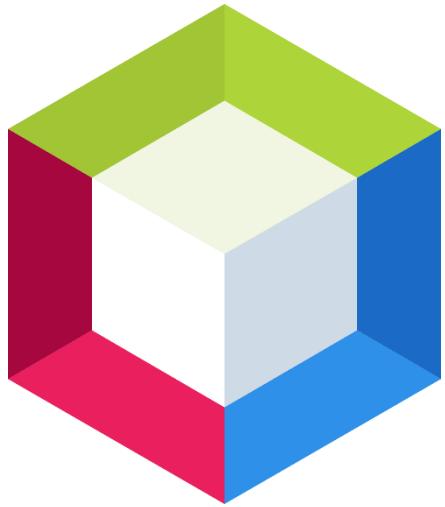


# TEMA 2. INSTALACIÓN Y USO DE ENTORNOS DE IDE

## 2.2. INSTALACIÓN Y USO DE ECLIPSE/NETBEANS

### ENTORNOS DE DESARROLLO

RAÚL PALAO  
25/10/2023



**APACHE NETBEANS**

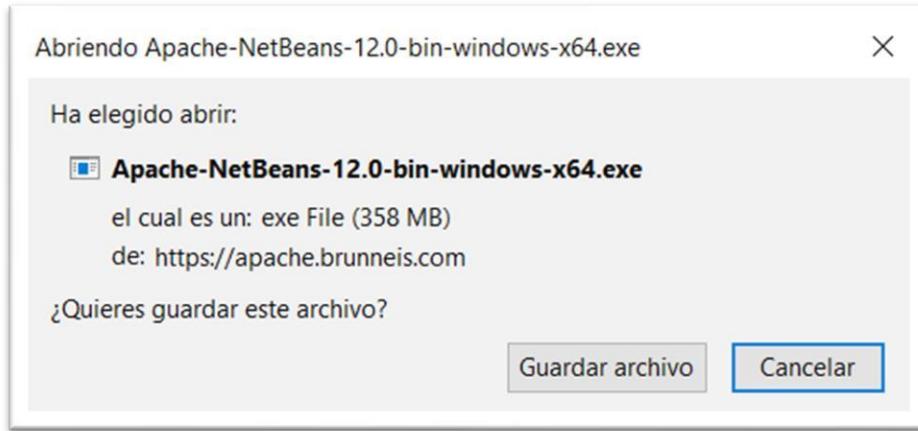
# Presentación de Netbeans

- **NetBeans** es un proyecto de **código abierto** con una comunidad en crecimiento.
- Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas.
- Está escrito en Java pero puede servir para “cualquier” otro lenguaje de programación.
- Inicialmente se denominó NetBeans y soportaba multitud de lenguajes.
- Desde la versión 9 se denomina **Apache Netbeans** y soporta de forma nativa únicamente:
  - Java, PHP, C/C++, HTML y JS.

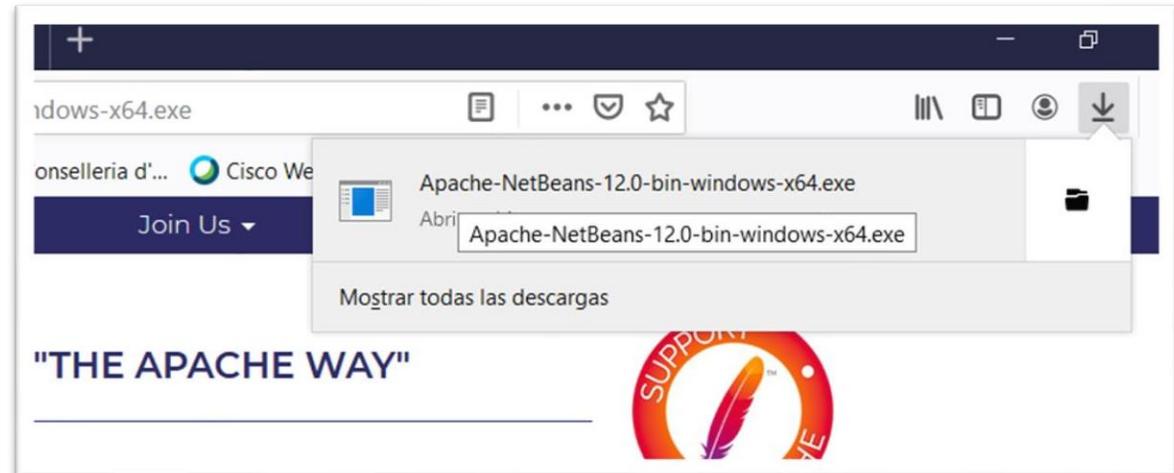


# Instalar Apache NetBeans

Después del JDK, vamos a instalar Apache NetBeans...



Guardamos e Instalamos...



# Instalar Apache NetBeans

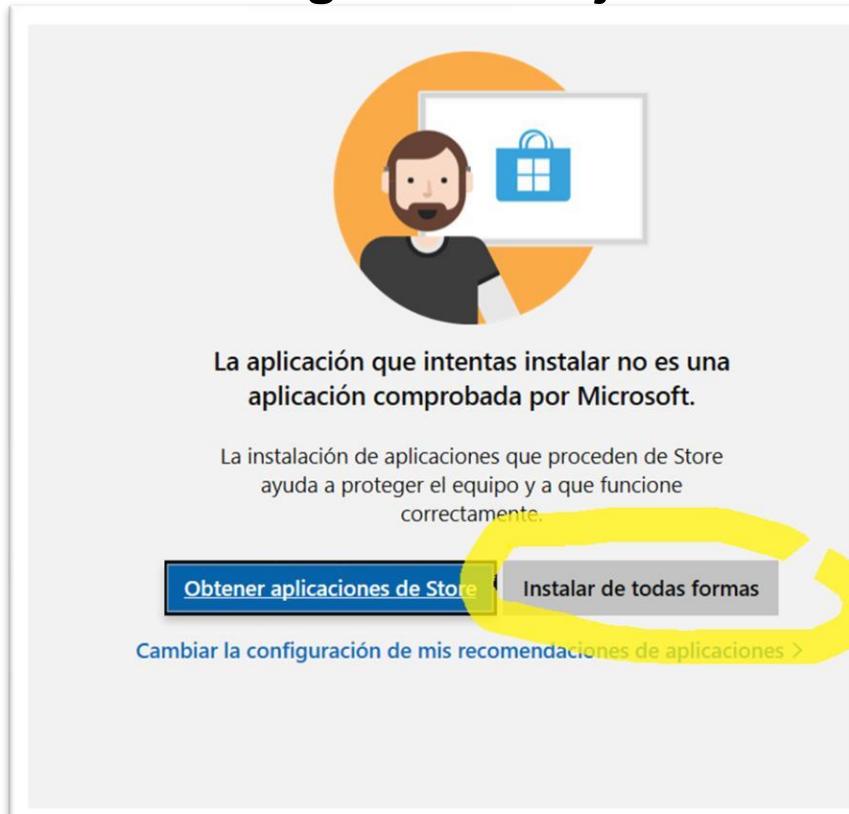
Desde la página <https://netbeans.apache.org>  
Ó <https://netbeans.apache.org/download/index.html>

The screenshot shows the Apache NetBeans homepage. At the top, there's a navigation bar with links like 'Más visitados', 'Aules', 'Conselleria d'Educació...', 'Mon tableau de bord...', 'Grammaire - Règles et...', 'Portal Educatiu - Cons...', 'CEFIRE - Conselleria d...', and 'Cisco Webex Meetings'. Below the navigation is the Apache NetBeans logo and the title 'Apache NetBeans 12.0'. A 'Find out more' button is visible. The main content area features a large image of a geometric cube with the text 'Apache NetBeans Fits the Pieces Together'. Below it, three boxes highlight features: 'Fast & Smart Editing', 'Java, JavaScript, PHP, HTML5, CSS, and More', and 'Cross Platform'. The 'Cross Platform' box is circled in yellow.

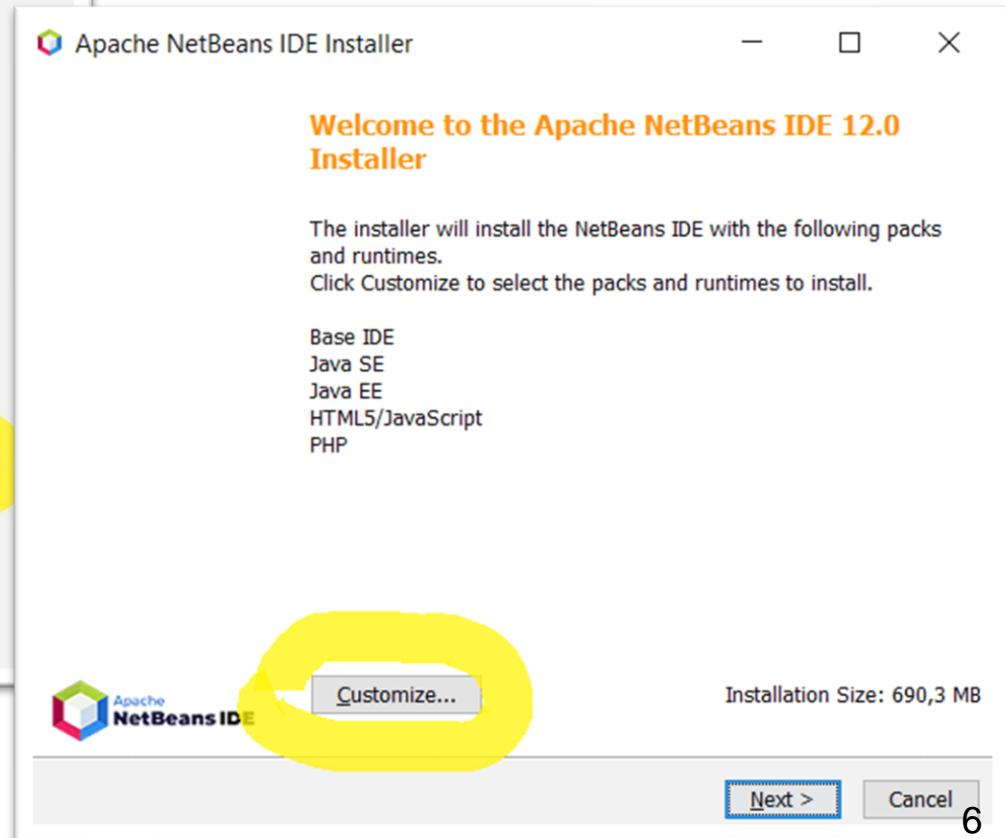
The screenshot shows the Apache NetBeans download page. At the top, there's a navigation bar with links like 'Conselleria d'Educació...', 'Mon tableau de bord...', 'Grammaire - Règles et...', 'Portal Educatiu - Cons...', 'CEFIRE - Conselleria d...', and 'Cisco Webex Meetings'. Below the navigation is the title 'Beans' and the 'Apache NetBeans 12.0' section with a 'Find out more' button. The main content area features a large image with the text 'Apache NetBeans Releases'. Below it, a box states 'Apache NetBeans is released four times a year. For details, see [full release schedule](#)'. A note in a box says: 'Our annual May/June release is a long-term support (LTS) release that benefits from our N remains available and supported for a year. Our other quarterly releases provide early acc consolidated in the subsequent LTS release.' The 'Apache NetBeans 12 LTS (NB 12.0)' section is highlighted with a yellow circle around the 'Download' button. Below it, a link 'Older releases' is visible.

# Instalar Apache NetBeans

Quizá te salga un mensaje similar....



...Aceptamos la instalación y personalizamos



# Instalar Apache NetBeans

En nuestro caso, las opciones marcadas son suficiente...

The image shows two windows related to the Apache NetBeans IDE 12.0 installation.

**Customize Installation Dialog:** This dialog is titled "Customize Installation". It contains a list of "Select packs and runtimes to install from the list below." The options are:

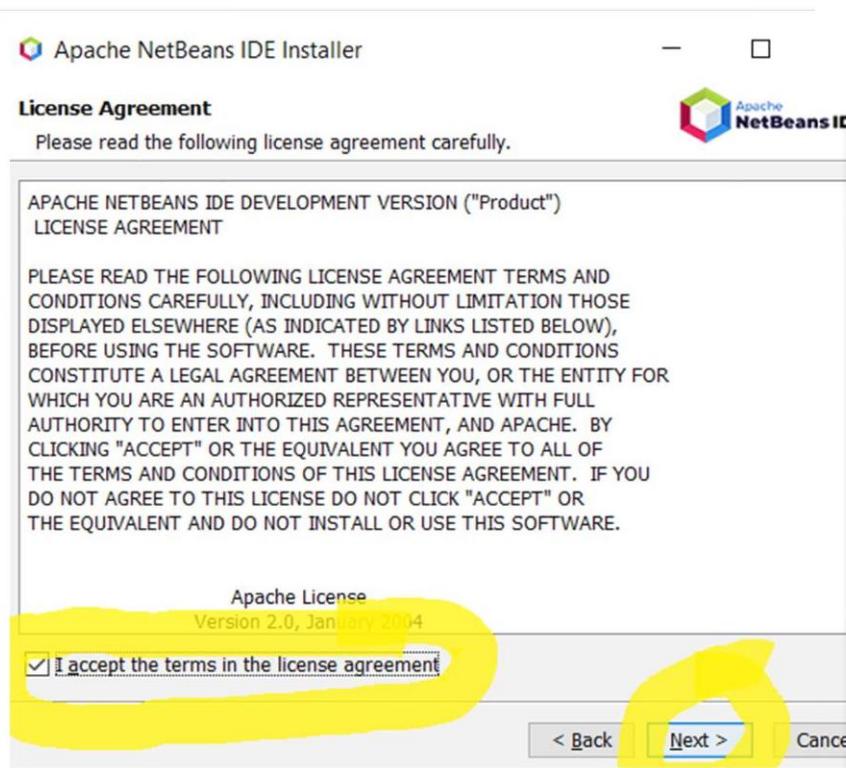
- Base IDE
- Java SE
- Java EE
- HTML5/JavaScript
- PHP

A yellow highlight is placed over the "Base IDE", "Java SE", and "HTML5/JavaScript" checkboxes. A blue highlight is placed over the "PHP" checkbox. Below the list, it says "Installation size: 476,4 MB". At the bottom are "OK" and "Cancel" buttons, with "OK" highlighted by a yellow circle.

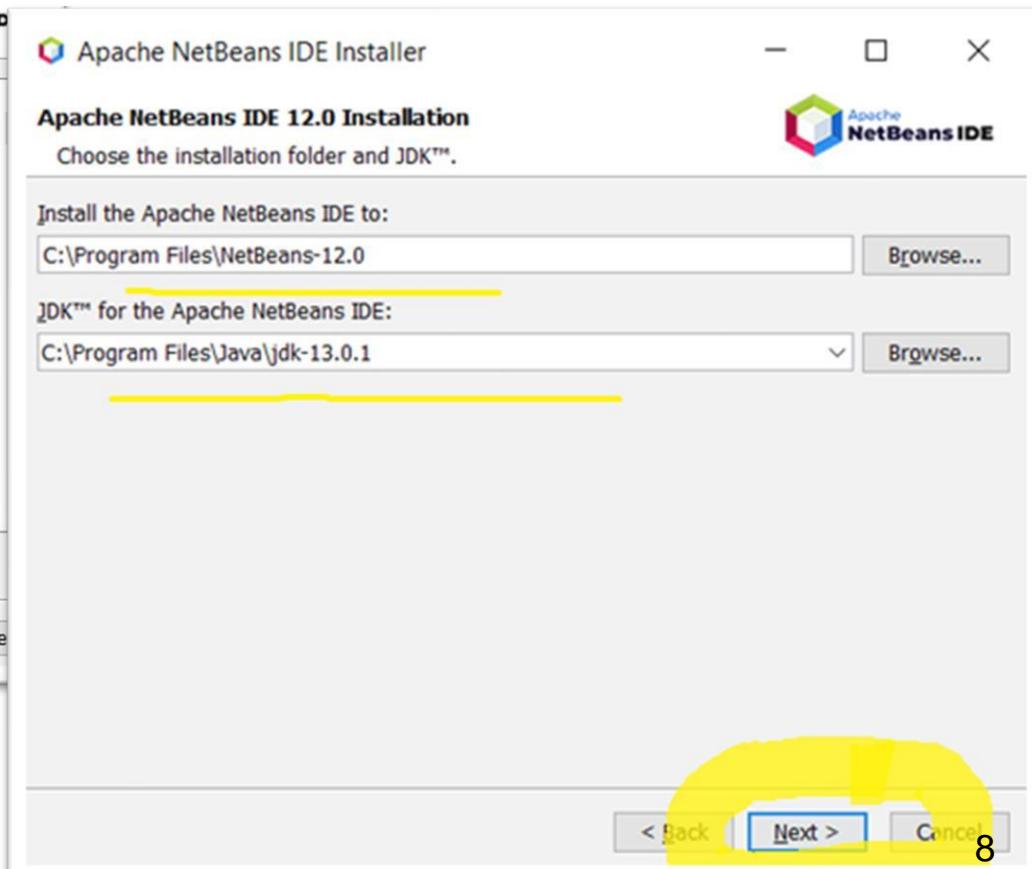
**Main Installer Window:** This window is titled "Apache NetBeans IDE Installer". It displays the message "Welcome to the Apache NetBeans IDE 12.0 Installer". It states: "The installer will install the NetBeans IDE with the following packs and runtimes. Click Customize to select the packs and runtimes to install." Below this, it lists the selected packs: "Base IDE", "Java SE", and "HTML5/JavaScript". At the bottom, it shows "Installation Size: 476,4 MB" and has "Next >" and "Cancel" buttons.

# Instalar Apache NetBeans

Aceptamos...



Es importante saber qué guardamos y dónde!



# Instalar Apache NetBeans

Instalamos...

The image shows two screenshots of the Apache NetBeans IDE Installer window. The left screenshot is titled 'Summary' and contains the following information:

- Summary: Click Install to start the installation.
- Base IDE Installation Folder: C:\Program Files\NetBeans-12.0
- Check for Updates: The NetBeans installer can automatically check for updates of installed plugins using your Internet connection.
- Total Installation Size: 476,4 MB

At the bottom are '< Back' and 'Install' buttons. The right screenshot is titled 'Setup Complete' and contains the following information:

- Setup Complete: Click Finish to complete the Apache NetBeans IDE setup.
- Installation completed successfully.**
- 5 updates successfully installed.
- To launch the IDE, use either the Start menu or the Apache NetBeans desktop icon.
- To change installed components and add NetBeans plugins, use Plugin Manager that is an integral part of Apache NetBeans IDE.

