



Índice

1. El entorno de desarrollo Eclipse	1
1.1. ¿Qué es Eclipse?	1
1.2. Trabajando con Eclipse	1
1.2.1. Creación de un proyecto	1
1.2.2. Creando clases.	2
1.2.3. Ejecutando el programa	2
1.2.4. Depuración de programas.	3
1.2.5. Otras herramientas interesantes.	3

1. El entorno de desarrollo Eclipse

1.1. ¿Qué es Eclipse?

Eclipse es una plataforma de desarrollo *open source* basada en Java. Es un desarrollo de IBM cuyo código fuente fue puesto a disposición de los usuarios. En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (*plug-in*). Hay *plug-ins* para el desarrollo de Java (JDT Java Development Tools) así como para el desarrollo en C/C++, COBOL, etc. La versión instalada en el laboratorio incluye el *plug-in* JDT.

1.2. Trabajando con Eclipse

Al ejecutar Eclipse aparece una ventana como la mostrada en la figura 1.

Eclipse contiene una serie de *perspectives*. Cada perspectiva proporciona una serie de funcionalidades para el desarrollo de un tipo específico de tarea. Por ejemplo la perspectiva Java combina un conjunto de *views* que permiten ver información útil cuando se está escribiendo código fuente, mientras que la perspectiva de depuración contiene vistas que muestran información útil para la depuración de los programas Java.

La barra de herramientas vertical (en la parte izquierda) muestra las perspectivas abiertas y permite pulsando sobre ellas cambiar de una a otra. La perspectiva activa se muestra en la barra del título en la parte superior de la ventana.

1.2.1. Creación de un proyecto

Eclipse permite organizar los ficheros en forma de proyecto. Para crear un proyecto Java procederemos del siguiente modo:

1. Seleccionando en el menú **File** → **New** → **Project** o pulsando con el botón derecho del ratón sobre la vista *Navigator* en la perspectiva *Resource* y seleccionando **New** → **Project**.
2. Aparece una ventana en la que se puede seleccionar el tipo de proyecto. En este caso se pulsa sobre **Java** en la parte izquierda y **Java project** en la derecha. Pulsar sobre el botón **Next**. Ver la figura 2
3. A continuación se piden los datos sobre el proyecto (nombre y ruta donde se almacenarán los ficheros asociados al proyecto). Una vez introducidos se pulsa sobre **Finish**. Véase la figura 3.
4. Eclipse abre automáticamente la perspectiva Java cuando se crea un proyecto Java. Se crea el directorio especificado con dos ficheros `.project` y `.classpath` que contienen información sobre el proyecto. La vista *Navigator* de la perspectiva *Resource* y la vista *Package Explorer* de la

Listing 1: Código generado utilizando el asistente

```
/*
 * Created on 13-ago-2003
 *
 * To change the template for this generated file go to
 * Window>Preferences>Java>Code Generation>Code and Comments
 */

/** Clase
 * @author juan
 *
 * Descripción:
 */
public class HolaMundo {

    public static void main(String [] args) {

    }

}
```

perspectiva Java contienen un directorio con el nombre del proyecto. Como se comentó anteriormente, se puede pasar de una perspectiva a otra pulsando sobre los botones de la barra vertical izquierda.

- Hay una perspectiva más asociada con Java, se puede abrir del siguiente modo: **Window** → **Open perspective** → **Java browsing**. En esta perspectiva aparecen views correspondientes al proyecto, a paquetes, a clases e interfaces y a los miembros de estas.

1.2.2. Creando clases.

Las clases se pueden crear de dos formas diferentes:

- Utilizando un asistente. Pulsamos con el botón derecho sobre el proyecto → **New** → **Class**. Aparece una ventana como la que se muestra en la figura 4. Los campos que nos interesan por ahora son: el nombre de la clase, el modificador, y si queremos que esta clase tenga un método `main(String[] args)`. Al pulsar sobre **Finish** (con los datos que se muestran en la figura 4 se crea un fichero `HolaMundo.java` con el código mostrado en el listado 1 (no será exactamente así ya que yo he personalizado el comentario que aparece). Si exploramos el contenido del directorio `c:\tmp\prac1` vemos que además del fichero `HolaMundo.java` hay otro fichero `HolaMundo.class`, este fichero es el que contiene el código compilado a partir de `HolaMundo.java` (Eclipse compila el proyecto cada vez que se guarda).
- Escribiendo directamente toda la clase: Pulsamos con el botón derecho sobre el proyecto → **New** → **File**. Se abre una ventana como la que se muestra en la figura 5 donde hay que poner el nombre del fichero. Al pulsar sobre **Finish** se crea un fichero vacío con el nombre `HolaMundo.java`.

No se tiene una preferencia por alguno de los dos métodos expuestos. Aunque, como recomendación, al principio es conveniente escribir toda la clase para saber lo que se está haciendo y además... ¡en el examen no habrá ningún asistente!

El asistente simplemente a creado un esqueleto ahora falta completar el código. Añadimos las líneas necesarias para completar el programa. El resultado se muestra en la figura 6.

1.2.3. Ejecutando el programa

Los programas se pueden ejecutar dentro de Eclipse.

