

# Programación II

## Seminario de introducción al entorno de desarrollo Eclipse

Curso 2010/2011

### Entorno Eclipse

Eclipse es una plataforma de desarrollo software genérica

- desarrollada por una comunidad “*open source*”
- que permite desarrollar entornos de desarrollo integrados para distintos lenguajes programación
- uno de los lenguajes soportados es el Java

Su instalación es muy sencilla:

1. visitar <http://www.eclipse.org> y pulsar “*Download Eclipse*”
  - versión en los laboratorios: “*Galileo SR1 Packages (v 3.5.1)*”
  - si se desea instalar esa versión pulsar en “*Older Versions*”
2. descargar “*Eclipse IDE for Java Developers*”
3. descomprimir en la carpeta donde queramos instalar Eclipse
4. ejecutar el programa “*eclipse*” que hay dentro de esa carpeta

### Uso del entorno de desarrollo Eclipse

Para programar utilizando Eclipse debemos crear un proyecto por programa (al igual que ocurría en Bluej).

El proyecto es un directorio donde se guarda toda la información del programa: código fuente, clases compiladas, documentación, etc.

## Creación del proyecto

Elegir **File=> New => Project** . En la pantalla que nos aparece elegimos el tipo de proyecto: **Java Project**.

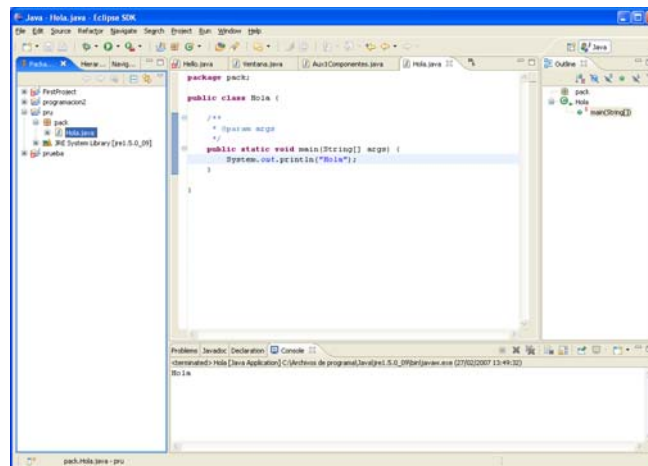
Eso nos llevará a la siguiente pantalla, en la que damos nombre al proyecto.

Las clases que añadamos al proyecto se pueden ordenar en paquetes. Para crear un paquete dentro de un proyecto:

- Botón derecho en la carpeta `src` (*sources*) del proyecto y elegimos **New => Package**.

Todos los proyectos Eclipse se almacenan en un directorio denominado *workspace*. Podemos cambiar la ubicación de dicho directorio haciendo **File => Switch Workspace**, asignándole la ubicación que nos interese.

## Entorno de trabajo del proyecto



## Creación de una clase

Para añadir una nueva clase al proyecto pinchamos con el botón derecho en el proyecto (o en cualquiera de los paquetes del proyecto) y elegimos **New => Class**. En la pantalla que aparece:

- darle un nombre a la clase
- elegir las opciones deseadas: visibilidad, clase de la que hereda, creación automática de constructores, etc.

Entre las opciones aparece la posibilidad de que la clase tenga un método `main`.

A continuación, escribimos el código de nuestra clase.

## Compilar y ejecutar el programa

Eclipse realiza la compilación del programa de forma automática:

- Los errores de compilación van apareciendo a la izquierda de la línea en la que se encuentra el error, o como un subrayado rojo en el punto del error. Situándonos encima, podemos ver una descripción del mismo.

Para ejecutar el proyecto:

- Botón derecho sobre la clase que tenga el método main y elegimos **Run As => Java Application**.

## Creación de archivos .jar

Para importar ficheros con código java a un paquete:

- Botón derecho sobre el paquete y elegimos **Import**.
- Elegimos **General => File System**
- Seleccionamos el directorio donde están los ficheros (**Browse**)
- Elegimos los ficheros que queremos importar

Para generar el archivo .jar de un paquete que contiene un conjunto de clases:

- Botón derecho sobre el paquete (o sobre el proyecto completo) y elegimos **Export**.
- En la pantalla que aparece elegimos **Java => JAR file**.
- En la nueva pantalla elegimos las clases que queremos añadir al archivo comprimido, le damos nombre y ubicación y pulsamos **Finish**.

## Añadir paquetes externos al proyecto

Los paquetes externos se añaden como archivos .jar.

Si queremos añadir a nuestro proyecto el paquete fundamentos:

- Generar un archivo .jar correspondiente al paquete
- Botón derecho sobre el proyecto y elegimos **Build Path => Configure Build Path**
- En la pestaña **Libraries** elegimos **Add External JARs**
- Buscamos el archivo .jar elegido, pinchamos sobre él y pulsamos **Abrir**.
- Pulsar **OK**

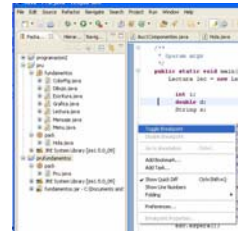
Acordarse de importar las clases a usar en el código de nuestro programa.

## Depuración con Eclipse

(1/3)

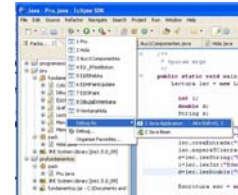
Antes de comenzar la depuración poner al menos un punto de ruptura

- Pulsar con el botón derecho en el borde de la ventana de edición y seleccionar **Toggle Breakpoint**



Comenzar la depuración desplegando el menú de depuración

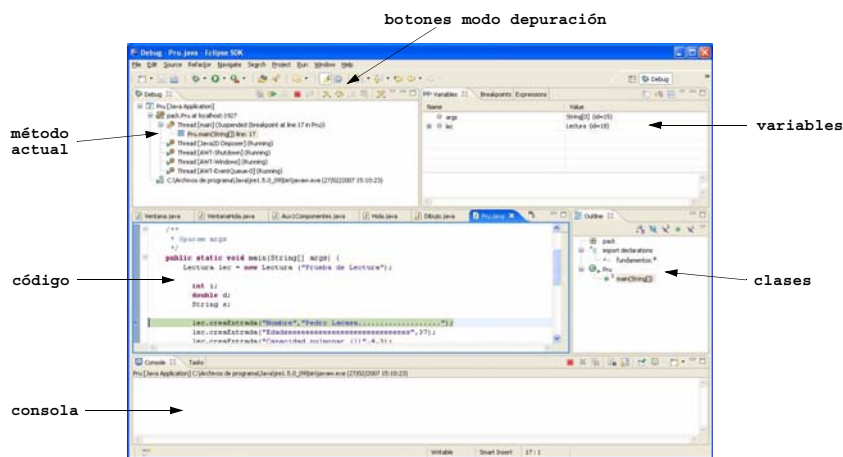
- Elegir **Debug As => Java Application**



## Depuración con Eclipse

(2/3)

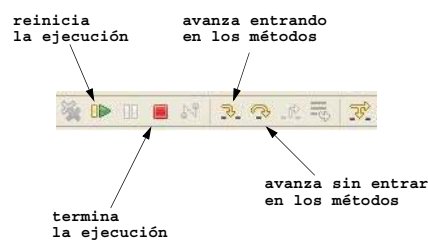
Durante la depuración cambia el entorno de trabajo:



## Depuración con Eclipse

(3/3)

El proceso de depuración se controla con los siguientes botones:



Una vez terminada la depuración volver a la perspectiva de edición

- pulsando el botón "Java" en la esquina superior derecha de la ventana