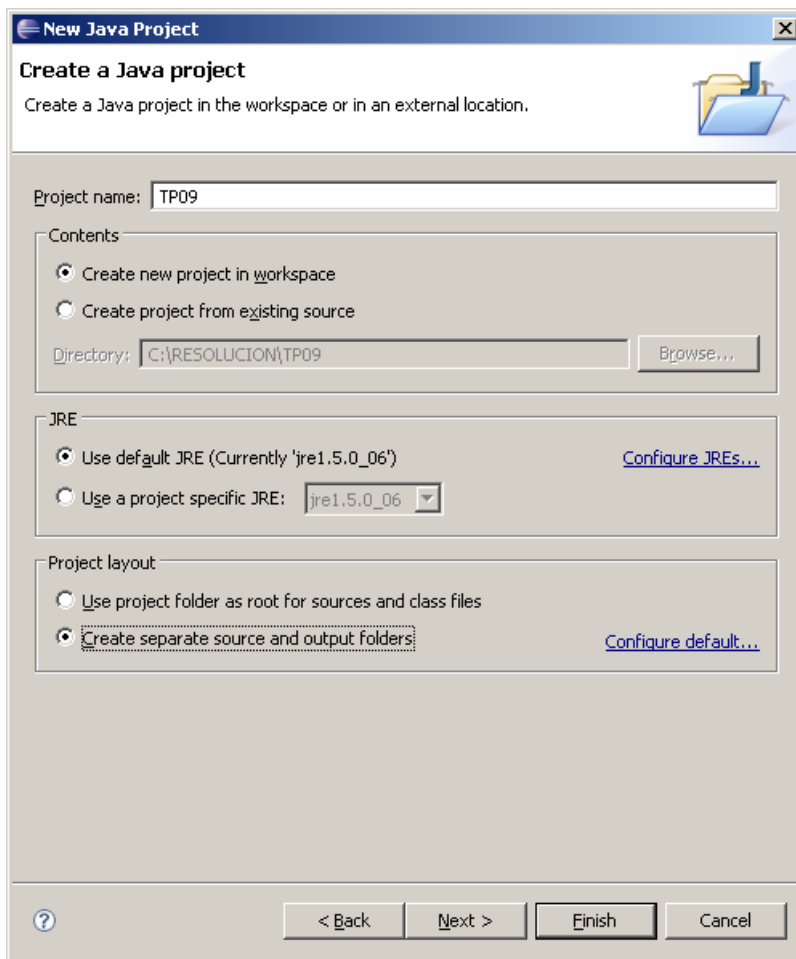


Fig. 5

### Instructivo para el uso del entorno de desarrollo Eclipse

Ingresamos un nombre de proyecto, por ejemplo TP09 para el caso del trabajo práctico nro. 9, y marcamos las opciones *Create new Project in workspace*, es decir, creamos un nuevo proyecto en el espacio de trabajo creado llamado RESOLUCION, y elegimos la opción *Create separate source and output folders*, para crear en forma automática dos carpetas; una carpeta fuente (src) para los archivos .java, y una carpeta de salida (bin) para los archivos compilados .class. De esta manera, Eclipse sabrá donde encontrar los archivos fuentes y compilados. (Fig. 4)



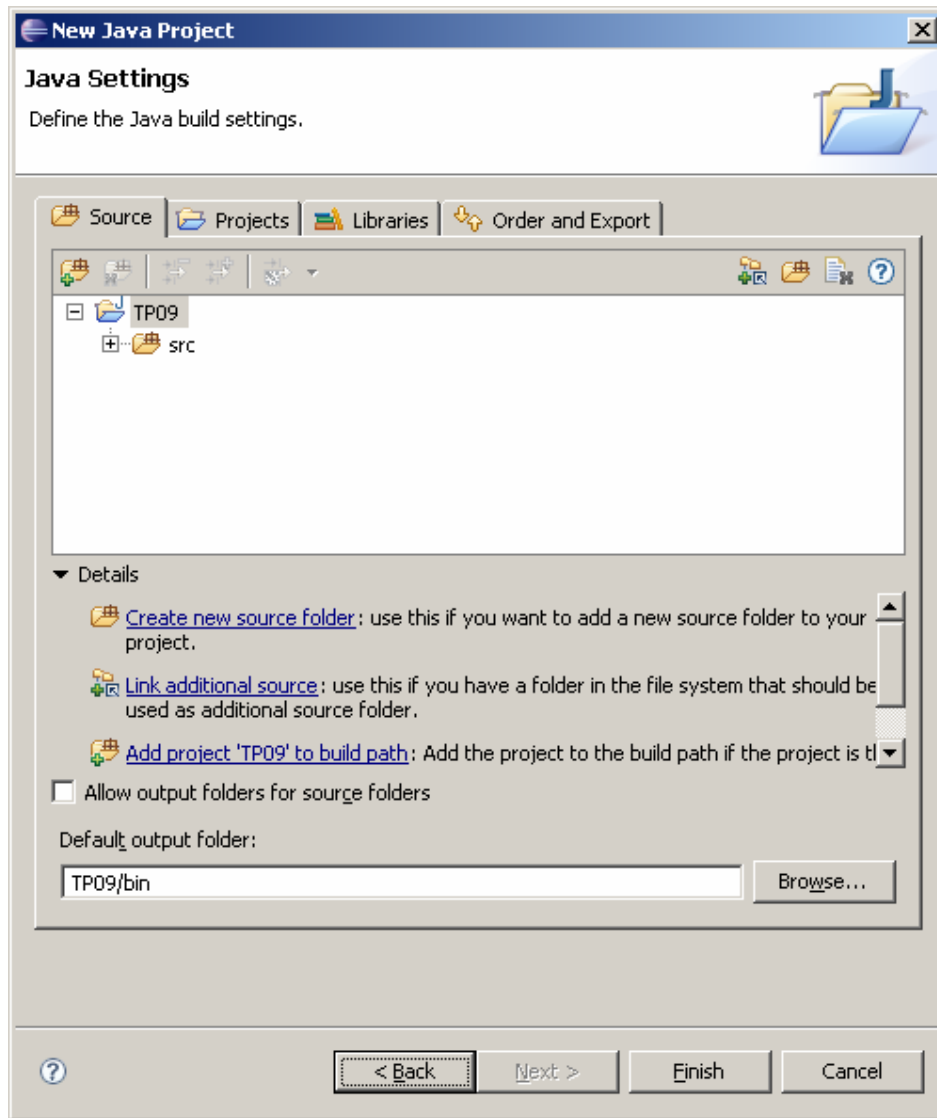
**Fig. 4**

Hacemos clic en el botón "Next >" del asistente.