At the bottom is a 'Finish' button.

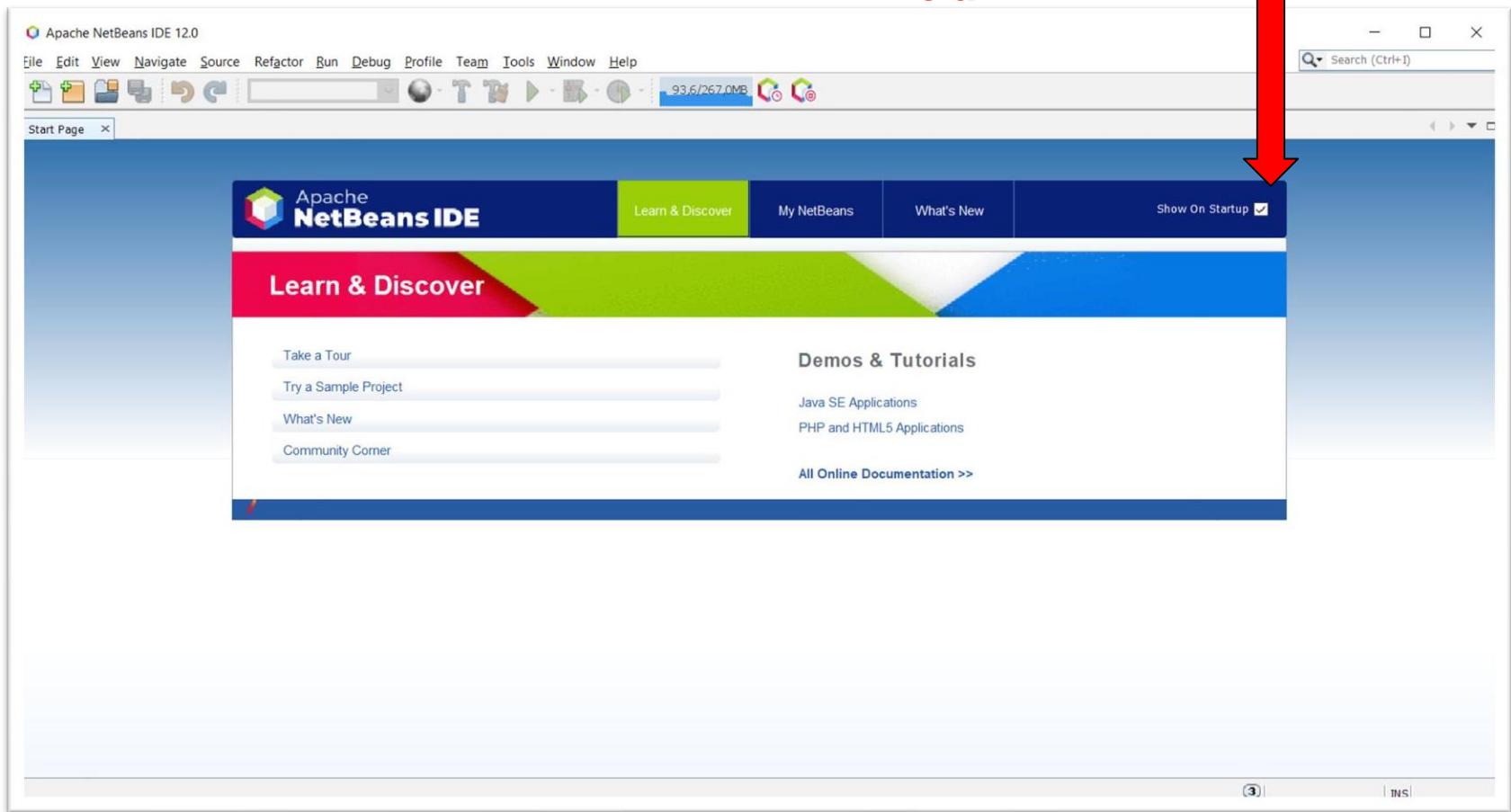
Entomo Netbeans.

Mi primer programa: HolaMundo.



# Netbeans. Pantalla inicial

Para que no salga la pantalla inicial



# Netbeans. Crear un nuevo proyecto

- El primer paso para generar un programa es crear un nuevo proyecto

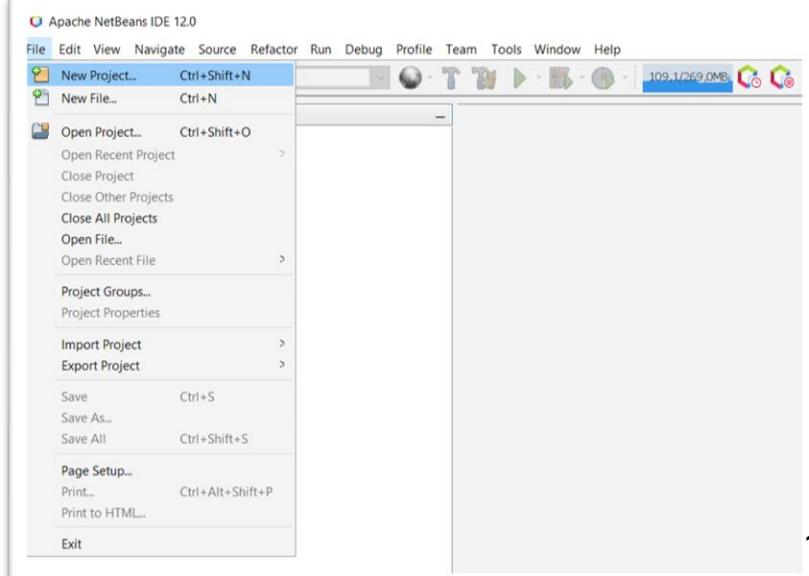
Un proyecto engloba uno o mas ficheros de código Java con relación entre si de manera que se pueden gestionar y almacenar de forma conjunta

- Para nuestros primeros ejercicios que solo ocupan un fichero no es necesario un proyecto individual

Agruparemos cada bloque temático en un proyecto

- Menú principal -> File -> New Project

Se iniciará el asistente de proyectos



# Netbeans. Crear un nuevo proyecto

The screenshot shows the 'New Project' wizard in Netbeans. The first step, 'Choose Project', is selected. In the 'Categories' list, 'Java with Ant' is highlighted with a red arrow. In the 'Projects' list, 'Java Application' is selected. A red callout box labeled 'Java -> Java application' points to this selection. The second step, 'Name and Location', shows 'Project Name' set to 'HolaMundo'. A yellow highlight surrounds the 'Create Main Class' checkbox, which is checked, and the value 'holamundo.HolaMundo' is entered. A red callout box labeled 'Activar' points to this checkbox. Below the 'Name and Location' panel, a note states: 'Por defecto los proyectos se guardan en C:\Users\<NombreUsuario>\Documents\NetBeansProjects\<NombreDelProyecto>'.

New Project

**Steps**

1. Choose Project
2. ...

**Choose Project**

Filter:

**Categories:**

- Java with Maven
- Java with Ant
- JavaFX
- NetBeans Modules
- HTML5/Javascript
- Samples

**Projects:**

- Java Application
- Java Class Library
- Java Project with Existing Sources

**Steps**

1. Choose Project
2. Name and Location

**Name and Location**

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Create Main Class

Por defecto los proyectos se guardan en  
C:\Users\<NombreUsuario>\Documents\NetBeansProjects\<NombreDelProyecto>

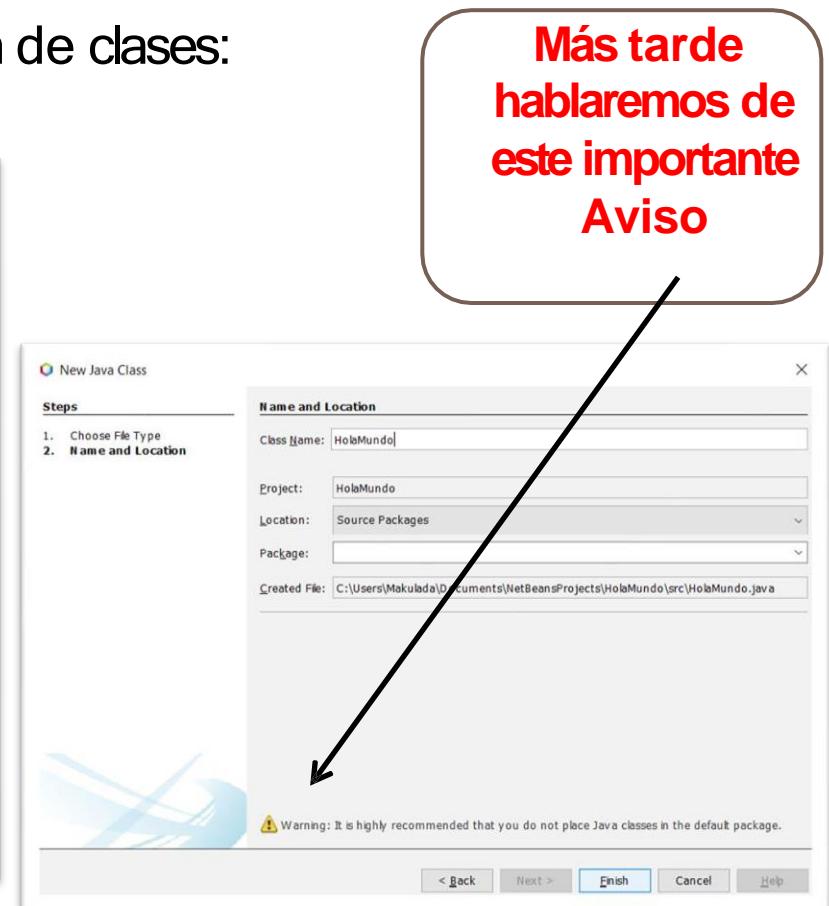
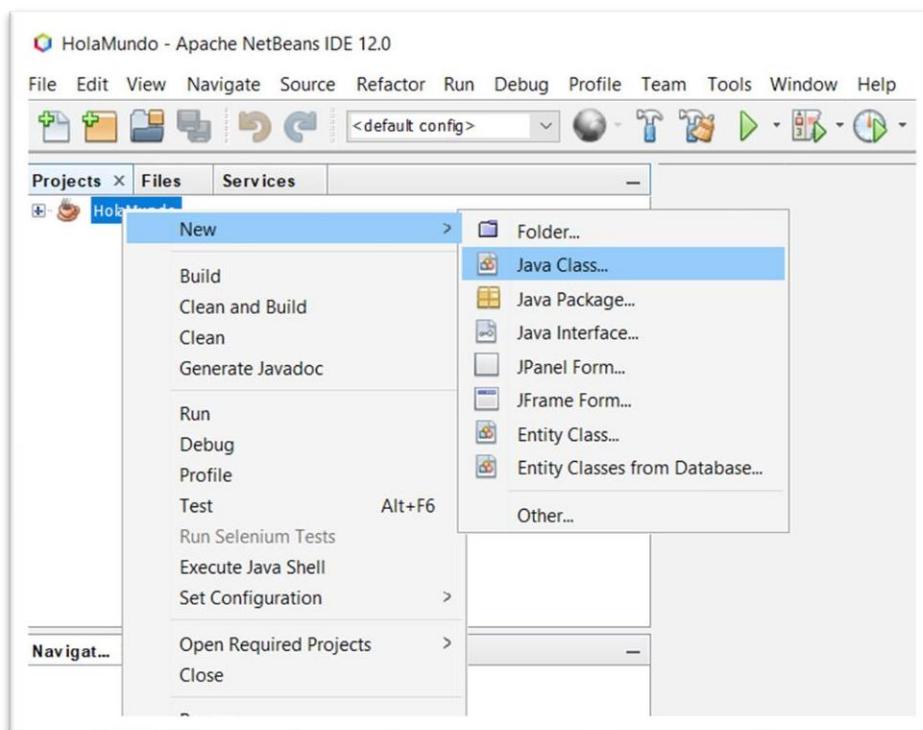
Activar

Java -> Java application

# Netbeans. Crear una nueva clase

- Para crear una nueva clase dentro del *package* por defecto, hay que situarse sobre el proyecto y pulsar el botón derecho  
New-> Java Class

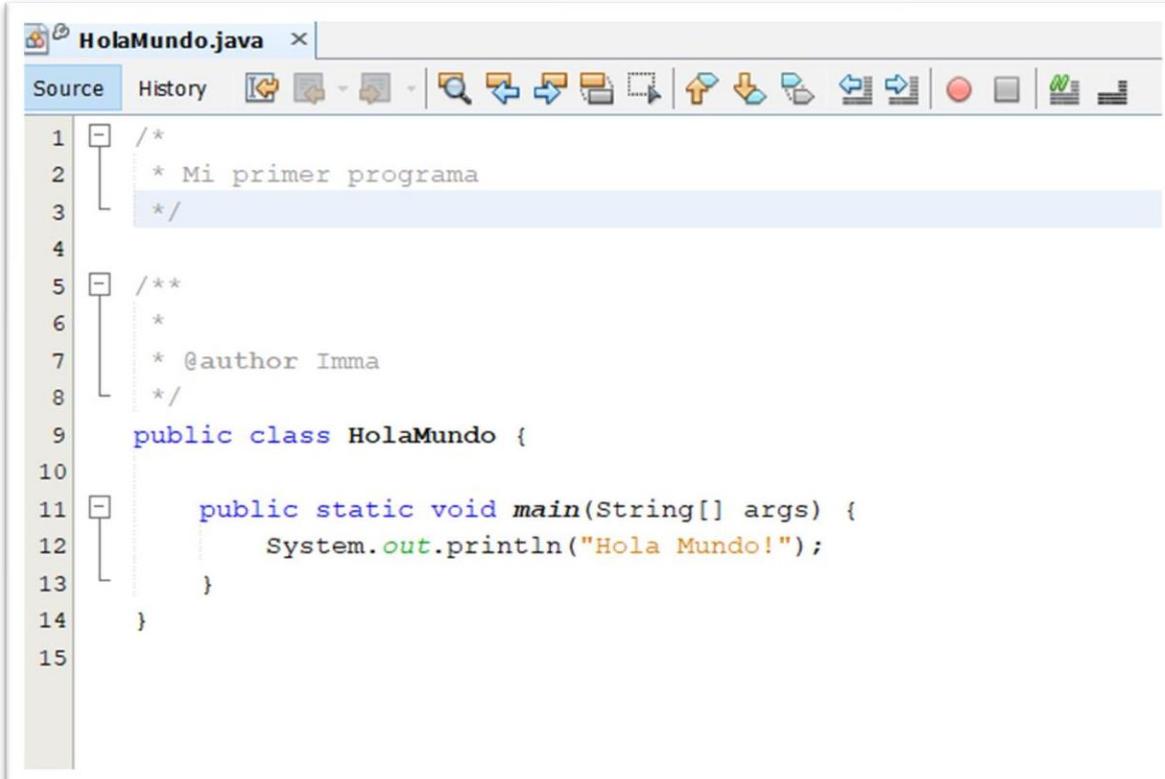
Se iniciará el asistente de creación de clases:



Más tarde  
hablaremos de  
este importante  
Aviso

# Netbeans. Escribir el código

- Introducir el código y corregir errores  
El entorno nos da facilidades para detectar errores



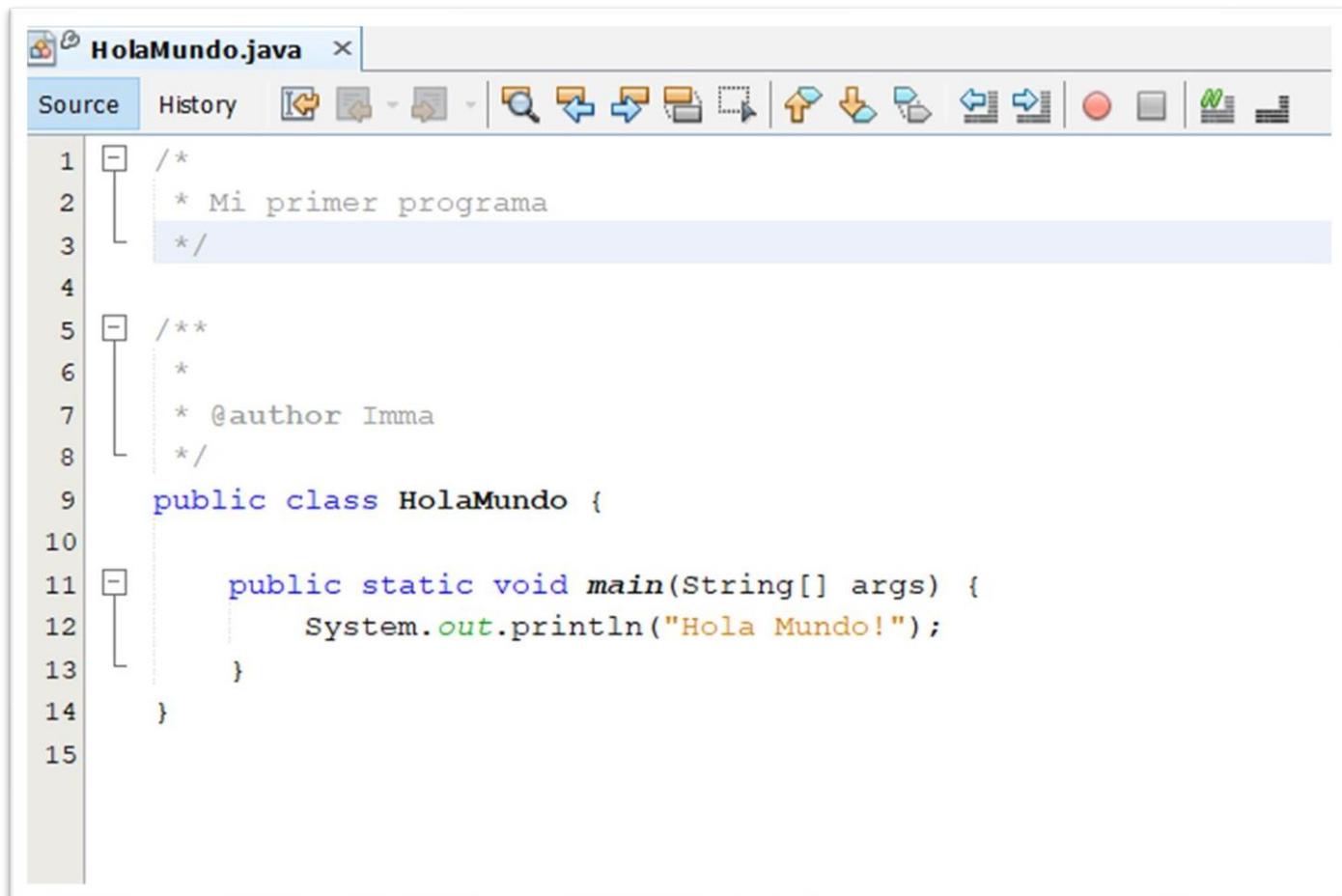
The screenshot shows the NetBeans IDE interface with the file `HolaMundo.java` open. The code editor displays the following Java code:

```
1  /*
2   * Mi primer programa
3   */
4
5  /**
6   *
7   * @author Imma
8   */
9  public class HolaMundo {
10
11     public static void main(String[] args) {
12         System.out.println("Hola Mundo!");
13     }
14 }
15
```

The code is syntax-highlighted, with comments in gray and strings in green. The NetBeans toolbar at the top includes icons for file operations like new, open, save, and build, as well as other developer tools.

