



Escuela de Ingeniería Informática



UNIVERSIDAD DE OVIEDO

INTRODUCCIÓN A ECLIPSE

Metodología de la programación
Curso 2020-2021

¿Qué es Eclipse?

- [Eclipse](#) es una comunidad de código abierto que se centra en la construcción de una *plataforma compuesta por marcos extensibles (Frameworks)* y herramientas para la construcción, despliegue y gestión del software.
- La [Fundación Eclipse](#), es una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios y servicios.
- [Proyectos eclipse](#), existen multitud de proyectos:
 - El IDE de java (JDK), aplicaciones de modelado, software para dispositivos, herramientas para la generación de informes, etc

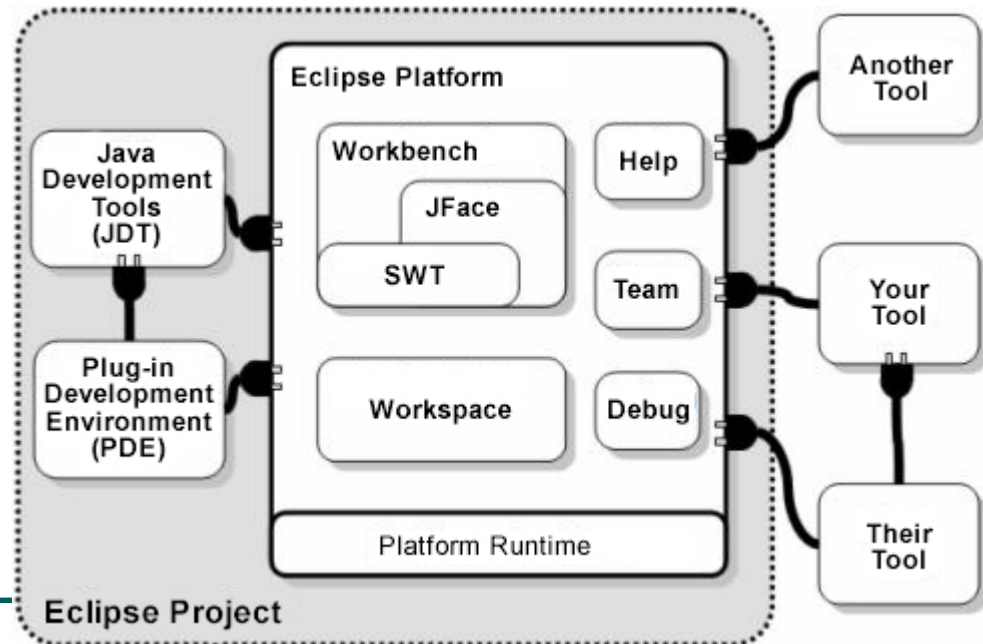
Arquitectura

- Esta basada sobre el concepto de **plug-in**
 - Código que extiende la funcionalidad del IDE.
 - Existen plug-ins para GUIs, pruebas, modelado ...

La **plataforma** está **desarrollada en java**

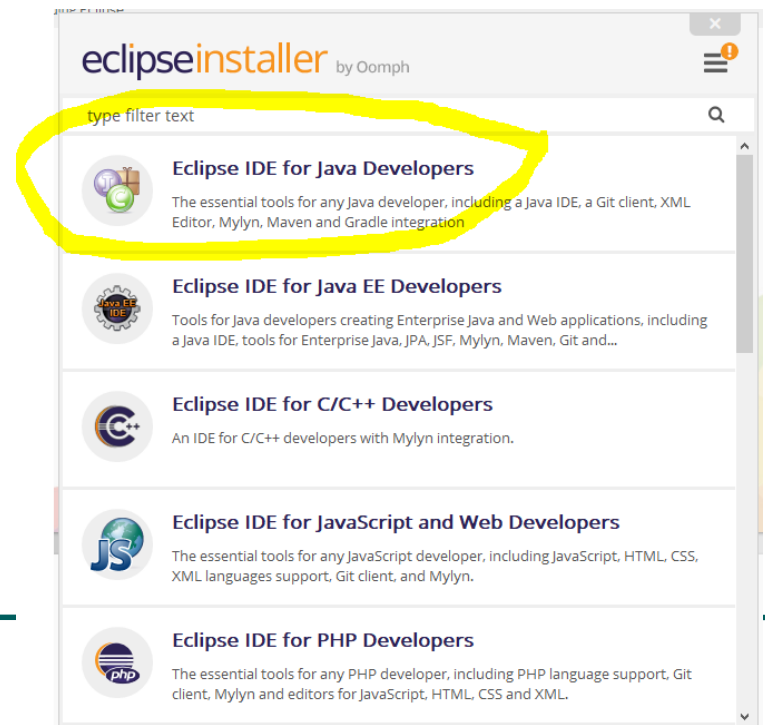
Soporta diferentes lenguajes:

Java, C/C++, Cobol, PHP, AspectJ, JavaScript ...



Distribución

- Descarga gratuita en la dirección:
<https://www.eclipse.org/downloads/packages/>
Versiones para Windows, Linux y Mac
- OS X.
- Cada versión (release) se libera anualmente y se identifica con un nombre.
 - **2020-12**
 - **2019-12**
 - Photon año 2018
 - Oxigen año 2017
 - Neon año 2016
 - Mars año 2015
- La instalación consiste en ejecutar el Instalador y seleccionar la versión.



Entorno de trabajo- Workbench

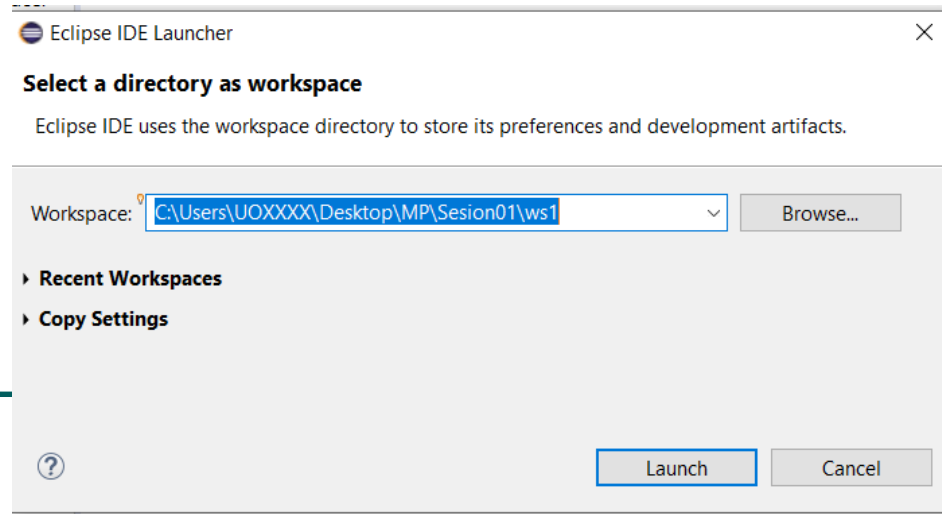
- El IDE (Integrated Development Environment) incluye:
 - Explorador de ficheros.
 - Editor.
 - Compilador.
 - Depurador.
- También incluye herramientas de ayuda al programador:
 - Refactorización (Refactoring)
 - Generación de código (Code generation)
 - Pruebas (Testing)

Workspace (Espacio de trabajo)

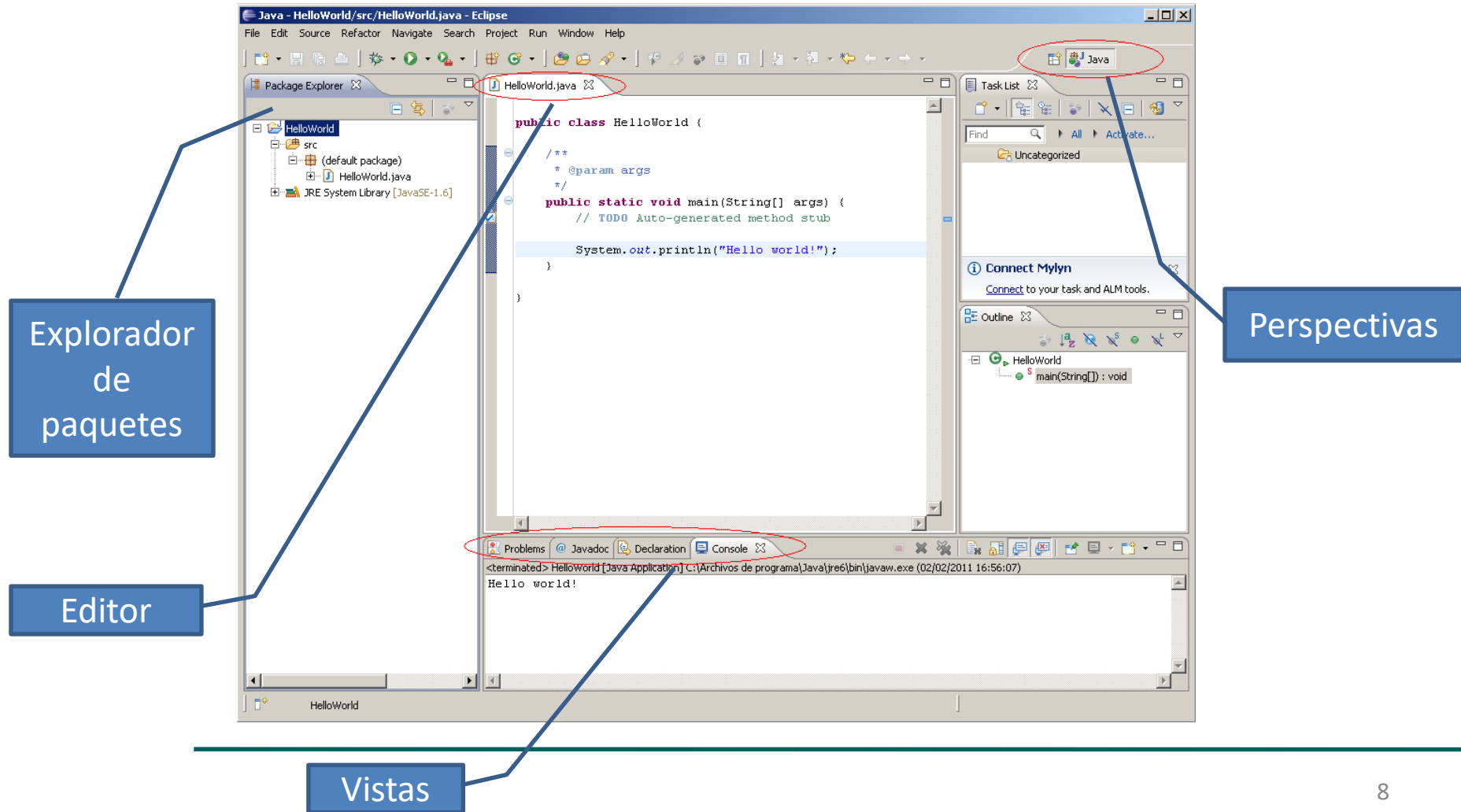
- Un **workspace** es una carpeta donde se almacena un conjunto de proyectos
- Guarda el estado de cada proyecto .
- Se pueden mantener tantos espacios de trabajo como se quiera.
- **Usaremos un workspace para cada sesión de laboratorio**
- Llamaremos al espacio de trabajo ws1 para la sesión 1.
- Después de cada clase de laboratorio se deben copiar los proyectos del workspace en un “lapiz” USB o mejor subirlos a OneDrive, Dropbox...