Obs.: En la carpeta creada (RESOLUCION), se generan dos archivos, son: .classpath y .project

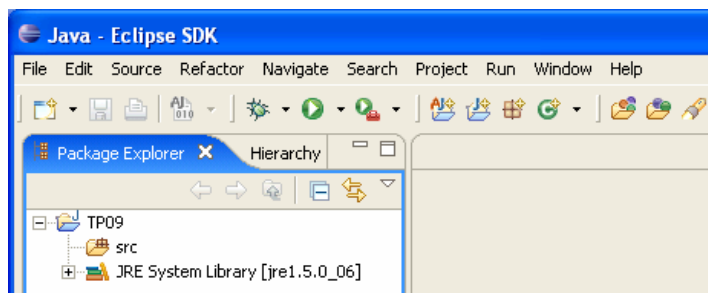
Una vez seleccionada las opciones indicadas, nos aparece la siguiente ventana *Java Settings* y pulsamos el botón "Finish" para terminar la configuración de nuestro proyecto. (Fig. 5)

**Instructivo para el uso del entorno de desarrollo Eclipse**



**Fig. 5**

Una vez terminada la configuración del proyecto, se visualiza en la ventana de Eclipse (vista de Package Explorer), el proyecto y la carpeta (src) creados. (Fig. 6)



**Fig. 6**

## Instructivo para el uso del entorno de desarrollo Eclipse

### 3) Incorporar los archivos creados

Para incorporar los archivos java creados hasta el momento, debería ser primero importado dentro de un proyecto de Eclipse. Esto puede hacerse desde el menú “File > Import...” (Fig. 7), o bien pulsando con el botón derecho en cualquier punto de la vista del “Package Explorer”. Para seleccionar un archivo o directorio hay que seleccionar “File System” en el sub menú de importación (Fig. 8).

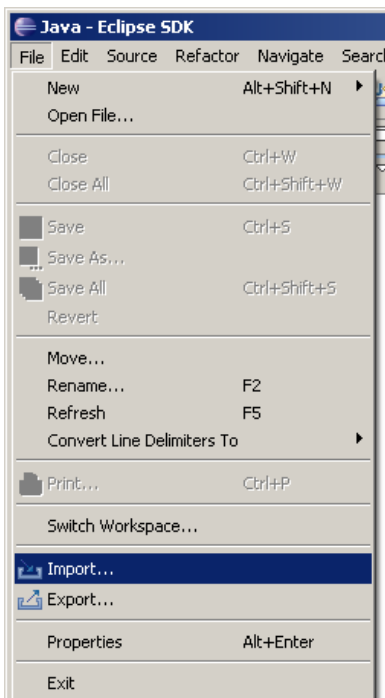


Fig. 7

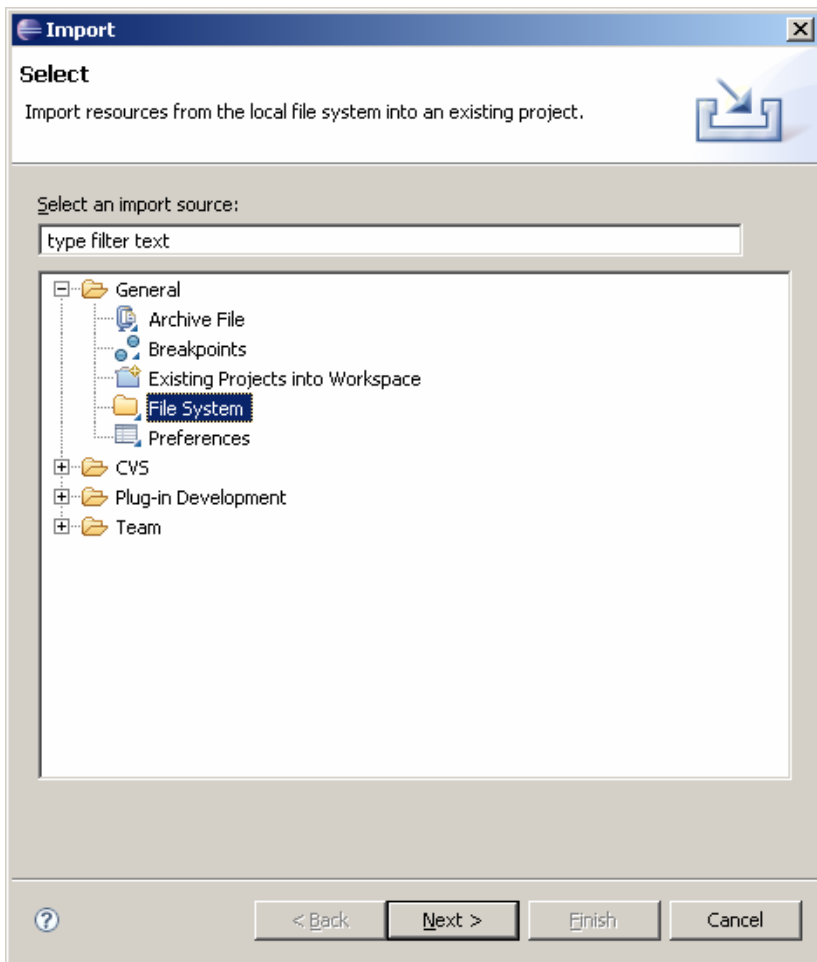
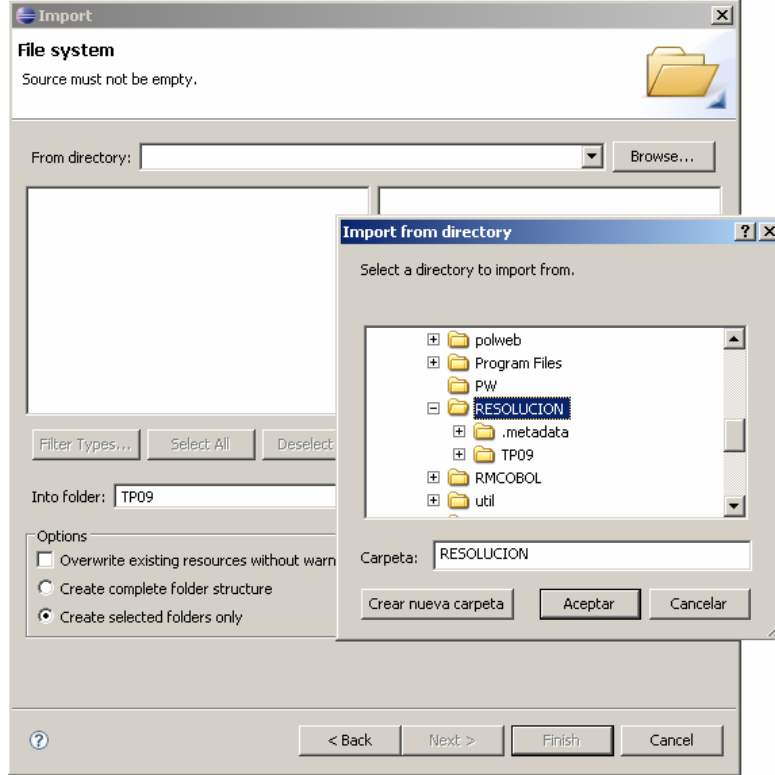


