

TE3.1 - Configuración de Eclipse y JetBrains IntelliJ IDEA

En esta tarea de aprendizaje, aprenderás a instalar el JDK y JREs en Windows y Linux, y a configurar Eclipse y JetBrains IntelliJ IDEA.

Recursos

- [Instalación del JDK en Linux](#)
- [Instalación del JDK en Windows](#)

Objetivos

- Conocer cómo instalar el JDK/JRE en Windows y Linux.
- Conocer cómo configurar Eclipse y JetBrains IntelliJ IDEA para utilizar un JDK.
- Conocer cómo configurar Windows y Linux a nivel de sistema para utilizar diferentes versiones de Java.
- Conocer cómo instalar y usar la herramienta SDKMan para instalar diferentes versiones de Java en Linux/MacOS/Windows.

Recursos

GIF Videos

- [Software crear GIFs animados para Windows](#)
- [Software crear GIFs animados para Linux](#)
- [Herramienta Online GIF](#)

Entrega

El documento justificativo de la realización de la tarea se realizará en formato [Markdown](#), el nombre del fichero será [readme.md](#) y estará dentro de la carpeta [UT3\TE3.1](#) dentro del repositorio oficial del alumno para la asignatura.

El fichero [readme.md](#) debe contener los siguientes apartados:

- Cada uno de los puntos de la tarea.

1. Configuración de Java en Windows y Linux

1. Revisa la configuración de tu máquina a través del terminal e indica la versión de Java que tienes instalada.

```
# Comprueba la versión de Java instalada
$> java -version
# Comprueba donde está instalado Java
$> where java
# Busca todas las versiones de Java instaladas
$> which java
```

⌚ Adjunta una imagen de los comandos anteriores y responde a las siguientes preguntas

```
Alumno_ciclo@PC-FP MINGW64 ~/Desktop
$ 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)

Alumno_ciclo@PC-FP MINGW64 ~/Desktop
$ where java
C:\Program Files\Common Files\Oracle\Java\javapath\java.exe

Alumno_ciclo@PC-FP MINGW64 ~/Desktop
$ which java
/c/Program Files/Common Files/Oracle/Java/javapath/java

Alumno_ciclo@PC-FP MINGW64 ~/Desktop
$
```

¿Qué versión de Java tienes instalada?

Java 21.0.4

¿Cuántas versiones de Java tienes instaladas? ¿Por qué?

Una, por que hasta ahora, nunca he necesitado más. Así que siempre me aseguraba de mantener solo la versión que utilizo.

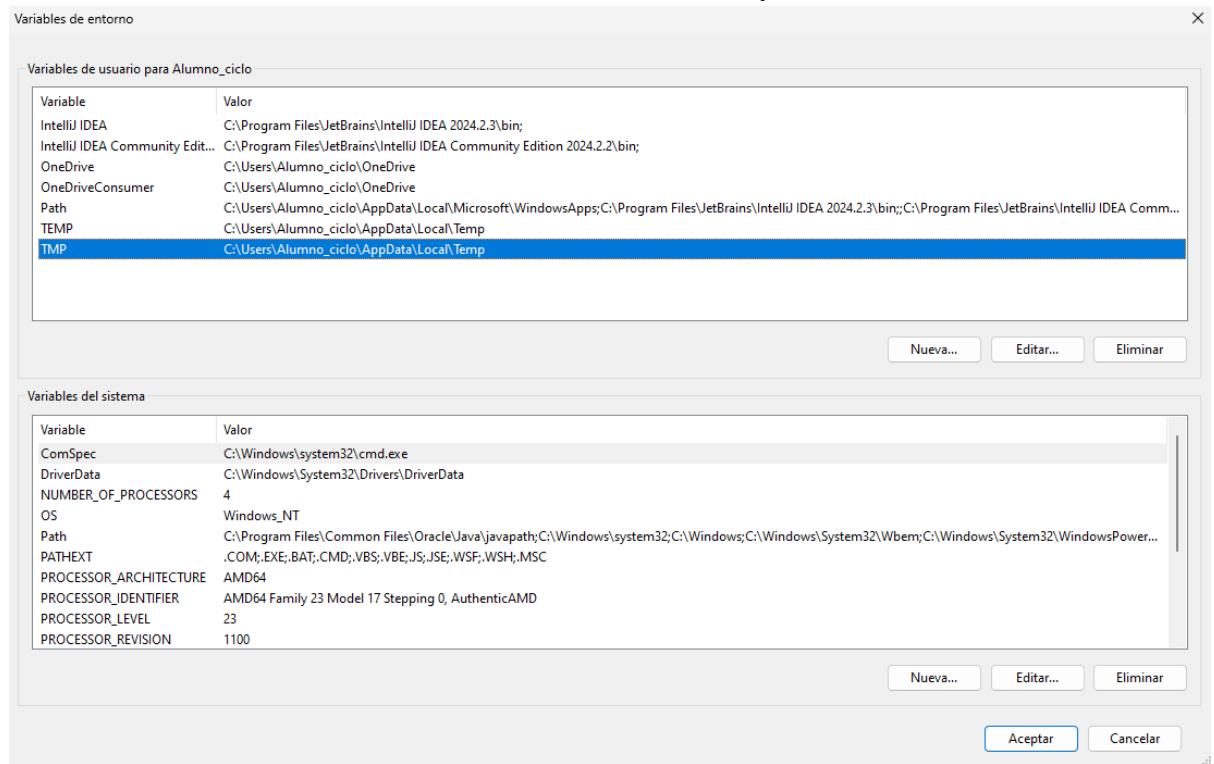
Si tienes más de una versión indica todas las versiones y rutas de instalación.