Arrancar el entorno

- Crea la **carpeta sesion01** dentro de **carpeta MP** y guarda aquí el material de esta sesión bajado del campus.
- Crea el espacio de trabajo (carpeta ws01) dentro de sesion01
`c:\Users\uoXXXXXX\desktop\MP\sesion01\ws1`
- Ejecuta Eclipse.
 - eclipse.exe (icono eclipse del escritorio)
- Selecciona el espacio de trabajo.

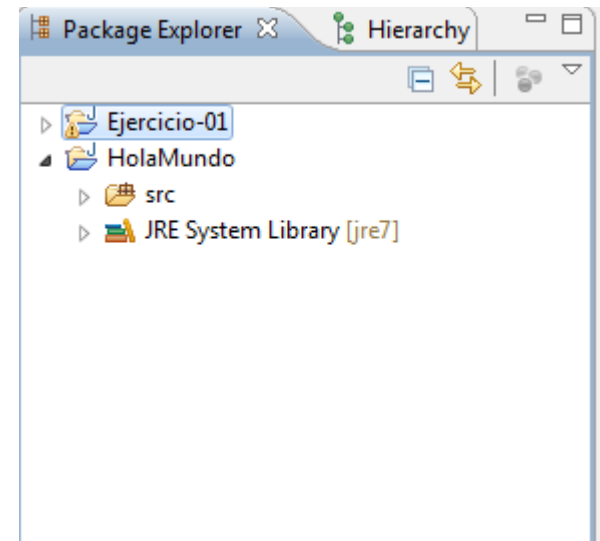


Entorno



Package Explorer

- Muestra el contenido del workspace (espacio de trabajo) actual.
 - Contiene un conjunto de proyectos
 - Cada proyecto contiene diversos packages y librerías.
 - Los Packages contienen tipos
- Está sincronizado automáticamente con el sistema de ficheros.
- **Presionar F5** para actualizar la sincronización.

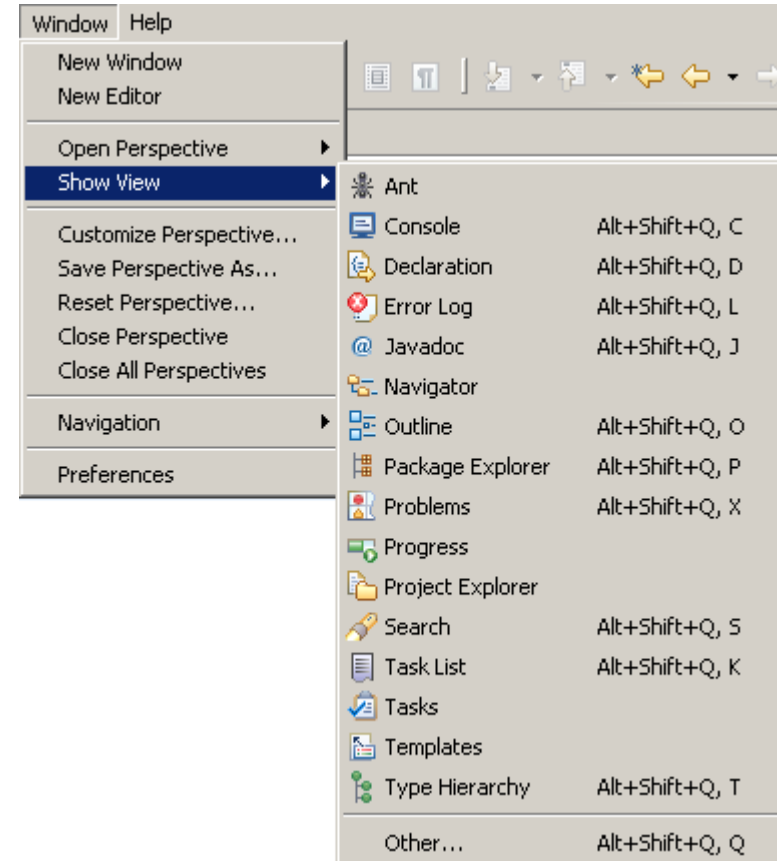


Paquete

- Un **paquete** puede ser definido como un **contenedor de tipos relacionados** (clases, interfaces, enumeraciones y anotaciones).
- Los componentes del mismo paquete están relacionados de la misma forma; por ejemplo, están enfocados a una función común (interfaz con el usuario, lógica de negocio, interfaz con bases de datos....)
- Los paquetes también proporcionan protección de acceso y gestión del espacio de nombres.

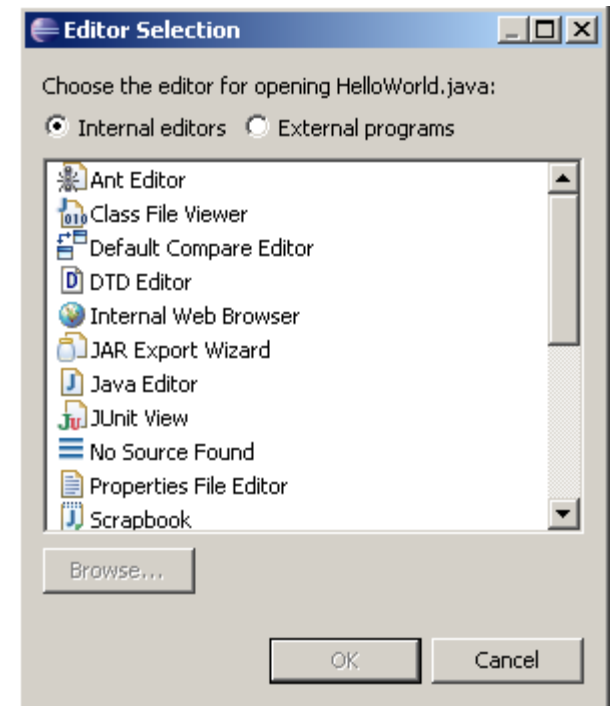
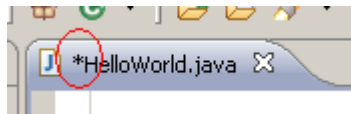
Vistas

- Es una ventana que muestra algo concreto del proyecto.
 - Consola de Java.
 - Errores de compilación.
 - etc...
- Se puede abrir cualquier vista desde el menú.
- Aparecen agrupadas y solo se puede ver una.
- No permite modificar.



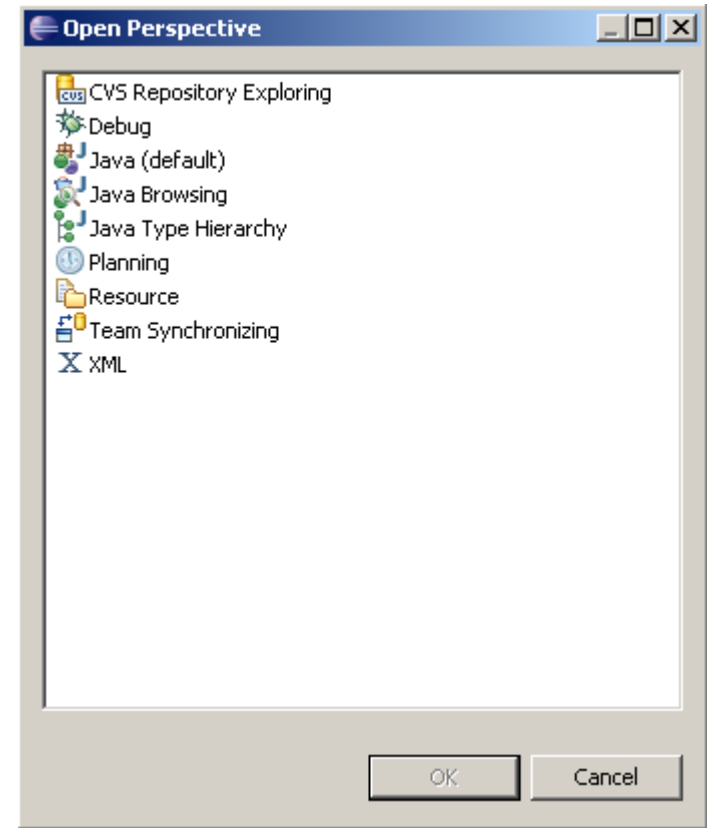
Editores

- Similares a las vistas.
- Múltiples editores, gráficos o de texto.
- Asociados por la extensión.
 - Configurable desde las preferencias.
- Es posible forzar el uso del editor que queramos.
- Un “*” al lado del nombre del archivo => sin salvar.



Perspectivas

- Adaptan el entorno para una tarea de alto nivel.
- Seleccionan un conjunto de vistas, editores y barras de herramientas adecuados a la tarea.
 - Java: Desarrollo en java.
 - Debug: Depuración.
 - CVS: Control de versiones.
 - etc...
- En cualquier momento se puede cambiar la perspectiva.