Fig. 8

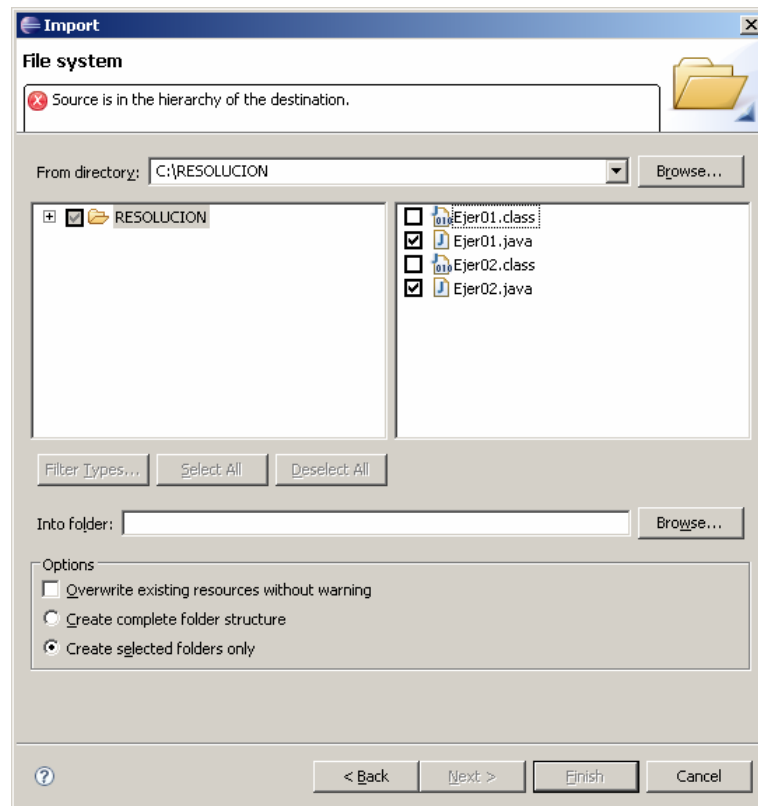
Entonces solo habrá que recorrer los directorios marcando los archivos que se deseen importar así como el proyecto y la carpeta destino. Es decir, pulsamos el botón “Browse...” de “From directory” y seleccionamos la carpeta RESOLUCION (Fig. 9).

### Instructivo para el uso del entorno de desarrollo Eclipse



**Fig. 9**

Seleccionada la carpeta, se nos despliega una subventana con los archivos que existen en la misma, clickeamos los archivos fuentes que deseamos importar (.java). (Fig. 10)



**Fig. 10**

### Instructivo para el uso del entorno de desarrollo Eclipse

Luego, pulsamos el botón “Browse...” de “Into Folder”, para elegir la carpeta destino, en este caso la subcarpeta *src* contenida dentro de la carpeta RESOLUCION, la cual contendrá los archivos fuentes de java. (Fig. 11)