- Con la perspectiva Java abierta, seleccionar en el menú: **Run** → **Run...**
- En la ventana que se abre, pulsar 2 veces sobre **Java Application**. En el panel **Main** se rellena la información tal y como aparece en la figura 7.
- Puesto que en el ejemplo propuesto se necesitan argumentos pulsamos sobre el panel **Arguments** y ponemos el argumento que vamos a pasar al programa. Un ejemplo se muestra en la figura 8.
- Una vez proporcionada la información necesaria (hay más paneles pero por ahora no es necesario conocerlos) se pulsa sobre el botón **Run**.
- Este programa simplemente muestra un mensaje por consola. La consola está integrada (es una vista más) dentro de Eclipse. Así tras ejecutar el programa, en la vista Consola veremos lo que se muestra en la figura 9.

Cuando el programa no requiere ningún ajuste especial (por ejemplo, no se requiere el paso de argumentos) se puede hacer de forma más rápida pulsando sobre **Run** → **Run as** → **Java Application**.

Hay barras de tareas para el acceso rápido a algunas funciones (entre ellas ejecutar). La figura 10 muestra algunas de ellas.

1.2.4. Depuración de programas.

Dentro del entorno de desarrollo de Eclipse se pueden depurar programas desarrollados en Java.

Window → **Open perspective** → **Debug**. Aparecen una serie de vistas similares a las de la figura 11.

Una vez hemos abierto la perspectiva de depuración podemos parar la ejecución del programa en una determinada línea (poniendo un *breakpoint*) e inspeccionar las variables locales.

- Para poner un breakpoint, en la vista donde se encuentra el código fuente seleccionamos la línea donde queremos que se detenga la ejecución y seleccionamos en el menú **Run** → **Add/remove Breakpoint**. Vemos que se muestra un punto azul en la parte izquierda de la línea. Ahora ya podemos lanzar el depurador.
- Seleccionamos en el menú **Run** → **Debug**. La ejecución del programa se detiene en el primer breakpoint. Una vez el programa está detenido, en una de las vistas podemos ver el valor de las variables o ver los breakpoints que hemos definido. En la figura 12 se muestra el programa detenido en una línea y se muestra la vista *Variables* con el contenido de una variable local.
- Una vez inspeccionado el código donde está el problema se puede optar por ejecutar el programa hasta que termine (**Run** → **Resume**) o terminar el programa inmediatamente (**Run** → **Terminate**)

1.2.5. Otras herramientas interesantes.

- El editor de Java ofrece correcciones a problemas encontrados mientras se escribe el código y tras compilar. El editor muestra que existen propuestas para la corrección de un problema o aviso mediante una bombilla visible en la parte izquierda del editor. Si se pulsa con el botón izquierdo sobre esta bombilla (o también mediante (**Edit** → **Quick Fix**)) se muestran las propuestas para el problema en la posición del cursor.
- Si se desean importar recursos a un proyecto se puede realizar del siguiente modo: en la vista *Navigator* se pulsa sobre el botón derecho y aparece un menú, se selecciona **Import** y aparece una ventana desde la que se puede seleccionar el directorio donde están los recursos y cuales se desean incorporar al proyecto.



- Desde un proyecto se pueden exportar todos o algunos de los ficheros que lo conforman. Para ello en la vista *Navigator* se pulsa sobre el botón derecho y aparece un menú, se selecciona **Export** y aparece una ventana en la que podemos indicar cómo se va a exportar (un fichero zip, tal cual aparecen en el proyecto, ...). La siguiente ventana sirve para seleccionar los ficheros que se desean exportar y a dónde.
- Si colocamos el ratón sobre un método (sin pulsar) se nos muestra la declaración del método (qué devuelve y qué parámetros acepta). Si colocamos el ratón (sin pulsar) sobre una variable aparece información sobre el tipo de la variable. Al escribir código podemos pulsar Ctrl + espacio y nos aparece un menú con posibles formas de finalizar la sentencia que se está escribiendo. Un ejemplo se muestra en la figura 13.

