

# PROGRAMACIÓN ORIENTADA A OBJETOS (Laboratorio de Prácticas)

Titulaciones de Grado en Ingeniería Informática, Grado en Ingeniería en Sistemas de Información y Grado en Ingeniería de Computadores

## Sesión 1 INTRODUCCIÓN AL DESARROLLO EN JAVA

### 1. Introducción al JDK

El entorno de desarrollo básico para trabajar con JAVA consiste únicamente en un conjunto de compiladores, utilidades y documentos llamado JDK (Java Development Kit).

Oracle proporciona los JDK para diversos sistemas operativos como Windows, Linux, Solaris, etc. y las directrices para que el resto de fabricantes puedan proporcionar sus propios JDK.



El JDK es un conjunto de herramientas en modo carácter, no es un entorno visual. Sin embargo, tomando como base las herramientas del JDK, existen varios entornos de desarrollo que facilitan la labor de programar en Java, entre los más importantes nos encontramos con NetBeans y Eclipse.

La versión oficial del JDK que está actualmente en uso es la 17 (llamada Java SE 17). Estas versiones del JDK se engloban en lo que se conoce como Java Standard Edition, que partieron de la versión 1.2

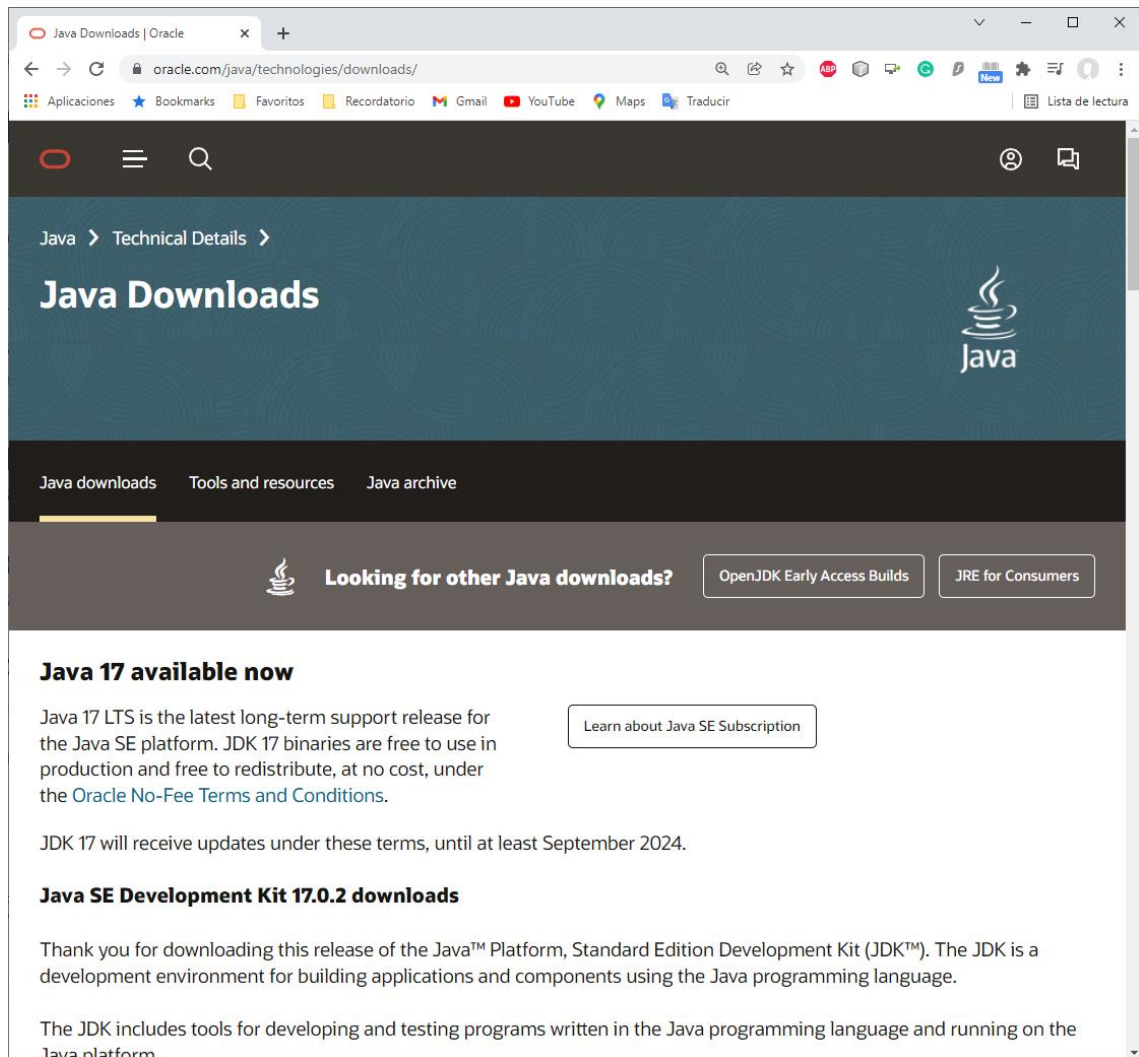
El entorno básico del JDK de Java que proporciona Oracle está formado por herramientas en modo texto, que son: *java*, intérprete que ejecuta programas en byte-code. *javac*, compilador de Java que convierte el código fuente en byte-code. *javap*, es un descompilador de byte-code a código fuente Java. *javadoc*, es un generador automático de documentos HTML a partir del código fuente Java. *appletviewer*, es un visualizador de applets.

