



Introducción al lenguaje Java

Origen y características

- Creado en 1995 por Sun Microsystems
- Java, en argot americano, es una taza de café (de ahí el logo)
- Pertenece al modelo de programación orientada a objetos, que permite representar elementos del mundo real en el código mediante el uso de clases.
- Utiliza el **Garbage Collector**, un programa interno que libera la memoria de los objetos que dejan de estar referenciados para que no sea necesario destruirlos al dejar de usarlos.
- Es un lenguaje interpretado, que puede ejecutarse en cualquier ordenador independientemente de hardware y SO siempre que tenga instalada la **Java Virtual Machine (JVM)**.





Origen y características

Uno de los lenguajes de programación más utilizados actualmente

(Índice de Redmonk en Junio de 2018, según usos en GitHub y consultas en StackOverflow)

- 1 JavaScript
- 2 Java
- 3 Python
- 4 PHP
- 5 C++
- 5 C#
- 7 CSS
- 8 Ruby
- 9 C
- 10 TypeScript
- 11 Swift
- 12 Objective-C
- 13 Scala
- 14 Shell
- 15 R
- 16 Go
- 17 PowerShell
- 18 Perl
- 19 Haskell
- 20 Kotlin



Origen y características

Uno de los lenguajes de programación más utilizados actualmente

(Índice de Redmonk en Enero de 2024, según usos en GitHub y consultas en StackOverflow)

- 1 JavaScript
- 2 Python
- 3 Java
- 4 PHP
- 5 C#
- 6 TypeScript
- 6 CSS
- 8 C++
- 9 Ruby
- 10 C
- 11 Swift
- 12 Go
- 12 R
- 14 Shell
- 14 Objective-C
- 16 Scala
- 17 Kotlin
- 18 PowerShell
- 19 Rust
- 19 Dart



La Máquina Virtual de Java (JVM)

A diferencia que en otros lenguajes como C, donde en la compilación del código se obtiene un programa ejecutable para un S.O. específico creado en lenguaje máquina, en Java la compilación del código traduce el código a un lenguaje intermedio llamado **ByteCode**.

En el S.O. donde se quiere ejecutar el código es necesario instalar la **Máquina Virtual**, que será la que se encargue de ejecutar el código.

La JVM toma el código en ByteCode y lo compila a su vez en código máquina nativo del S.O. en la que se está ejecutando. Por esto un programa en Java puede ser ejecutado en cualquier S.O. que tenga una implementación de la JVM.