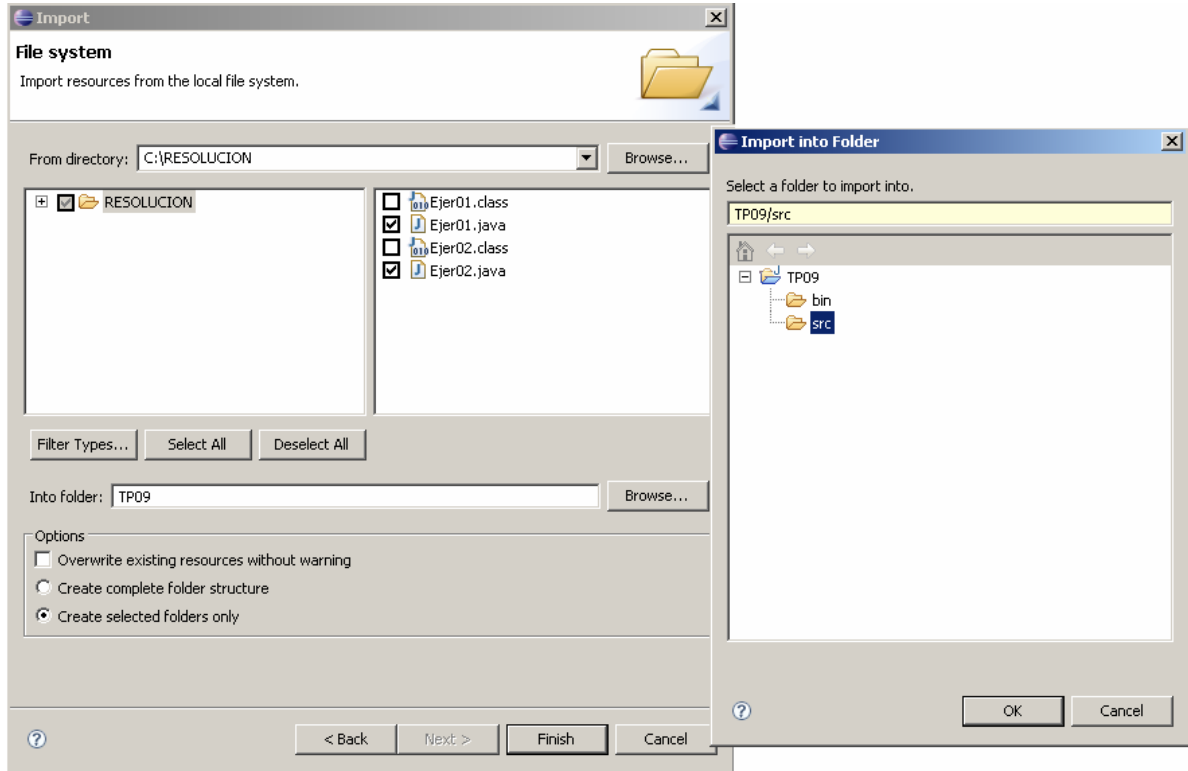


Fig. 11

Es importante tener en cuenta que todo elemento que se importa en Eclipse se duplica. Esto significa que no borrará el archivo original, de manera que seguirá teniendo una copia de reserva.

Una vez terminado esto, los archivos importados, se observaría en la vista del Package Explorer, de la siguiente forma:

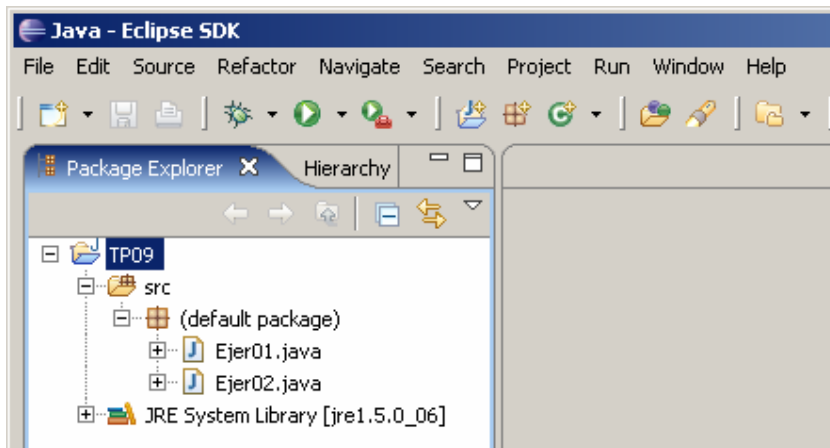


Fig. 12

## Instructivo para el uso del entorno de desarrollo Eclipse

### 4) Compilar y ejecutar

Ahora se describen las funciones de ayuda básicas para la programación java en el entorno Eclipse. Es en esta parte nos daremos cuenta que usar Eclipse para programar en Java, ahorra gran cantidad de tiempo y esfuerzo.

Es importante tener en cuenta que en Eclipse los errores de compilación se muestran en tiempo real subrayando el fragmento de código adecuado con una línea roja. Y además el entorno automáticamente compila los archivos salvados. Así pues, no será necesario pasar por el tedioso y lento proceso de compilar - observar los errores - corregir los errores.

Los errores pueden encontrarse fácilmente porque se muestran además como marcas rojas en el margen derecho del editor de código Java. También los errores y advertencias presentes en archivos ya guardados se muestran dentro de la vista de problemas (Problems). Haciendo click en cualquiera de los dos tipos de marcadores de error llevará automáticamente hasta la línea en que el error está presente. Las advertencias (warnings) se muestran de la misma manera, pero con marcas amarillas. (Fig. 13)

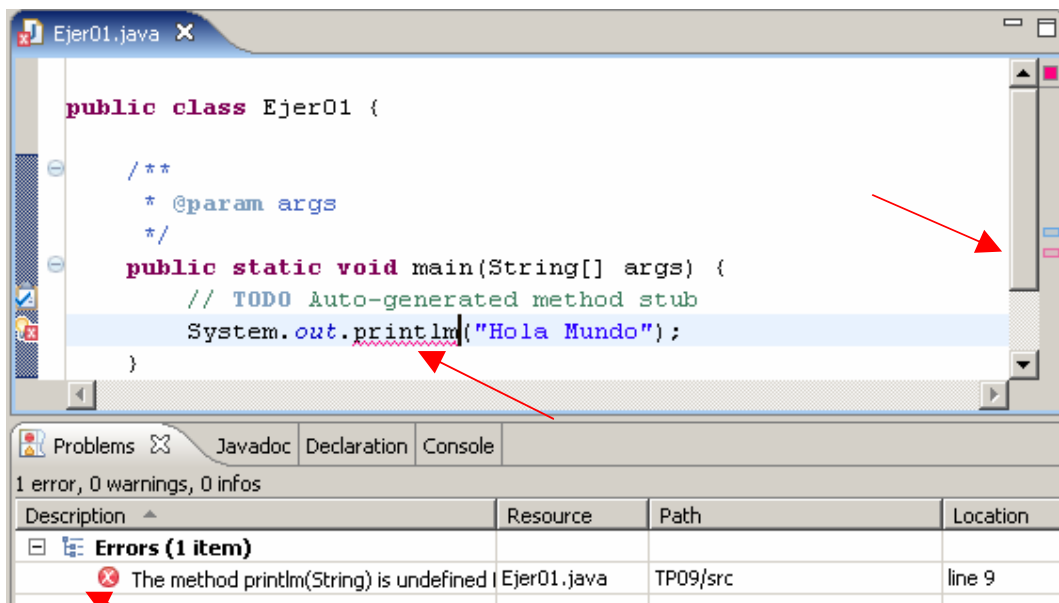


Fig. 13

### Instructivo para el uso del entorno de desarrollo Eclipse

Para ejecutar un programa dentro de Eclipse hay que elegir en la ventana Package Explorer, el archivo fuente java que deseamos ejecutar. Entonces seleccionamos "Run > Run..." del menú principal y elegimos *Java Application*. (Fig. 14)