# Netbeans. Escribir el código

- Algunos comentarios sobre el código...



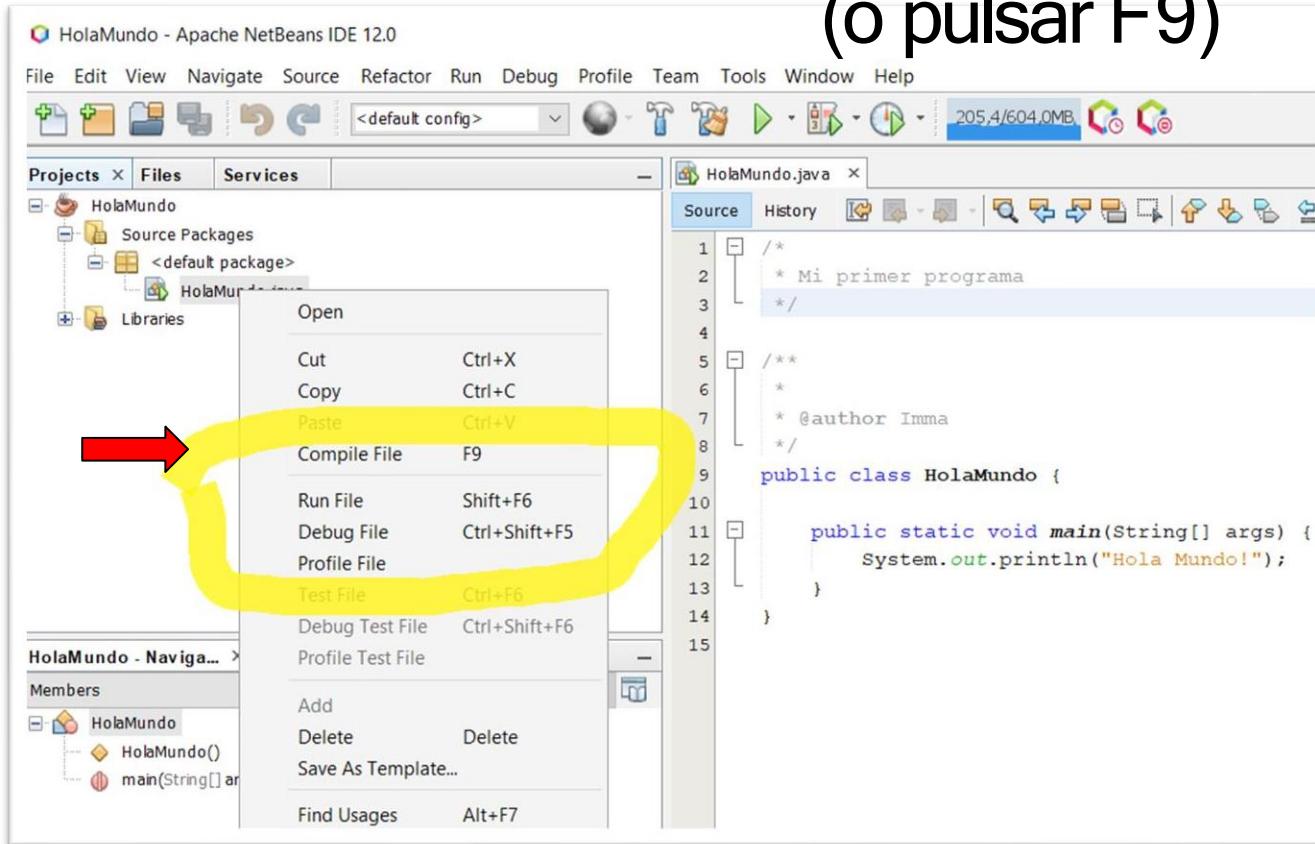
The screenshot shows the NetBeans IDE interface with the file `HolaMundo.java` open. The code editor displays the following Java code:

```
1  /*
2   * Mi primer programa
3   */
4
5  /**
6   *
7   * @author Imma
8   */
9  public class HolaMundo {
10
11     public static void main(String[] args) {
12         System.out.println("Hola Mundo!");
13     }
14 }
15
```

The code includes multi-line comments and a single-line author annotation. The `main` method is present, which prints "Hola Mundo!" to the console.

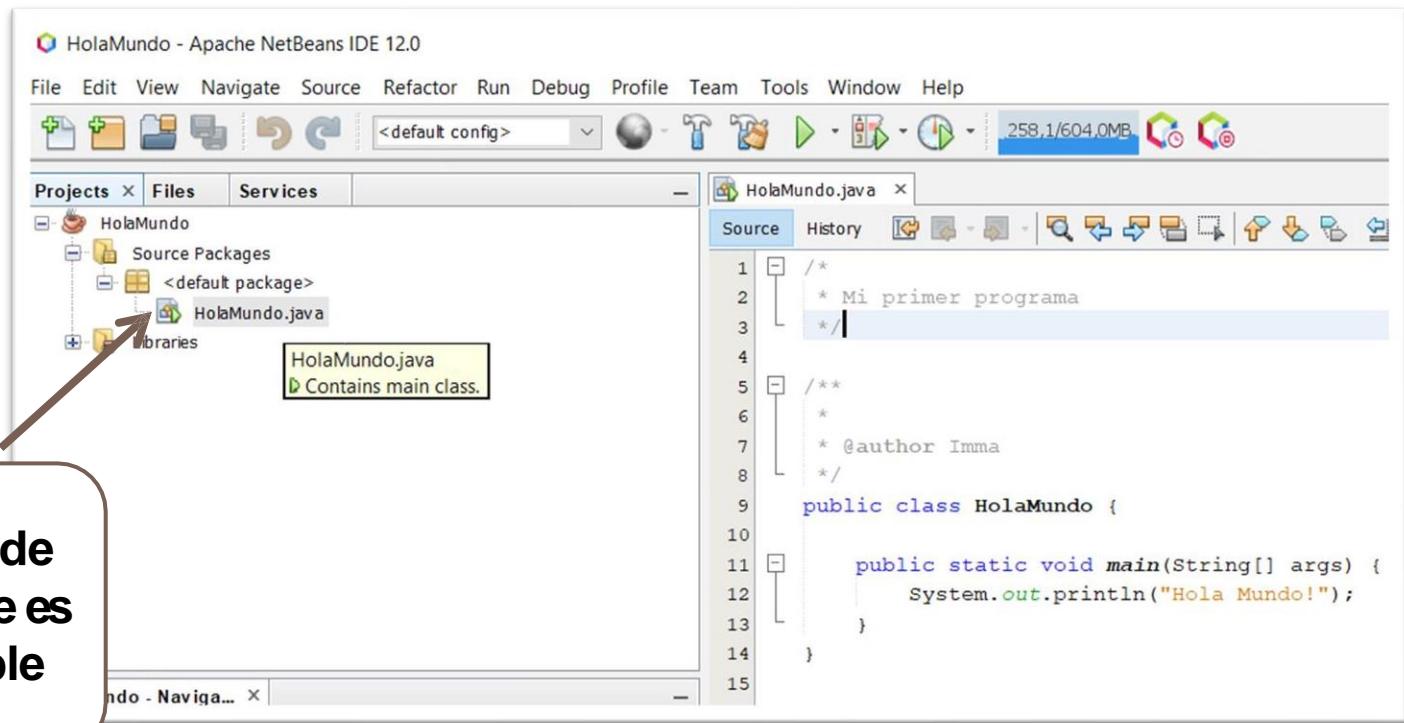
# Netbeans. Compilar

- Para compilar la clase hay que situarse sobre ella y pulsar el botón derecho -> Compile File (o pulsar F9)



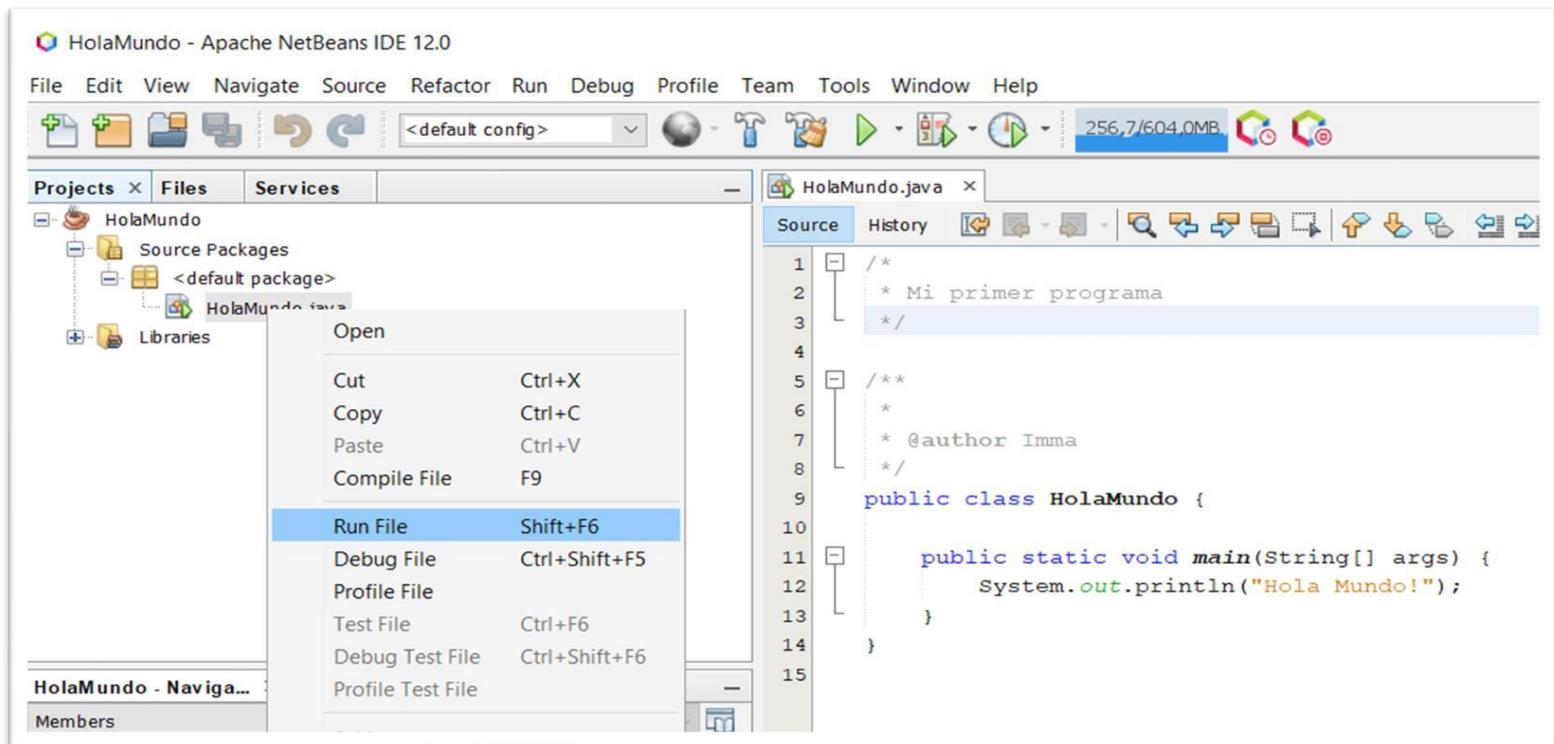
# Netbeans. Ejecutar1

- En Java solo podemos ejecutar clases que tengan un método *main* (con un formato similar al visto)



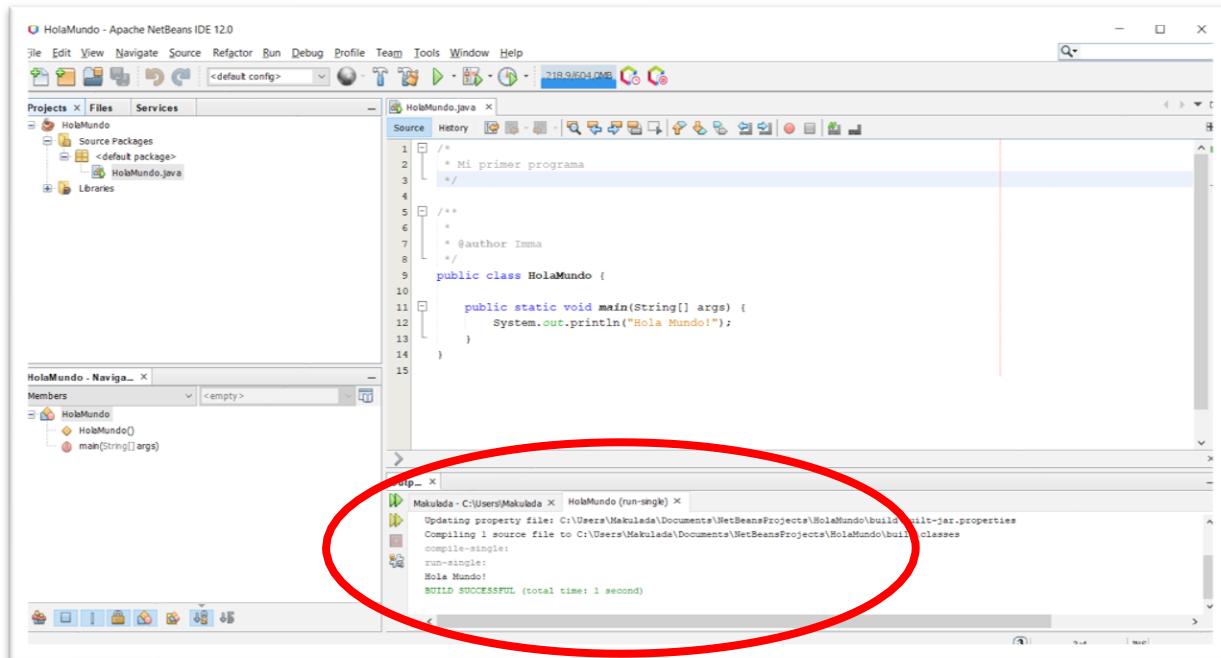
# Netbeans. Ejecutar2

- Para ejecutar la clase hay que situarse sobre ella y pulsar el botón derecho -> Run File  
(Mayus + F6)



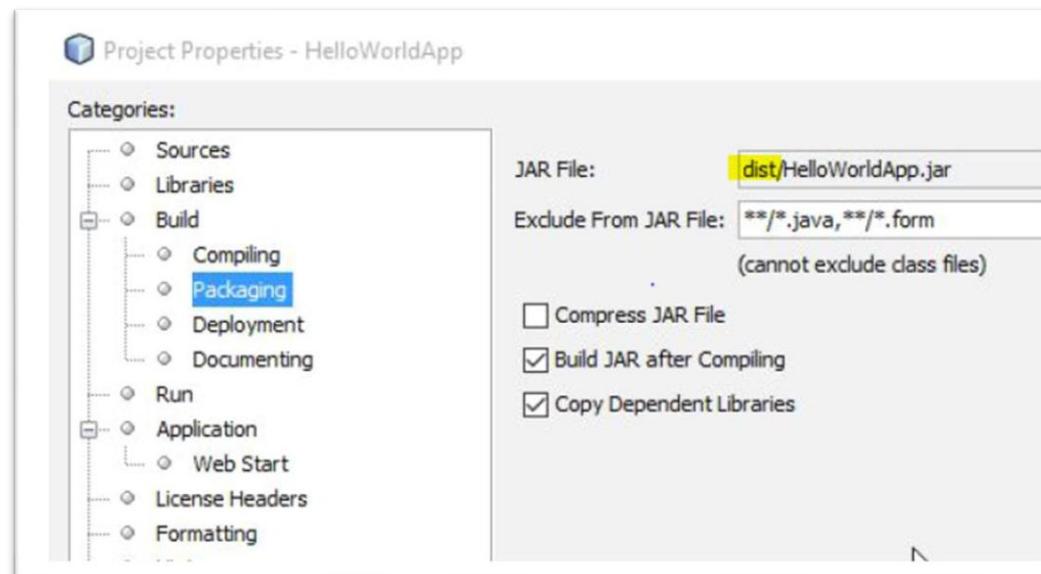
# Netbeans. Ejecutar3

- Siempre que ejecutamos una clase el Netbeans la compila automáticamente
- La ventana Output nos muestra la ejecución