COMO SE CREA UN NUEVO PROYECTO

Nomenclatura para proyectos

- **Nombre de los proyectos**

nombre_apellido1_apellido2_session01_game

- Los proyectos creados en la sesión 1 llevarán *sesion1*
- Siempre en minúscula y sin tildes
- En inglés

- **Nombre de los proyectos a entregar**

nombre_apellido1_apellido2_session01_task_game

- **Nombre del paquete**

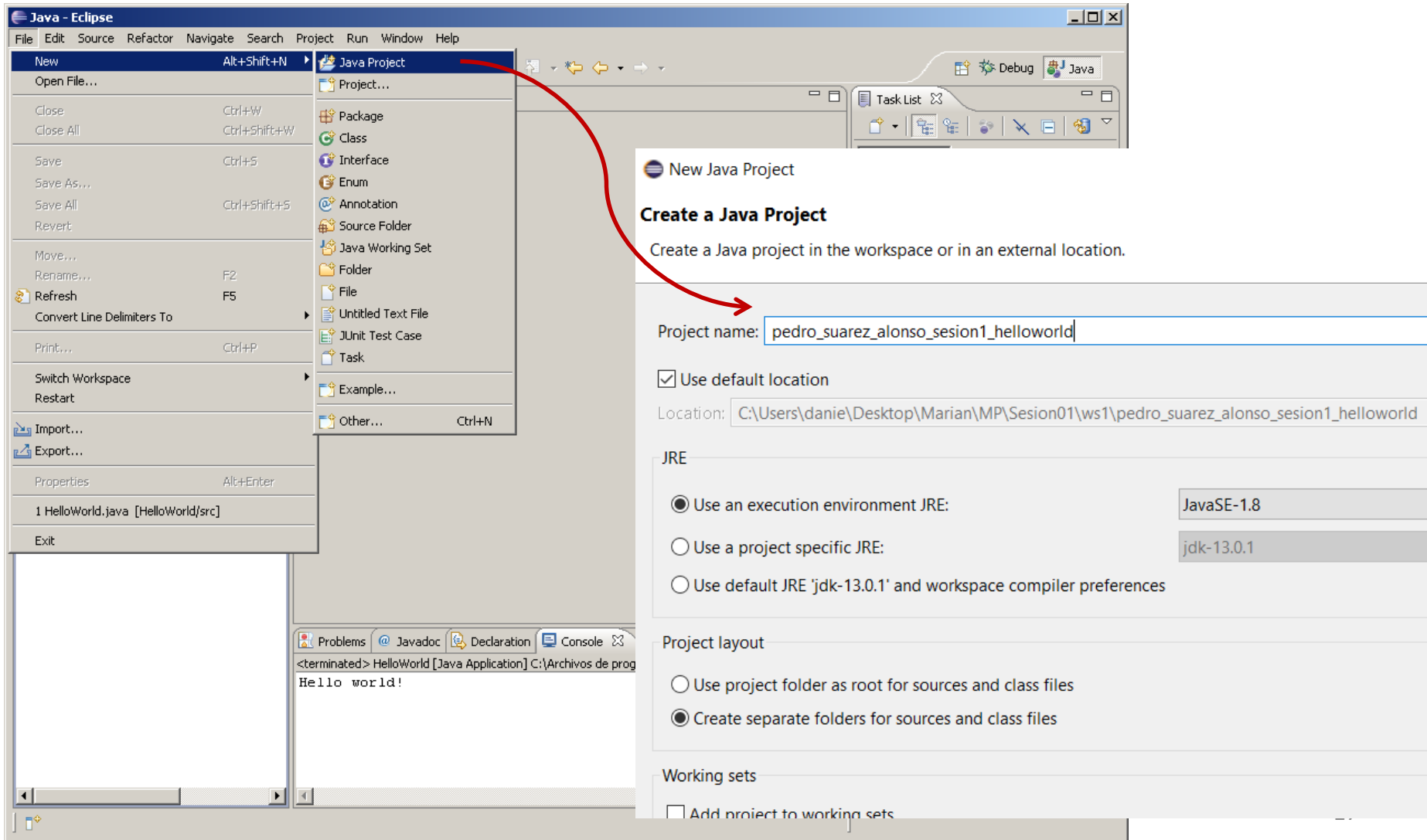
uo.mp.s1.game

- Dentro de una carpeta (src o test)
- Las clases se distribuyen en paquetes si son varias. Su último nombre depende del contenido.
- Se suelen incluir paquetes como ***model*** , ***service***, ***etc.***
- Ejemplo: *uo.mp.s1.game.model*

Ejercicio

- Crear un programa llamado helloworld que al ejecutarlo muestre por pantalla “Hola Mundo”
 - Se crea un proyecto:
 - nombre_apellido1_apellido2_sesion1_helloworld
 - Se crea un paquete en la carpeta src
 - uo.mp.s1.helloworld
 - Se crea una **clase con el método estático Main**
 - HelloWorld

Crear un nuevo proyecto



Crear un nuevo proyecto

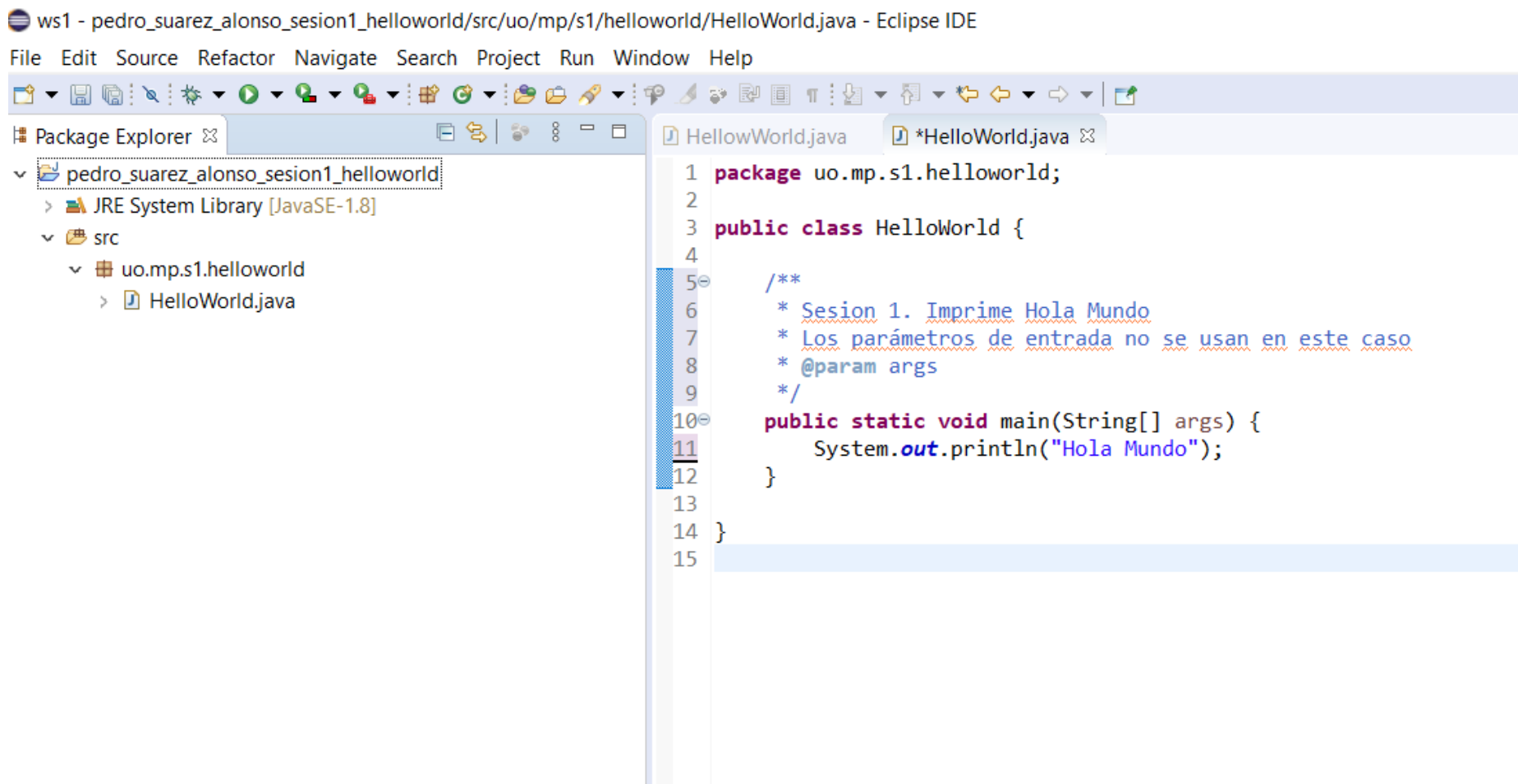
The screenshot shows the Eclipse IDE interface. On the left, the 'File' menu is open, with the 'New' option selected. A red arrow points from the 'Class' option in the 'New' submenu to the 'New Java Class' dialog box on the right.

New Java Class Dialog:

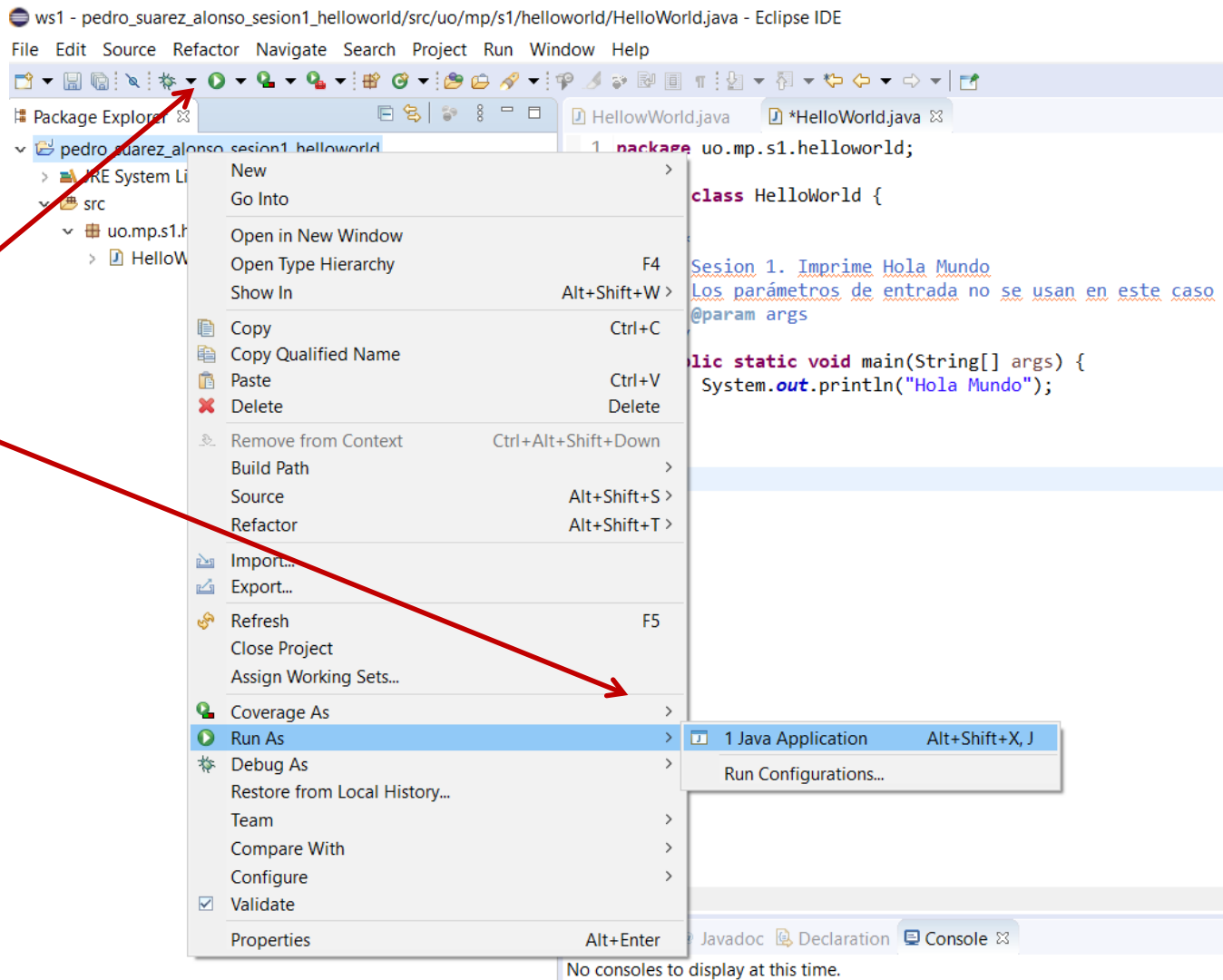
- Source folder:** pedro_suarez_alonso_sesion1_helloworld/src
- Package:** uo.mp.s1.helloworld (circled in red)
- Enclosing type:** (empty)
- Name:** HelloWorld (circled in red)
- Modifiers:** ☒ public, ☐ package, ☐ private, ☐ protected, ☐ abstract, ☐ final, ☐ static
- Superclass:** java.lang.Object
- Interfaces:** (empty)
- Which method stubs would you like to create?:** ☒ public static void main(String[] args) (circled in red), ☐ Constructors from superclass, ☒ Inherited abstract methods
- Do you want to add comments? (Configure templates and default value [here](#))**: ☐ Generate comments

Buttons at the bottom: **Finish** and **Cancel**.