Java Virtual Machine

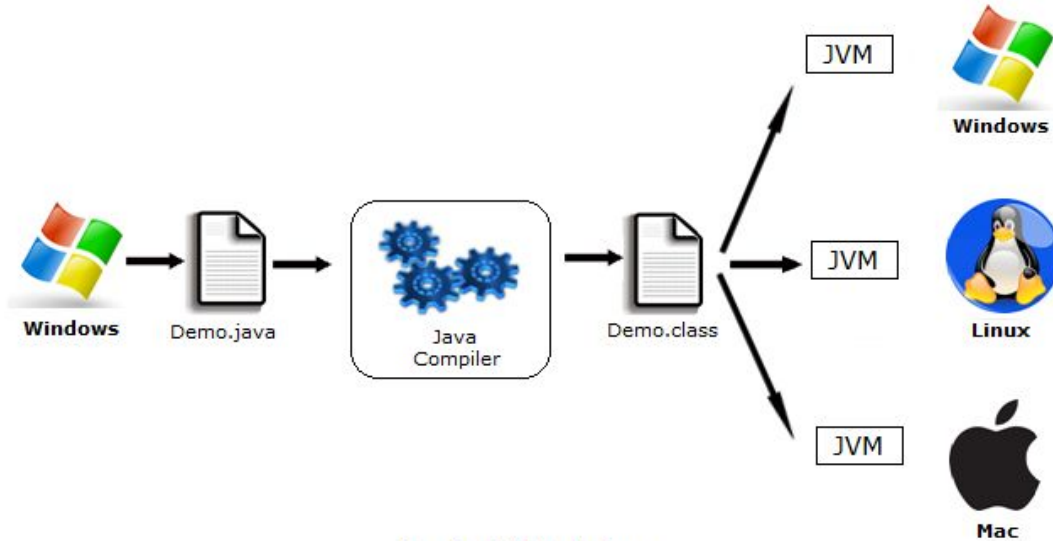


fig : Portability in Java



Java Runtime Environment y Java Development Kit

El entorno de ejecución de Java (JRE) contiene los elementos necesarios para la ejecución de Java en el S.O.





El entorno de desarrollo de Java (JDK) contiene, además del JRE, los elementos necesarios para el desarrollo en Java (el compilador, Javadoc para crear documentación, ...)

Si sólo instalamos el JRE podremos ejecutar Java pero no compilarlo con lo que en entornos de desarrollo se instala el JDK y en despliegues de producción (al cliente) se instala el JRE.

Instalación de Java

Paso 1: Comprobar la instalación de Java en el equipo.

Buscar en el menú “Agregar y quitar programas” si existe una versión instalada de Java
(Panel de Control -> Programas y características para Windows 10)

Organizar ▼				
Nombre	Editor	Se instaló el	Tamaño	Versión
 Java 8 Update 191	Oracle Corporation	20/11/2018	102 MB	8.0.1910.12
 Java 8 Update 191 (64-bit)	Oracle Corporation	20/11/2018	118 MB	8.0.1910.12
 Java SE Development Kit 8 Update 191	Oracle Corporation	20/11/2018	335 MB	8.0.1910.12
 Java SE Development Kit 8 Update 191 (64-bit)	Oracle Corporation	20/11/2018	353 MB	8.0.1910.12

Instalación de Java

Paso 2: Instalar el JDK de Java 8

Buscar en Google JDK 8

Instalar el JDK de la versión de Windows

Java SE Development Kit 8 - Downloads - Oracle





<https://www.oracle.com/.../java/javase/.../jdk8-downloads-213315..>

Java SE Development Kit 8u191		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.97 MB	jdk-8u191-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.92 MB	jdk-8u191-linux-arm64-vfp-hflt.tar.gz
Linux x86	170.89 MB	jdk-8u191-linux-i586.rpm
Linux x86	185.69 MB	jdk-8u191-linux-i586.tar.gz
Linux x64	167.99 MB	jdk-8u191-linux-x64.rpm
Linux x64	182.87 MB	jdk-8u191-linux-x64.tar.gz
Mac OS X x64	245.92 MB	jdk-8u191-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	133.04 MB	jdk-8u191-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.28 MB	jdk-8u191-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	134.04 MB	jdk-8u191-solaris-x64.tar.Z
Solaris x64	92.13 MB	jdk-8u191-solaris-x64.tar.gz
Windows x86	197.34 MB	jdk-8u191-windows-i586.exe
Windows x64	207.22 MB	jdk-8u191-windows-x64.exe

Instalación de Java

Paso 3: Comprobación de archivos de instalación

En el directorio de instalación de Java (C:\Program Files (x86)\Java\jdk1.8.0_191\bin\ tendremos que ver los ficheros del paquete de desarrollo.

 java.exe	30/07/2018 12:56	Aplicación	227 KB
 javac.exe	30/07/2018 12:56	Aplicación	18 KB
 javacpl.exe	30/07/2018 12:56	Aplicación	86 KB
 javadoc.exe	30/07/2018 12:56	Aplicación	18 KB

Los que nos interesan son los ejecutables *java.exe* y *javac.exe*



Configuración de variables de entorno

Para poder acceder a los ejecutables javac.exe y java.exe desde cualquier directorio modificaremos la variable PATH para añadir la ubicación de dichos ejecutables.