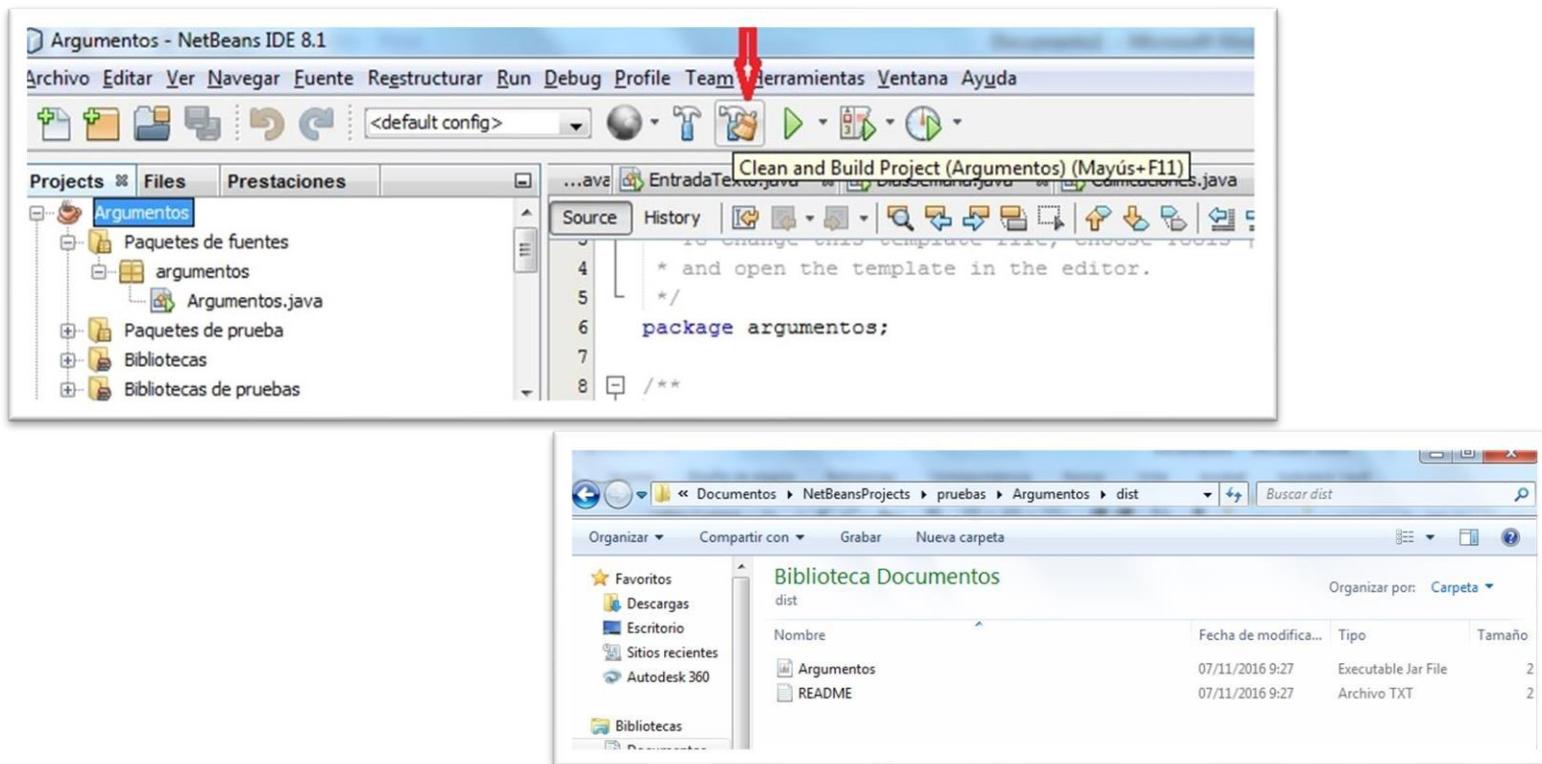
# Crear ficheros JAR desde NetBeans

- Puedo ver donde se guarda el JAR en las propiedades del Proyecto
  - > Project properties > Packaging



# Crear ficheros JAR desde NetBeans

## □ Y solicitarlo



# Paquetes en Java (package)

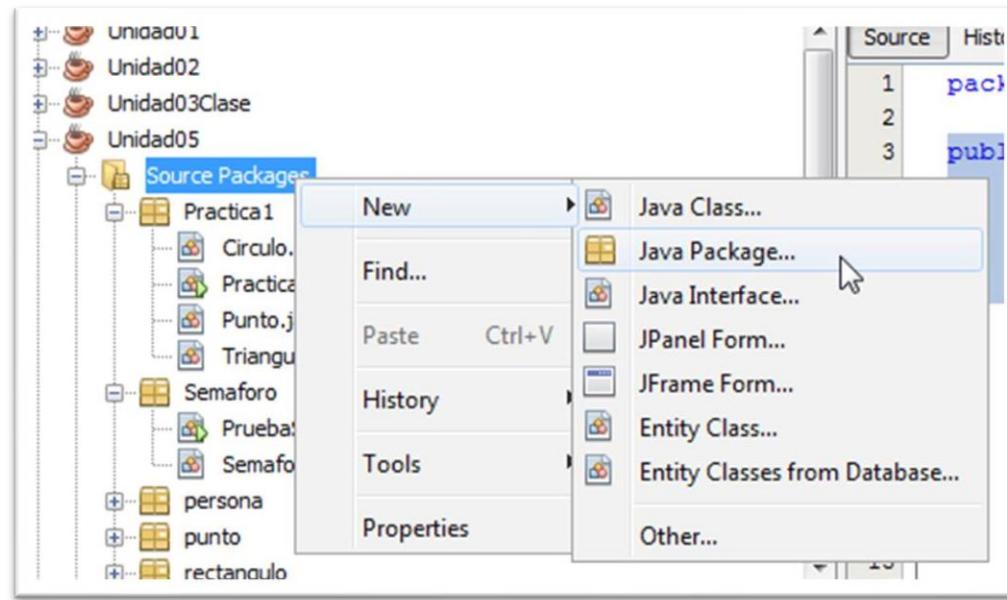
- Los paquetes son una forma de organizar grupos de clases
- Un paquete contiene un conjunto de clases relacionadas bien por finalidad, por ámbito o por herencia
- En el IDE NetBeans
  - un proyecto nuevo se crea en un subdirectorio que tiene el nombre del proyecto
  - Al crear packages dentro del proyecto creamos una estructura de carpetas dentro del proyecto

# Paquetes en Netbeans

- Cuando no se especifica el nombre del paquete al que pertenece una clase, esta pasa a pertenecer al “paquete por defecto”

Aunque esto no genera errores de compilación es recomendable declarar paquetes para todos los componentes de nuestro programa Java

Con el botón derecho  
solicitamos un  
nuevo paquete.



# Paquetes en Netbeans

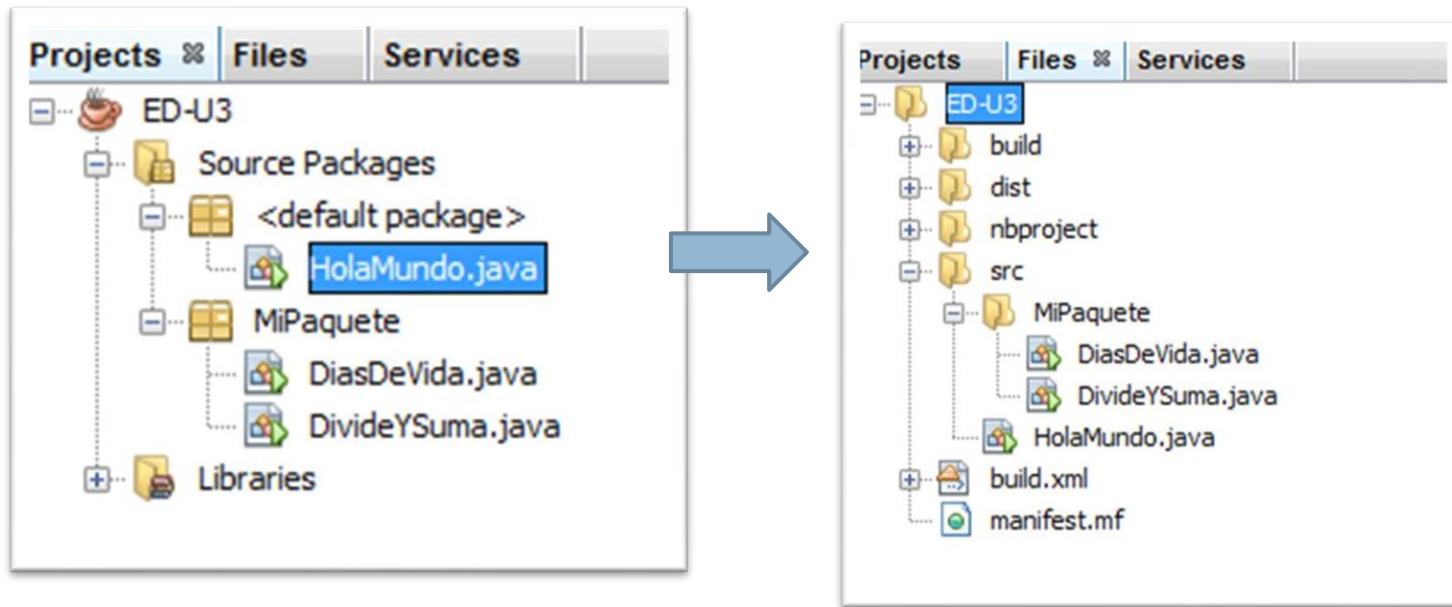
- Se crea la subcarpeta con el nombre del paquete
- Y aparece la instrucción package en la primera línea del fichero

The screenshot shows the Netbeans IDE interface. On the left, the 'Source Packages' panel is open, displaying a tree structure of Java projects and their files. A red circle highlights the 'Semaforo' package, which contains two files: 'PruebaSemaforo.java' and 'Semaforo.java'. Another red circle highlights the 'PruebaSemaforo.java' file. On the right, the code editor displays the content of 'PruebaSemaforo.java'. The first line of code is 'package Semaforo;', also circled in red. The code then defines a class 'PruebaSemaforo' with a main method that creates three instances of the 'Semaforo' class and calls their 'PonColor' and 'DimeColor' methods.

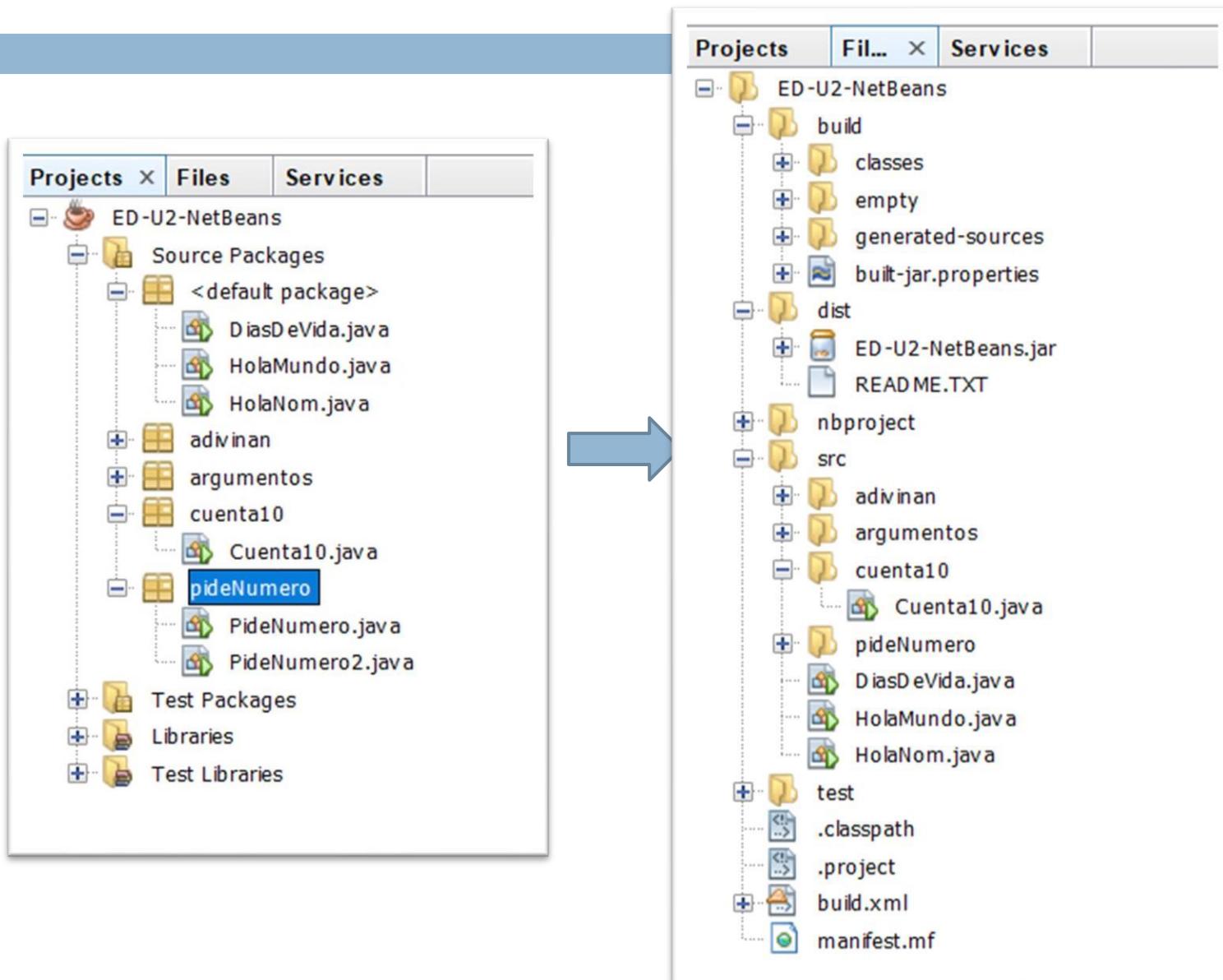
```
java2
dad03Clase
dad05
Source Packages
  Práctica1
  Semaforo
    PruebaSemaforo.java
    Semaforo.java
  persona
    Persona.java
    PersonaApp.java
  punto
  rectángulo
  triángulo
Libraries

1 package Semaforo;
2
3 public class PruebaSemaforo {
4     public static void main (String[] args) {
5         Semaforo MiSemaforo = new Semaforo();
6         Semaforo SemaforoDeMiCalle = new Semaforo();
7         Semaforo OtroSemaforo = new Semaforo();
8
9         MiSemaforo.PonColor("Rojo");
10        OtroSemaforo.PonColor("Verde");
11
12        System.out.println( OtroSemaforo.DimeColor() );
13        System.out.println( SemaforoDeMiCalle.DimeColor() );
14
15        if (MiSemaforo.DimeColor().equals("Rojo"))
16            System.out.println ("No Pasar");
```

# Paquetes en Netbeans

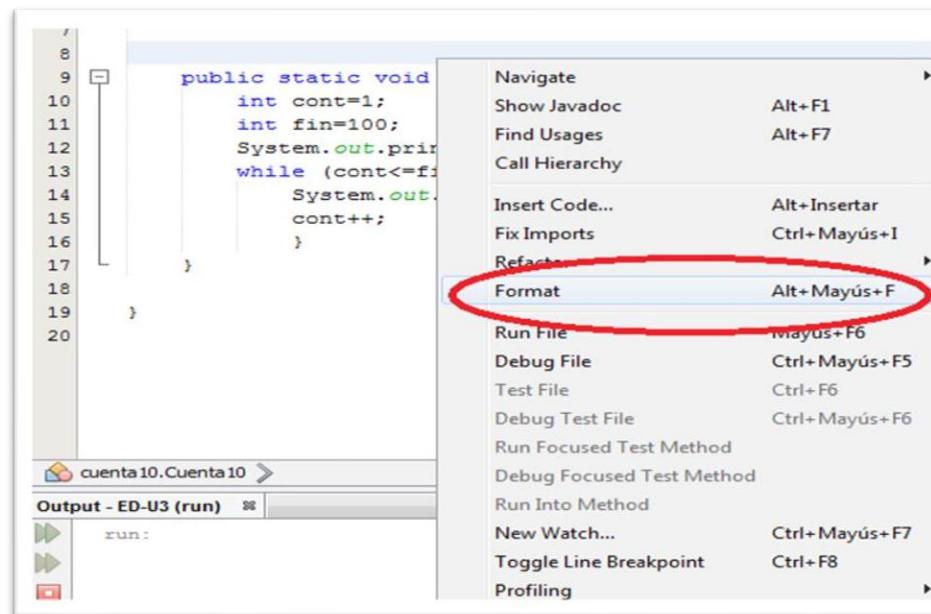


# Carpetas en un proyecto Netbeans



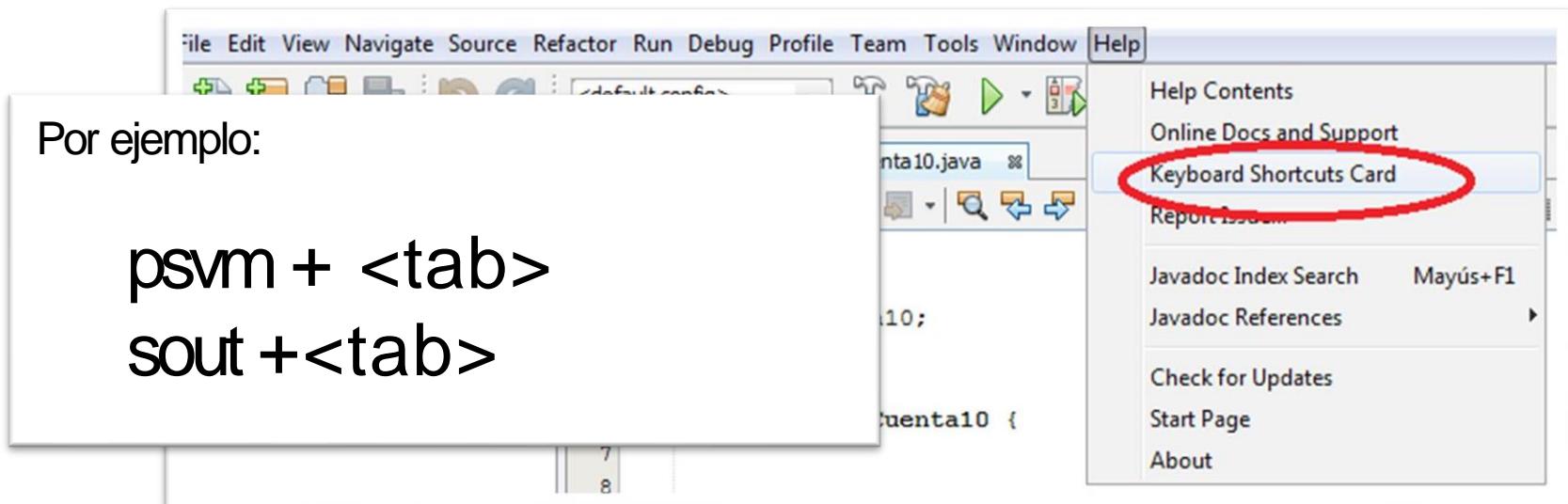
# AutoFormato en Netbeans

- Netbeans nos maqueta los ficheros con corrección  
Botón derecho > Format



# Teclado rápido en Netbeans

- Netbeans me ofrece combinaciones rápidas de teclado
- Puedo consultarlas en
  - Menu > Help > Keyboard Shortcuts Card



Por ejemplo:

psvm + <tab>

sout +<tab>

Entomo Netbeans.  
Depurar en Netbeans.



# Depurar en Netbeans

- Depurar nos permite ejecutar el programa parando en las líneas de código que queramos.
- Nos permite:
  - Encontrar errores.
  - Parar el flujo de ejecución.
  - Ver a tiempo real el estado de las variables.

