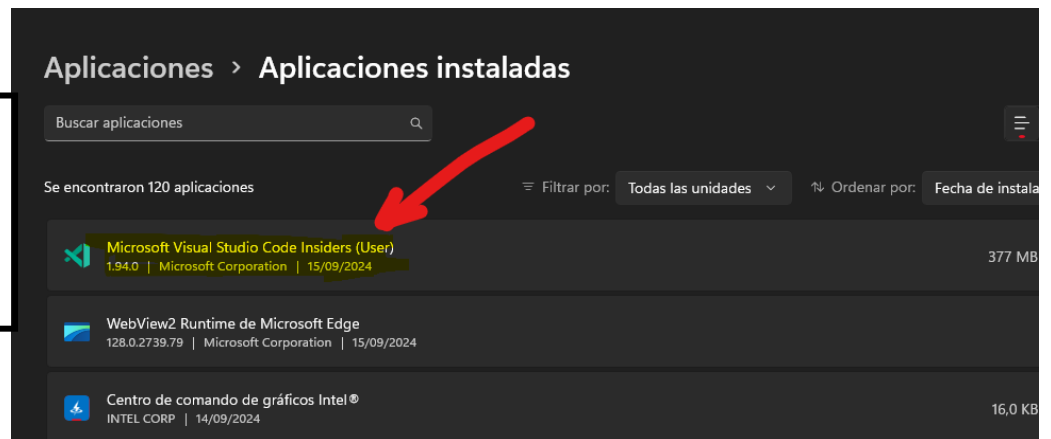


RAÚL VICARIO  
GARCÍA  
71482159E



## 1. ¿Qué es la JVM?

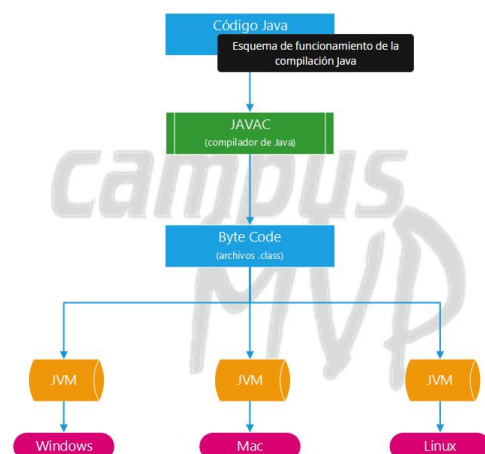
La JVM (Máquina Virtual de Java), es una parte de la plataforma Java. Su propósito principal es proporcionar un entorno de ejecución independiente de la plataforma para los programas escritos en Java.

Tradicionalmente, el compilador de un lenguaje de alto nivel se encargaba de traducir ese lenguaje "sencillo" en lenguaje máquina, directamente utilizable por el computador a través del sistema operativo. Es decir, cuando compilamos un programa en C lo que obtenemos es un programa ejecutable, por ejemplo para Windows, que este sistema operativo es capaz de ejecutar directamente contra el procesador, en un lenguaje "entendible" por este.

Sin embargo, lenguajes modernos como Java, lo que hacen es utilizar un paso intermedio entre estos dos estados: entre el código de alto nivel en el que escribimos las aplicaciones y el de bajo nivel que sale del proceso de compilación.

Cuando compilas una aplicación escrita en lenguaje Java, en realidad éste **no se compila a lenguaje máquina**, directamente entendible por el sistema operativo, sino a un **lenguaje intermedio denominado ByteCode**.

Entre el *ByteCode* y el sistema operativo se coloca un componente especial llamado **Máquina virtual** que es el que realmente va a ejecutar el código. Esta idea, por cierto, no tiene nada que ver con las máquinas virtuales a las que estamos acostumbrados hoy en día que ejecutan sistemas operativos completos, sino que es un concepto mucho más antiguo.



```
PS C:\Users\theof> java -version
java version "21.0.4" 2024-07-16 LTS
Java(TM) SE Runtime Environment (build 21.0.4+8-LTS-274)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.4+8-LTS-274, mixed mode, sharing)
```

## 2. ¿Qué es la JDK? ¿Qué componentes la forman?

La JDK, o Java Development Kit, es un conjunto de herramientas y bibliotecas necesarias para desarrollar aplicaciones en Java. Es una distribución completa que incluye todo lo necesario para compilar, ejecutar y depurar programas Java. Aquí te detallo los componentes principales que forman parte de la JDK:

