

Tema 1A – Desarrollo en Java

Unai Aguilera Irazabal

unai.aguilera@deusto.es

Grado en Ingeniería Informática

Facultad de Ingeniería – Universidad de Deusto

Reflexiona

- ¿Qué herramientas necesitamos para desarrollar programas?
 - ¿Y en concreto para Java?
- ¿Conoces el proceso de instalación y uso de estas herramientas?
- ¿Qué sabes sobre el proceso de compilación en Java?
- ¿Conoces alguna configuración necesaria cuando se instala Java en un entorno de desarrollo o ejecución?
- ¿Por qué crees que es necesario saber configurar Java para su uso fuera de Eclipse?


Descarga del JDK (I)

- Para poder desarrollar en Java es necesario tener instalado el Java Development Kit (JDK).
 - Contiene las herramientas tanto para compilar como para ejecutar programas Java.
 - No debe confundirse con la versión que contiene solamente la máquina virtual (JRE – Java Runtime Environment) y que está orientada a ser utilizada para desplegar y/o usuarios finales.
- La versión más reciente del JDK (mantenida por Oracle) se encuentra disponible aquí
 - <https://www.oracle.com/java/technologies/javase-downloads.html>
 - Sin embargo, esta versión de Oracle tiene una licencia de uso que se debe tener en cuenta en el caso de realizar programas comerciales
- Existen también versiones del JDK mantenidas por otras organizaciones que no tienen esas restricciones de uso
 - <https://adoptium.net/>

Descarga del JDK (II)

- Para esta práctica, vamos a descargar la versión OpenJDK 17 (LTS) obtenida de <https://adoptium.net/>

Download Temurin™ for Windows x64

 Latest LTS Release

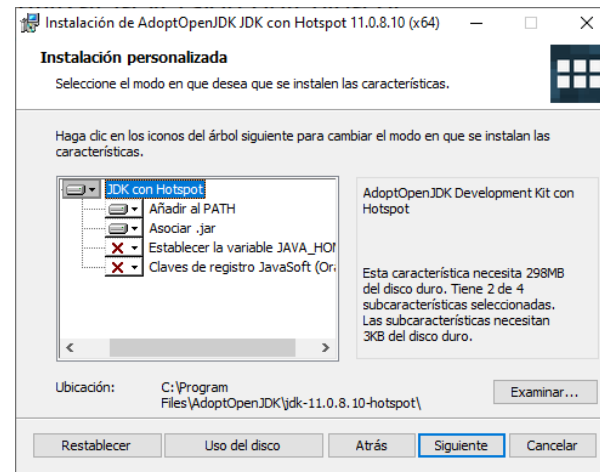
jdk-17.0.4.1+1

Other platforms and versions ↗

Release Archive 📁

Instalación en Windows

- Cuando lanzamos el instalador es importante dejar marcadas las opciones de "Asociar al PATH" y "Asociar al .jar".
 - La opción de "Asociar al PATH" configura las variables de entorno de Windows para poder lanzar los comandos "java" y "javac" correctamente desde el "Símbolo del sistema".



- Continuamos el proceso y esperamos a que finalice la instalación.

Variable de entorno PATH (I)

- Recordemos que en los Sistemas Operativos (Windows, Linux, etc.), los ejecutables pueden ser encontrados y ejecutados porque existe una lista de directorios conocida como PATH.
- En el caso de Windows esta variable se configura en las variables de entorno.
 - Si hemos marcado la opción correcta en el proceso de instalación anterior, las variables de entorno habrán sido actualizadas para incluir el directorio de Java.
- Podemos comprobar en la línea de comandos de Windows si la variable PATH está correctamente configurada utilizando el comando