# Depurar en Netbeans

- Primero debemos establecer un “breakpoint”, que será el punto donde queremos que pare nuestro programa.
- Podemos establecerlo mediante doble click:



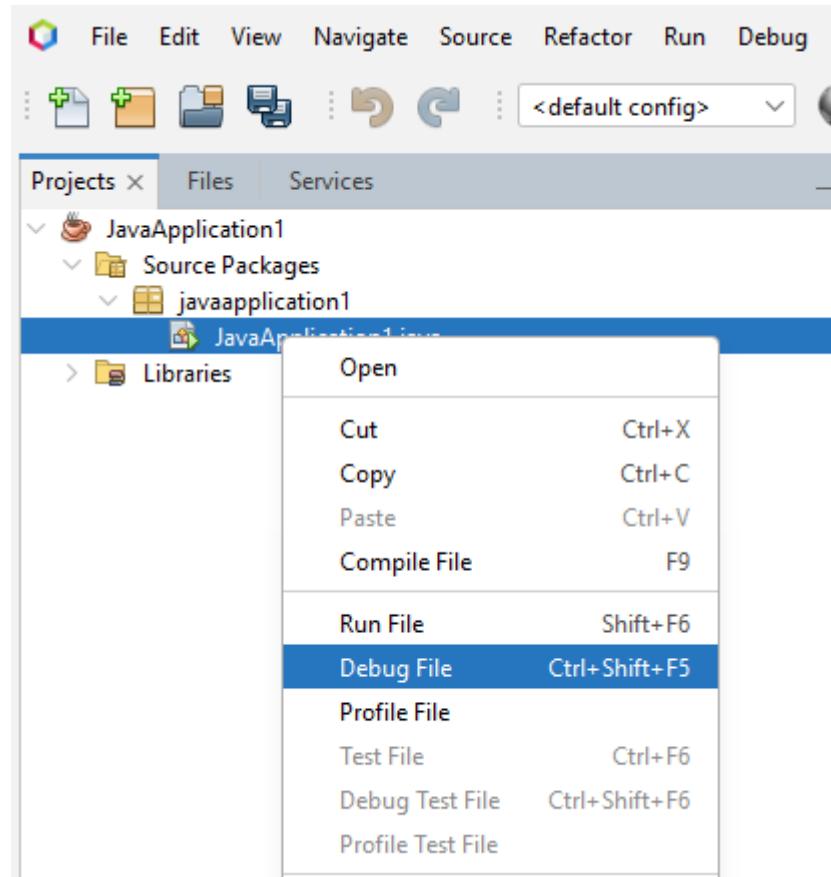
```
10  */
11  public class JavaApplication1 {
12
13  /**
14  * @param args the command line arguments
15  */
16  public static void main(String[] args) {
17      for (int i=0; i<10; i++) {
18          System.out.println(i);
19      }
20  }
21
22 }
```

The code editor shows a Java program named JavaApplication1. A double-clicked breakpoint is visible on line 18, where the println statement is located. The line is highlighted with a red background, and the word 'println' is highlighted in green. The code is as follows:

```
10  */
11  public class JavaApplication1 {
12
13  /**
14  * @param args the command line arguments
15  */
16  public static void main(String[] args) {
17      for (int i=0; i<10; i++) {
18          System.out.println(i);
19      }
20  }
21
22 }
```

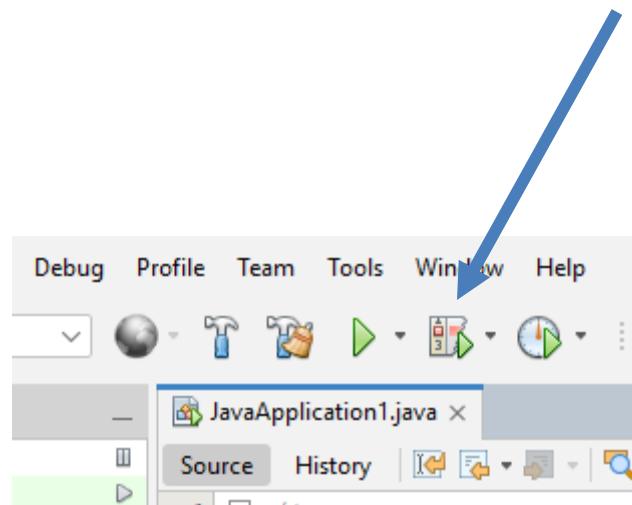
# Depurar en Netbeans

- Podemos acceder al depurado clicando en:
  - Botón derecho en la clase JAVA > Debug File.



# Depurar en Netbeans

- Si queremos acceder al debugger de la última clase también podemos clicar en este botón:



# Depurar en Netbeans

- El flujo para la ejecución en el breakpoint.  
Podemos ver el estado actual de las variables:

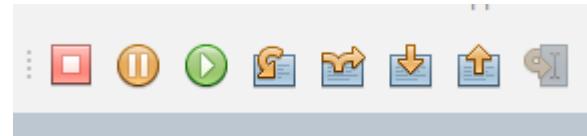
```
9 * gauthor Raul
10 */
11 public class JavaApplication1 {
12
13 /**
14 * @param args the command line arguments
15 */
16 public static void main(String[] args) {
17     for (int i=0; i<10; i++){
18         System.out.println("i");
19     }
20 }
21
22 }
```

javaapplication1.JavaApplication1 > main > for (int i = 0; i < 10; i++) >

Name	Type
Static	String[]
args	String[]
i	int

# Depurar en Netbeans

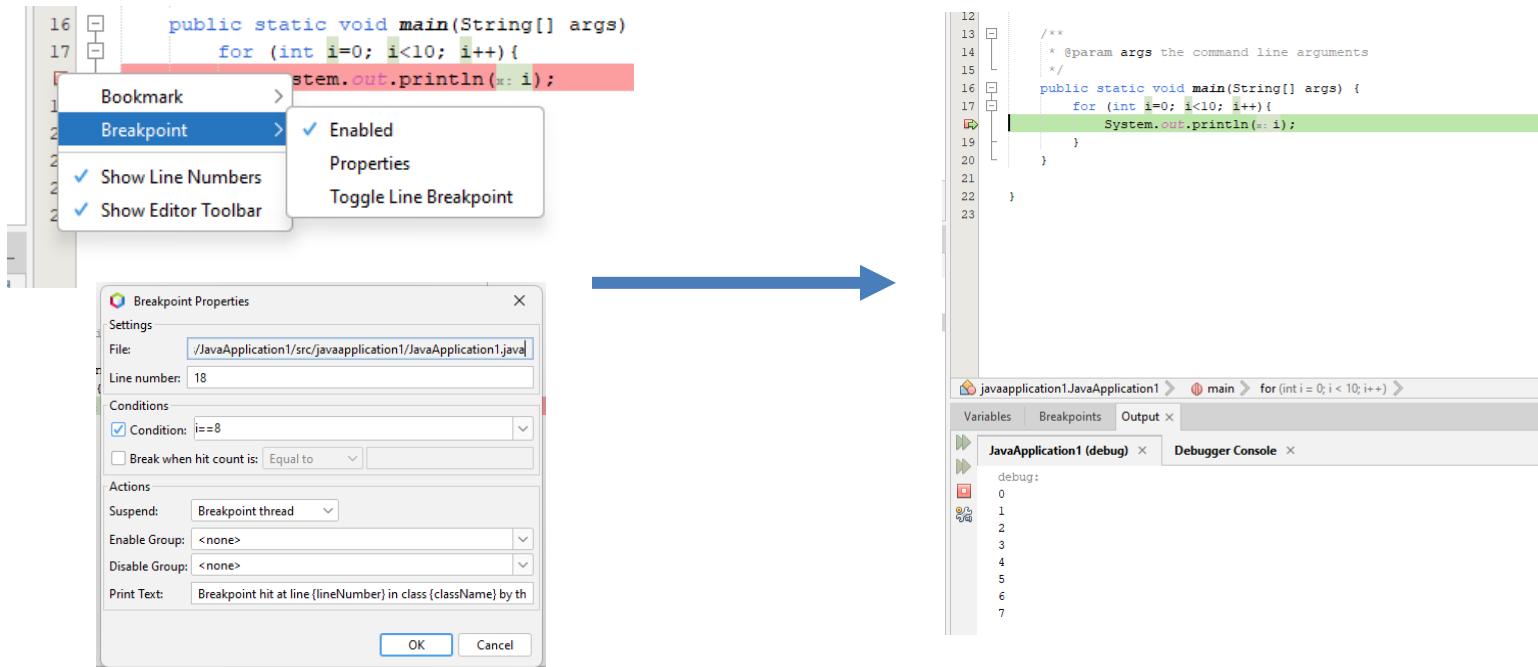
- En la parte superior podemos ver una serie de botones:



- Para el flujo de ejecución y sale del programa.
- Permite continuar hasta el siguiente breakpoint.
- Step Into: entra a la función interior.
- Step Over: salta a la siguiente línea del código.

# Depurar en Netbeans

- Puedo establecer condiciones dentro de un breakpoint.
  - Botón derecho breakpoint > Properties.
- En este caso parará cuando imprima el 7.





ECLIPSE

# Presentación de Eclipse

- El antecesor de Eclipse fue **VisualAge de IBM** y se construyó en el lenguaje **Smalltalk** en un entorno de desarrollo llamado **Envy**.
- Con la aparición de Java en la década de los 90, IBM desarrolló una maquina virtual dual, válida tanto para Smalltalk y Java.
- La rápida expansión de Java obligaron a IBM a plantearse el **abandono de esta máquina virtual dual** y la construcción de una nueva plataforma basada en Java desde el principio.
- El producto final resultante fue Eclipse, que ya había costado unos 40 millones de dólares a IBM en el año 2001.



# Presentación de Eclipse

- De hecho, el **nombre de Eclipse** fue elegido porque el objetivo era crear un IDE capaz de "**eclipsar a Visual Studio**" de Microsoft.
- Rumores dicen que la razón por la que Sun Microsystems no quiso asociarse con ellos fue porque el juego de palabras "eclipsar el sol" (Sun = Sol, en inglés) **no gustó al dpto. de marketing de Sun Microsystems**.
- Además, de que el nombre se encontraba ya utilizado en internet por un equipo de fútbol femenino de Illinois ([eclipse.org](http://eclipse.org)).
- En los apuntes, os retamos a averiguar cuánto pagó la Fundación Eclipse por ese dominio.



# Instalar Eclipse

## Desde la página

<https://www.eclipse.org/downloads/>

The screenshot shows the Eclipse Foundation's Downloads page. At the top, there's a navigation bar with links to Projects, Supporters, Collaborations, Resources, and The Foundation, along with a prominent orange "Download" button. Below the navigation, there's a large graphic of a smartphone with a download arrow pointing to it. To the right of the graphic, the text reads "Download Eclipse Technology that is right for you". A yellow banner at the bottom encourages users to participate in the State of Open Source Survey, offering a \$100 gift card for participation. Two main download sections are visible: one for "Install your favorite desktop IDE packages" (with a blue arrow pointing to the "Learn More" button) and another for "The Eclipse Temurin™ project" (with a grey arrow pointing to the "Learn More" button). Both sections include "Download" buttons.

ECLIPSE FOUNDATION

Projects   Supporters   Collaborations   Resources   The Foundation

Download

Download Eclipse Technology that is right for you

Participate in the State of Open Source Survey

Take a few moments to share how your organisation uses open source in the annual State of Open Source Survey and be entered for a chance to win a \$100 gift card.

Take the Survey

Install your favorite desktop IDE packages

Learn More   Download x86\_64

Download Packages | Need Help

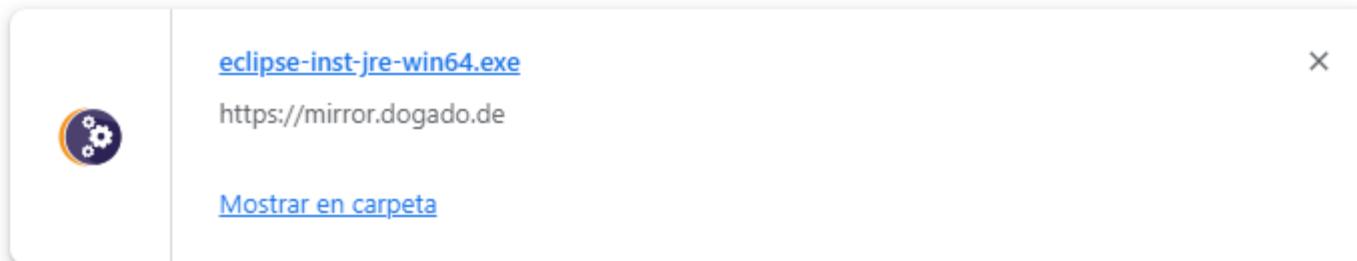
TEMURIN

The Eclipse Temurin™ project provides high-quality, TCK certified OpenJDK runtimes and associated technology for use across the Java™ ecosystem.

Learn More   Download

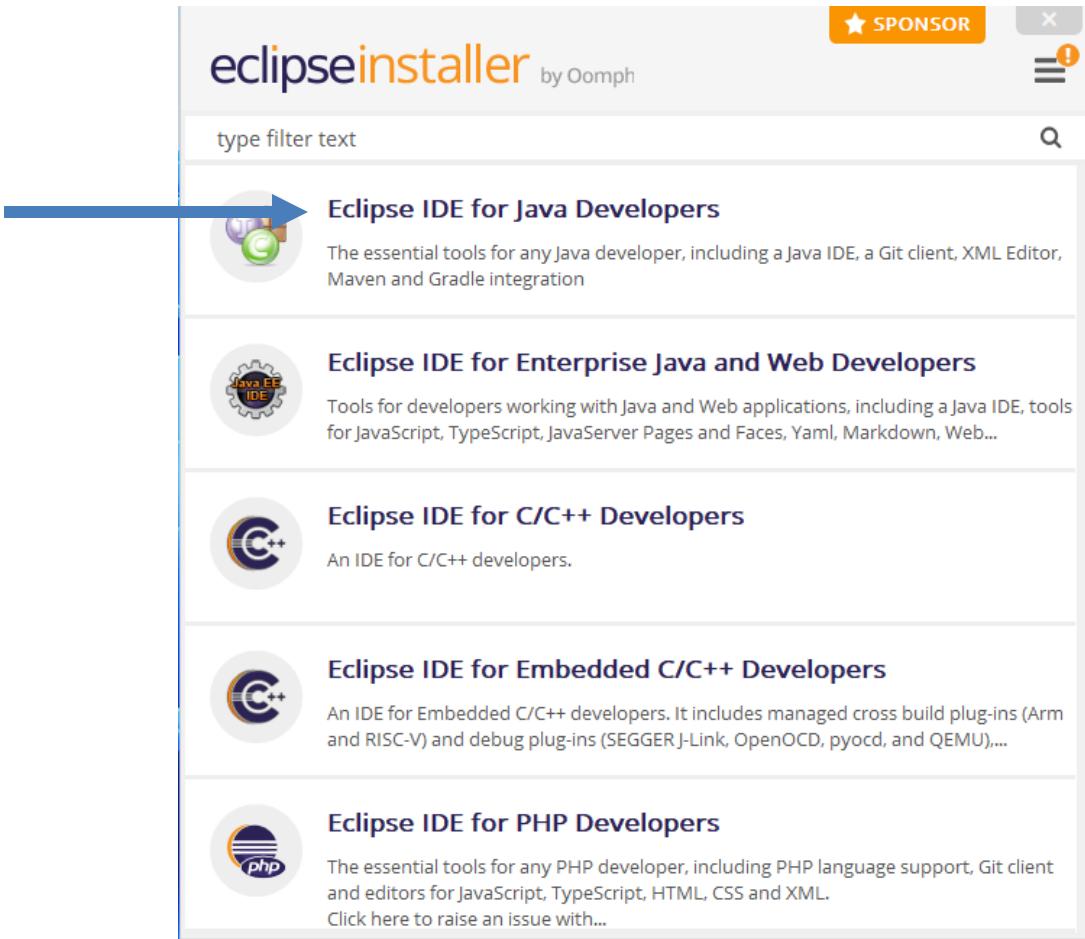
# Instalar Eclipse

Después del JDK, vamos a instalar Eclipse...



# Instalar Eclipse

Seleccionamos la versión para “Java Developers”:



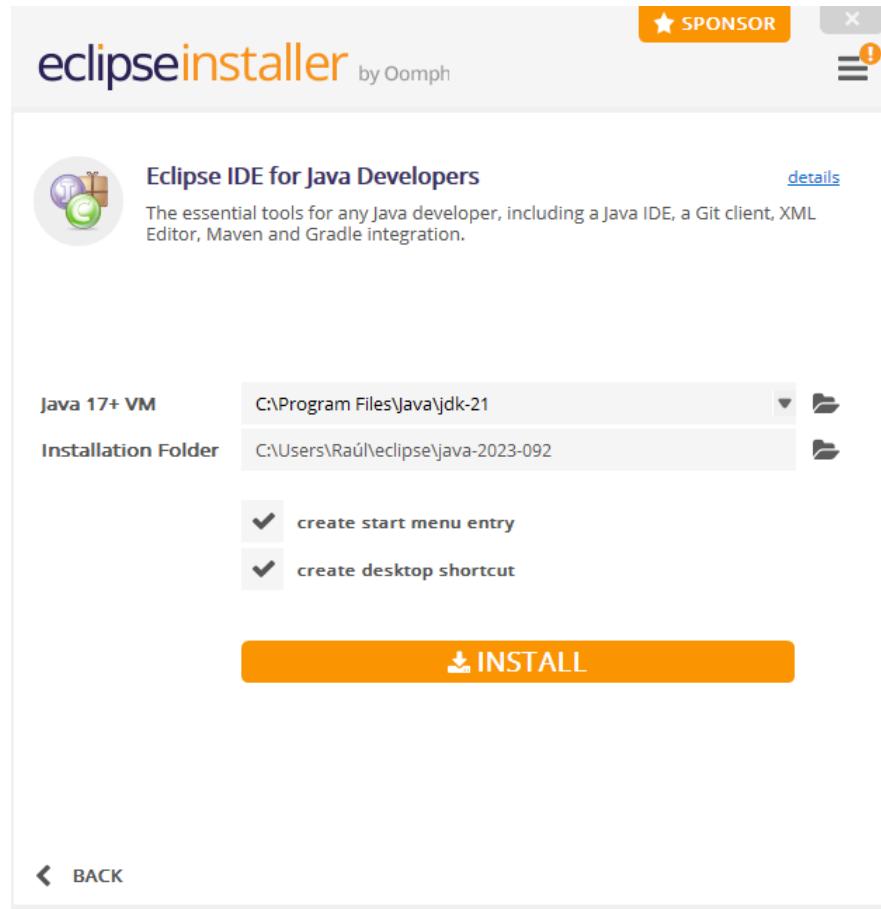
The screenshot shows the **eclipseinstaller** website interface. At the top, there's a search bar with placeholder text "type filter text" and a magnifying glass icon. To the right of the search bar are a "SPONSOR" button with a star icon, a menu icon, and a help icon. Below the search bar, there are five cards, each representing a different Eclipse IDE variant:

- Eclipse IDE for Java Developers**: This card features a green icon with a white 'E' and Java-related symbols. It describes the essential tools for Java developers, including a Java IDE, Git client, XML Editor, Maven, and Gradle integration.
- Eclipse IDE for Enterprise Java and Web Developers**: This card features a grey icon with a gear and Java EE text. It describes tools for Java and Web applications, including support for JavaScript, TypeScript, JavaServer Pages, and Faces.
- Eclipse IDE for C/C++ Developers**: This card features a grey icon with a 'C' and '+' symbol. It describes an IDE for C/C++ developers.
- Eclipse IDE for Embedded C/C++ Developers**: This card features a grey icon with a 'C' and '+' symbol. It describes an IDE for Embedded C/C++ developers, mentioning managed cross build plug-ins for Arm and RISC-V, and debug plug-ins like SEGGER J-Link, OpenOCD, pyocd, and QEMU.
- Eclipse IDE for PHP Developers**: This card features a grey icon with a blue 'php' logo. It describes the essential tools for PHP developers, including PHP language support, Git client, and editors for various web technologies.