Editar para añadir el código



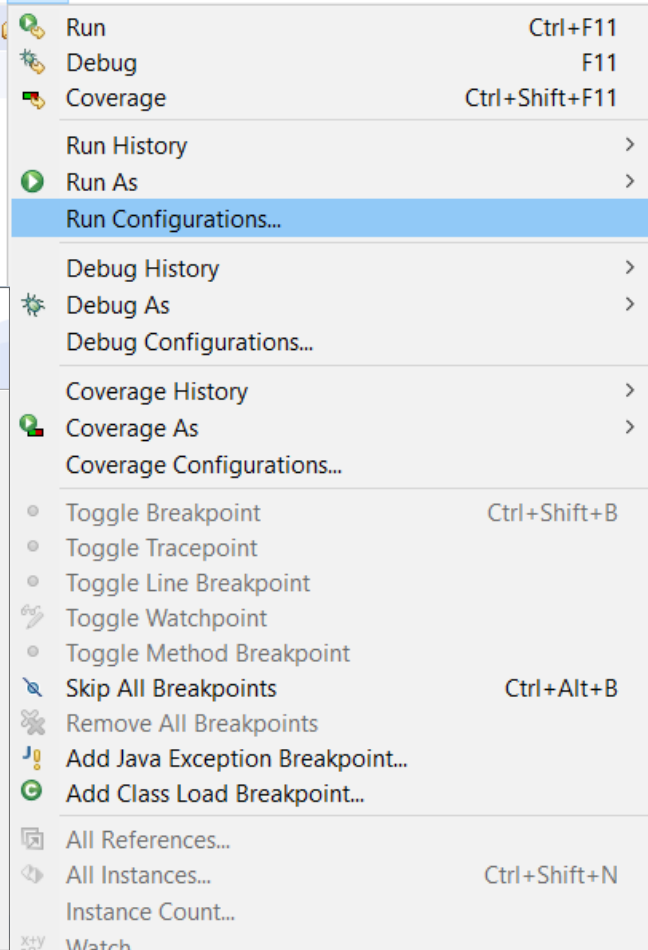
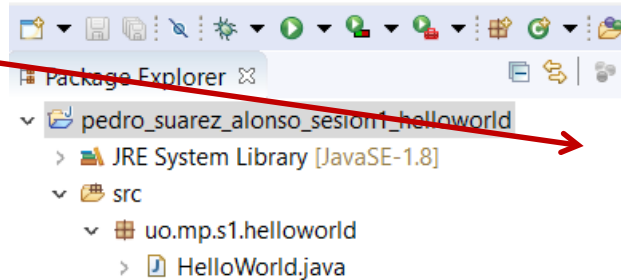
Ejecutar y ver el resultado



Utilizar parámetros

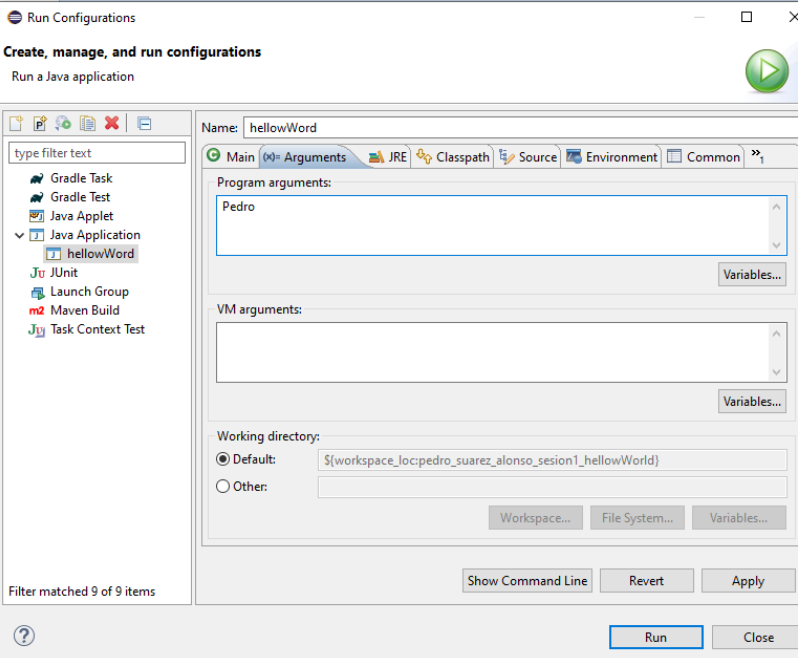
ws1 - pedro_suarez_alonso_sesion1_helloworld/src/uo/mp/s1/helloworld/HelloWorld.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



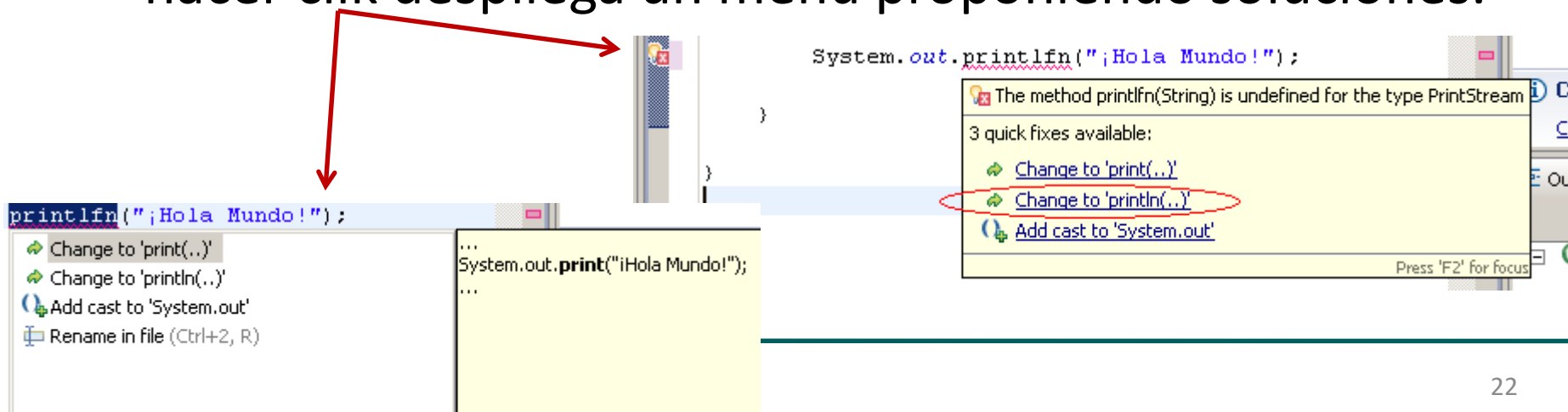
ndo
no se usan en este caso

```
ng[] args) {  
    " + args[0]);
```



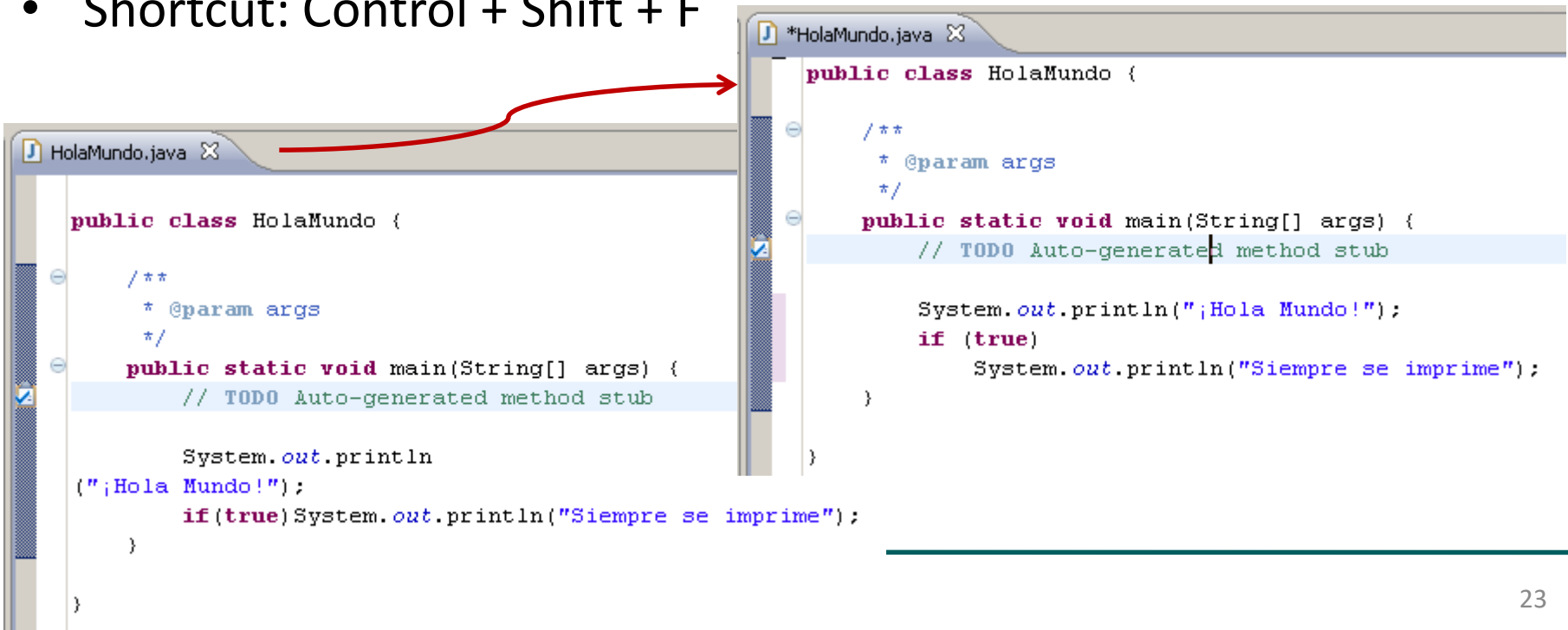
Detección y corrección de errores

- Similar a un corrector ortográfico.
- Subraya el error con una línea roja ondulada (amarilla si es un warning).
- Situando el ratón sobre la línea muestra una descripción del error y puede proponer soluciones.
- A la izquierda muestra un icono de error en el que al hacer clic despliega un menú proponiendo soluciones.