Para acceder a ellas: Segundo botón encima de “Este Equipo” -> propiedades -> “Configuración avanzada del sistema” -> “Variables de entorno”

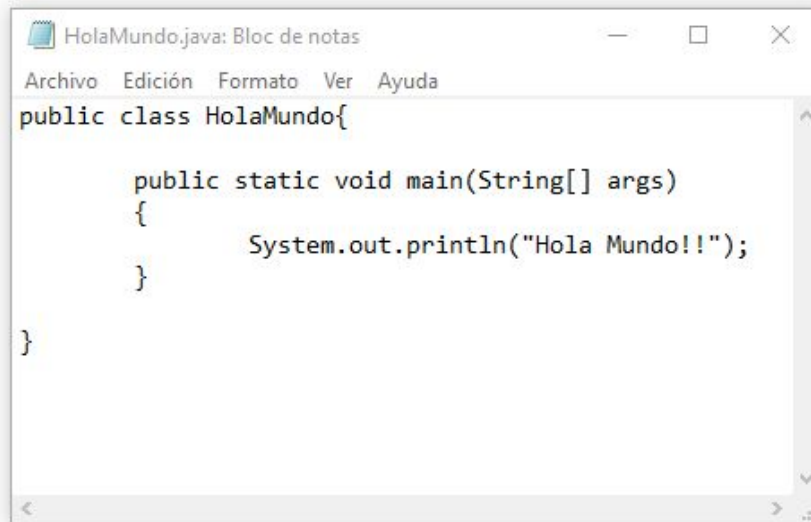
En la parte inferior (Variables de usuario) -> Editar la variable Path y añadir una línea con la ruta del JDK



C:\Program Files (x86)\Java\jdk1.8.0_191\bin

Compilación y Ejecución

Paso 1: Creación de un programa sencillo (Notepad o Notepad ++)

A screenshot of a Notepad window titled "HolaMundo.java: Bloc de notas". The window has a menu bar with "Archivo", "Edición", "Formato", "Ver", and "Ayuda". The text area contains the following Java code:

```
public class HolaMundo{  
  
    public static void main(String[] args)  
    {  
        System.out.println("Hola Mundo!!");  
    }  
}
```

The code is formatted with indentation for the main method. The window has standard Windows window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.



Compilación y Ejecución

Paso 2: Compilación

Abrimos la consola de comandos y nos situamos en la ubicación del fichero HolaMundo.java

Compilamos el archivo java mediante el comando **javac** seguido del nombre del fichero con la extensión

```
C:\Ejercicio_Java>javac HolaMundo.java  
C:\Ejercicio_Java>
```

Tras la compilación veremos un fichero HolaMundo.class en el directorio donde está el fichero HolaMundo.java



Compilación y Ejecución

Paso 3: Ejecución

Desde el directorio donde se encuentran el fichero `HolaMundo.java` y `HolaMundo.class` ejecutaremos el programa mediante el comando ***java*** seguido del nombre de la clase sin extensión, viendo por consola el mensaje esperado.

```
C:\Ejercicio_Java>java HolaMundo
Hola Mundo!!

C:\Ejercicio_Java>
```



Eclipse

Eclipse es un IDE (Entorno de Desarrollo Integrado) que facilita el desarrollo de aplicaciones en Java pudiendo gestionar la estructura del proyecto, agregar y moverse en los diversos ficheros que lo componen o pudiendo depurar los errores que se produzcan. Gracias a los IDE se reduce el tiempo necesario para el desarrollo de las aplicaciones cuando alcanzan un cierto tamaño.

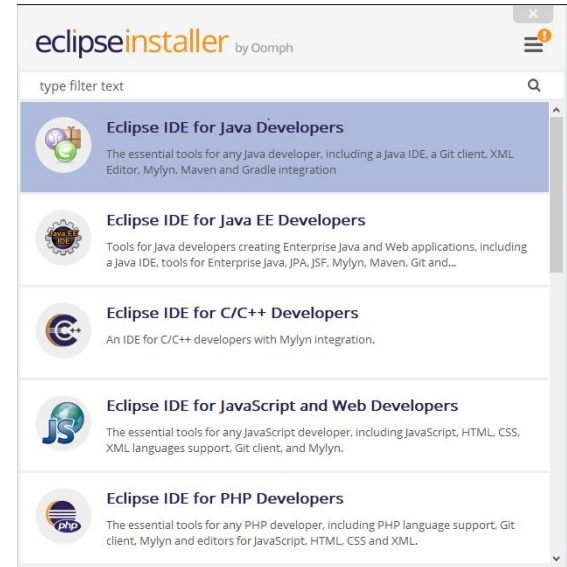
Además del desarrollo en Java, Eclipse permite el desarrollo en otros lenguajes (PHP, Javascript, HTML,...)