A blue arrow points to the first card, highlighting the "Eclipse IDE for Java Developers".

# Instalar Eclipse

Seleccionamos la carpeta donde se encuentra el ‘bin’ de java y la de destino de instalación:



Entomo Eclipse.

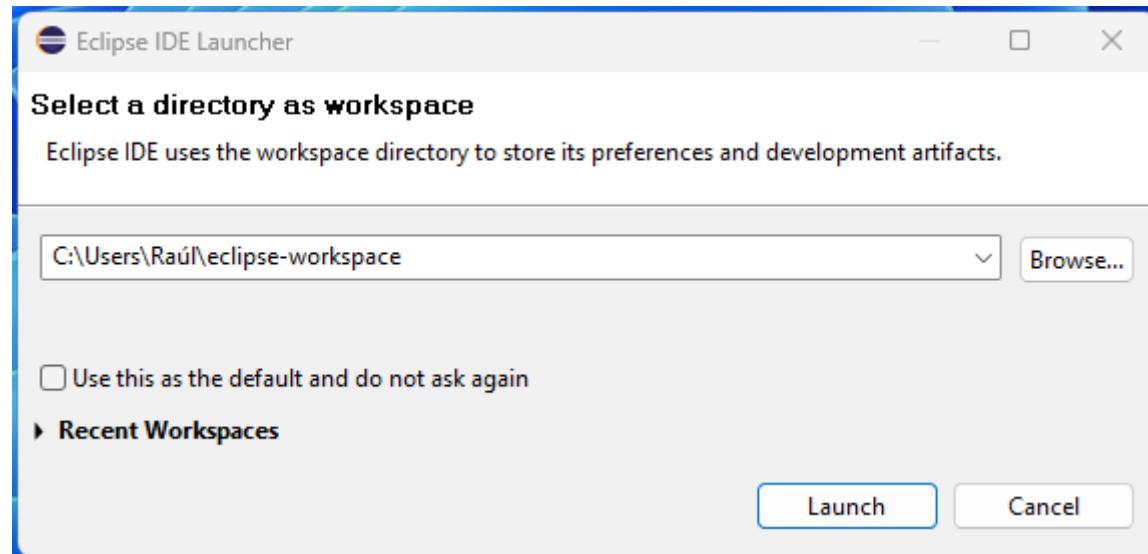
Mi primer programa: HolaMundo.

eclipse



# Eclipse. Pantalla inicial

Seleccionamos la carpeta donde se guardarán los proyectos:



# Eclipse. Crear un nuevo proyecto

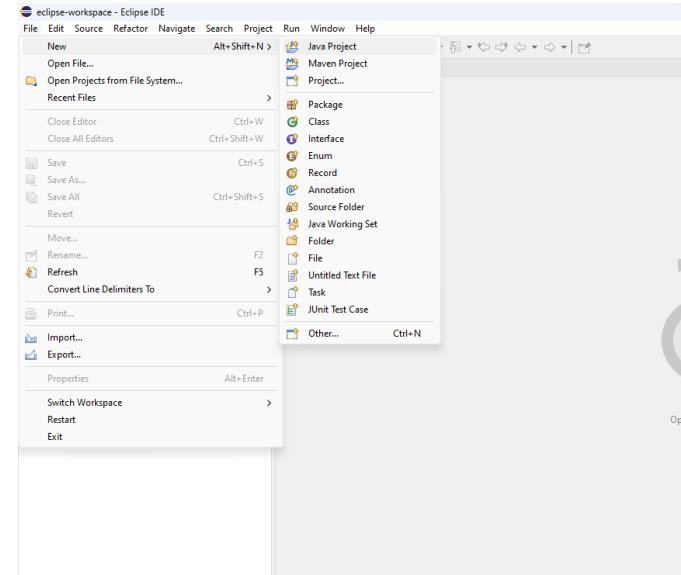
- El primer paso para generar un programa es crear un nuevo proyecto

Un proyecto engloba uno o mas ficheros de código Java con relación entre si de manera que se pueden gestionar y almacenar de forma conjunta

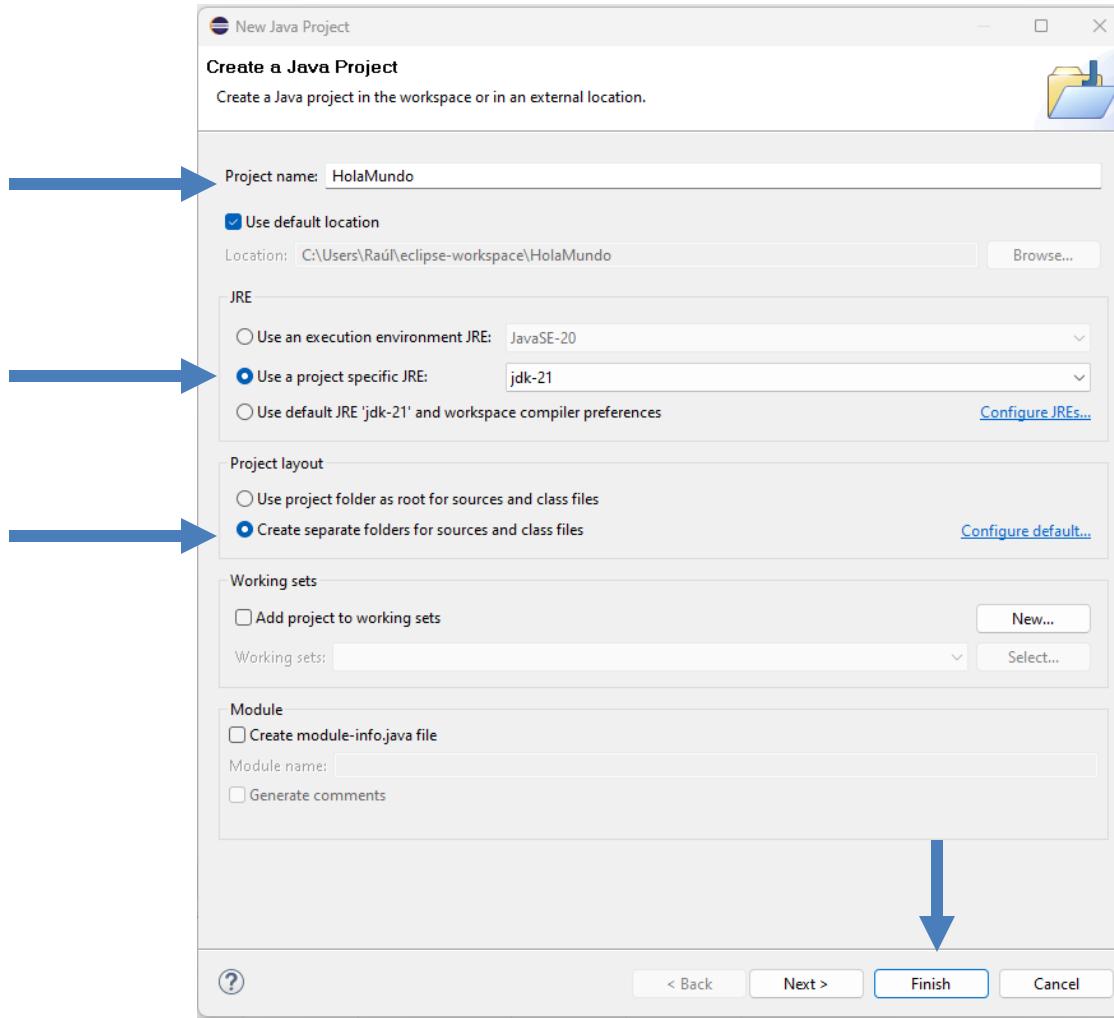
- Para nuestros primeros ejercicios que solo ocupan un fichero no es necesario un proyecto individual

Agruparemos cada bloque temático en un proyecto

- Menú principal -> File -> New -> Java project

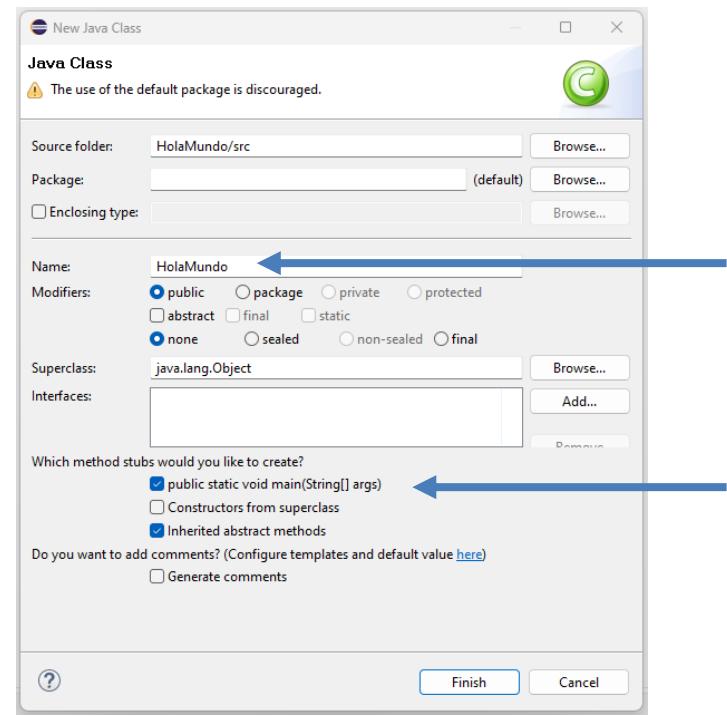
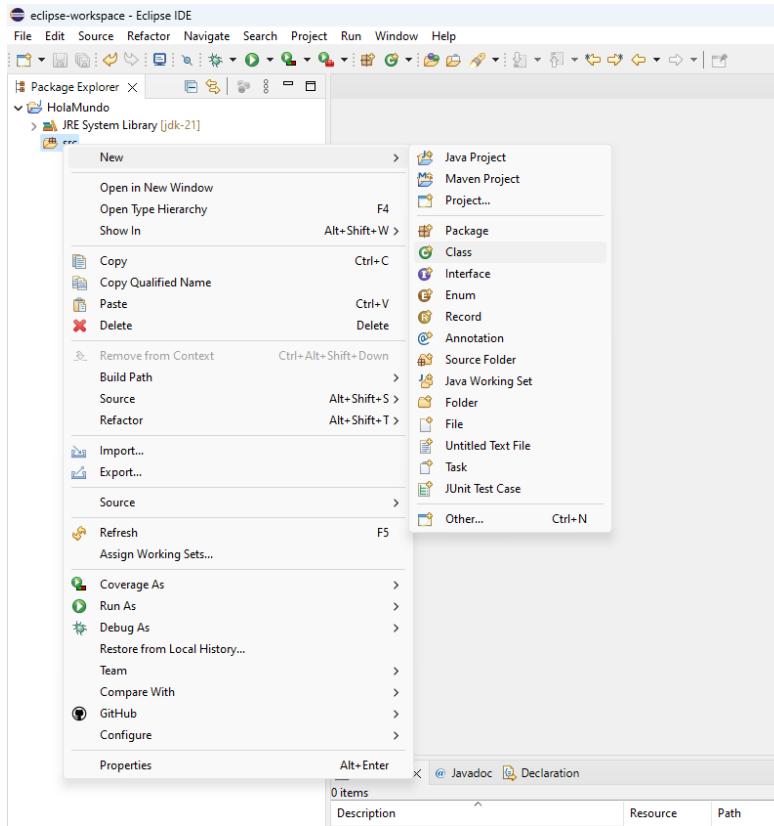


# Eclipse. Crear un nuevo proyecto



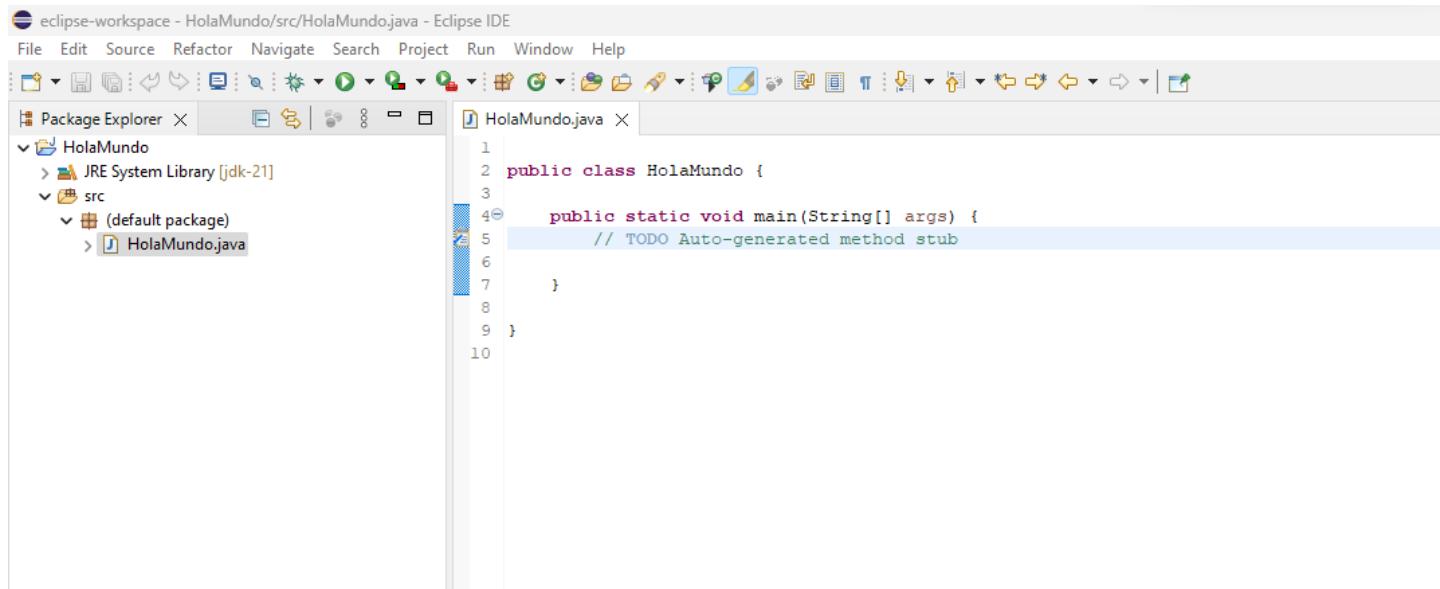
# Eclipse. Crear una nueva clase

- Para crear una nueva clase dentro del *package* por defecto, hay que situarse sobre el proyecto y pulsar el botón derecho  
New-> Java Class  
Se iniciará el asistente de creación de clases:



# Eclipse. Escribir el código

- Los comentarios se pueden introducir con “//”



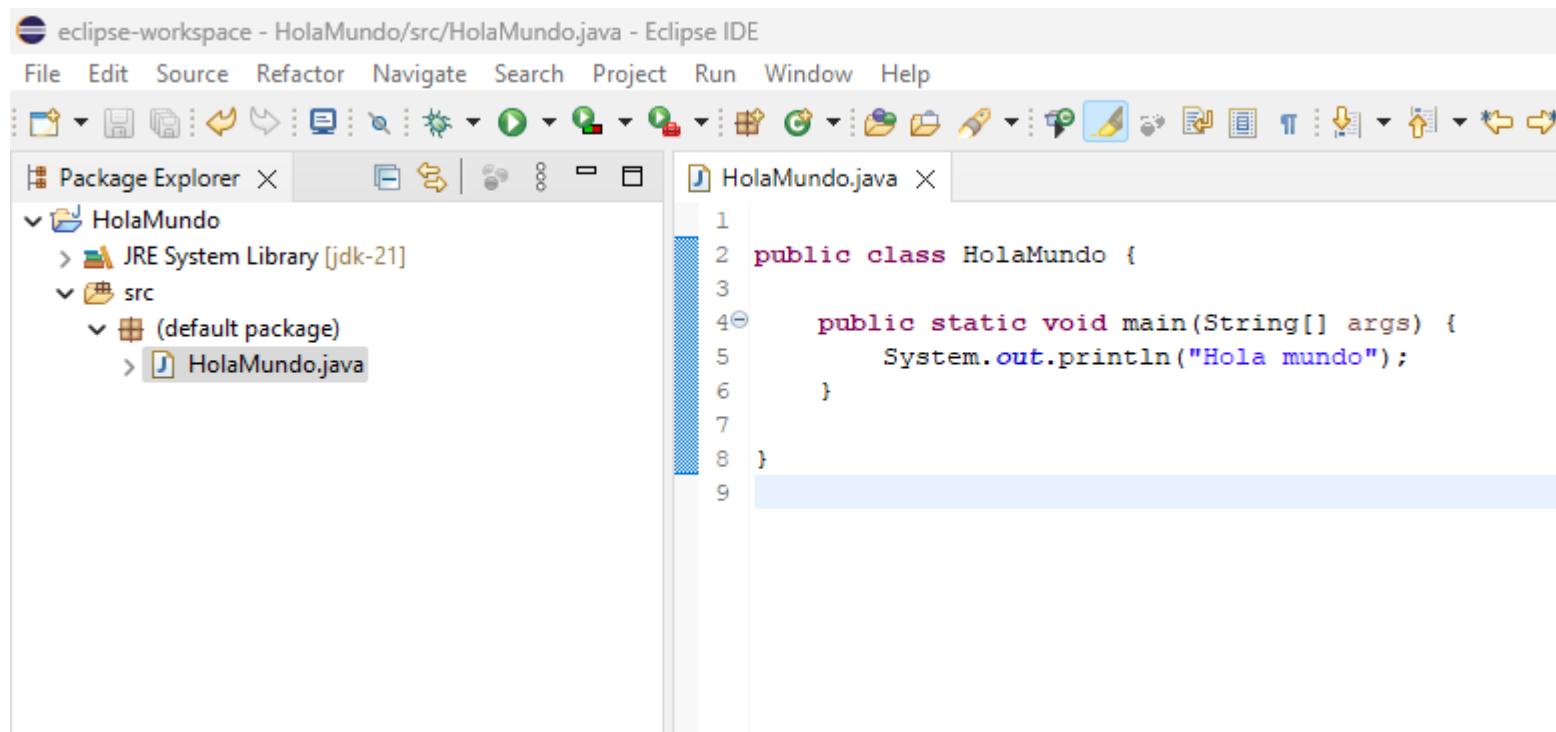
The screenshot shows the Eclipse IDE interface. The title bar reads "eclipse-workspace - HolaMundo/src/HolaMundo.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, and Cut. The Package Explorer view on the left shows a project named "HolaMundo" with a JRE System Library [jdk-21] and a src folder containing a default package with a file named "HolaMundo.java". The main view displays the code for "HolaMundo.java":

```
1
2 public class HolaMundo {
3
4     public static void main(String[] args) {
5         // TODO Auto-generated method stub
6
7     }
8
9 }
10
```

The line " // TODO Auto-generated method stub" is highlighted with a light blue background.