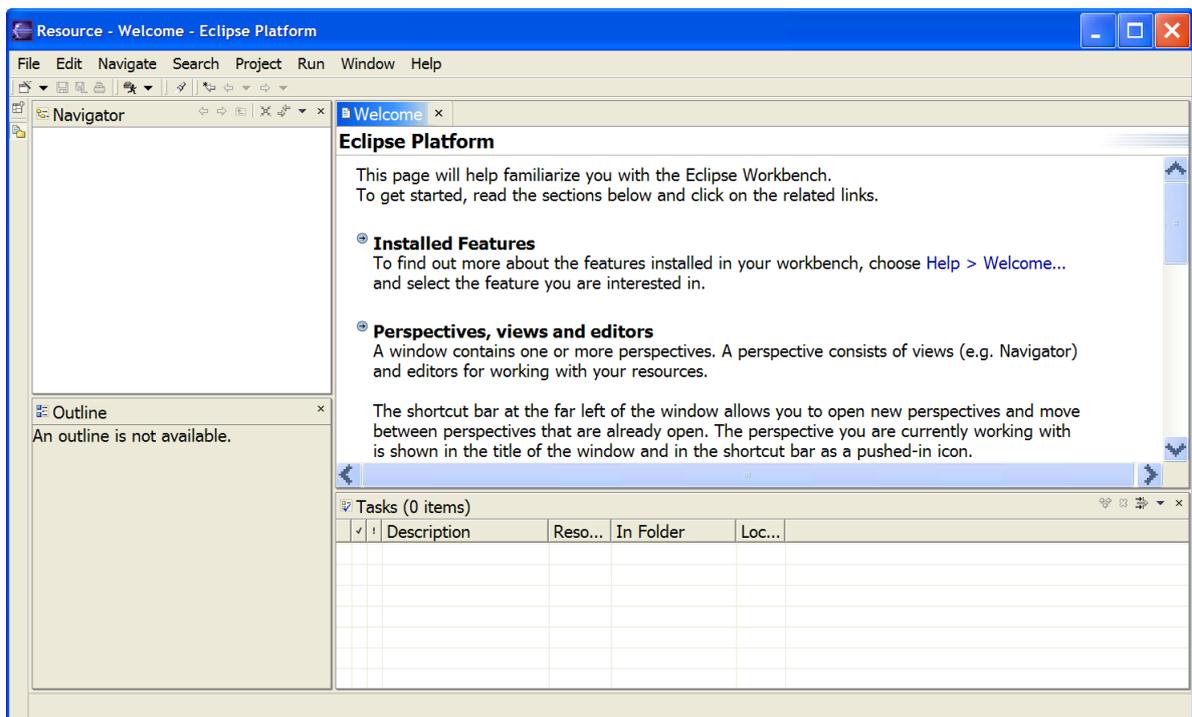


Figura 1: El entorno Eclipse

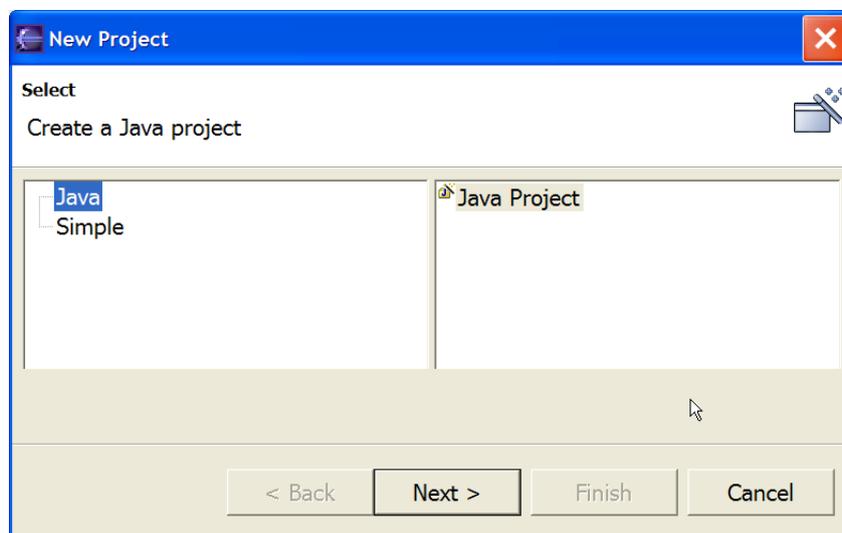


Figura 2: Selección del tipo de proyecto

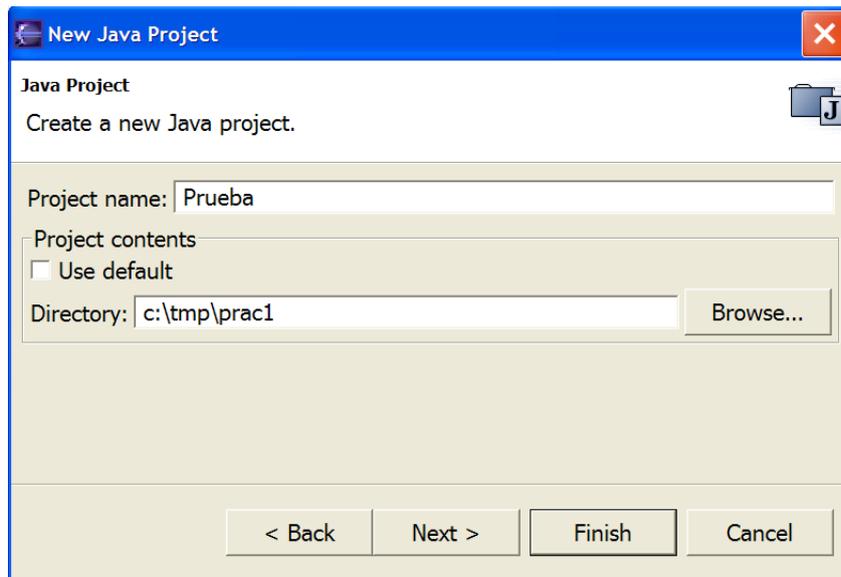