Alternativas al uso de Eclipse pueden ser Netbeans, IntelliJ IDEA o JCreator.

Instalación de Eclipse

Paso 1: Descargar el IDE de <https://www.eclipse.org/>

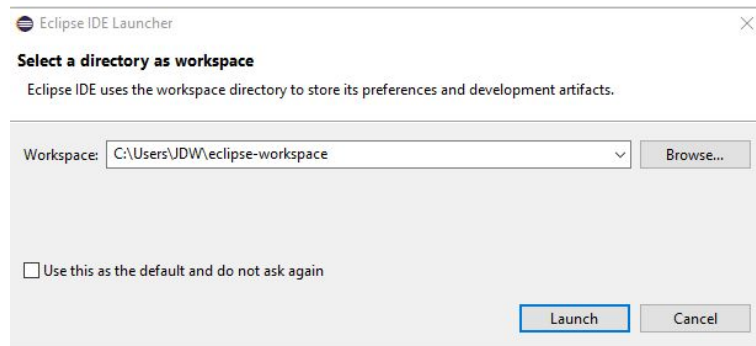
Paso 2: Realizar la instalación para el entorno de desarrollo JAVA



Instalación de Eclipse

Paso 3: Arrancar Eclipse

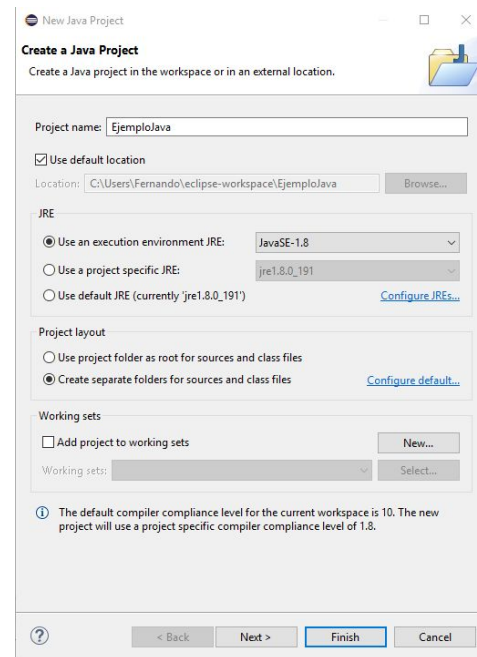
Nos pedirá que elijamos el Workspace, que será el directorio de trabajo donde se guardarán por los proyectos que creemos.



Ejemplo HolaMundo en Eclipse

Con Eclipse abierto crear un nuevo proyecto java:

File → New → Java Project










Ejemplo HolaMundo en Eclipse

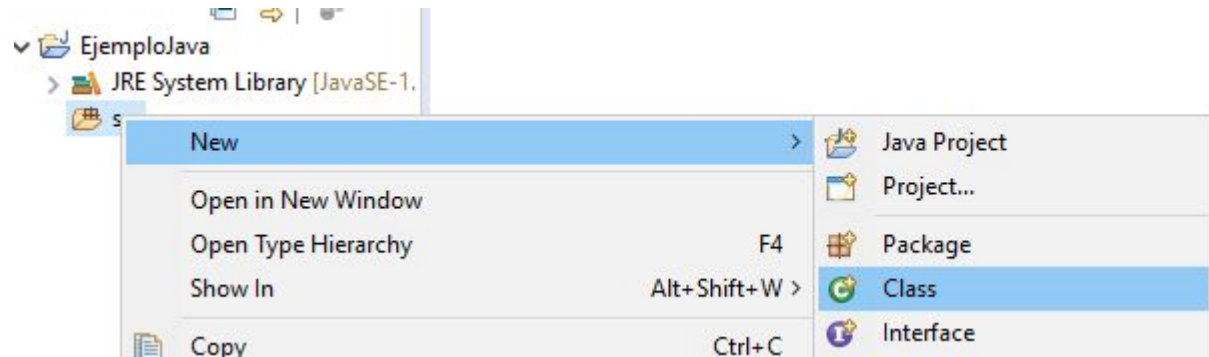
En el directorio del workspace nos habrá creado un directorio con el nombre dado y una estructura de subcarpetas en las que destacan dos:

- src: En donde se guardan los ficheros .java
- bin: En donde se generan los ficheros .class

Nombre	Fecha de modifica...	Tipo	Tamaño
 .settings	20/11/2018 21:05	Carpeta de archivos	
 bin	20/11/2018 21:05	Carpeta de archivos	
 src	20/11/2018 21:05	Carpeta de archivos	
 .classpath	20/11/2018 21:05	Archivo CLASSPA...	1 KB
 .project	20/11/2018 21:05	Archivo PROJECT	1 KB

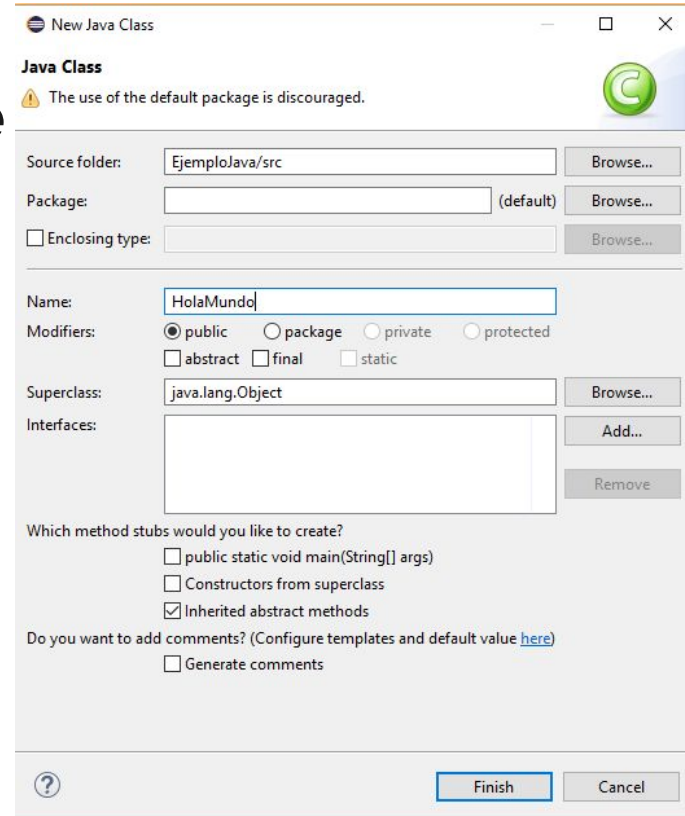
Ejemplo HolaMundo en Eclipse

Sobre el directorio src (en Eclipse) pulsaremos el segundo botón → New → Class



Ejemplo HolaMundo en Eclipse

En la ventana de creación de esta clase pondremos el nombre HolaMundo



The screenshot shows the 'New Java Class' dialog box in the Eclipse IDE. The dialog is titled 'New Java Class' and has a warning icon and text: 'The use of the default package is discouraged.' The 'Source folder' is set to 'EjemploJava/src'. The 'Package' is set to '(default)'. The 'Enclosing type' is unchecked. The 'Name' is 'HolaMundo'. The 'Modifiers' are 'public' (selected), 'package', 'private', 'protected', 'abstract', 'final', and 'static'. The 'Superclass' is 'java.lang.Object'. The 'Interfaces' list is empty. The 'Which method stubs would you like to create?' section has 'public static void main(String[] args)', 'Constructors from superclass', and 'Inherited abstract methods' (checked). The 'Do you want to add comments?' section has 'Generate comments' (unchecked). The 'Finish' button is highlighted.