El entorno de programación básico consiste en un compilador que convierta el código fuente Java a byte-code y el intérprete Java para ejecutar los programas (máquina virtual), también podremos necesitar un navegador que pueda ejecutar applets. No obstante, se necesita un editor para escribir el código fuente, podemos utilizar desde un bloc de notas a un entorno de desarrollo que nos proporcione herramientas para organizar el código, corrector de errores en tiempo real, depurador, entorno visual, documentación sensible al contexto, etc.

## 2. Instalación

Para instalar el jdk y el entorno NetBeans hay que realizar los siguientes pasos:

- Instalación:
  1. Descargar desde la Web oficial de Java en Oracle <https://www.oracle.com/java/technologies/downloads/> los archivos del jdk y la documentación.



2. Ejecutar el fichero descargado. Este fichero creará el directorio “C:\Archivos de programa\Java\jdkX”, a no ser que le indiquemos que se instale en otra carpeta.
3. Colocar el fichero zip de la documentación en la carpeta donde tengamos el jdk instalado.
4. Descargar el NetBeans en su última versión. <https://netbeans.apache.org/>
5. Ejecutar el fichero de instalación, esto creará la carpeta “Netbeans”.

### 3. El entorno de desarrollo integrado NetBeans

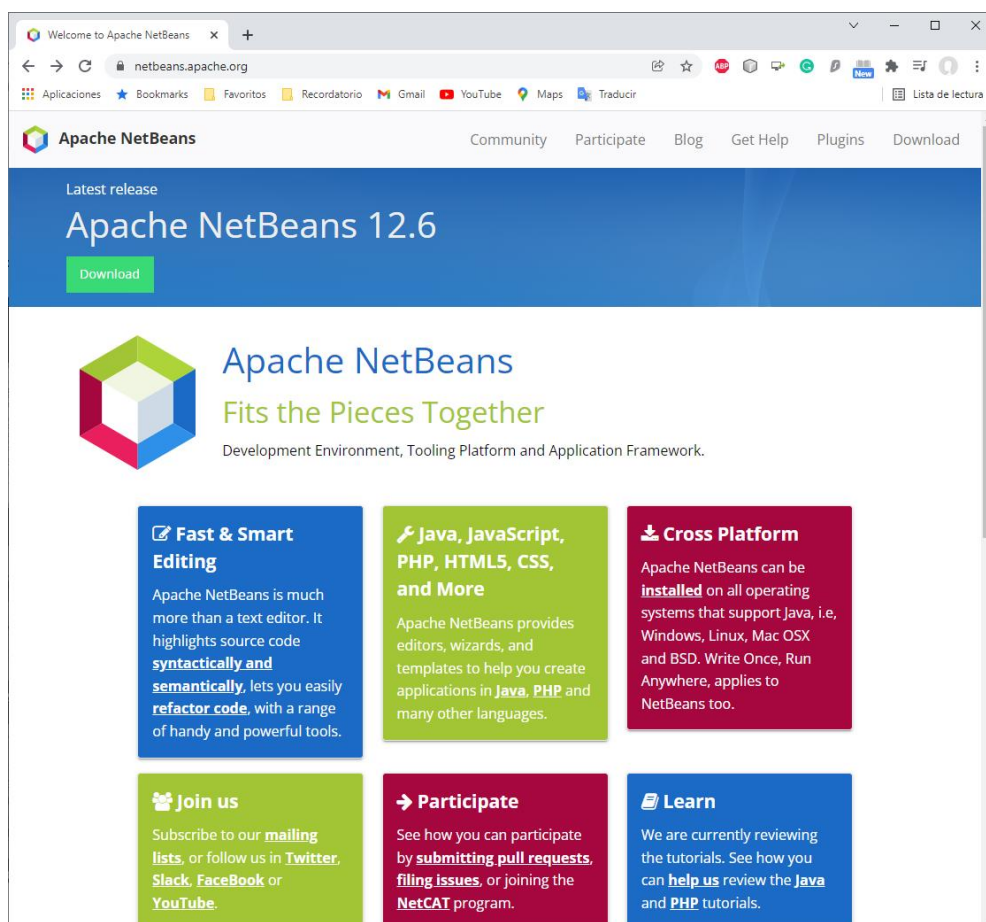
Normalmente el desarrollo de programas en Java se hace utilizando un entorno de desarrollo integrado (IDE por sus iniciales en inglés), que facilite la edición del programa, su compilación, depuración de errores, pruebas, etc. Ejemplos de entornos de desarrollo son: Eclipse, NetBeans, IntelliJ IDEA, JDeveloper, etc.

Para realizar los programas Java de este laboratorio utilizaremos NetBeans, una plataforma de código abierto que nació en 2000, como un proyecto de Sun Microsystems, que continuó desarrollando Oracle y que ahora lo gestiona la fundación Apache.