Figura 3: Datos sobre el proyecto

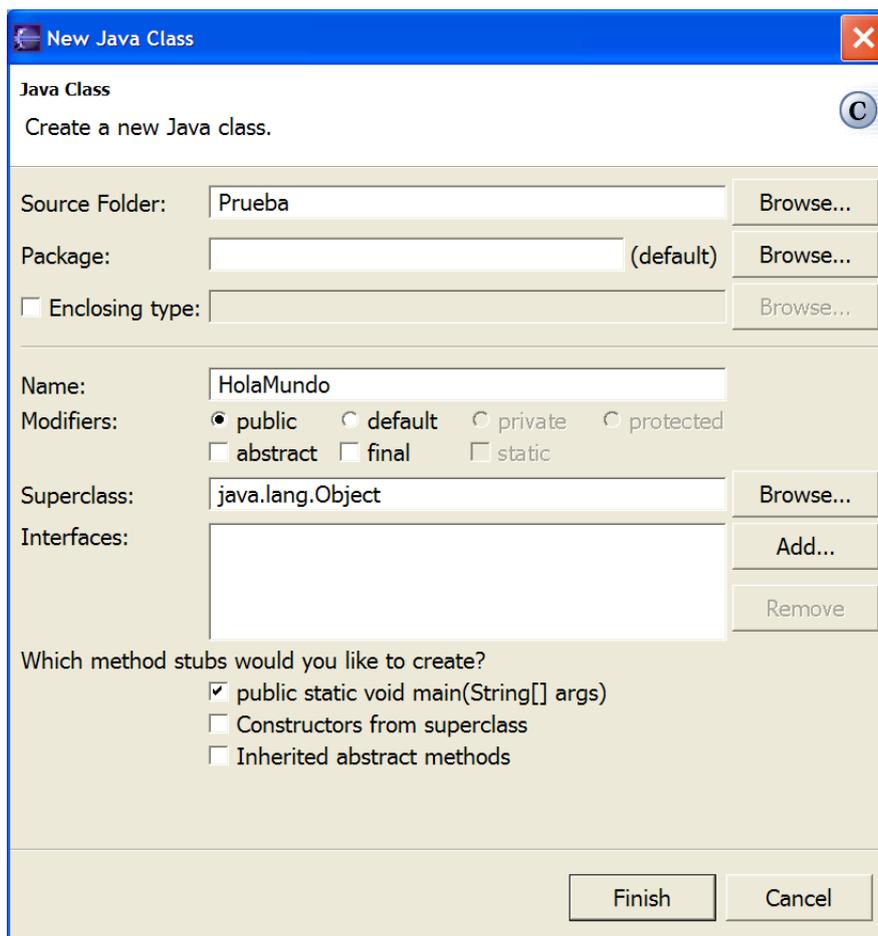


Figura 4: Creación asistida de una clase

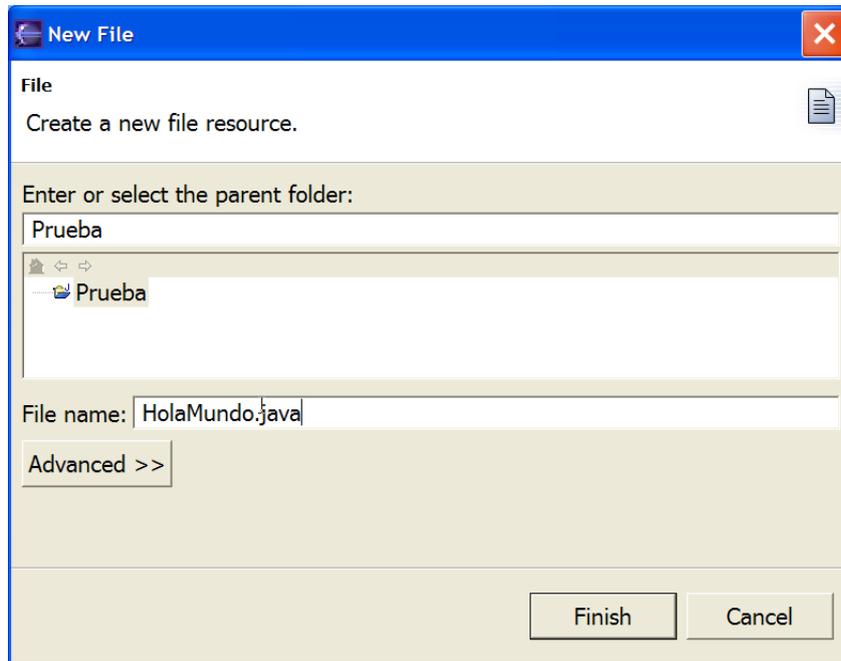


Figura 5: Creación de un fichero vacío

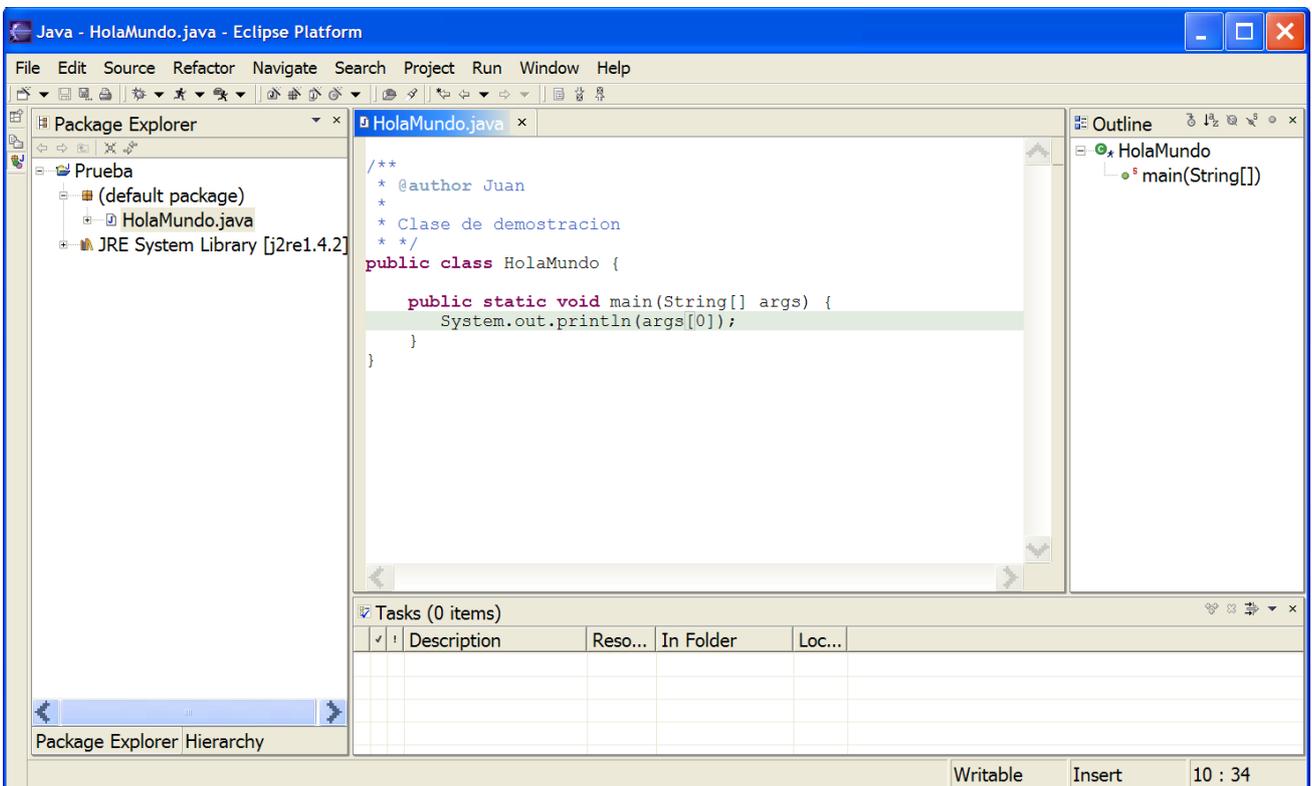