New Java Class

Java Class

The use of the default package is discouraged.

Source folder: EjemploJava/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: HolaMundo

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

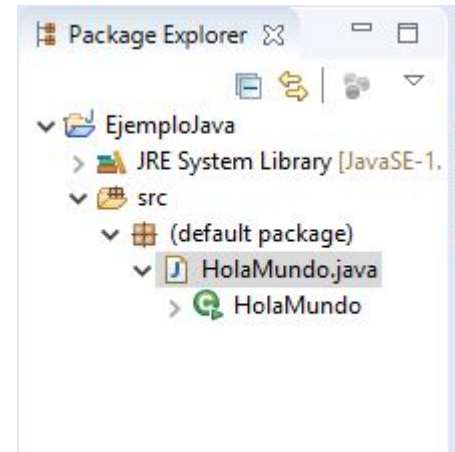
☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

Finish Cancel

Ejemplo HolaMundo en Eclipse

Tras la creación se genera un fichero .java con la estructura siguiente en el explorador de paquetes





Ejemplo HolaMundo en Eclipse

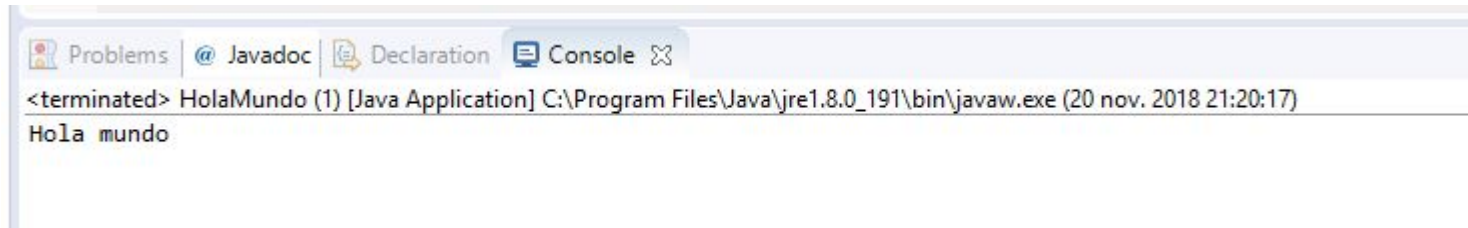
En la parte de visualización de la clase modificaremos el código con los datos del HolaMundo

```
public class HolaMundo {  
    public static void main(String[] args)  
    {  
        System.out.println("Hola mundo");  
    }  
}
```

Ejemplo HolaMundo en Eclipse

Para ejecutar el código pulsaremos con el segundo botón sobre la clase HolaMundo.java
→ Run As → Java Application

En la parte inferior de Eclipse, en la ventana Console, veremos el resultado de la ejecución del programa



The screenshot shows the Eclipse IDE's Console window. The title bar includes tabs for 'Problems', 'Javadoc', 'Declaration', and 'Console'. The 'Console' tab is active, displaying the output of a Java application. The output text is: '<terminated> HolaMundo (1) [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (20 nov. 2018 21:20:17)' followed by 'Hola mundo' on a new line.