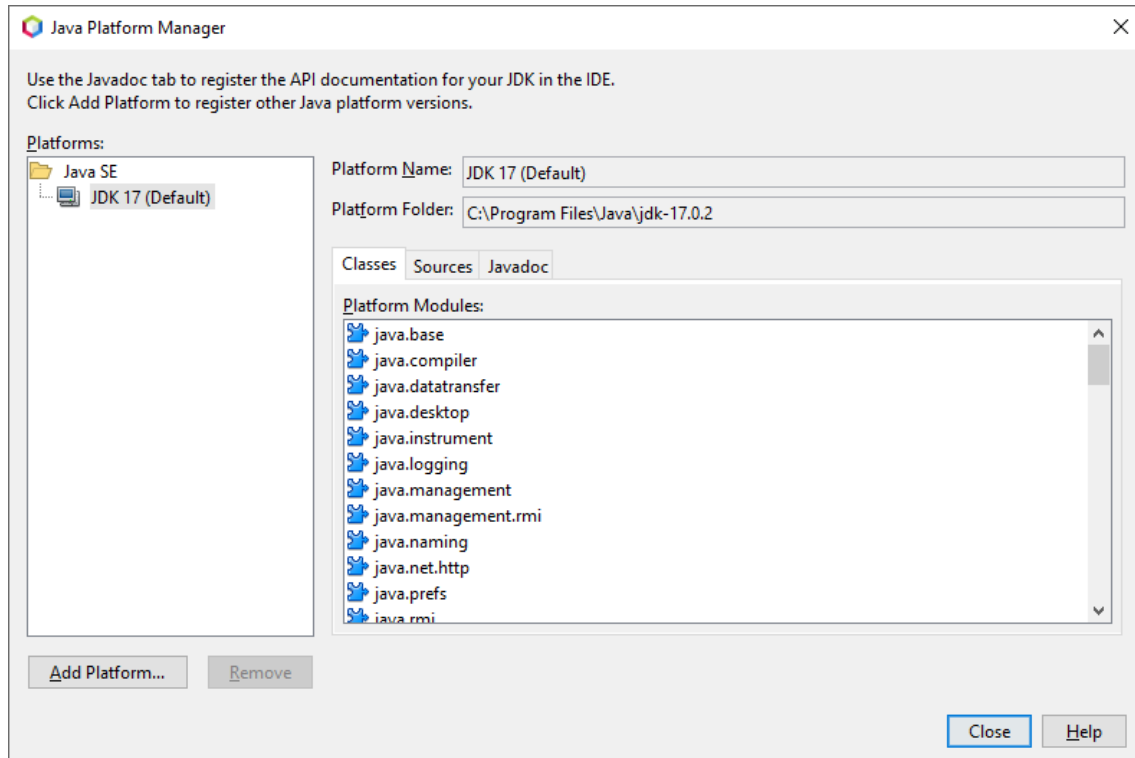
NetBeans está escrito en Java y pensado para este lenguaje, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. Desde Julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL).

Lee más información en <http://es.wikipedia.org/wiki/Netbeans>.

NetBeans se puede descargar e instalar de forma independiente del JDK. La última versión, la puedes descargar en <https://netbeans.apache.org/download/index.html>.



La documentación de Java, imprescindible para consultar clases, métodos, etc. hay que descargarla aparte. Se trata de un archivo grande que no hace falta descomprimir. Solamente guardar en la carpeta de instalación del JDK y si NetBeans no lo detecta configurar su ruta (Tools->Java Platforms):



## 4. Ejemplos

### 4.1 HolaMundo

Esta sesión estará dedicada a la familiarización con el entorno IDE de programación. Para ello crearemos nuestro primer **proyecto** NetBeans y vamos a escribir, compilar y ejecutar nuestro primer **programa** en Java:

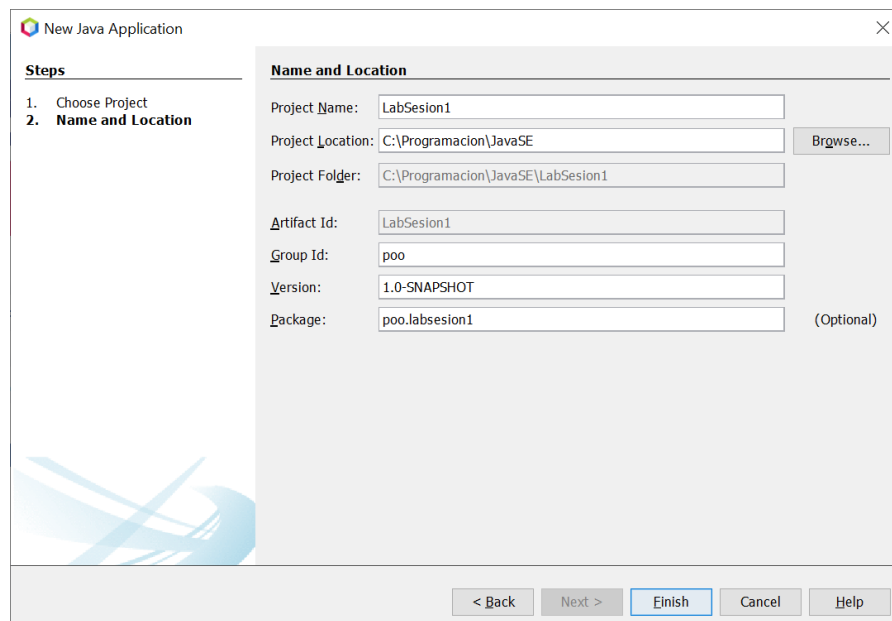
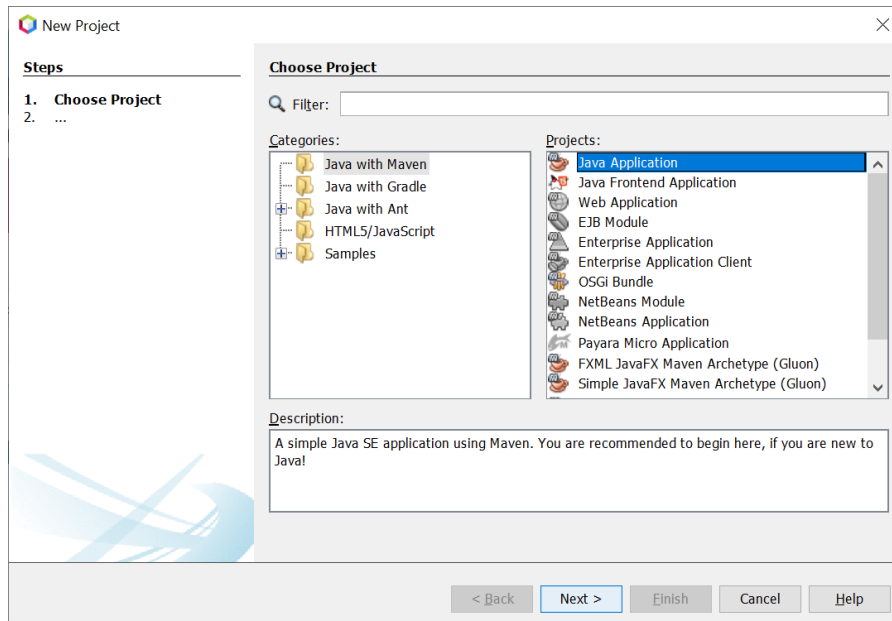
Cuando se desarrolle este programa es importante aprender los conceptos de proyecto, paquete y clase.

Para la creación de proyectos en NetBeans se pueden utilizar los siguientes gestores de proyectos: Maven, Gradle y Ant. En nuestro caso utilizaremos Maven por ser uno de los más populares.