Figura 6: La perspectiva Java con las vistas mostrando una clase.

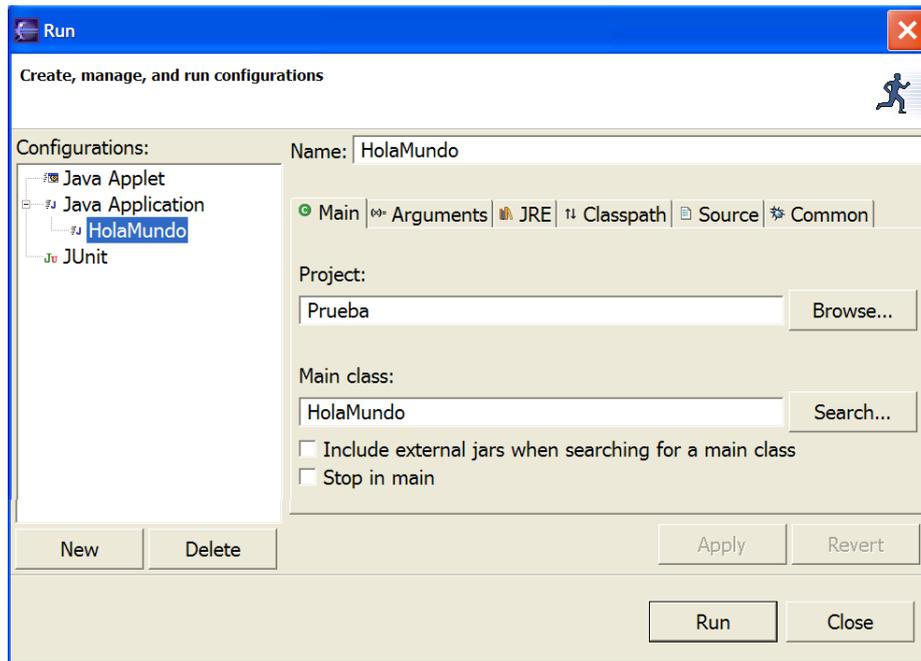


Figura 7: Configuración para ejecutar la aplicación: proyecto y clase con el main