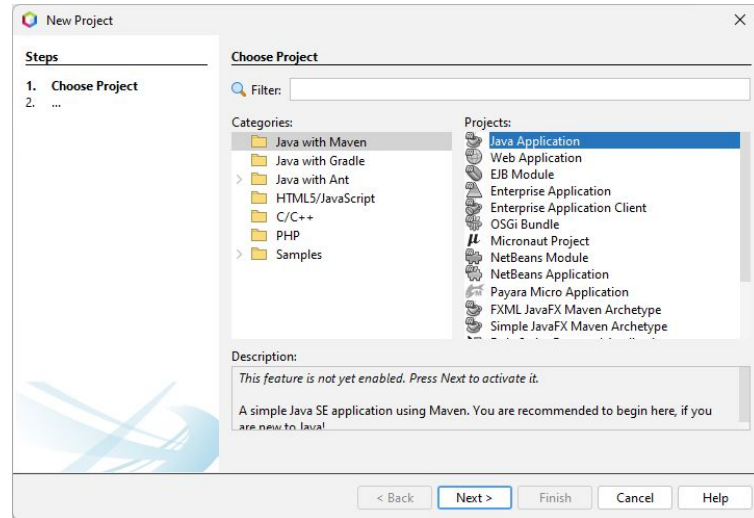
```
<terminated> HolaMundo (1) [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (20 nov. 2018 21:20:17)
Hola mundo
```


Ejemplo HolaMundo en NetBeans

Para crear un nuevo proyecto:

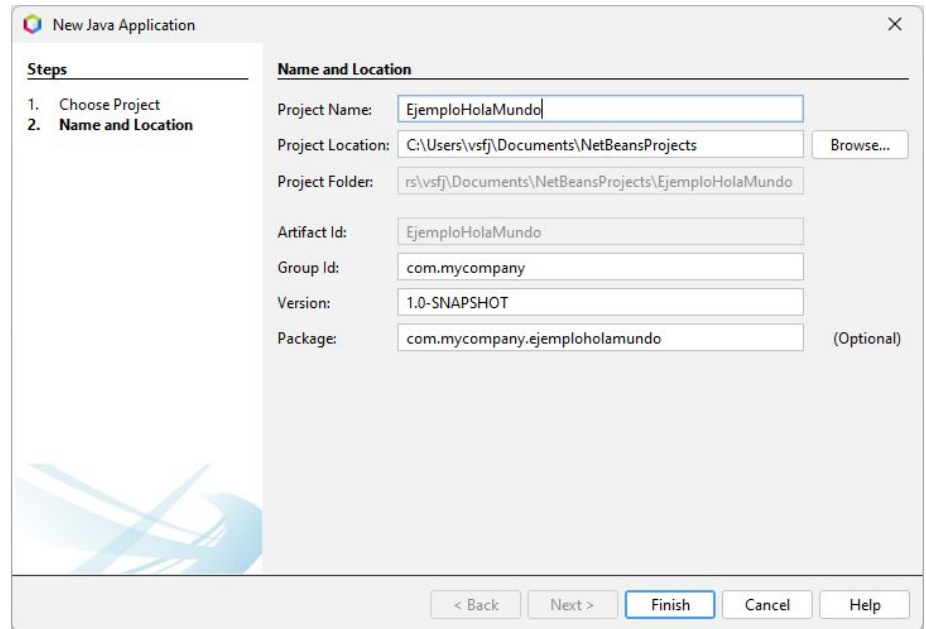
File->New Project

Escogemos Java Application de Maven



Ejemplo HolaMundo en NetBeans

Escogemos el nombre y la
localización del proyecto

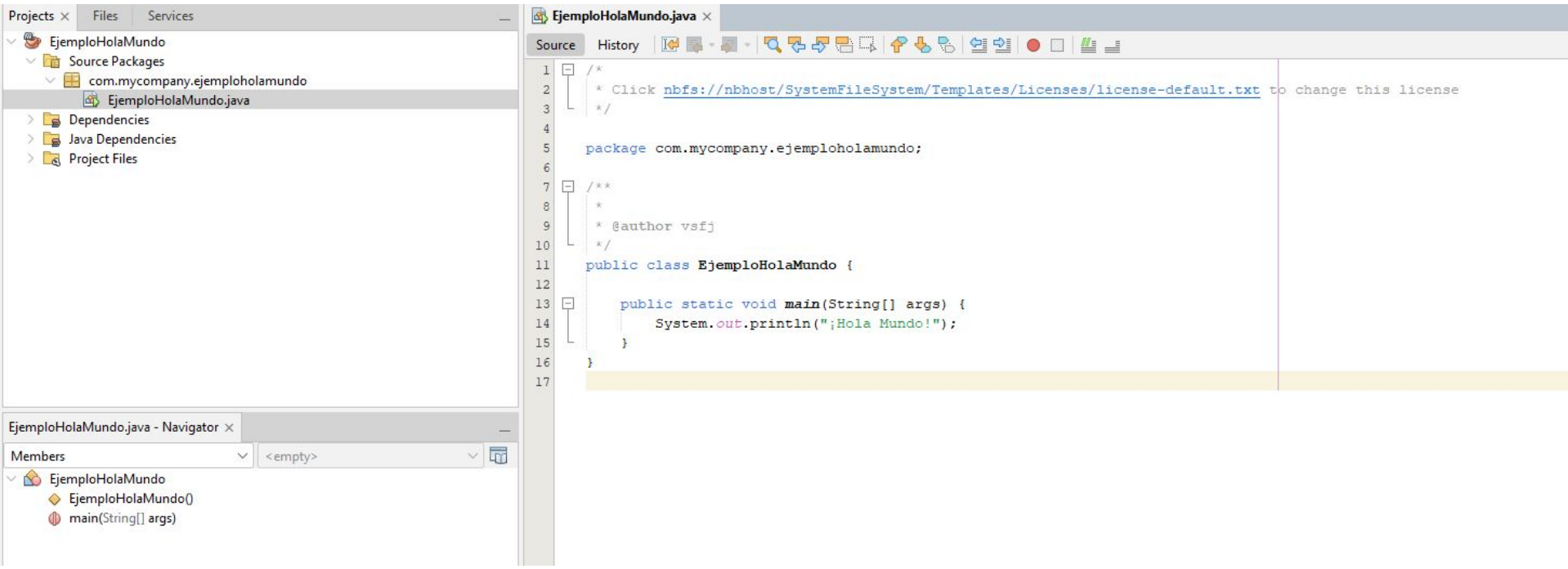


The screenshot shows the 'New Java Application' dialog box in NetBeans. The 'Steps' panel on the left indicates that step 2, 'Name and Location', is the current step. The main panel contains several input fields for project configuration:

- Project Name:** EjemploHolaMundo
- Project Location:** C:\Users\vsfj\Documents\NetBeansProjects (with a 'Browse...' button)
- Project Folder:** rs\vsfj\Documents\NetBeansProjects\EjemploHolaMundo
- Artifact Id:** EjemploHolaMundo
- Group Id:** com.mycompany
- Version:** 1.0-SNAPSHOT
- Package:** com.mycompany.ejemploholamundo (Optional)

At the bottom of the dialog, there are five buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue border), 'Cancel', and 'Help'.

Ejemplo HolaMundo en NetBeans



The screenshot displays the NetBeans IDE interface. On the left, the Project Explorer shows a project named 'EjemploHolaMundo' with a source package 'com.mycompany.ejemploholamundo' containing the file 'EjemploHolaMundo.java'. Below this, the 'EjemploHolaMundo.java - Navigator' pane shows the class 'EjemploHolaMundo' and its 'main(String[] args)' method. The main editor window shows the source code of 'EjemploHolaMundo.java' with the following content:

```
1  /*  
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license  
3   */  
4  
5   package com.mycompany.ejemploholamundo;  
6  
7   /**  
8    *  
9    * @author vsfj  
10   */  
11  public class EjemploHolaMundo {  
12  
13      public static void main(String[] args) {  
14          System.out.println("¡Hola Mundo!");  
15      }  
16  }  
17
```

Ejemplo HolaMundo en NetBeans

En la parte superior tenemos los botones para crear el proyecto, limpiar y construir, ejecutar (normal y paso a paso)...



Para ejecutar pulsamos el botón de 'play'



Ejemplo HolaMundo en NetBeans

En la parte inferior nos mostrará la ejecución correcta en la ventana de output