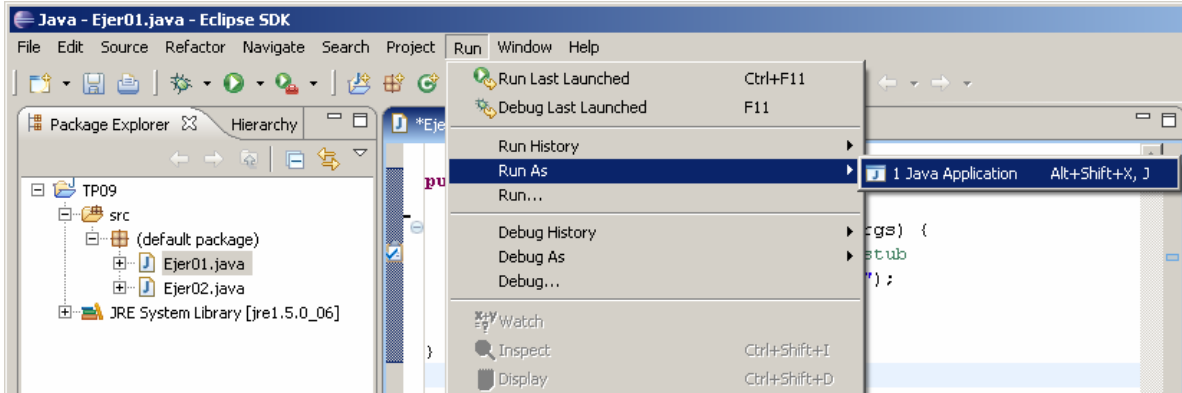



Fig. 14

o también, seleccionar desde botón  y elegir "Run As > Java Application" (Fig. 15)

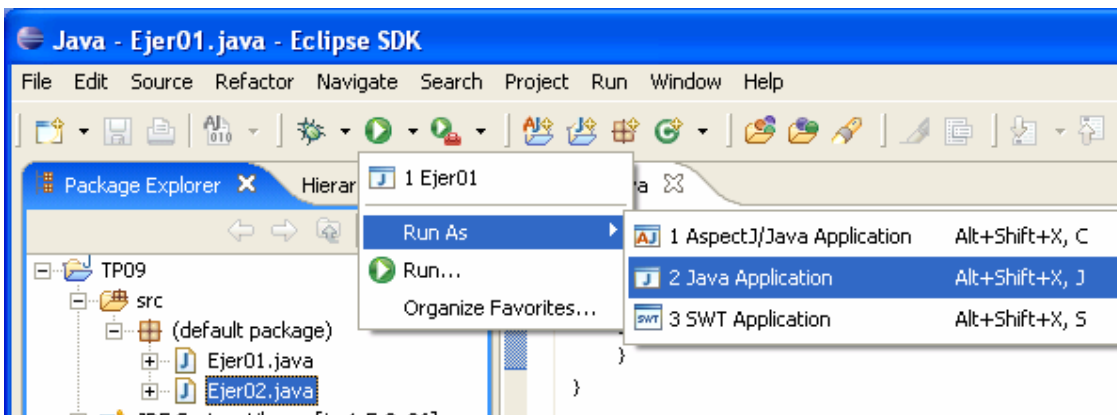


Fig. 15

La ejecución del programa se visualizará en la consola de salida, denominada

Console (Fig. 16)

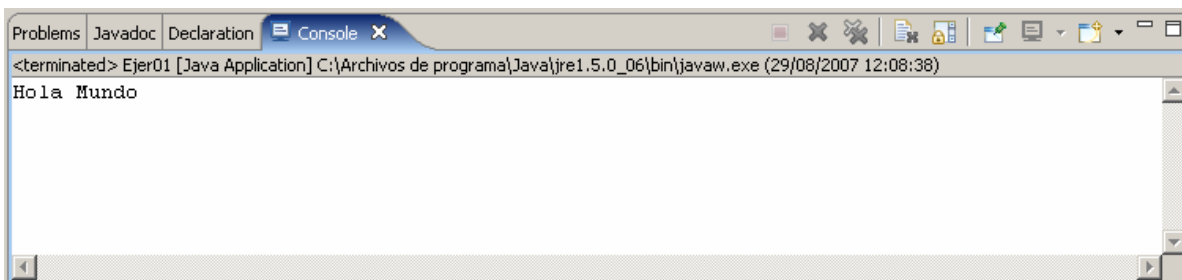


Fig. 16

## Instructivo para el uso del entorno de desarrollo Eclipse

### 5) Creación de una nueva clase

Las clases de Java son los archivos ".java" que contienen el código fuente y que serán posteriormente compilados en archivos ".class". Estos archivos Java tienen que ser almacenados dentro de la carpeta fuente recientemente creada (llamada "src").

Entonces, haciendo clic derecho en la carpeta fuente "src" dentro de la vista del Package Explorer, aparecerá un menú contextual. Seleccionando "New >" en dicho menú se mostrará una lista con los diversos elementos de Java que pueden ser creados.

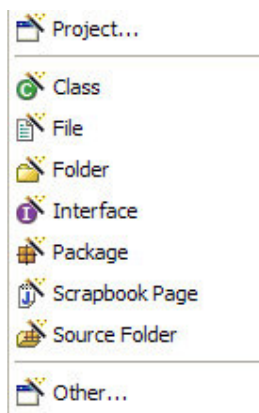


Fig. 17

En este caso, seleccionaremos la opción Class (Fig. 18)