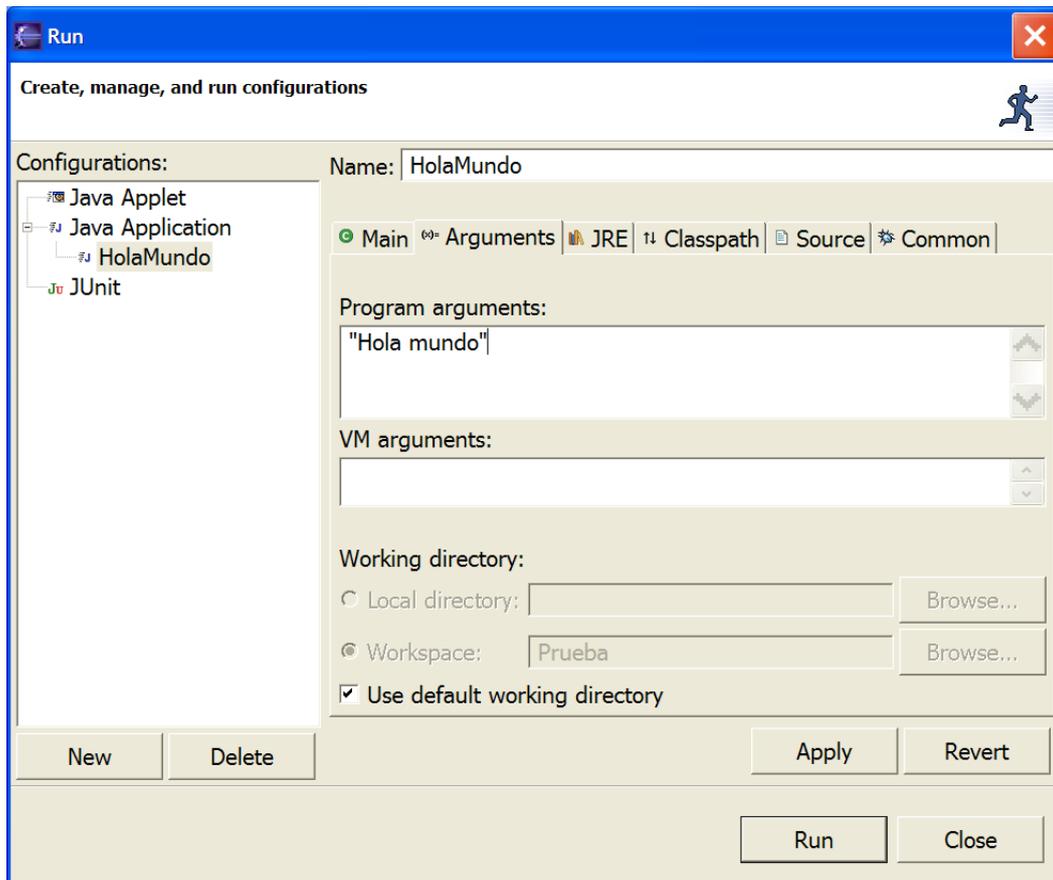


Figura 8: Argumentos para enviar al programa y argumentos para enviar a la Máquina Virtual.

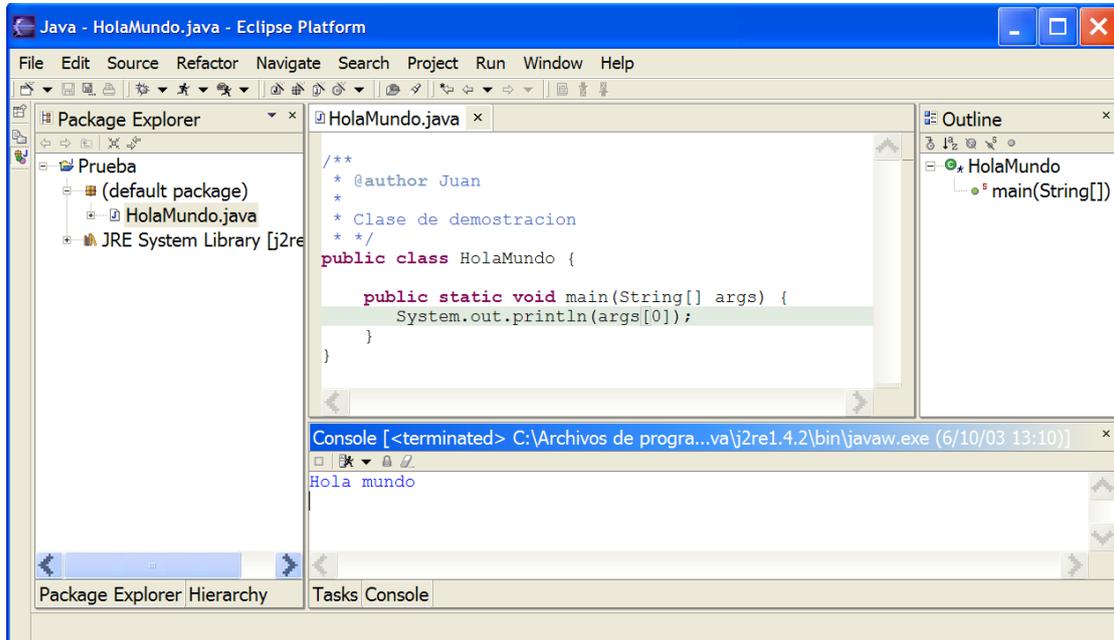


Figura 9: Consola con el resultado de la ejecución.