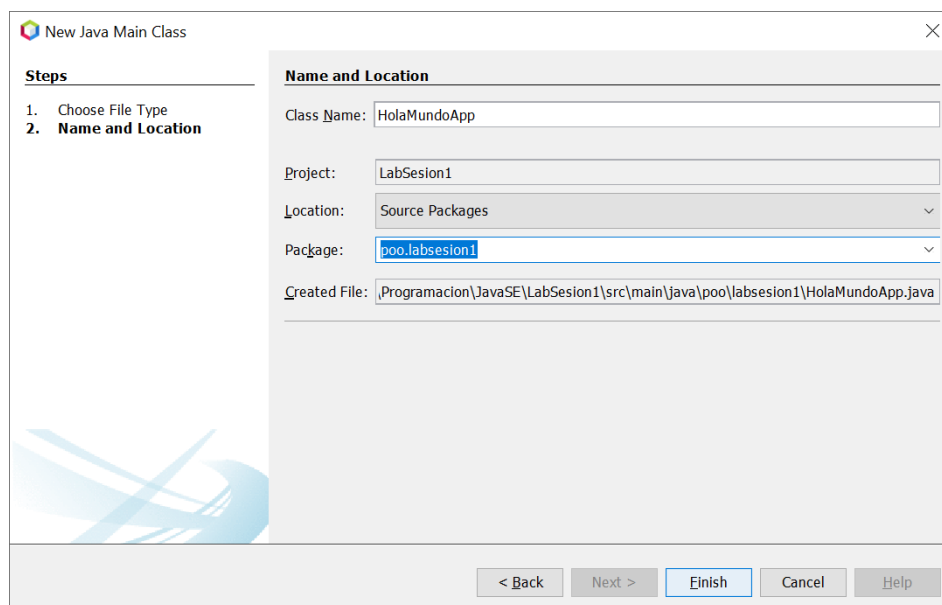
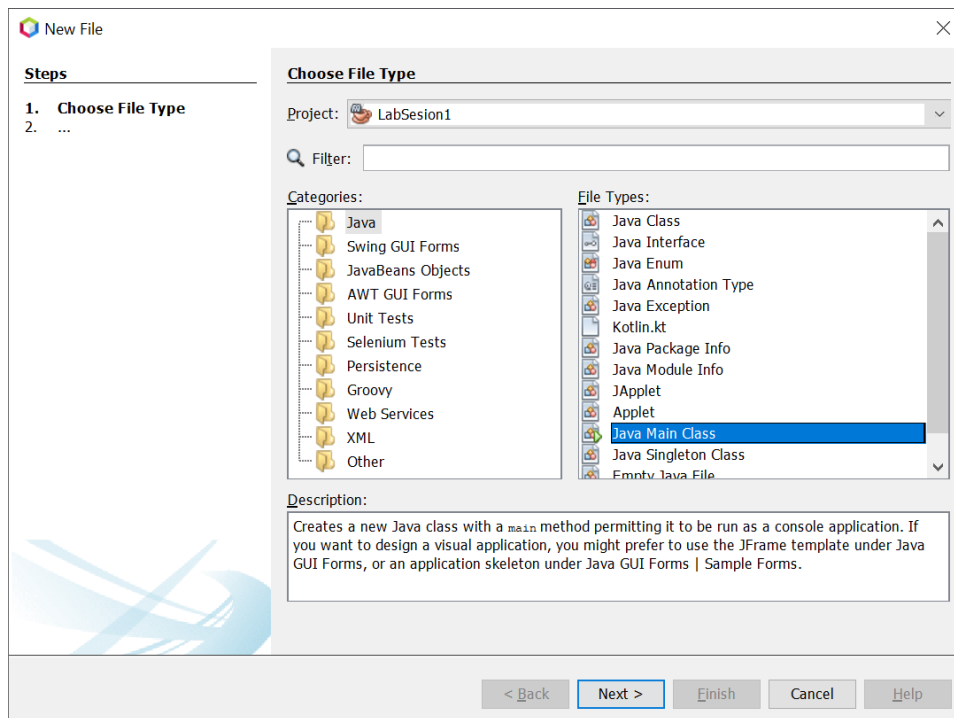
Para obtener más información sobre Maven consultar su página web:

<https://maven.apache.org/>

Por lo tanto, empezaremos creando un nuevo proyecto en NetBeans con Maven llamado LabSesion1:



Una vez creado el proyecto seleccionamos el tipo de fichero que queremos crear:



Escribimos lo que aparece en negrita:

```
package poo.labsesion1;

public class HolaMundoApp {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        System.out.println("Hola Mundo");
    }

}
```

## Ejercicios:

1. Comprobar que la documentación de Java está correctamente referenciada en NetBeans, viendo si funciona la ayuda de Java (Ayuda -> Búsqueda de índice de Javadoc).
2. Modificar el programa HolaMundoApp, cometiendo errores de escritura (falta del “;”, paréntesis no balanceados, palabras reservadas mal escritas (“estatic”), métodos no existentes (“println”), etc... para ver cómo NetBeans ayuda a escribir programas sintácticamente correctos.
3. Buscar las carpetas y archivos donde NetBeans está guardando nuestro trabajo.
4. Crear uno o varios “paquetes” en el proyecto, con programas similares al del ejemplo, y observar cómo se guardan ahora en las carpetas y ficheros.