1. **Compilador de Java (javac):** El compilador convierte el código fuente de Java (.java) en bytecode (.class), que es el formato intermedio que la JVM puede ejecutar.
2. **Java Runtime Environment (JRE):** La JDK incluye el JRE, que proporciona la JVM, las bibliotecas de clase estándar y otras componentes necesarias para ejecutar aplicaciones Java. Esto significa que, además de permitir la compilación, la JDK también te permite ejecutar y probar tus aplicaciones durante el desarrollo.
3. **Herramientas de Depuración:** Incluye herramientas para depurar aplicaciones Java, como jdb (Java Debugger) que permite examinar el estado de la aplicación en ejecución y realizar operaciones de depuración.
4. **Herramientas de Documentación:** Proporciona herramientas como javadoc, que generan documentación API en formato HTML a partir de comentarios especiales en el código fuente de Java.
5. **Herramientas de Administración y Monitorización:** Incluye herramientas como jps (Java Process Status) para mostrar información sobre los procesos Java en ejecución, jstat para estadísticas de rendimiento, y jconsole para la monitorización y administración de aplicaciones Java en ejecución.
6. **Bibliotecas y APIs de Java:** Incluye una amplia gama de bibliotecas y APIs que proporcionan funcionalidades adicionales como colecciones, entrada/salida, red, y más.
7. **Herramientas de Construcción y Gestión de Proyectos:** A veces incluye herramientas como javap (un descompilador de bytecode) y jar (para empaquetar archivos .class en archivos JAR).

## 3. ¿Cuál es la diferencia entre JDK y JRE?

- **JDK** es el paquete completo que incluye tanto las herramientas para desarrollar (compilador, depurador, etc.) como las herramientas para ejecutar aplicaciones (JRE).
- **JRE** es solo el entorno necesario para ejecutar aplicaciones Java y no incluye herramientas de desarrollo como el compilador.

Si eres un desarrollador de Java, necesitas el JDK. Si solo deseas ejecutar aplicaciones Java, el JRE es suficiente, debemos reseñar, como dijimos en la primera sesión de prácticas de Programación III, que son cosas completamente

distintas.

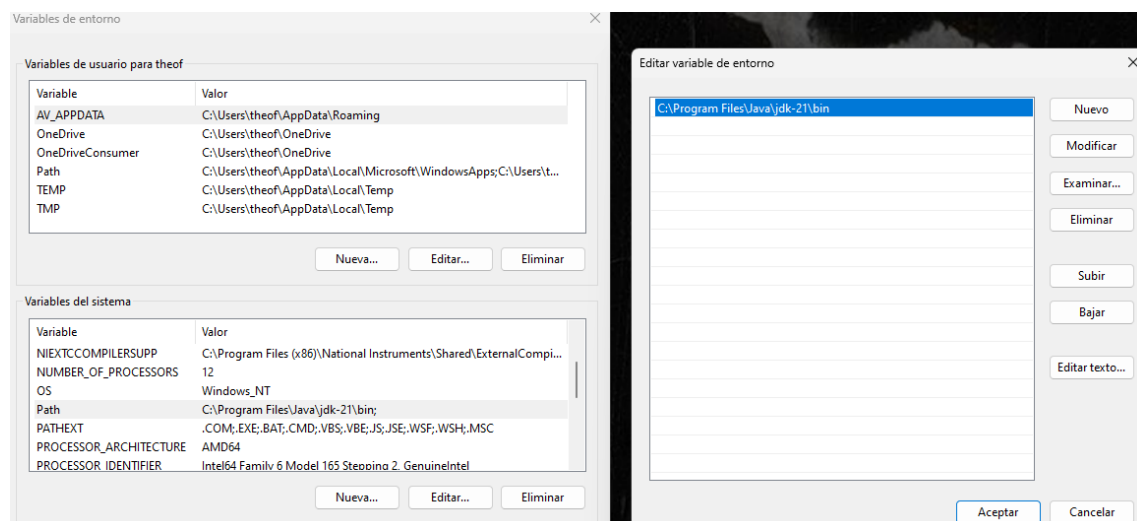
#### **4. ¿Qué es una variable de entorno? ¿Cómo se cambia su valor en tu OS desde la terminal?**

Una variable de entorno es una variable que contiene información del sistema y que puede ser utilizada por los programas en ejecución para obtener información sobre el entorno en el que se ejecutan. Estas variables suelen almacenar información como la configuración del sistema, la ruta de acceso a directorios importantes, o parámetros que afectan el comportamiento de las aplicaciones.

Para hacer cambios permanentes:

1. Abrir el Panel de Control y ve a Sistema y seguridad > Sistema.
2. Haz clic en Configuración avanzada del sistema en el panel izquierdo.
3. En la ventana Propiedades del sistema, haz clic en el botón Variables de entorno.
4. En la sección Variables del sistema o Variables del usuario, selecciona la variable que deseas cambiar y haz clic en Editar. Si la variable no existe, puedes crear una nueva.
5. Modifica el valor de la variable y haz clic en Aceptar para guardar los cambios.

Para aplicar los cambios en la línea de comandos después de hacer modificaciones permanentes, es posible que necesites cerrar y volver a abrir la ventana del terminal o cmd o la powershell de windows.



```
Microsoft Windows [Versión 10.0.22631.4169]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\theof>java --version
java 21.0.4 2024-07-16 LTS
Java(TM) SE Runtime Environment (build 21.0.4+8-LTS-274)
Java HotSpot(TM) 64-Bit Server VM (build 21.0.4+8-LTS-274, mixed mode, sharing)

C:\Users\theof>javac --version
javac 21.0.4
```

### **5. ¿Qué propósito tiene la variable de entorno PATH? ¿Qué valor tiene en tu equipo (rutas)? ¿Qué separador se usa en tu OS?**

La variable de entorno PATH tiene un propósito crucial en los sistemas operativos: define una lista de directorios que el sistema busca para encontrar los ejecutables de los comandos que ingresas en la terminal o línea de comandos. Cuando introduces un comando, el sistema operativo utiliza la variable PATH para buscar en estos directorios para encontrar el archivo ejecutable correspondiente.

```
C:\Users\theof>echo %PATH%
C:\Program Files\Java\jdk-21\bin;C:\Users\theof\AppData\Local\Microsoft\WindowsApps;C:\Users\theof\AppData\Local\Programs\Microsoft VS Code Insiders\bin
```

En Windows se usa como separador “;”.