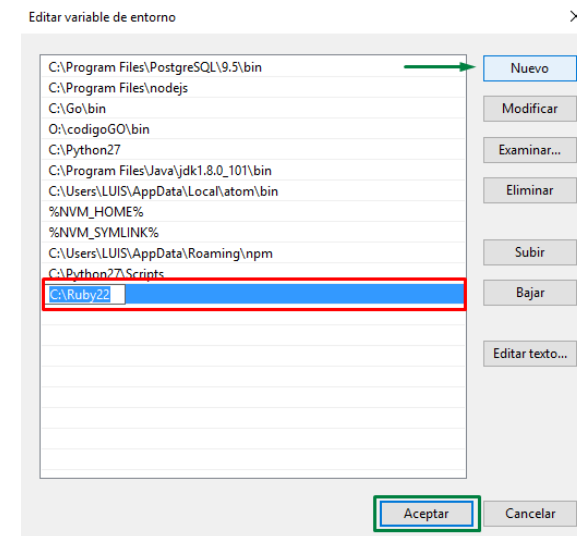
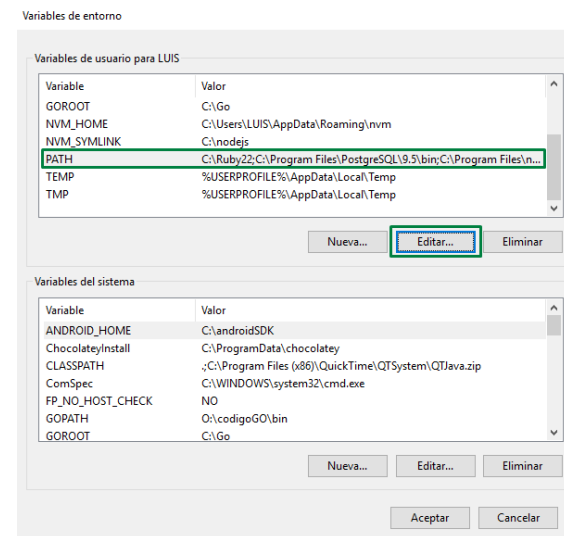
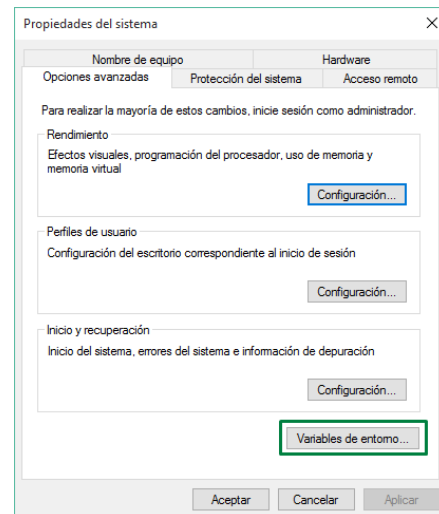
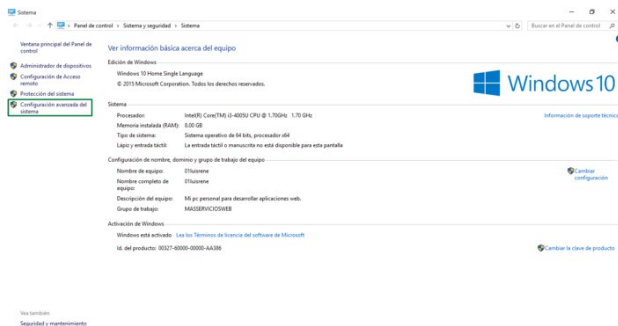
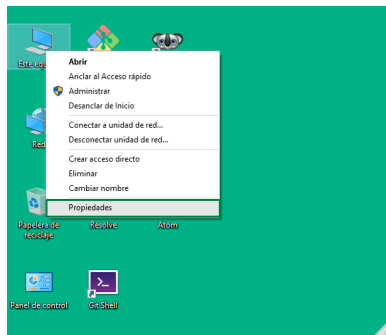
`echo %PATH%`

```
C:\Users\unai.aguilera>echo %PATH%
C:\Program Files\apache-maven-3.6.3\bin;C:\Program Files\AdoptOpenJDK\jdk-11.0.10-hotspot\bin;C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\OpenSSH\;C:\Windows\CCM;C:\Program Files\doxygen\bin;C:\Program Files\Graphviz\bin;C:\Users\unai.aguilera\AppData\Local\Programs\Python\Python39\Scripts\;C:\Users\unai.aguilera\AppData\Local\Programs\Python\Python39\;C:\Users\unai.aguilera\AppData\Local\Microsoft\WindowsApps;C:\Users\unai.aguilera\AppData\Local\Programs\Microsoft VS Code\bin;C:\Users\unai.aguilera\AppData\Local\Programs\Git\cmd
C:\Users\unai.aguilera>
```

- Podemos observar la lista de directorios separados por ";".
- En el caso de Linux/Mac podemos observar el PATH actual con el comando `echo $PATH` y veremos que en estos sistemas operativos se utilizan dos puntos ":" como separador en vez de ";".