## 4.2 Argumentos

La siguiente aplicación muestra el uso de los argumentos en la línea de llamada de un programa. La sintaxis utilizada por Java para poder realizarlo se hace a través del método main:

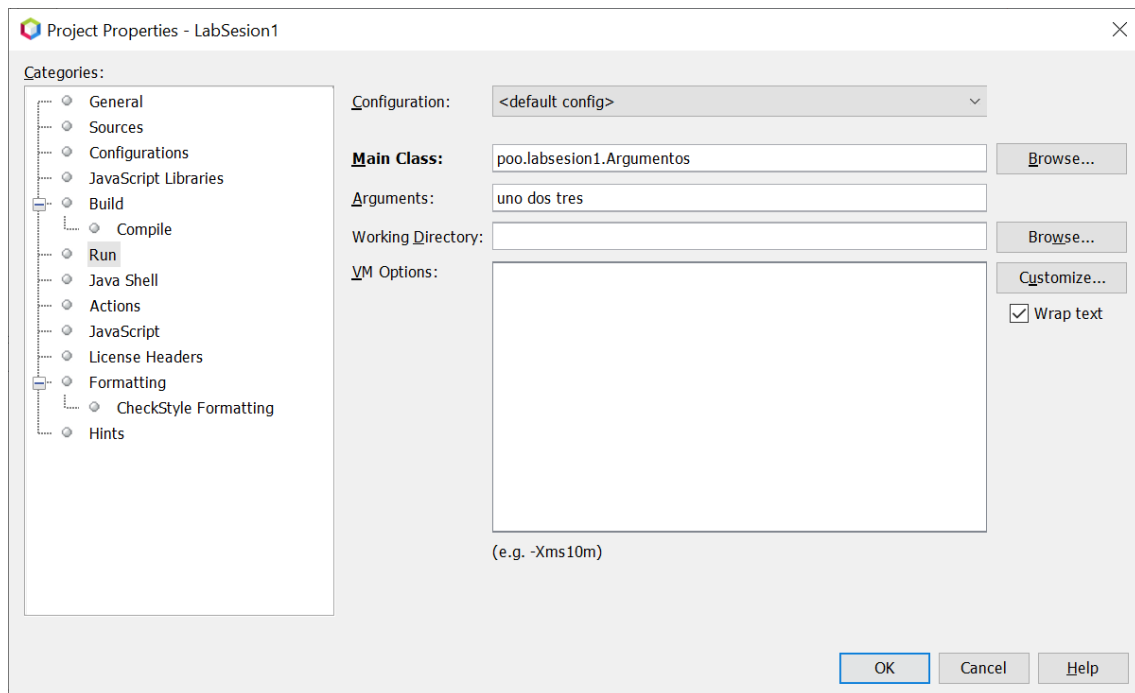
```
public static void main( String args[] ) {  
    . . .  
}
```

En él se puede apreciar que recibe un array de objetos de tipo String llamado args.

Para codificar el programa, dentro del proyecto anterior crearemos un nuevo fichero llamado “Argumentos”. El código de la aplicación es el siguiente:

```
package poo.labsesion1;  
  
public class Argumentos {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        for (int i = 0; i < args.length; i++) {  
            System.out.println(args[i]);  
        }  
    }  
}
```

Para ejecutarlo tendremos que cambiar las propiedades del proyecto para que admita argumentos:



Ahora solo tendremos que ejecutar el proyecto.

### Ejercicio:

1. Realizar un nuevo programa llamado “HolaTal” que salude a la persona cuyo nombre se pasa como argumento del programa. Codificar el programa en un nuevo paquete llamado “poo.ejercicio”.