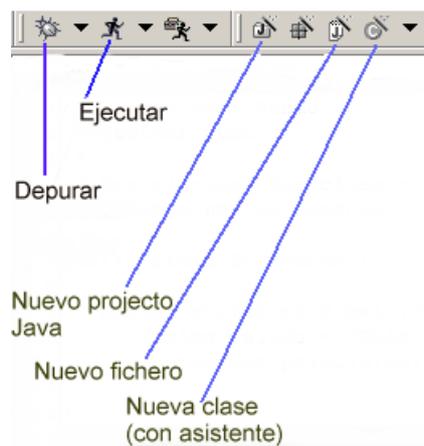


Figura 10: Barra de herramientas.

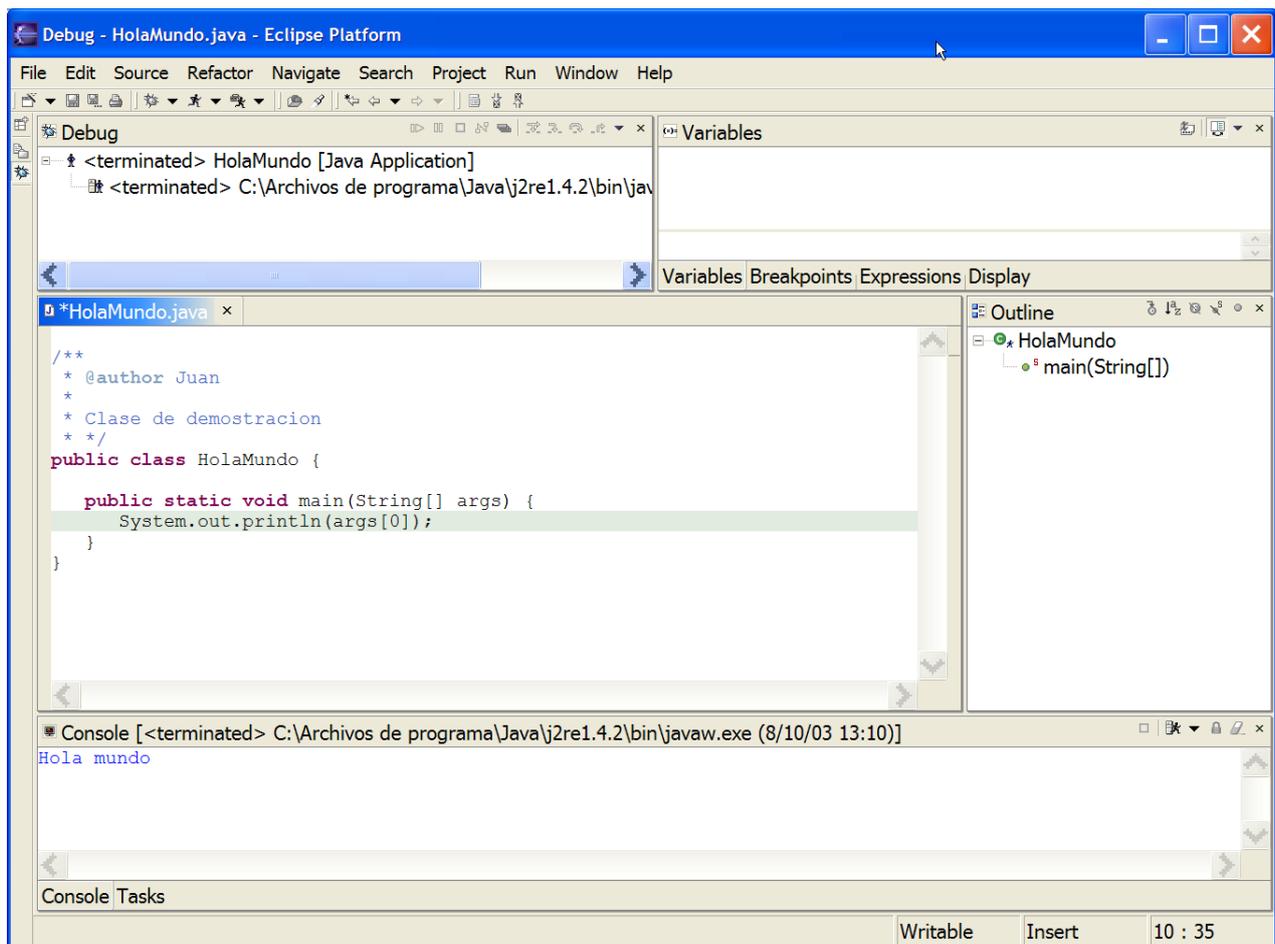


Figura 11: La perspectiva *Debug*.