Variable de entorno PATH (II)

- Recordemos que en los Sistemas Operativos (Windows, Linux, etc.), los ejecutables pueden ser encontrados y ejecutados porque existe una lista de directorios conocida como PATH.
- En el caso de Windows esta variable se configura en las variables de entorno.
 - Si hemos marcado la opción correcta en el proceso de instalación anterior, las variables de entorno habrán sido actualizadas para incluir el directorio de Java.
- En caso de necesitar modificar las variables de entorno manualmente en Windows 10



Se debe añadir una nueva entrada a la lista del PATH, al directorio de JAVA. Se debe tener cuidado de no borrar por error las entradas existentes.

Comprobación de la instalación

- Para comprobar si el JDK está instalado correctamente debemos abrir una ventana de línea de comandos.
- Probamos si el compilador está disponible. Debe aparecer la versión que hemos instalado.

`javac --version`

```
PS C:\Users\unai.aguilera> javac --version
javac 11.0.10
PS C:\Users\unai.aguilera>
```

- También podemos comprobar si está accesible la JVM.

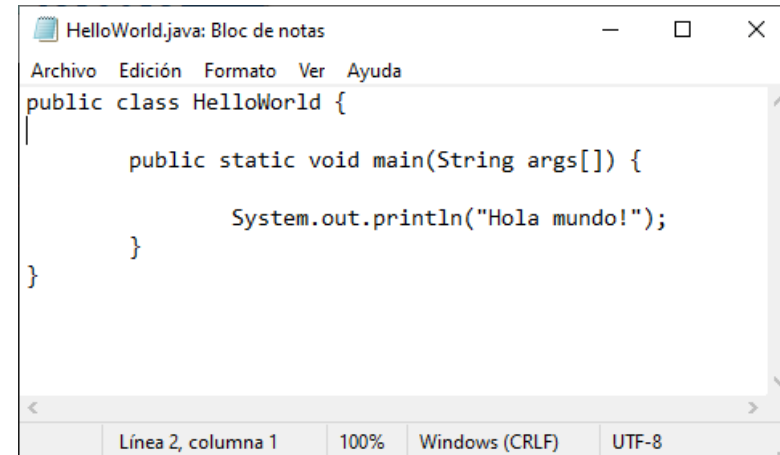
`java --version`

```
PS C:\Users\unai.aguilera> java --version
openjdk 11.0.10 2021-01-19
OpenJDK Runtime Environment AdoptOpenJDK (build 11.0.10+9)
OpenJDK 64-Bit Server VM AdoptOpenJDK (build 11.0.10+9, mixed mode)
PS C:\Users\unai.aguilera> |
```

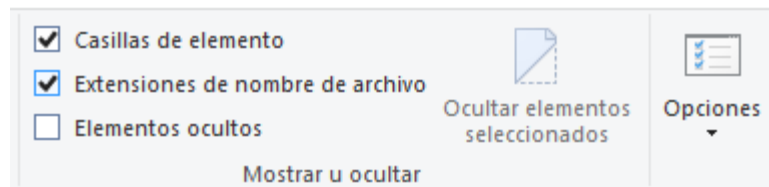

Escribiendo un programa

- Vamos a utilizar un editor de texto sencillo (Notepad/bloc de notas) para editar nuestro primer programa y probar el proceso de compilación.

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

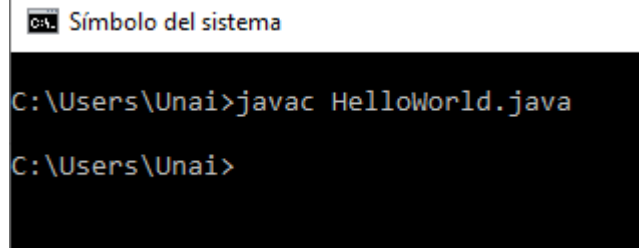


- Cuidado con la extensión del fichero. Debe ser .java.
- Para evitar problemas conviene activar la vista de extensiones en el explorador de Windows.