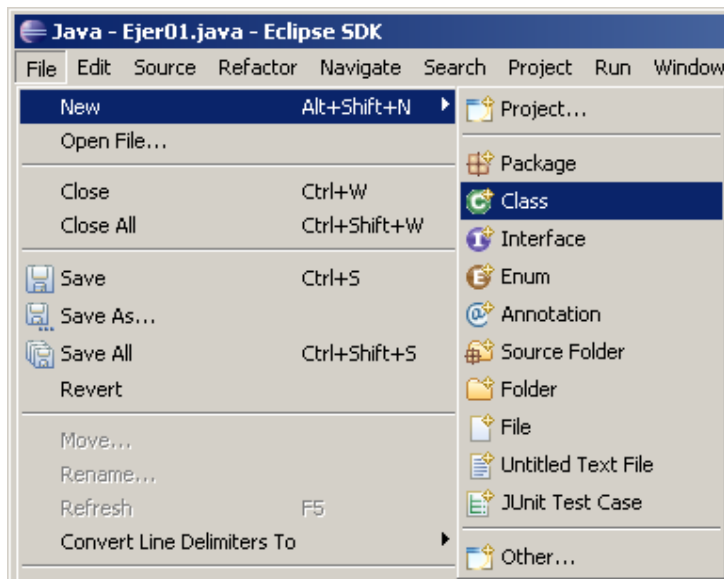


Fig. 18

Algunos de dichos elementos pueden ser creados también realizando clic sobre los iconos del menú de la parte superior de la pantalla. (Fig. 19)

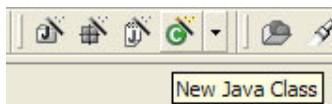


Fig. 19

### Instructivo para el uso del entorno de desarrollo Eclipse

La carpeta fuente especificada debería ser la carpeta recientemente definida (src). El último campo obligatorio que debería ser rellenado antes de proceder a la creación de la clase Java es el propio nombre de la clase. Las convenciones de Java sugieren que el nombre de una clase debería comenzar con mayúscula. (Fig. 20)

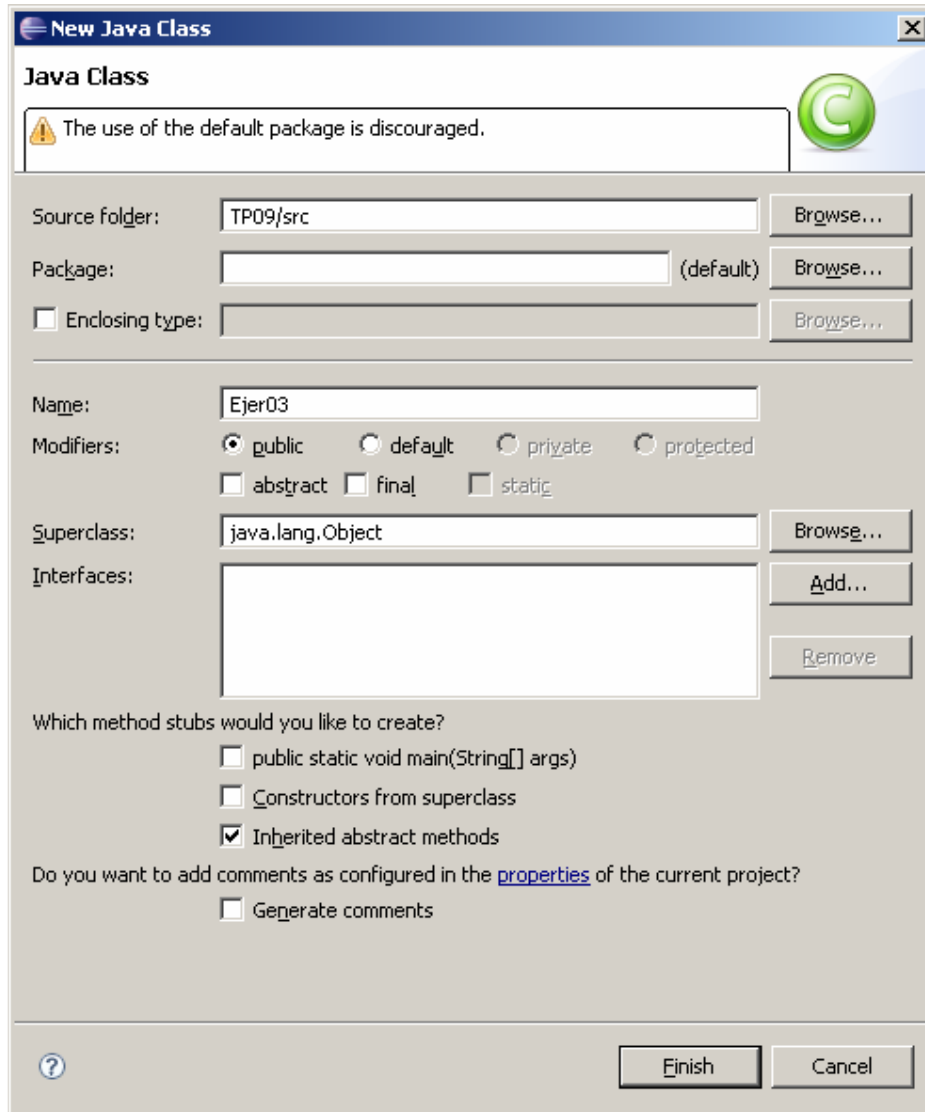


Fig. 20

Si se desea que la nueva clase contenga un método "main" (es decir, el punto inicial de ejecución del programa), puede añadirse dicho método automáticamente sólo con marcar la casilla con la opción apropiada.

Es interesante destacar que los archivos compilados ".class" no son visibles en la ventana "Package Explorer". Sin embargo, basta con escribir y guardar un archivo ".java" para que se cree un archivo ".class" resultante de compilar el archivo fuente anterior.

### Instructivo para el uso del entorno de desarrollo Eclipse

---

Una vez presionado el botón <Finish>, se podrá editar la clase recientemente creada en la ventana de edición de Eclipse (Fig. 21).