# Eclipse. Escribir el código

- El código a introducir sería el siguiente:

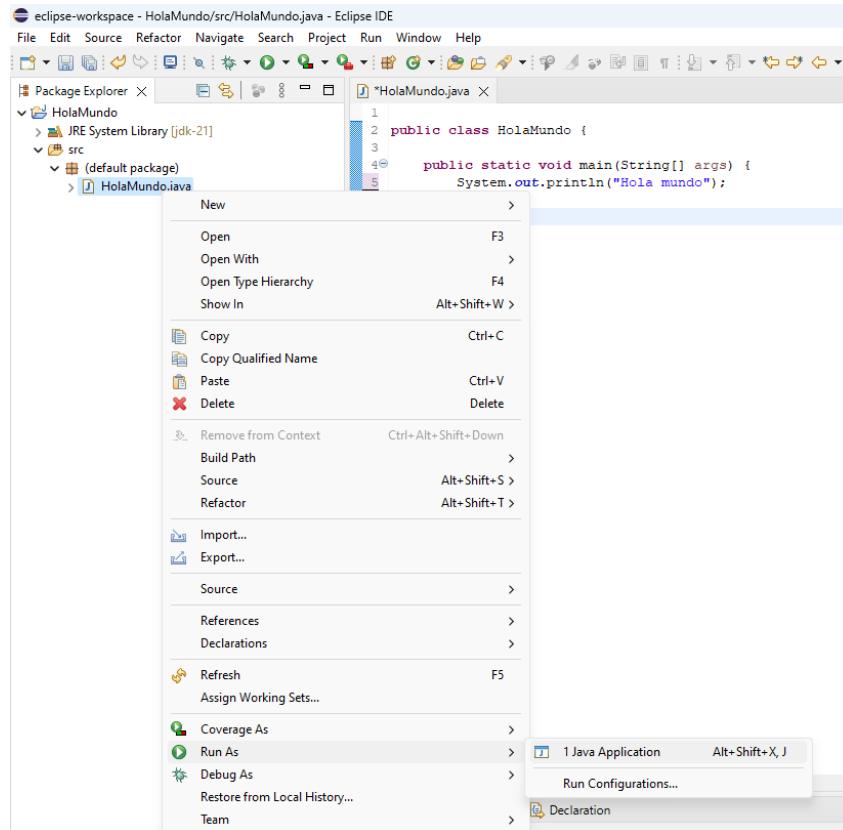


The screenshot shows the Eclipse IDE interface. The title bar reads "eclipse-workspace - HolaMundo/src/HolaMundo.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations like Open, Save, Copy, Paste, and Find. The Package Explorer view on the left shows a project named "HolaMundo" with a JRE System Library [jdk-21] and a src folder containing a (default package) with a file named "HolaMundo.java". The HolaMundo.java editor view on the right displays the following Java code:

```
1 public class HolaMundo {  
2     public static void main(String[] args) {  
3         System.out.println("Hola mundo");  
4     }  
5 }  
6  
7 }  
8  
9 }
```

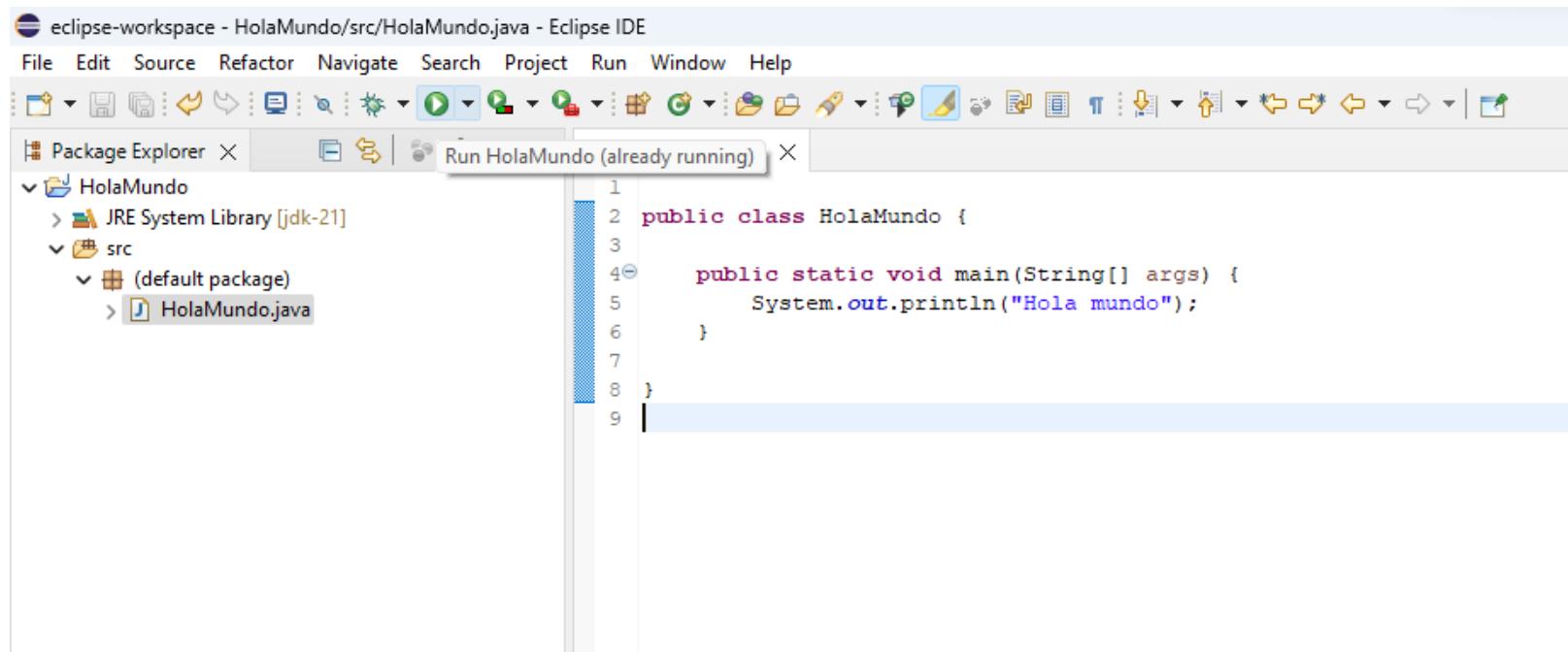
# Eclipse. Ejecutar1

- Para ejecutar la clase hay que situarse sobre ella y pulsar el botón derecho -> Run As -> Java App.



# Eclipse. Ejecutar2

- Una vez realizado lo anterior como mínimo una vez ya podemos lanzarlo desde el “PLAY” de la barra superior.



# Eclipse. Ejecutar3

- Siempre que ejecutamos una clase Eclipse la compila automáticamente
- La ventana "Console" nos muestra la ejecución.

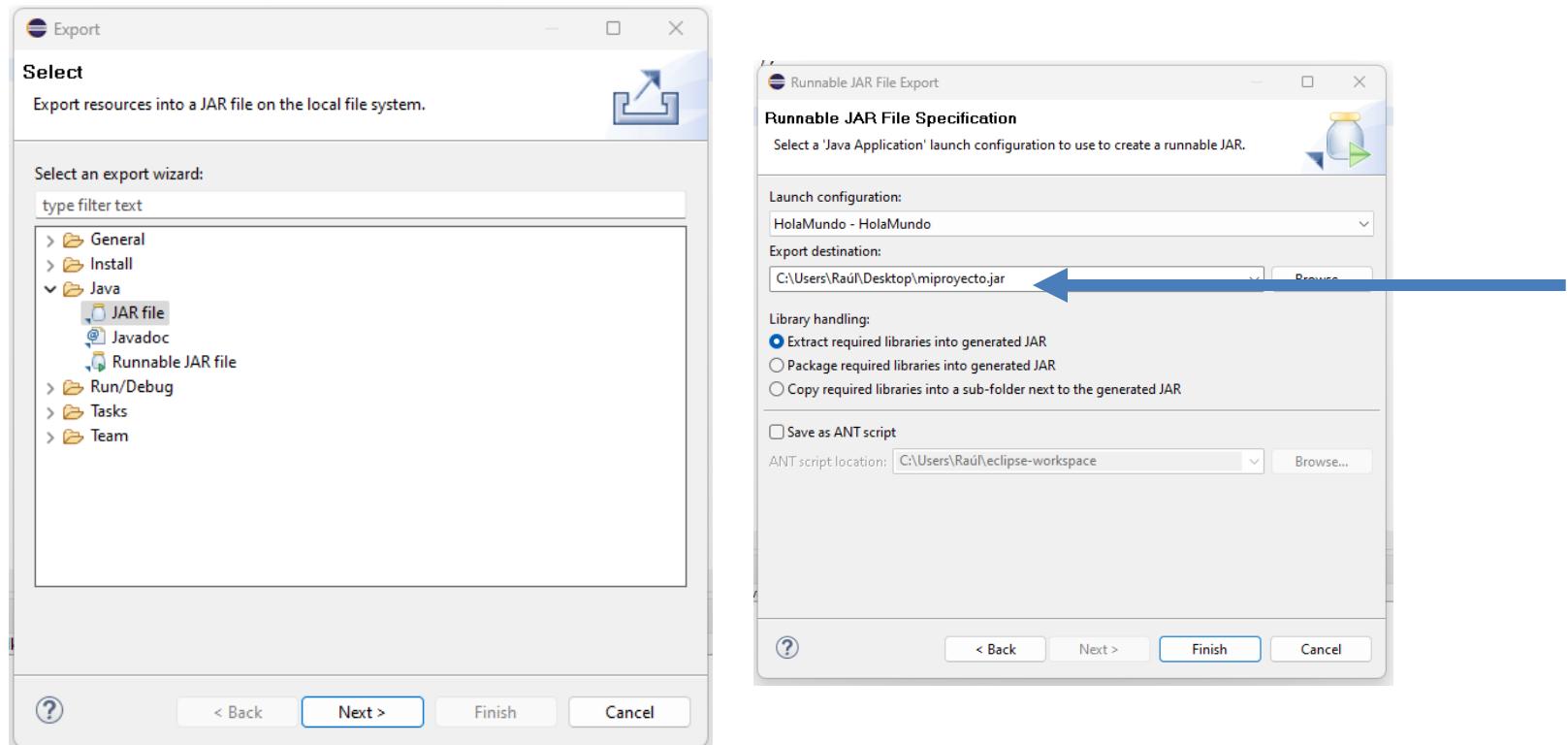
The screenshot shows the Eclipse IDE interface. The title bar reads "eclipse-workspace - HolaMundo/src/HolaMundo.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The toolbar has various icons for file operations. The left sidebar is the Package Explorer, showing a project named "HolaMundo" containing an "src" folder which in turn contains a "(default package)" folder with a file named "HolaMundo.java". The main editor window displays the Java code:

```
1 public class HolaMundo {  
2     public static void main(String[] args) {  
3         System.out.println("Hola mundo");  
4     }  
5 }  
6  
7 }  
8 }
```

The bottom right corner shows the Console view with the output: "terminated> HolaMundo [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (25 oct 2023, 1)  
Hola mundo".

# Crear ficheros JAR desde Eclipse

- Puedo crear un JAR del proyecto mediante botón derecho en el proyecto
  - > Export > Java > Runnable JAR file



# Paquetes en Java (package)

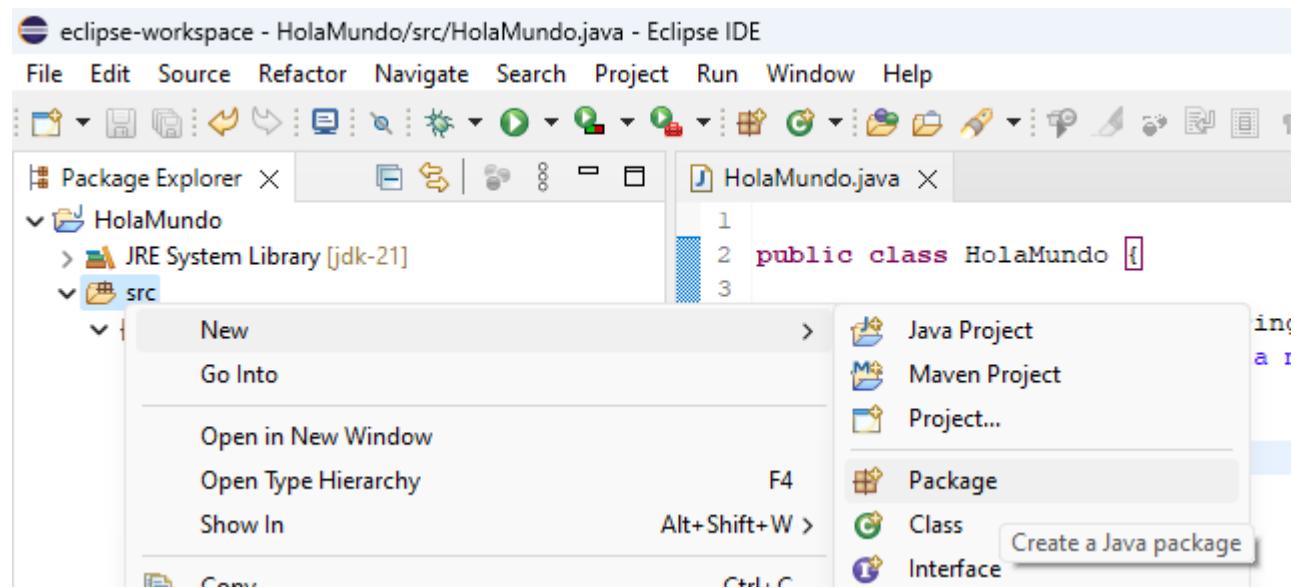
- Los paquetes son una forma de organizar grupos de clases
- Un paquete contiene un conjunto de clases relacionadas bien por finalidad, por ámbito o por herencia
- En el IDE Eclipse
  - un proyecto nuevo se crea en un subdirectorio que tiene el nombre del proyecto
  - Al crear packages dentro del proyecto creamos una estructura de carpetas dentro del proyecto

# Paquetes en Eclipse

- Cuando no se especifica el nombre del paquete al que pertenece una clase, esta pasa a pertenecer al “paquete por defecto”

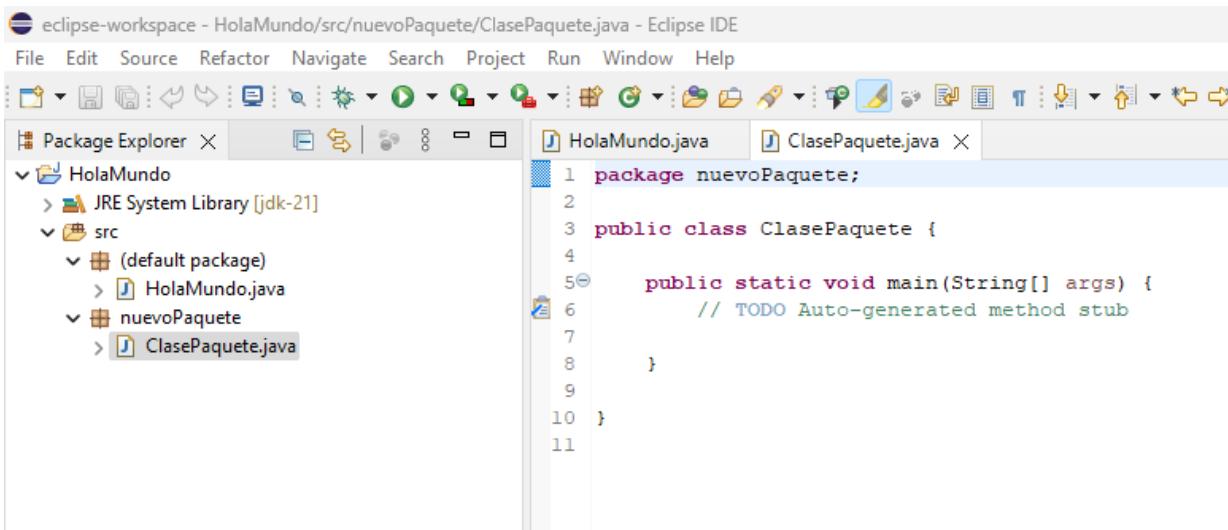
Aunque esto no genera errores de compilación es recomendable declarar paquetes para todos los componentes de nuestro programa Java

Con el botón derecho  
solicitamos un  
nuevo paquete.



# Paquetes en Eclipse

- Se crea la subcarpeta con el nombre del paquete
- Este paquete no tiene clases dentro. La creamos con botón derecho en el paquete.
- Vemos que la primera línea define que estamos dentro del paquete.