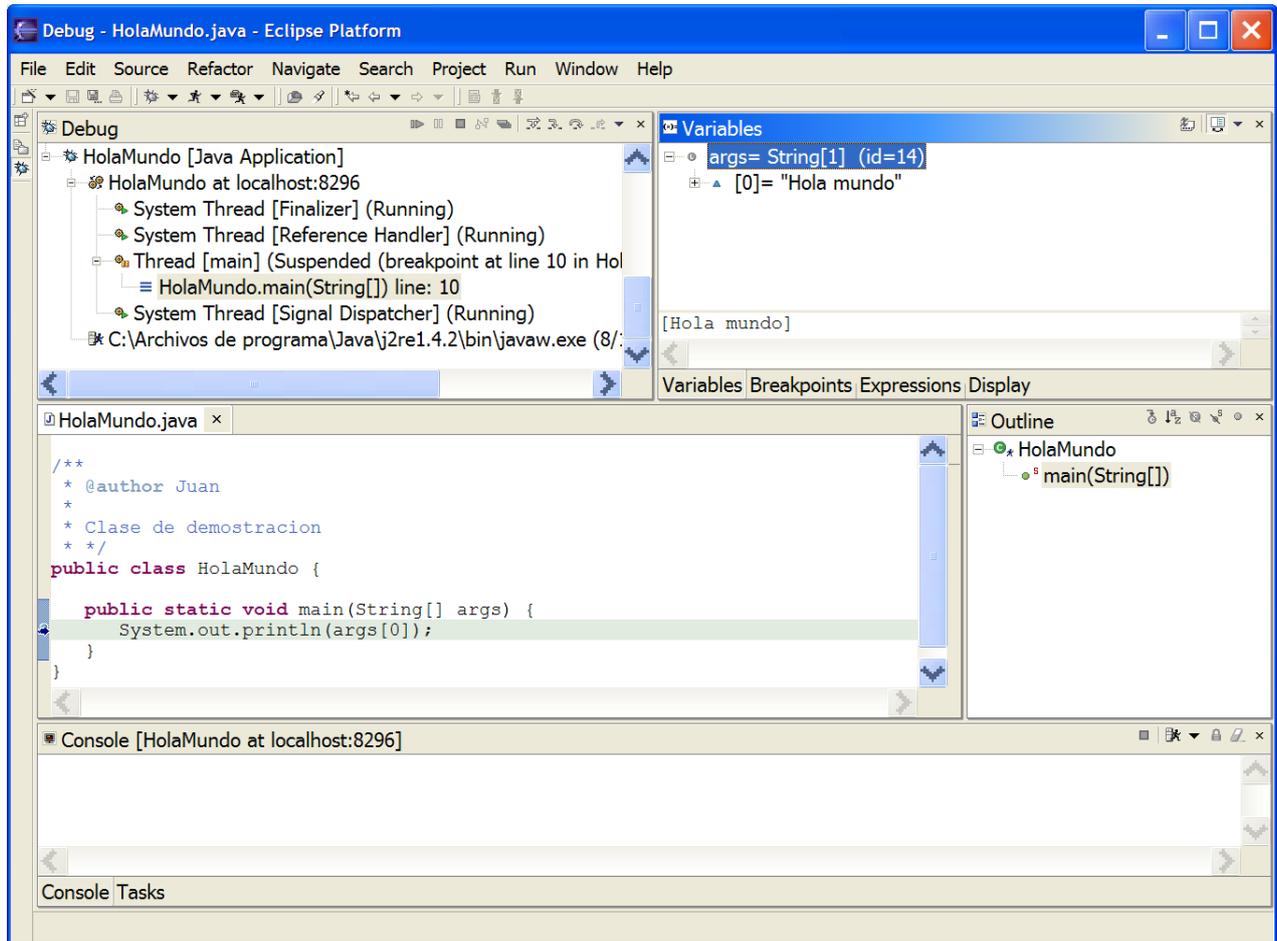


Figura 12: La perspectiva *Debug* en acción.

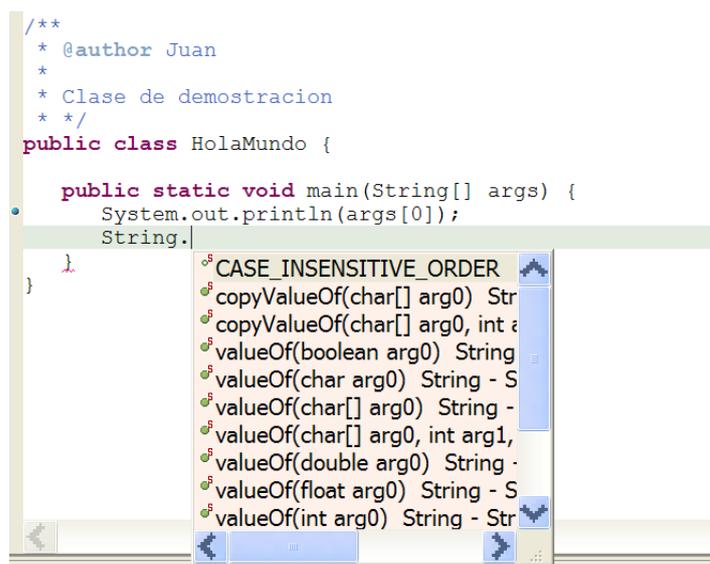


Figura 13: Asistente de código (pulsando Ctrl + espacio).