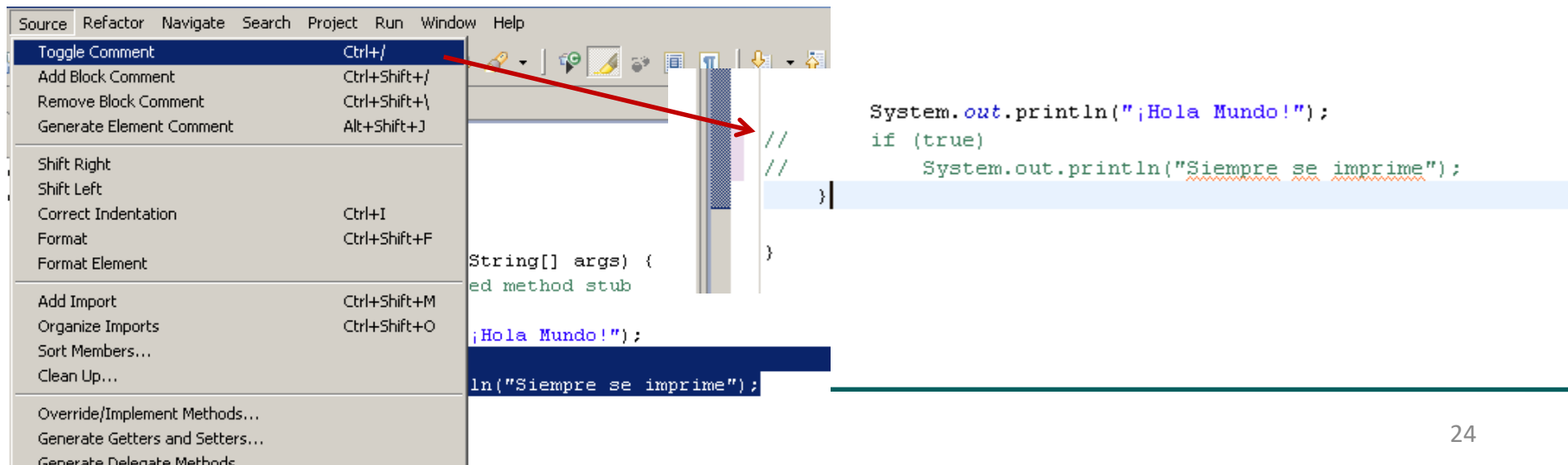
Manipulación del código fuente

- Opciones agrupadas en el menú “Source”
- Source → Format : Formatea e indenta el código fuente.
- Shortcut: Control + Shift + F



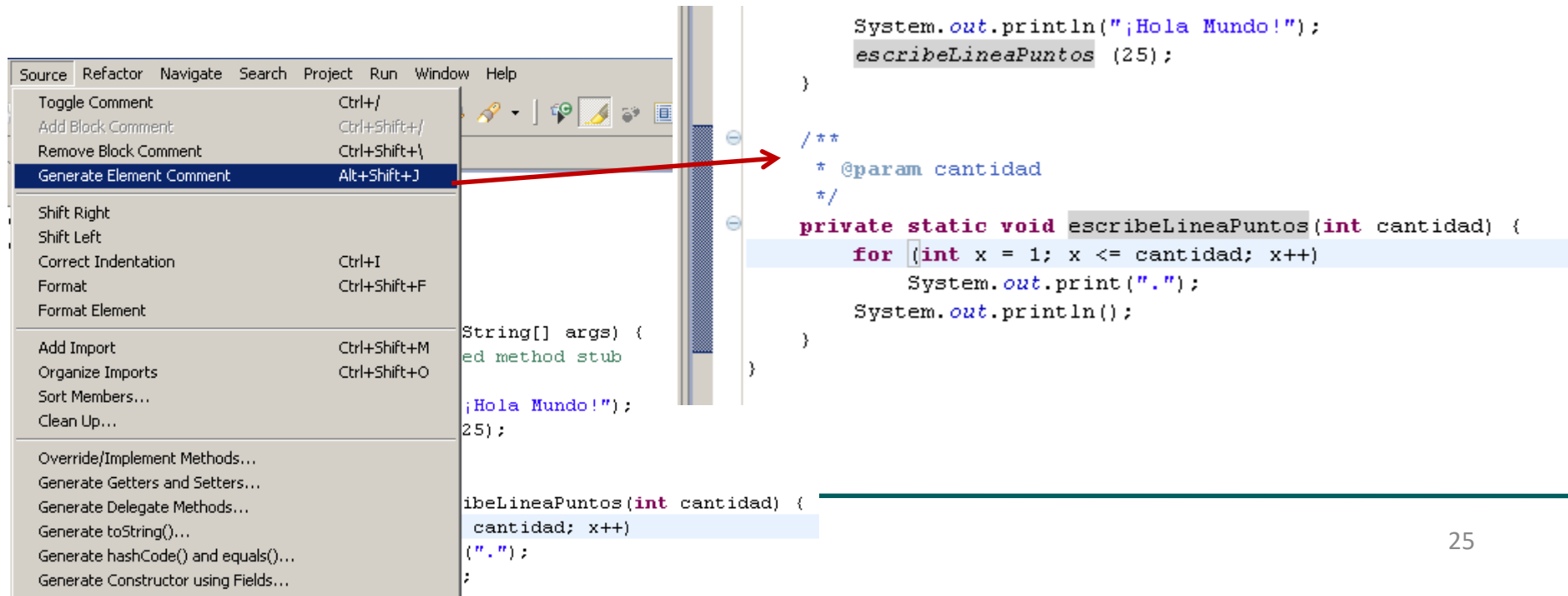
Código fuente: Comentar bloques

- Comentar y (des) comentar bloques de código previamente seleccionados.
- Source → Toggle Comment : Utiliza `//`
Shortcut: Control + /
- Source → Add Block Comment: Utiliza `/* ... */`



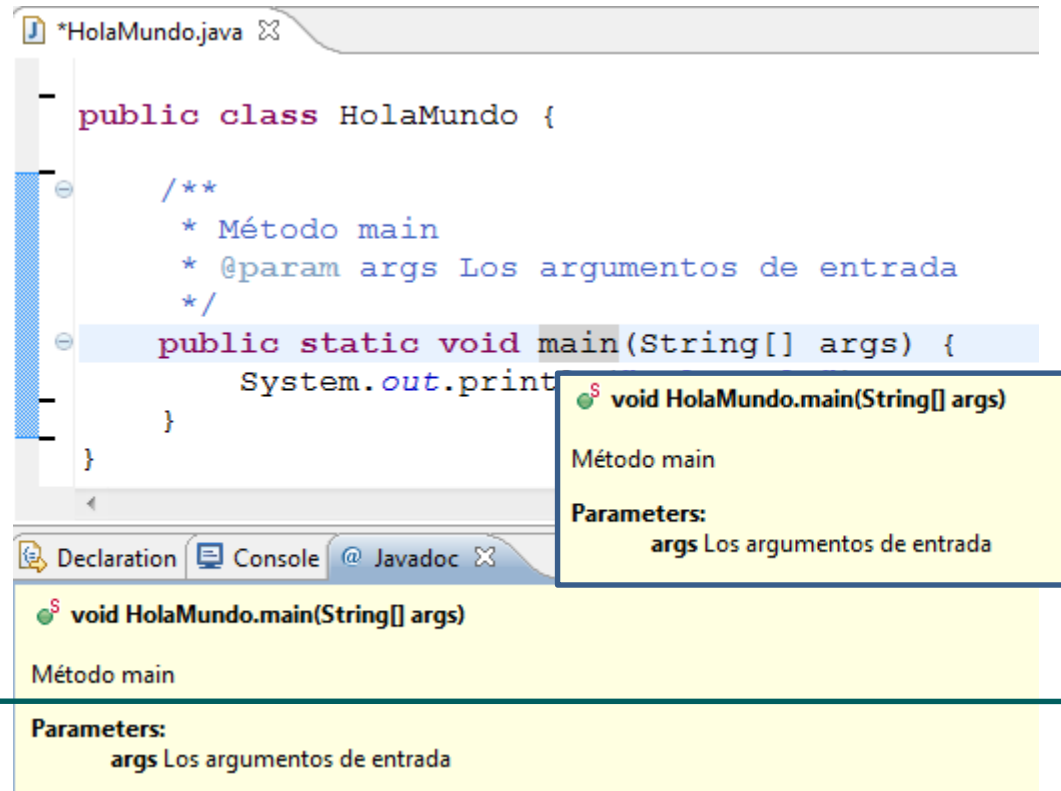
Código fuente: Comentarios Javadoc

- Añadir bloques apropiados para generar Javadoc a un elemento.
 - Source → Generate Element Comment
 - O escribiendo `/**` y presionando enter



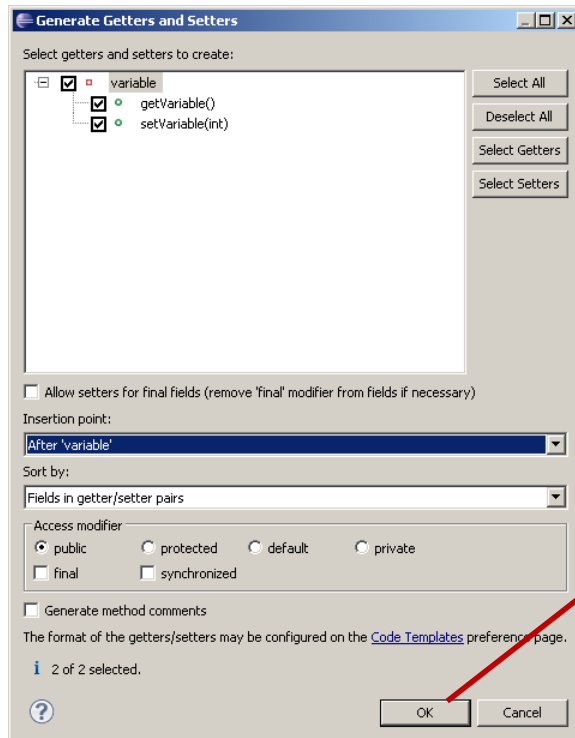
Comentarios Javadoc

- La información de Javadoc es usada:
 - En la **vista Javadoc** cuando se selecciona un elemento
 - En la **ventana emergente** cuando se coloca el ratón sobre un elemento.