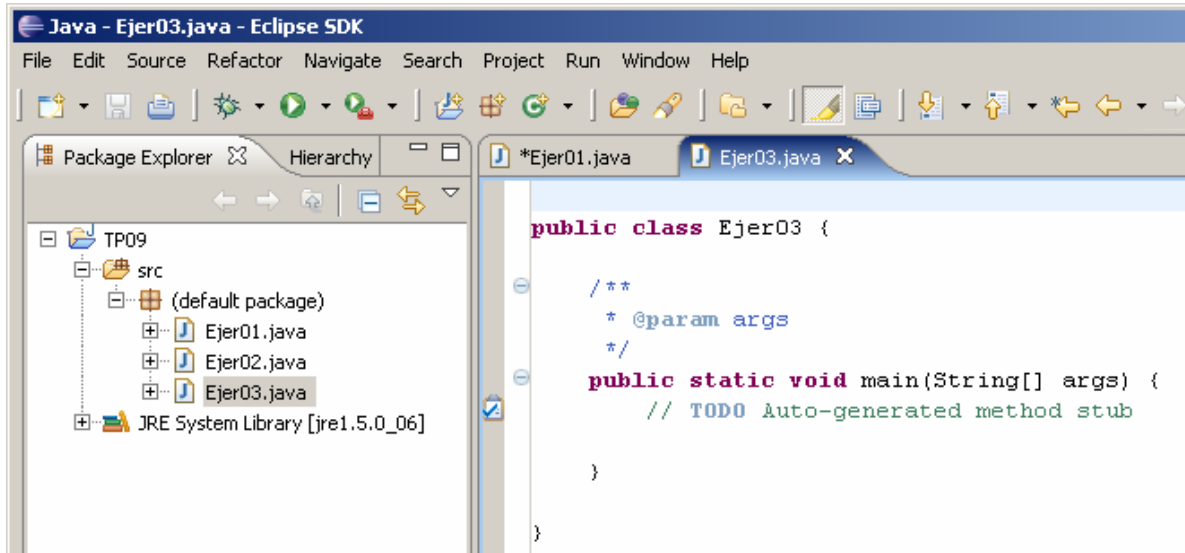


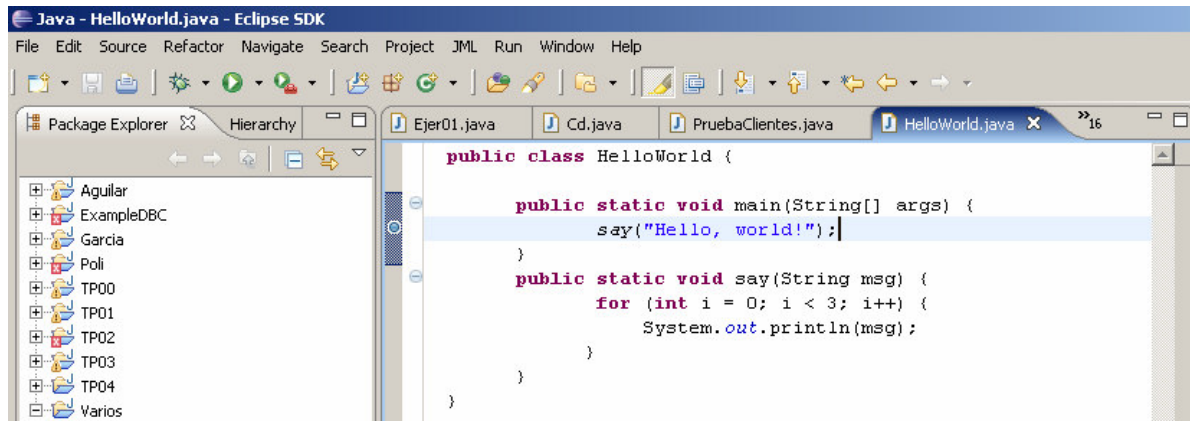
Fig. 21

# **ANEXO: DEPURACION EN ECLIPSE**

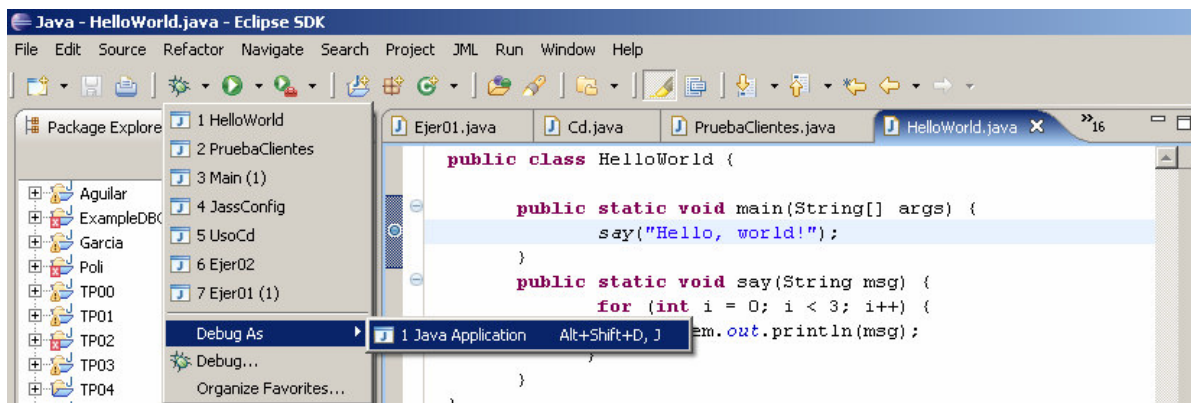
## Depuración de programas en Java

Realizar los siguientes pasos:

1. Establecer un punto de ruptura en el código para que el depurador suspenda la ejecución: para ello, hacer doble clic en el margen gris en el lado izquierdo del editor. Un punto azul aparecerá, indicando un corte activo.