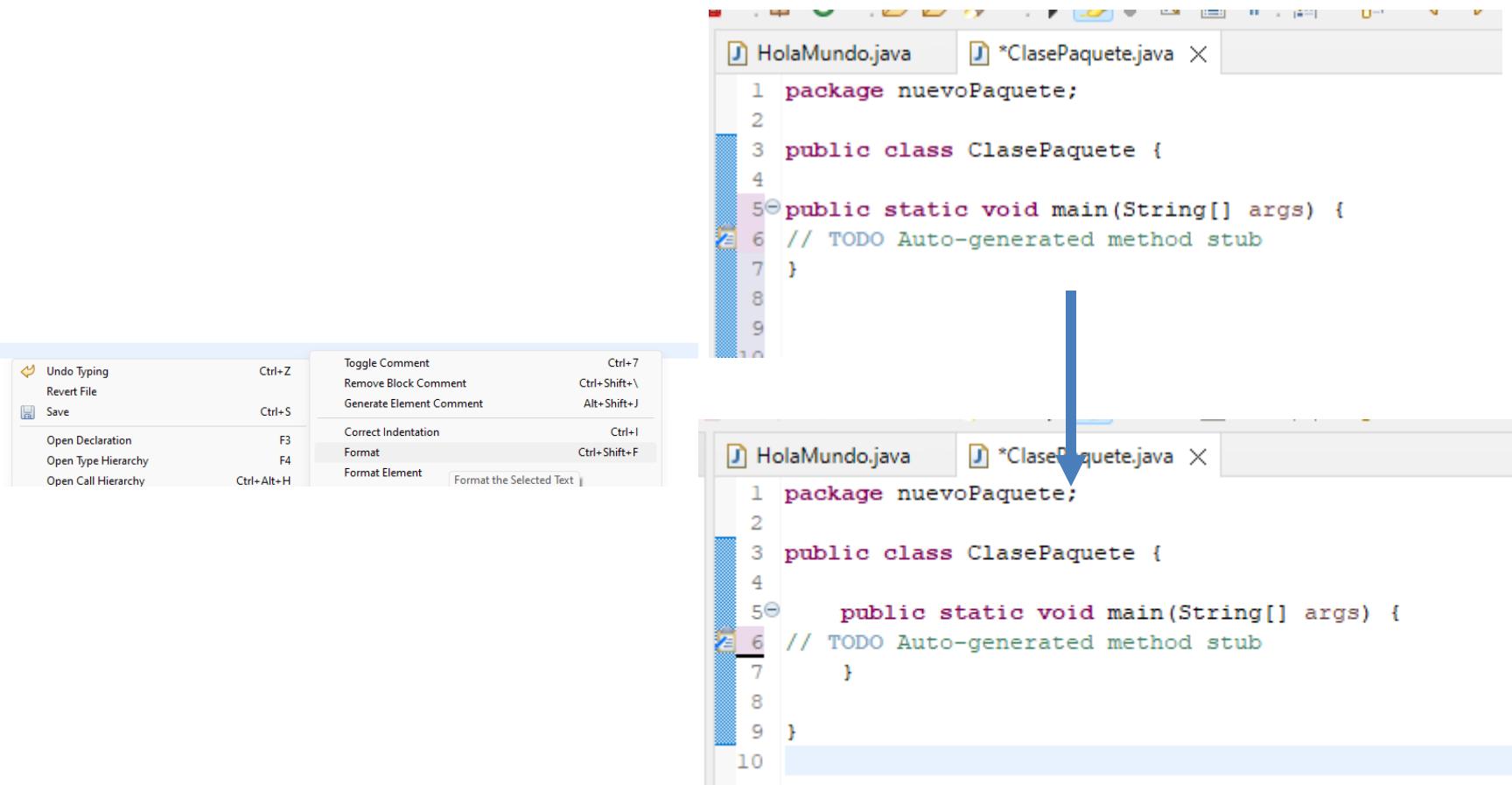


The screenshot shows the Eclipse IDE interface. The top bar displays "eclipse-workspace - HolaMundo/src/nuevoPaquete/ClasePaquete.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations. The left side features the Package Explorer view, which shows a project named "HolaMundo" with a "src" folder containing a "(default package)" folder with "HolaMundo.java" and a "nuevoPaquete" folder with "ClasePaquete.java". The right side shows two tabs: "HolaMundo.java" and "ClasePaquete.java". The code editor displays the following Java code:

```
1 package nuevoPaquete;
2
3 public class ClasePaquete {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7
8     }
9
10 }
```

# AutoFormato en Eclipse

- Netbeans nos maqueta los ficheros con corrección  
Botón derecho > Source > Format



# Teclado rápido en Eclipse

- Netbeans me ofrece combinaciones rápidas de teclado
- Puedo consultarlas en
  - Menu > Help > Show active keybindings...

Activate Editor	F12
Activate Task	Ctrl+F9
Add Block Comment	Ctrl+Shift+ /
Add Import	Ctrl+Shift+ M
Add Javadoc Comment	Alt+Shift+ J
Add Text Block	Ctrl+Shift+ '
All Instances	Ctrl+Shift+ N
Backward History	Alt+Left
Build All	Ctrl+B
Change Method Signature	Alt+Shift+ C
Close	Ctrl+F4
Close All	Ctrl+Shift+ F4
Collapse	Ctrl+Numpad_Subtract
Collapse All	Ctrl+Shift+Numpad_Divide
Commit...	Ctrl+#
Content Assist	Ctrl+Space
Context Information	Ctrl+Shift+Space
Copy	Ctrl+Insert
Copy Lines	Ctrl+Alt+Down
Correct Indentation	Ctrl+I
Coverage	Ctrl+Shift+F11
Coverage Eclipse Application	Alt+Shift+E, E
Coverage JUnit Plug-in Test	Alt+Shift+E, P
Coverage JUnit Test	Alt+Shift+E, T
Coverage Java Application	Alt+Shift+E, J
Coverage RAP JUnit Test	Alt+Shift+E, R

Entomo Eclipse.  
Depurar en Eclipse.

eclipse

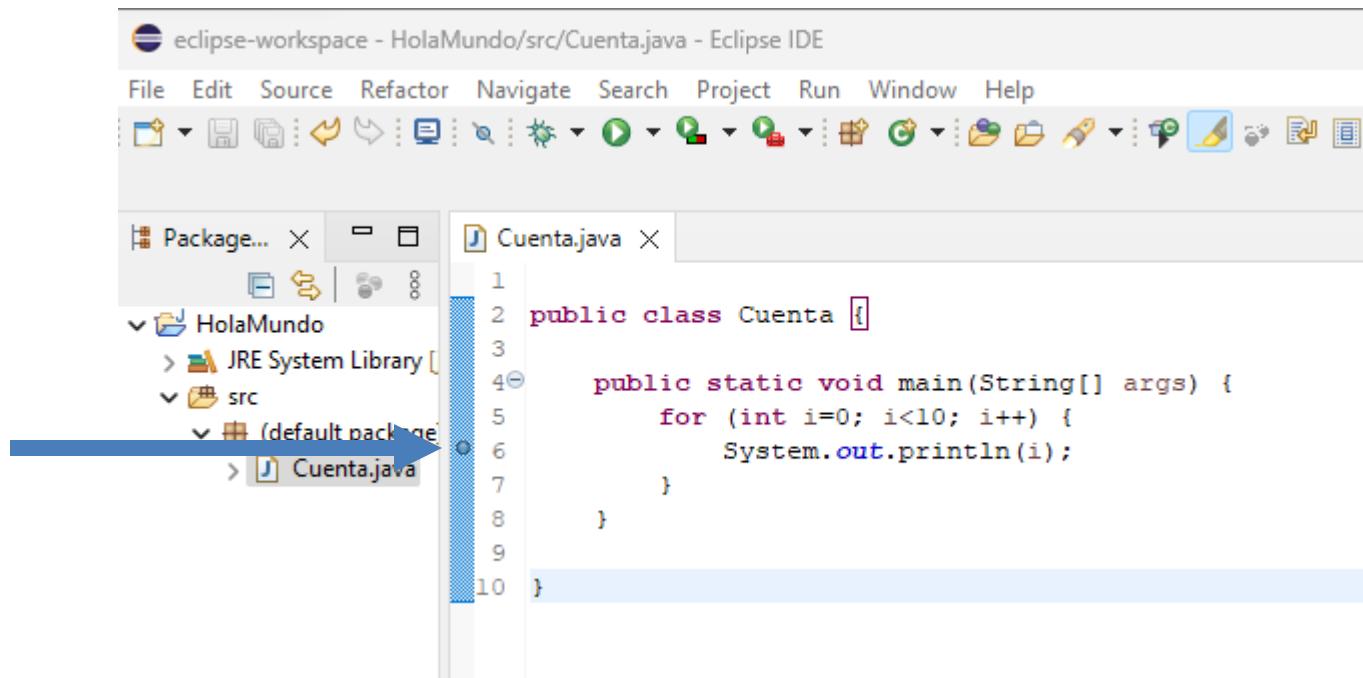


# Depurar en Eclipse

- Depurar nos permite ejecutar el programa parando en las líneas de código que queramos.
- Nos permite:
  - Encontrar errores.
  - Parar el flujo de ejecución.
  - Ver a tiempo real el estado de las variables.

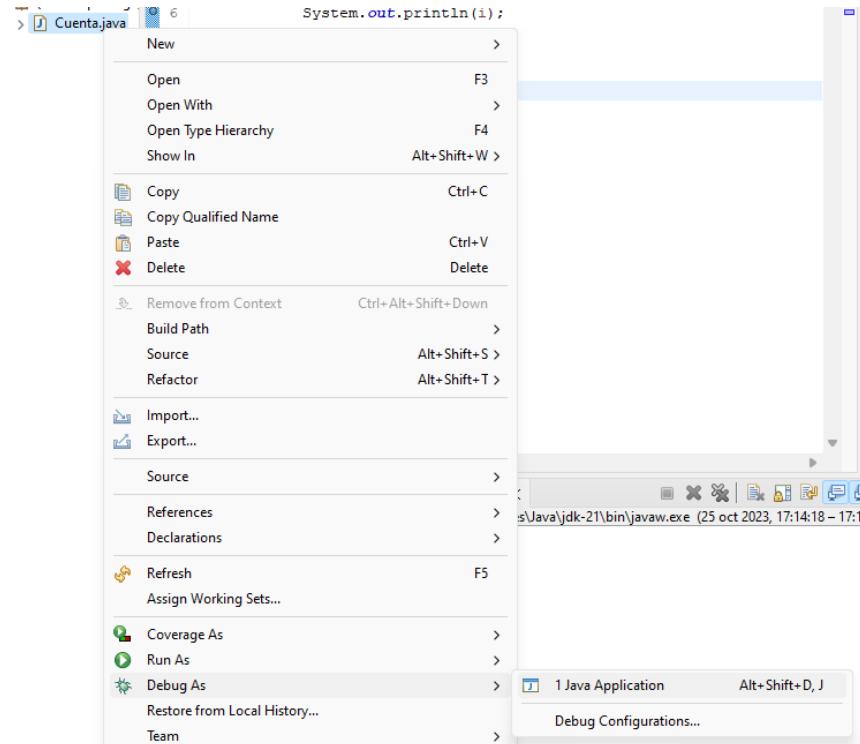
# Depurar en Eclipse

- Primero debemos establecer un “breakpoint”, que será el punto donde queremos que pare nuestro programa.
- Podemos establecerlo mediante doble click:



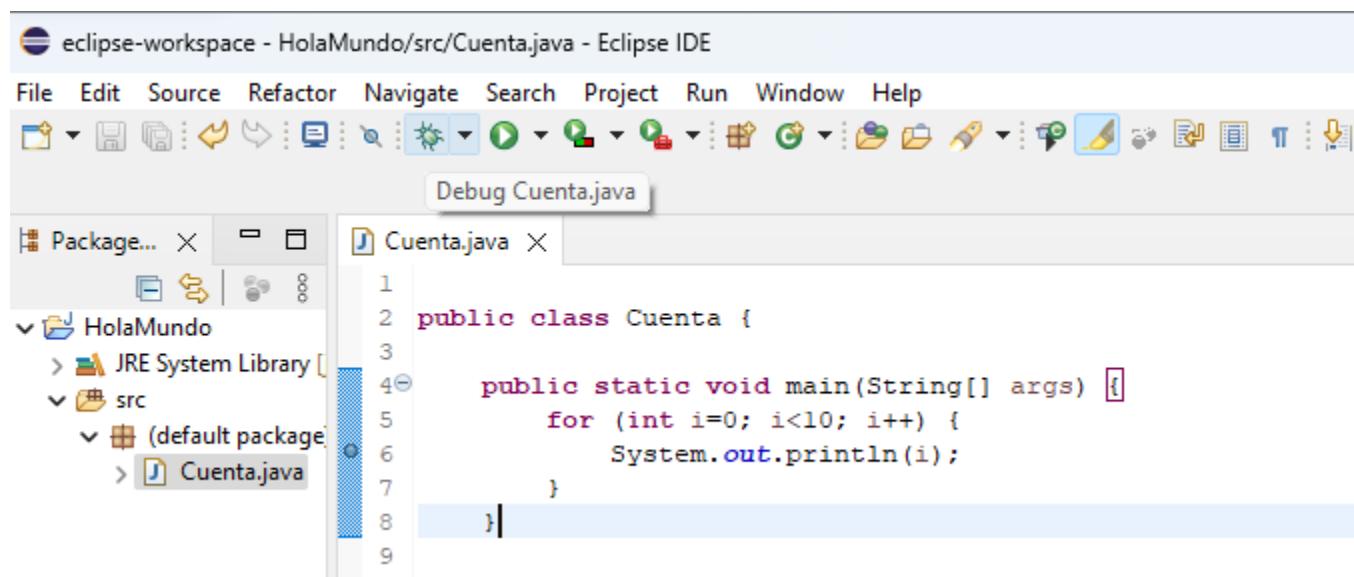
# Depurar en Eclipse

- Podemos acceder al depurado clicando en:
  - Botón derecho en la clase JAVA > Debug As > Java Application.



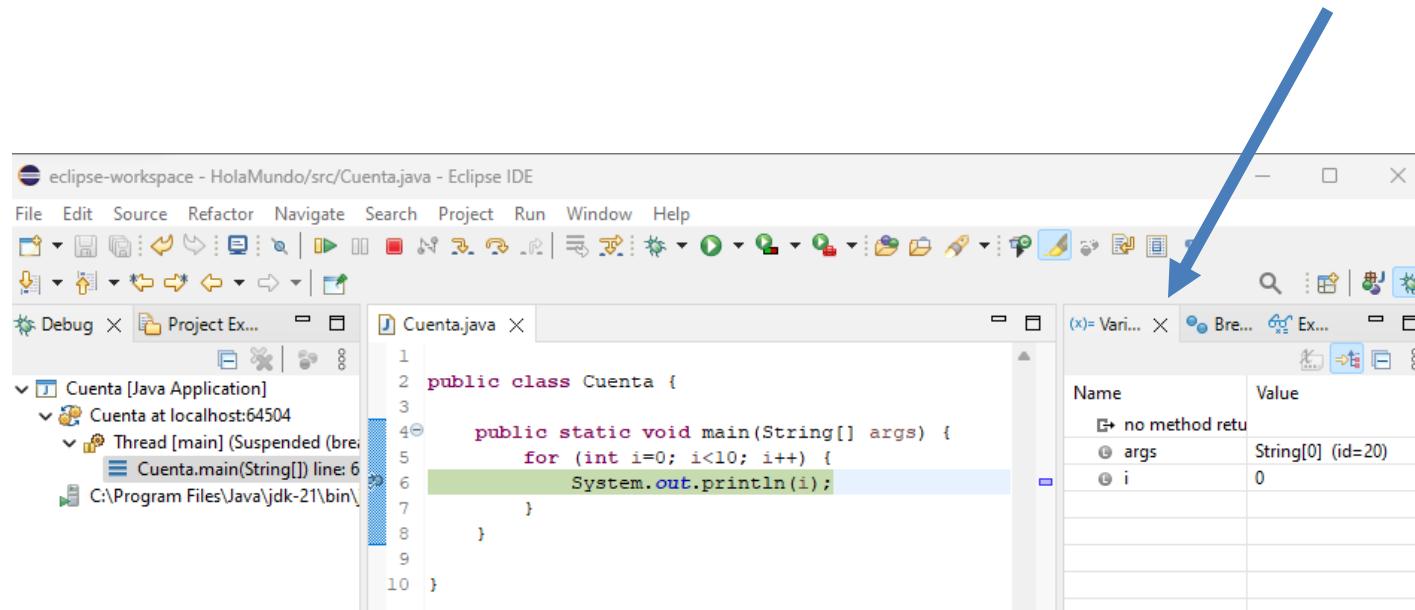
# Depurar en Eclipse

- Si queremos acceder al debugger de la última clase también podemos clicar en la “araña”.



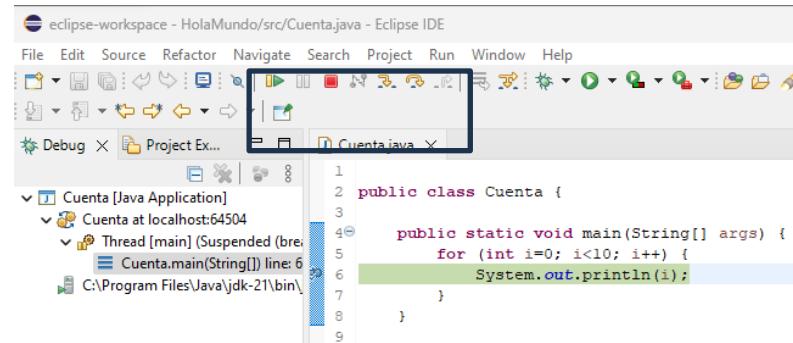
# Depurar en Eclipse

- El flujo para en la línea y podemos ver el estado actual de las variables:



# Depurar en Eclipse

- En la parte superior podemos ver una serie de botones:



The screenshot shows the Eclipse IDE interface. At the top, there's a toolbar with various icons. Below it is a menu bar with File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. The main area has a 'Debug' tab selected. On the left, there's a project tree showing 'Cuenta [Java Application]' and a thread list with 'Thread [main] (Suspended (breakpoint))'. In the center, there's a code editor with the following Java code:

```
1 public class Cuenta {  
2     public static void main(String[] args) {  
3         for (int i=0; i<10; i++) {  
4             System.out.println(i);  
5         }  
6     }  
7 }  
8  
9
```

A green line highlights the line 'System.out.println(i);' in the code editor, indicating it is the current line being debugged.



- Permite continuar hasta el siguiente breakpoint.



- Para el flujo de ejecución y sale del programa.



- Step Into: entra a la función interior.



- Step Over: salta a la siguiente línea del código.

# Depurar en Eclipse

- Puedo establecer condiciones dentro de un breakpoint.
  - Botón derecho breakpoint > Breakpoint properties.
- En este caso parará cuando imprima el 7.