Código fuente: Generar código

- Utiliza plantillas para añadir código. Ver `Source`
- Por ejemplo para generar métodos get y set:
 - `Source` → `Generate Getters and Setters ...`



```
private int variable;

public int getVariable() {
    return variable;
}

public void setVariable(int variable) {
    this.variable = variable;
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
}
```

Proyecto plane

- Crearemos un nuevo proyecto para cargar las clases Plane y Person desarrolladas en IP y usarlas.
 - Creamos nuevo proyecto nombre_apell1_apell2_sesion1_plane
 - Creamos paquete `uo.mp.s1.plane.model`
 - Copiamos en él las clases Plane y Person
 - Creamos paquete `uo.mp.s1.plane.ui`
 - Creamos en él la clase Main con el método estático main
 - Dentro del método main
 - Creamos un piloto
 - Creamos un avión pasando ese piloto, un identificador y combustible
 - Imprimimos por consola los datos del avión

Proyecto plane

ws1 - pedro_suarez_alonso_sesion1_plane/src/uo/mp/s1/plane/ui/Main.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- pedro_suarez_alonso_sesion1_helloworld
- pedro_suarez_alonso_sesion1_plane
 - JRE System Library [JavaSE-1.8]
 - src
 - uo.mp.s1.plane.ui
 - Main.java
 - uo.mp.s1.plane1.model
 - Person.java
 - Plane.java

```
1 package uo.mp.s1.plane.ui;
2
3 import uo.mp.s1.plane1.model.Person;
4 import uo.mp.s1.plane1.model.Plane;
5
6 public class Main {
7
8     public static void main(String[] args) {
9         Person pilot = new Person("Fernando Alonso",36);
10        Plane myPlane = new Plane(pilot,'Z',10000);
11        System.out.println("Datos del avión");
12        System.out.println(myPlane);
13    }
14 }
15
16 }
17
```

Problems Javadoc Declaration Console

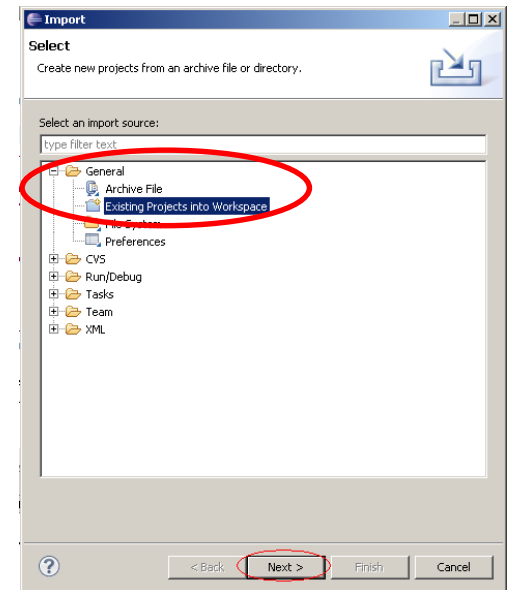
<terminated> Main [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (27 ene. 2020 23:32:01)

Datos del avión
Z 10000 Nombre: FERNANDO ALONSO Edad: 36

COMO SE ABRE UN PROYECTO YA CREADO

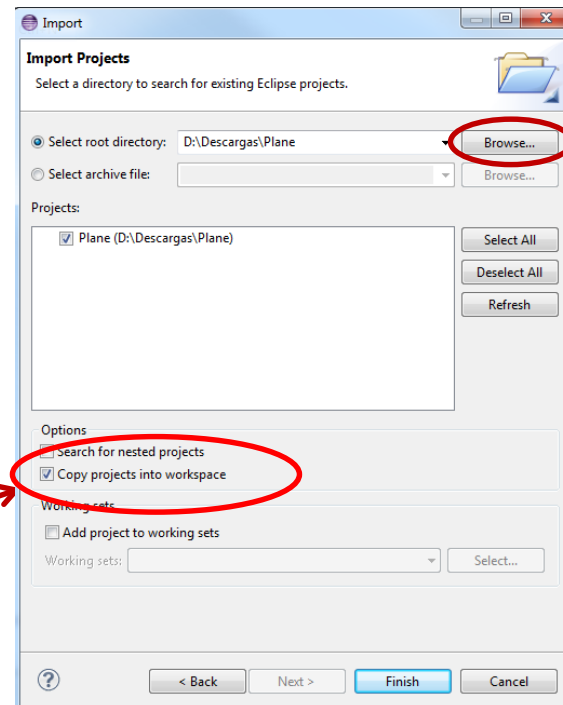
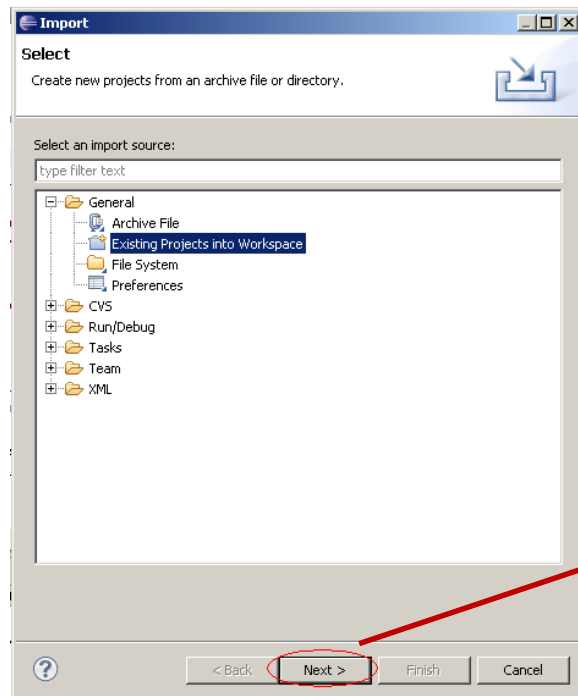
Importar un proyecto

- Menú File → Import
 - /General/Existing Projects into Workspace...
 - Dos posibilidades
 1. Seleccionar directorio
 - La carpeta con el proyecto
 2. Seleccionar fichero
 - El proyecto comprimido
- > Más cómodo usar segunda opción



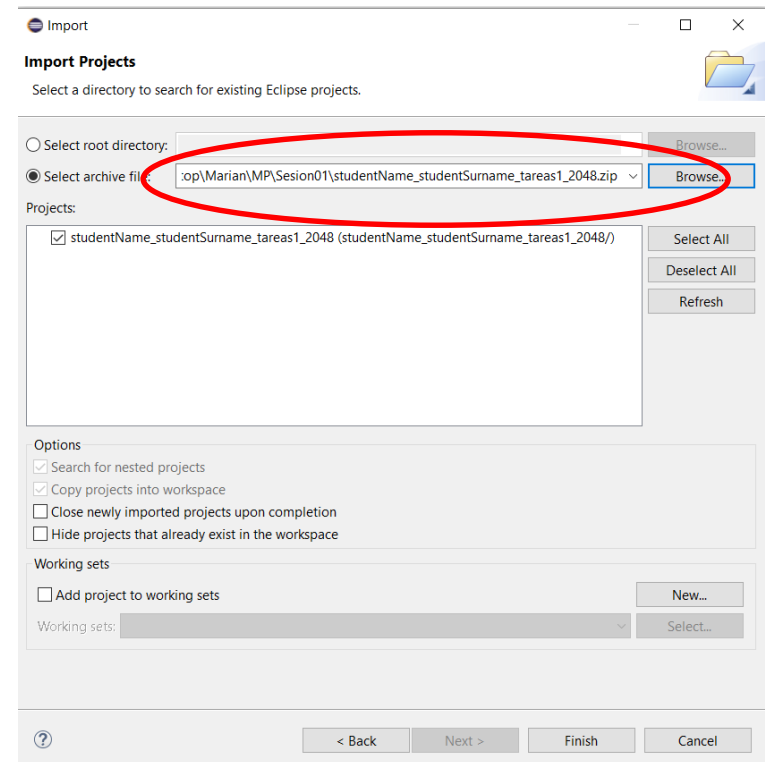
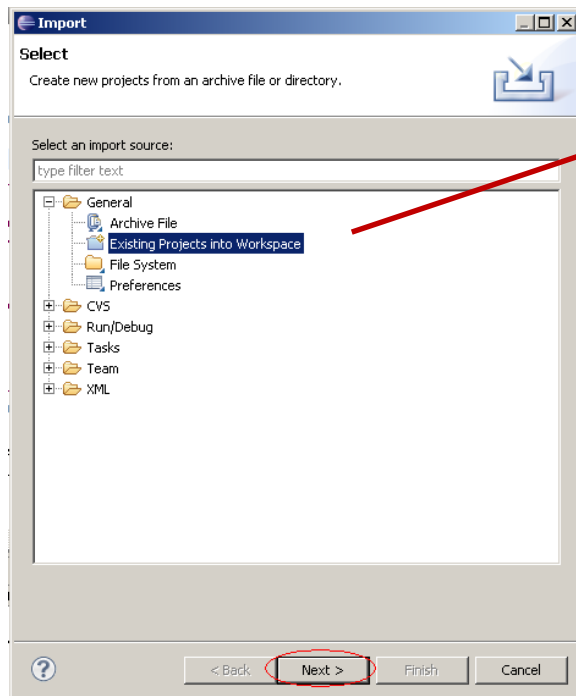
Importar un proyecto

- Si se importa proyecto sin comprimir es importante hacer una copia en el espacio de trabajo
- Si se importa proyecto comprimido ya la hace el sistema