Compilando y ejecutando

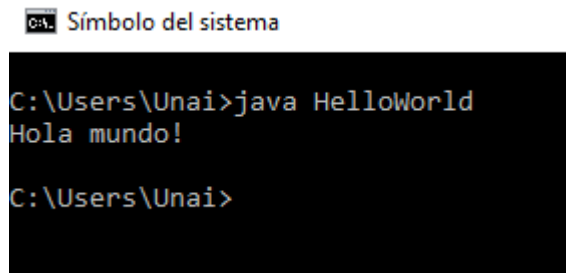
- Una vez que hemos guardado el fichero de código debemos compilarlo.



Simbolo del sistema

```
C:\Users\Unai>javac HelloWorld.java  
C:\Users\Unai>
```

- Como resultado de la compilación se genera un fichero `.class` que se puede ejecutar en la máquina virtual de Java.
- Es importante recordar que al comando Java se le pasa el nombre de la clase que contiene el main (no el nombre del fichero).



Simbolo del sistema

```
C:\Users\Unai>java HelloWorld  
Hola mundo!  
C:\Users\Unai>
```

Práctica IA – Parte I

- Realiza un programa que pida al usuario 10 números enteros, calcule su media y la imprima por consola. Incluye el programa en un clase llamada *Practica0*.
- Debes escribir el programa con un editor de texto y compilarlo y ejecutarlo utilizando la línea de comandos.
- Si ves símbolos raros a la hora de imprimir los datos por consola:
 - Revisa la codificación en la que has escrito los ficheros fuentes. Se recomienda utilizar UTF-8.
 - Usa los parámetros necesarios durante la compilación y ejecución para la codificación UTF-8.

Práctica I A – Parte II

- Realiza un nuevo programa basado en la Parte I que esté organizado de la siguiente forma:
 - Clase *Media* que proporcione un método *calcularMedia(...)* que reciba un array de 10 números enteros y devuelva un valor de tipo float con la media calculada para dichos valores.
 - La clase anterior debe estar situada en un paquete llamado *practica0.estadisticas*.
 - Clase *Practica0*, que debe contener el método *main* donde se leerán 10 números enteros de teclado, se introducirán en un array y se pasarán al método *calculaMedia* de la clase *Media* para obtener el valor medio e imprimirlo posteriormente por pantalla.
 - Esta clase *Práctica0* debe estar situada en el paquete *practica0*.
- Compila y ejecuta el programa obtenido en línea de comandos. Ten en cuenta que ahora el programa está compuesto de varios ficheros/clases.

Classpath

- La máquina virtual de Java necesita saber en qué directorios del sistema debe mirar para buscar las clases que el programa debe cargar para su ejecución.
- Lo usual es especificarla en la línea de comandos con un parámetro opcional para el compilador o la JVM

```
javac -classpath ruta1;ruta2 fichero_a_compilar
```

```
java -cp ruta1;ruta2 clase_principal
```

- Pero también se puede especificar como una variable de entorno en la configuración de Windows 10 o en la línea de comandos (se borra al reiniciar el CMD)

```
set CLASSPATH="ruta1;ruta2;..."
```

Práctica IA- Parte III

- Vamos a modificar la estructura de la práctica para separar las clases en distintos paquetes no situados en el mismo directorio.
- Mueve la clase principal a un directorio distinto, de tal forma que la estructura de directorios de la práctica quede de la siguiente manera

libreria	<i>libreria</i> es un directorio (no un paquete) que contiene los paquetes <i>practica0</i> y <i>estadisticas</i> en una jerarquía. La clase <i>Media</i> debe estar en el paquete <i>practica0.estadisticas</i> .
practica0	
estadisticas	
main	<i>main</i> es un directorio (no un paquete), que contiene el paquete <i>practica0</i> donde se encuentra la clase <i>Practica0</i> .
practica0	

- Compila y ejecuta el programa utilizando esta estructura de directorios.
- Ten en cuenta que ahora le tienes que decir a Java donde encontrar los ficheros necesarios tanto durante la compilación como durante la ejecución.