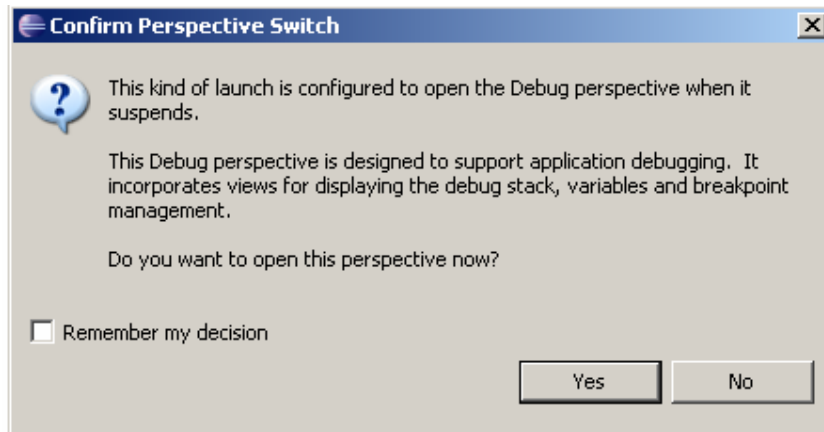


2. Seleccionar el programa a realizar debug y elegir “Run – Debug As... - Java Application” del menú principal.



### Instructivo para el uso del entorno de desarrollo Eclipse

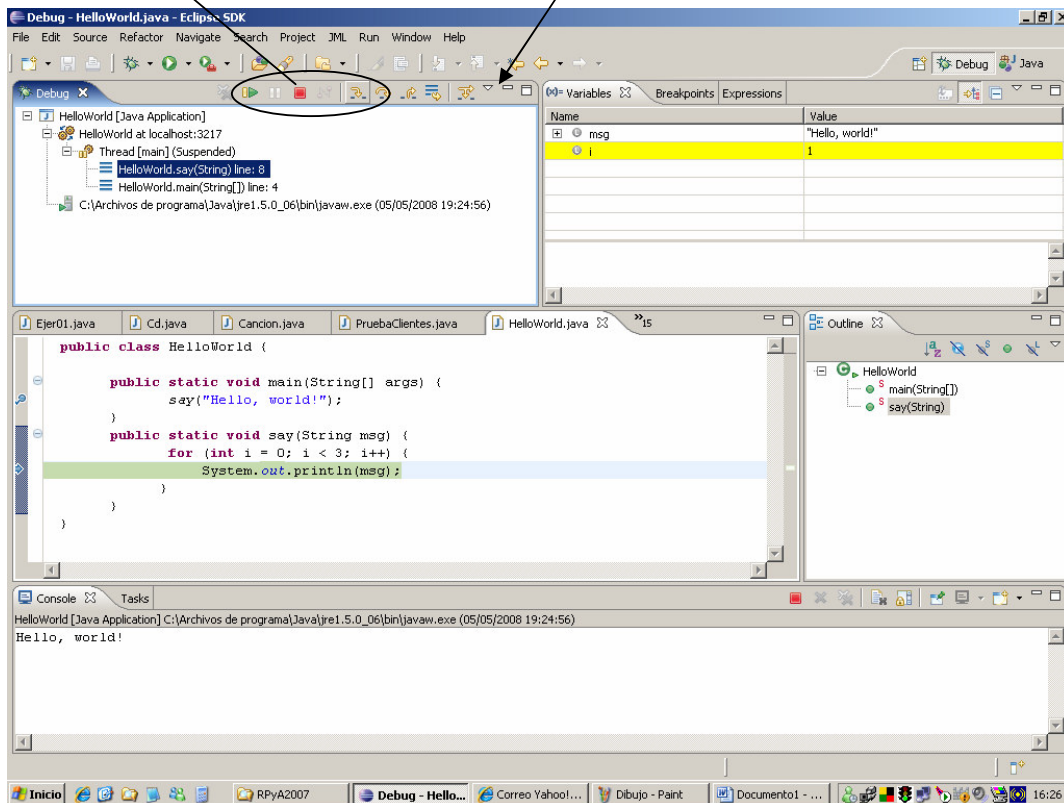
- Eclipse va a iniciar el programa y advierte que va a cambiar la vista a perspectiva Debug. Aceptar el cambio de perspectiva.



- Seleccionar de la barra de botones de Debug, el botón *Step Into* para avanzar en la ejecución del debug y observar los distintos valores de las variables:

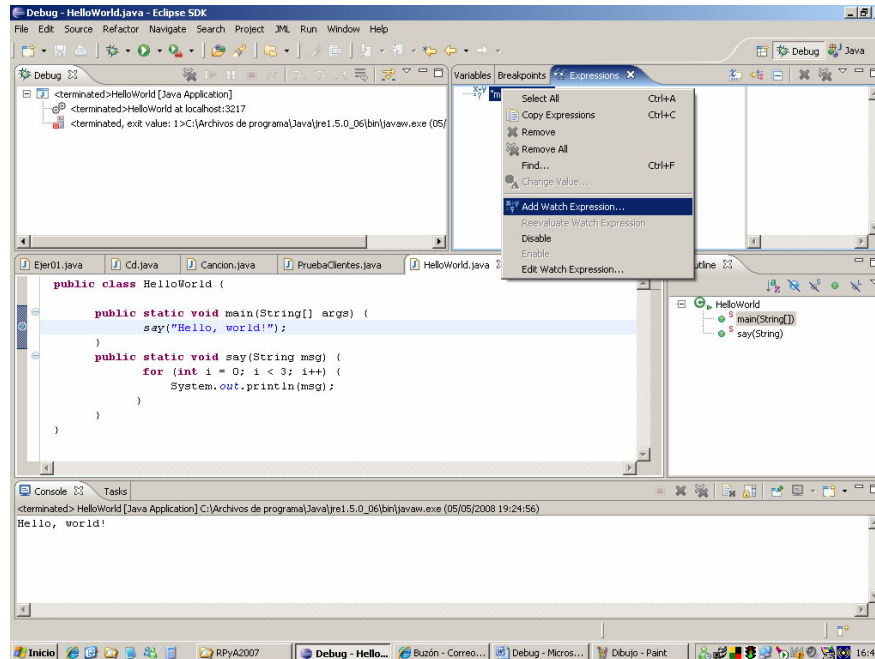
Cuadro de Botones p/Debug: Step With Filters, Step Into, Step Over, Sten Return

Lista de variable del programa



## Instructivo para el uso del entorno de desarrollo Eclipse

5. Para agregar nuevas expresiones/variables y verificar su valor, seleccionar de la solapa *Expressions* y *Add Watch Expresión...*, como se ve en la figura siguiente:



---

**Instructivo para el uso del entorno de desarrollo Eclipse**

---

**REFERENCIAS:**

- Eclipse Tutorial (Enrique Serrano): <http://eclipsetutorial.forge.os4os.org/in1.htm>
- Manual para el desarrollador:  
[http://www.saig.es/descargas/manuales/Kosmo\\_manual\\_desarrollador.pdf](http://www.saig.es/descargas/manuales/Kosmo_manual_desarrollador.pdf).
- Debugging a Java Program with Eclipse (David Gallardo):  
[http://www.developer.com/java/other/article.php/10936\\_2221711\\_1](http://www.developer.com/java/other/article.php/10936_2221711_1)
- Eclipse Tutorial:  
<http://eclipsetutorial.forge.os4os.org/ejecutar.htm>