### **6. ¿Qué ruta se debe incluir en el PATH cuando se instala la JDK?**

Se debe incluir el directorio “bin” en la ruta de la instalación del JDK en mi propio caso, me correspondería “C:\Program Files\Java\jdk-21\bin”.

Este directorio contiene los ejecutables necesarios para desarrollar y ejecutar aplicaciones Java.

### **7. ¿Qué es la variable JAVA\_HOME y cuál es su propósito? ¿Cuál es su valor en tu sistema?**

La variable de entorno JAVA\_HOME es una variable que se utiliza para definir la ubicación de la instalación del JDK (Java Development Kit) en el sistema. Su propósito principal es proporcionar una forma estandarizada para que las herramientas y aplicaciones puedan encontrar la instalación del JDK y acceder a sus componentes, como los compiladores (javac) y las herramientas de ejecución (java).

```
Microsoft Windows [Versión 10.0.22631.4169]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\theof>echo %JAVA_HOME%
C:\Program Files\Java\jdk-21\bin

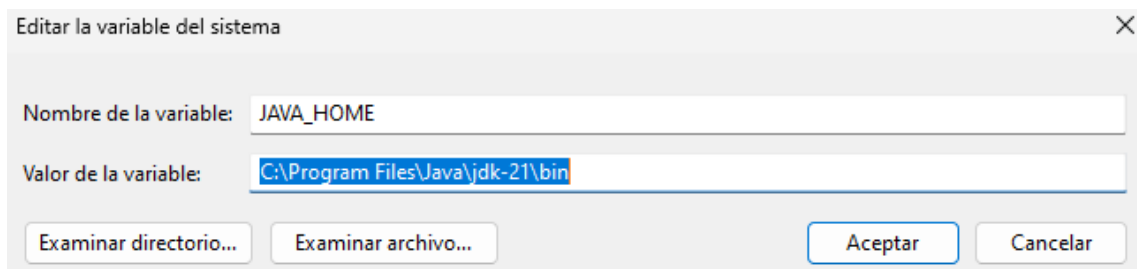
C:\Users\theof>
```

## 8. ¿Qué es la variable de entorno CLASSPATH y cuál es su propósito?

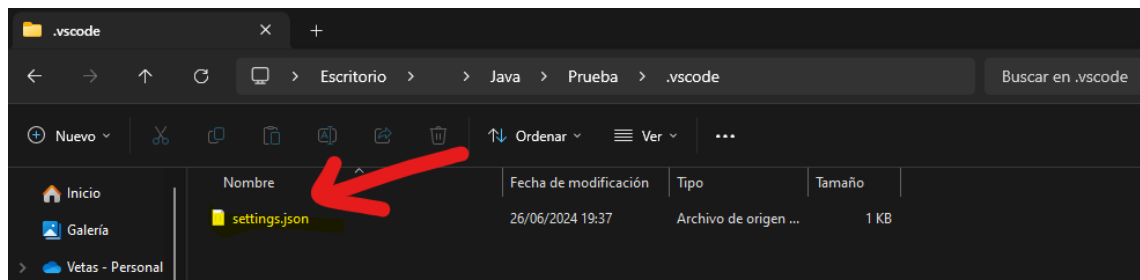
La variable de entorno **CLASSPATH** es una configuración importante en el entorno de desarrollo Java. Su propósito principal es definir una lista de directorios y archivos JAR que el sistema debe buscar para encontrar clases y recursos cuando se ejecuta un programa Java.

## 9. ¿Cómo se podría cambiar la JDK que utiliza un proyecto en VSCode?

- **Actualizar JAVA\_HOME:** Cambia la variable de entorno global JAVA\_HOME para que apunte al nuevo JDK.



- **Configurar en VSCode:** Usa el archivo settings.json en la carpeta .vscode de tu proyecto para definir el JDK específico para ese proyecto.



## 10. Haz un Hola Mundo en VSCode en tu equipo y describe que ocurre cuando pulsas el botón play (los comandos que se ejecutan en la terminal al hacerlo).

Con la PowerShell de Windows, se busca el directorio donde se aloja el .java, luego se acude a la JVM y utiliza el código para ejecutarlo con el “-cp”.

```
PS C:\Users\theof\Desktop\PRUEBA\HolaMundo> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\theof\Desktop\PRUEBA\HolaMundo\bin' 'App'
Hello, World!
PS C:\Users\theof\Desktop\PRUEBA\HolaMundo>
```