Importar un proyecto

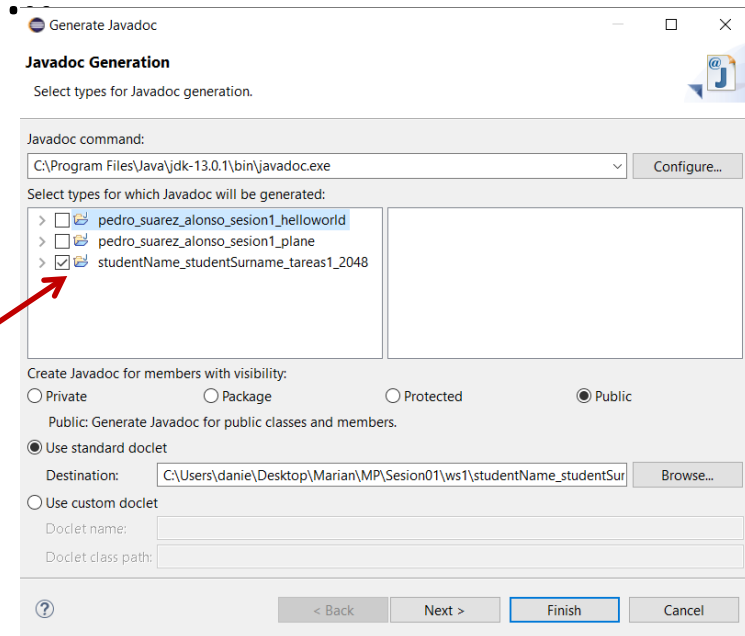
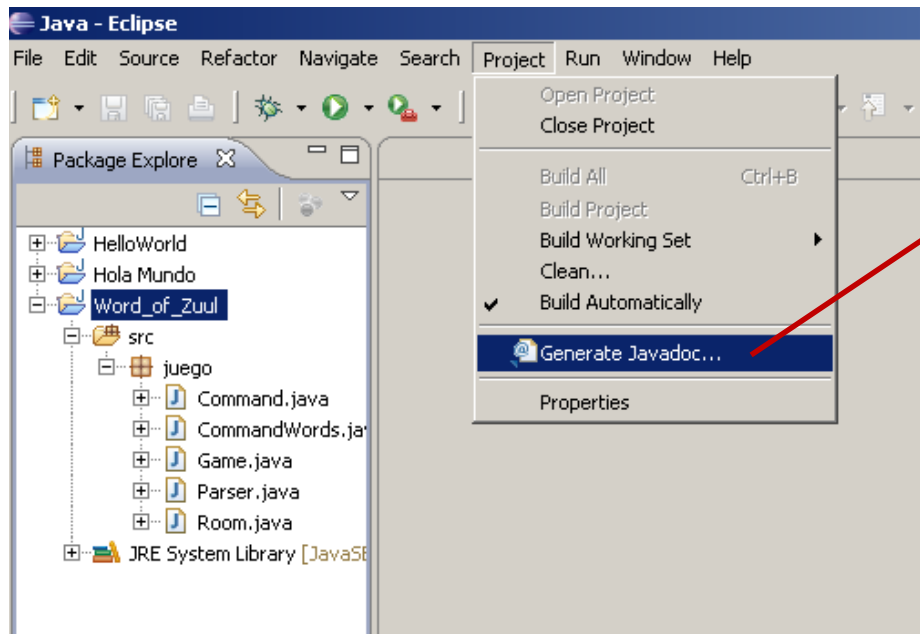
- Importaremos el proyecto para la tareas1 2048



COMO SE GENERA LA DOCUMENTACIÓN CON JavaDoc

Generar Javadoc

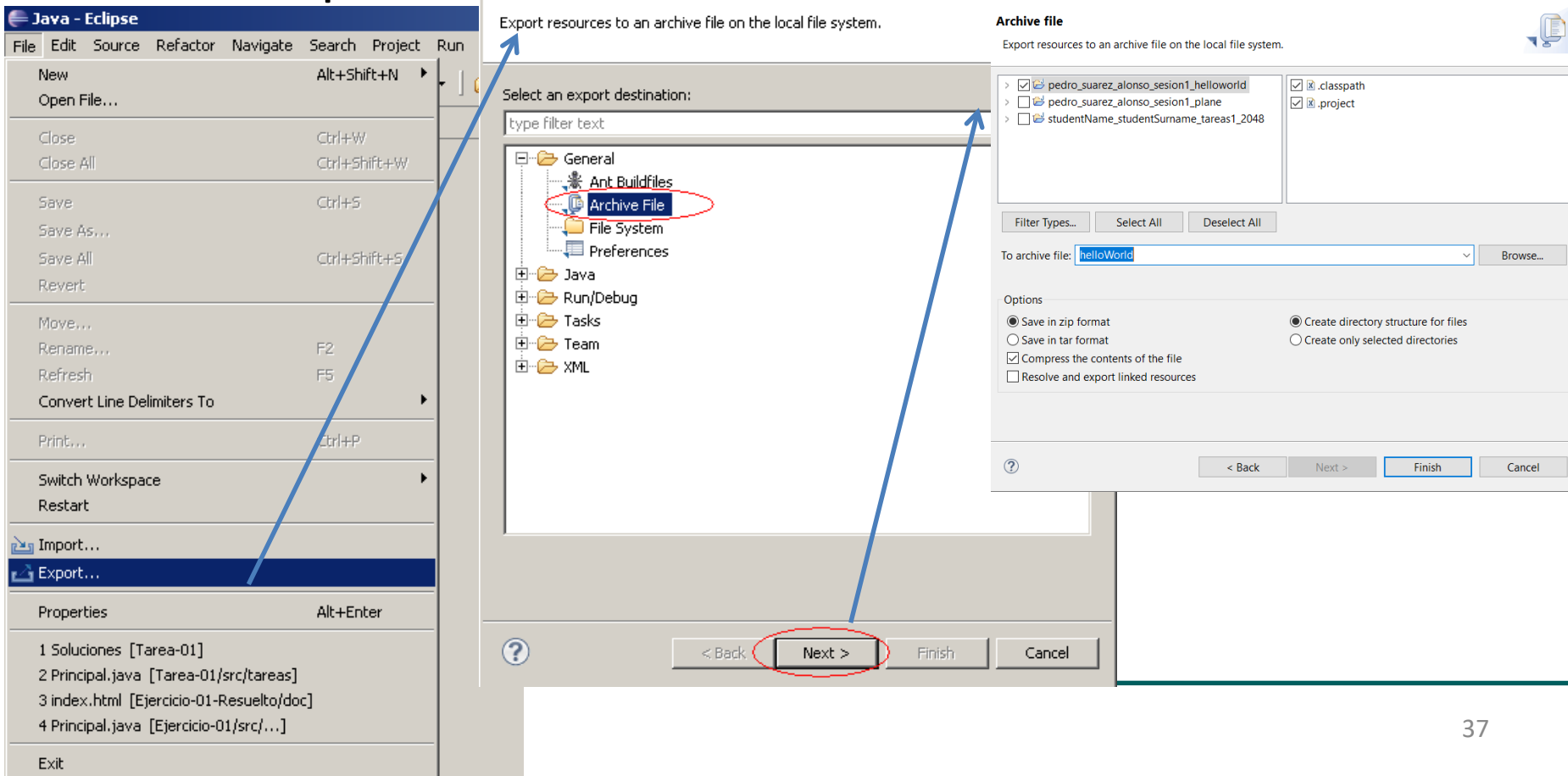
- Me permite generar el Javadoc a partir de los comentarios apropiados del código fuente.
 - Project → Generate Javadoc .



**COMO SE EXPORTA UN PROYECTO Y
SE CREA UN FICHERO COMPRIMIDO**

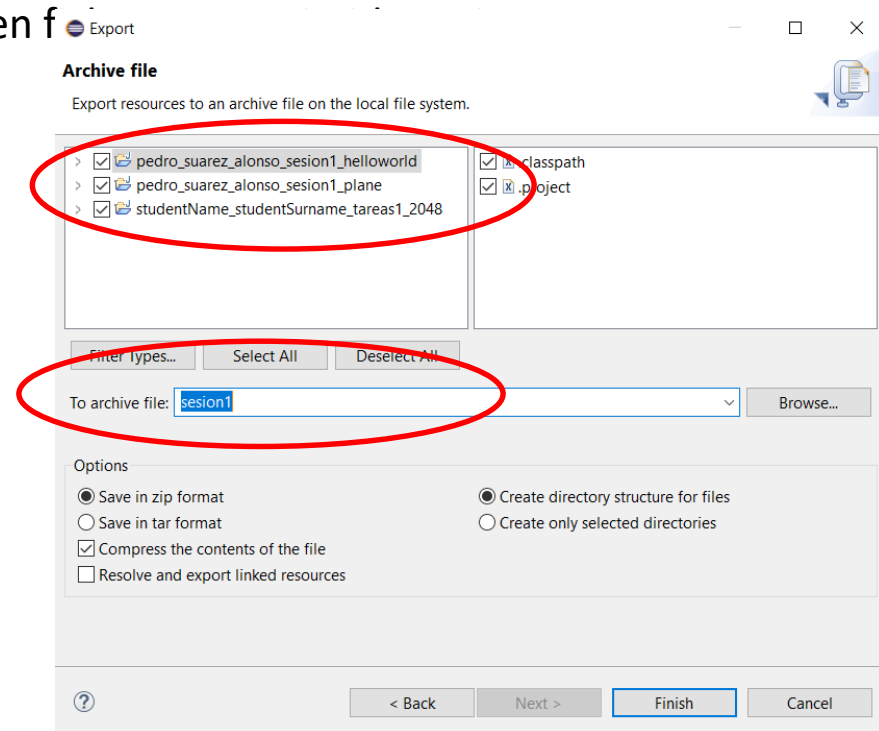
Exportar proyecto

- Permite copiar el proyecto a un archivo comprimido.
- File → Export



Exportar todos

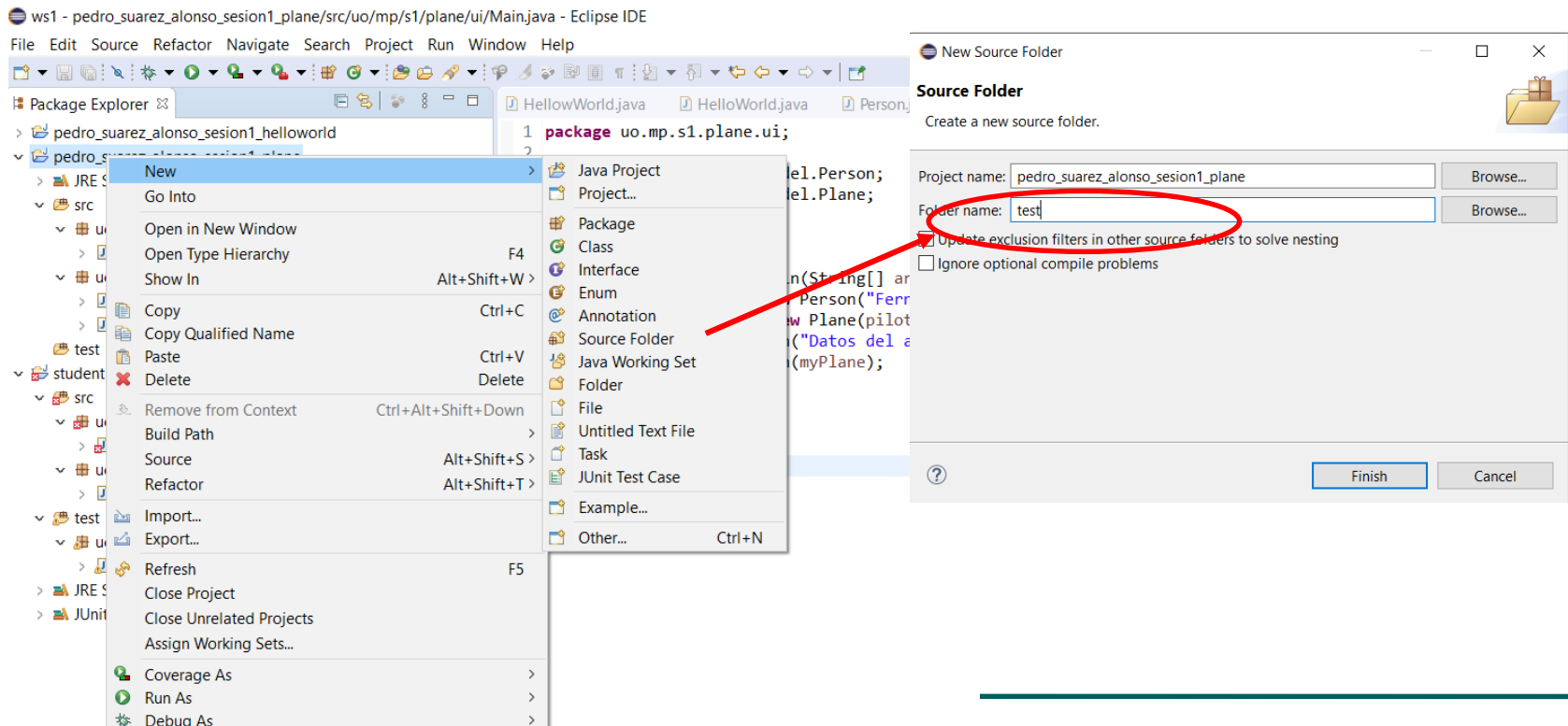
- Cuando se finalice la sesión
 - Se exportan TODOS los proyectos juntos en f
 - **Se guarda el comprimido en la nube**
- En casa
 - Se crea carpeta MP
 - Se crea carpeta sesion01 en MP
 - Se crea carpeta ws1 en sesion01
 - Se arranca eclipse y se le asigna sw1
 - Si no se hubiera creado sw1 lo crea eclipse al indicarle ese espacio
 - Se importan del fichero comprimido los proyectos



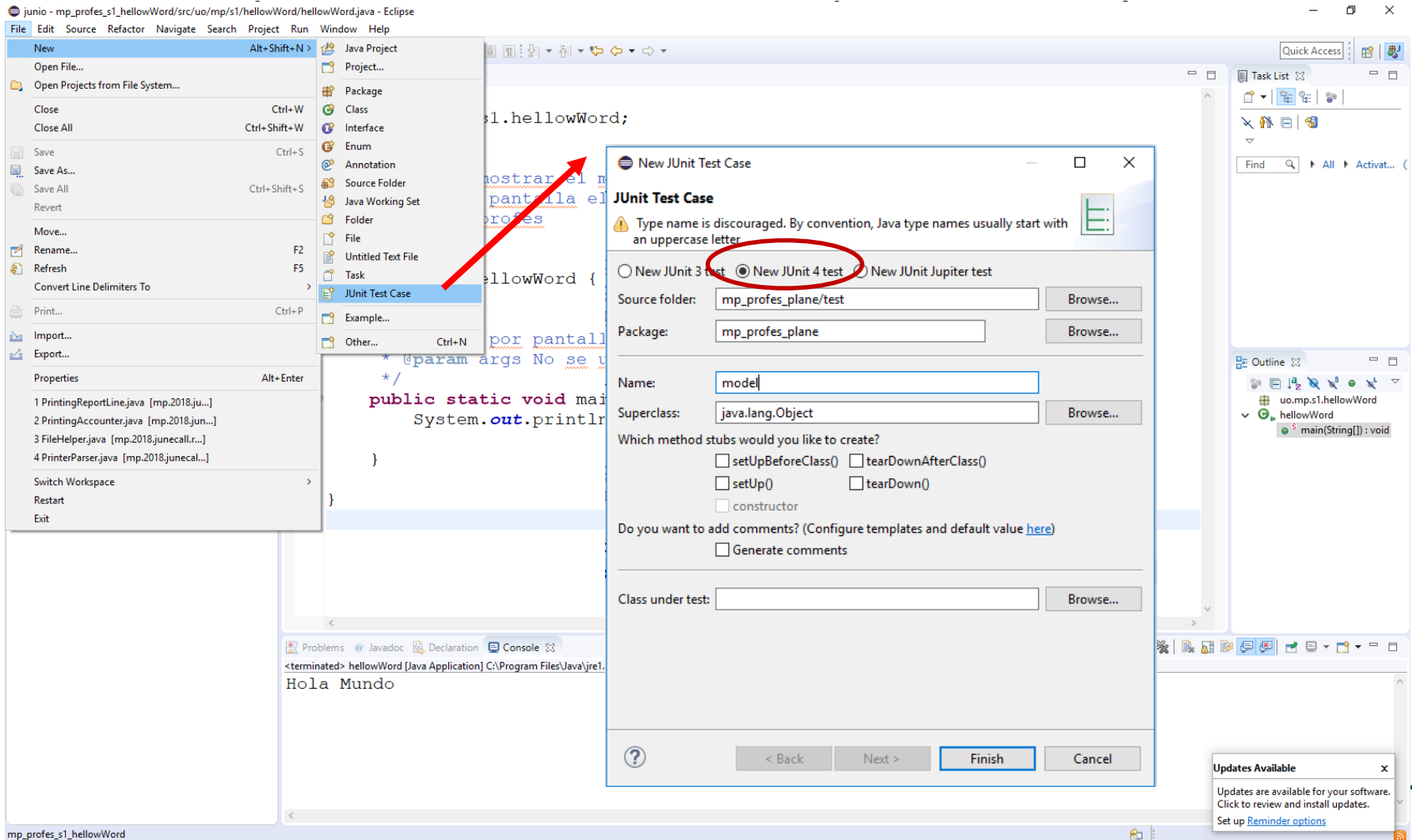
COMO SE CREAN PRUEBAS

Pruebas con JUnit

- Se debe crear una carpeta (source folder) llamada test
- Dentro de esta carpeta se crea un paquete con el mismo nombre que el paquete donde está la clase a probar más el nombre de la clase.



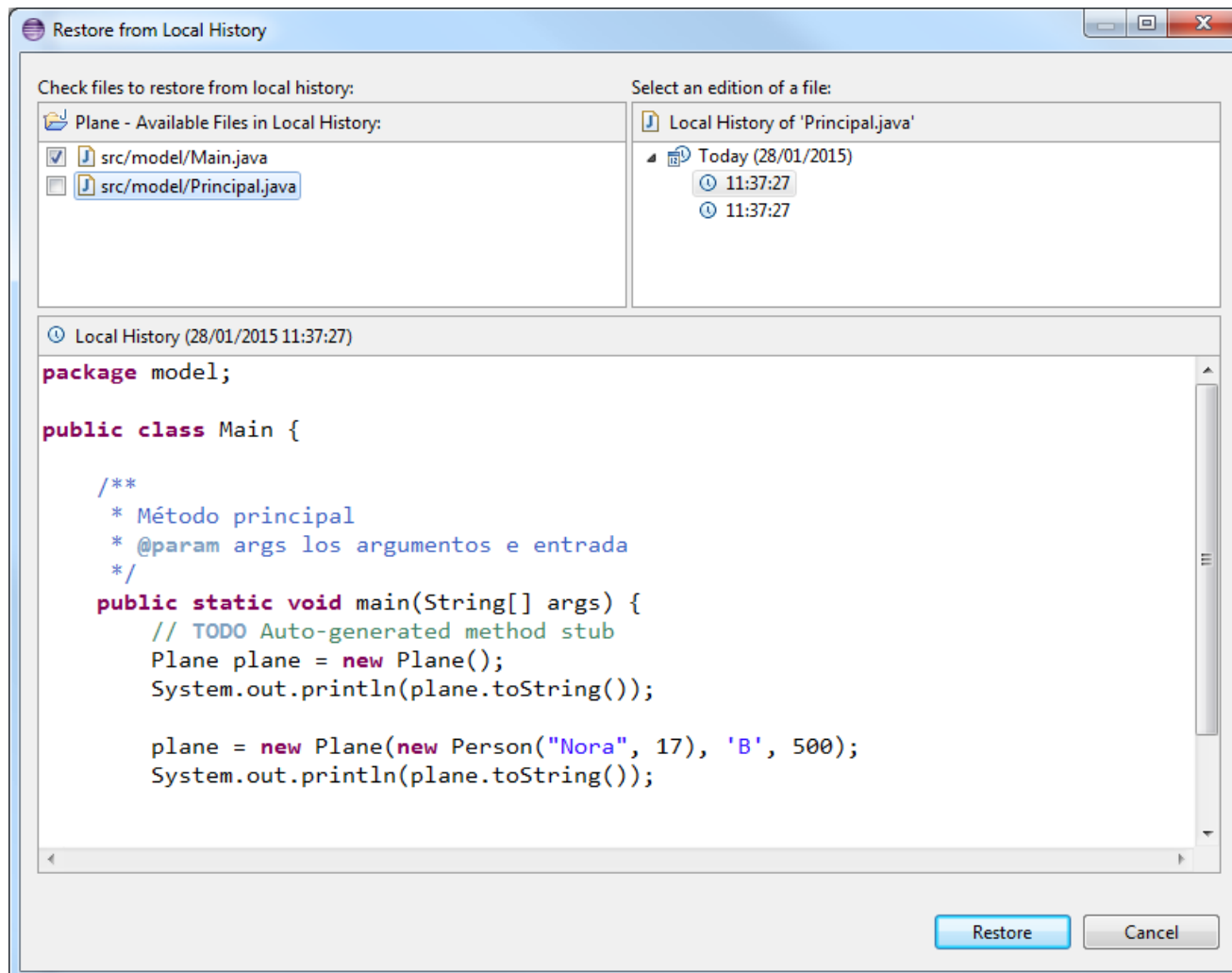
Pruebas con JUnit



REEMPLAZAR VERSIONES

Reemplazar versión (Local History)

- Permite sustituir un archivo por una versión anterior.



Tarea a entregar

- Para renombrar un proyecto, paquete o tarea. Sobre él se pulsa botón derecho y opción
 - Refactor/rename
- Habrá que renombrar el proyecto original que usamos en la tarea
 - nombre_apellido1_apellido2_tareaS1_game
- Sólo se entrega el proyecto pedido
 - Exportarlo con el mismo nombre
 - Subirlo al campus virtual

Ejercicios

- Los ejercicios deben estar acabados **2 días antes** de la siguiente clase de laboratorio.
- **Podrán ser usados en la siguiente clase.**
- **Podrán ser usados en los exámenes de laboratorio.**
- **Todas las tareas de trabajo autónomo (no presencial) que se pidan, deberán subirse al campus virtual.**
- Más de 2 tareas inválidas o no entregadas supone **la pérdida de la evaluación continua**

Enlaces y Bibliografía

- ***Eclipse Documentation***
<http://help.eclipse.org/Mars/index.jsp>
- ***Getting Started with Eclipse***
<https://dzone.com/refcardz/getting-started-eclipse>
- ***Eclipse in Action - A Guide for Java Developers***
David Gallardo; Ed Burnette; Robert McGovern.
Manning Publications, 2003