Java 21.0.4 => C:\Program Files\Commen Files\Oracle\Java\javapath\java.exe

2. Variables de entorno.

⌚ Adjunta una imagen de las variables de entorno de tu sistema, tanto a nivel de usuario como a nivel de sistema.

- Muestra a través de interfaz (Ventana de windows) (Usuarios y sistema)



- Muestra a nivel de comandos (Solo usuario) (set)

```
Alumno_ciclo@PC-FP MINGW64 ~/Desktop
$ set
ACLOCAL_PATH=/mingw64/share/aclocal:/usr/share/aclocal
ALLUSERSPROFILE='C:\ProgramData'
APPDATA='C:\Users\Alumno_ciclo\AppData\Roaming'
BASH=/usr/bin/bash
BASHPROMPT=checkwinsize:cmdhist:complete_fullquote:expand_aliases:extquote:force_f
ignore:globasciiranges:globskipdots:hostcomplete:interactive_comments:login_shel
l:patsub_replacement:progcomp:promptvars:sourcpath
BASH_ALIASES=()
BASH_ARGC=([0]="0")
BASH_ARGV=()
BASH_CMDS=()
BASH_LINENO=()
BASH_LOADABLES_PATH=/usr/local/lib/bash:/usr/lib/bash:/opt/local/lib/bash:/usr/p
kg/lib/bash:/opt/pkg/lib/bash::
BASH_SOURCE=()
BASH_VERSINFO=([0]="5" [1]="2" [2]="37" [3]="1" [4]="release" [5]="x86_64-pc-msy
s")
BASH_VERSION='5.2.37(1)-release'
COLUMNS=80
COMMONPROGRAMFILES='C:\Program Files\Common Files'
COMPLETION_PATH='C:/Program Files/Git/mingw64/share/git/completion'
COMPUTERNAME=PC-FP
COMP_WORDBREAKS=$'\t\n'[@><; |&(:'
COMSPEC='C:\Windows\system32\cmd.exe'
CONFIG_SITE=/etc/config.site
CommonProgramW6432='C:\Program Files\Common Files'
DIRSTACK=()
DISPLAY=needs-to-be-defined
DriverData='C:\Windows\System32\Drivers\DriverData'
EUID=197609
EXEPATH='C:\Program Files\Git'
GIT_EXEC_PATH='C:/Program Files/Git/mingw64/libexec/git-core'
GROUPS=()
HISTFILE=/c/Users/Alumno_ciclo/.bash_history
HISTFILESIZE=500
HISTSIZE=500
```

```

HOMEPATH='\\Users\\Alumno_ciclo'
HOSTNAME=PC-FP
HOSTTYPE=x86_64
IFS=$' \t\n'
INFOPATH=/mingw64/local/info:/mingw64/share/info:/usr/local/info:/usr/share/info
:/usr/info:/share/info
LC_CTYPE=es_ES.UTF-8
LINES=24
LOCALAPPDATA='C:\\Users\\Alumno_ciclo\\AppData\\Local'
LOGONSERVER='\\PC-FP'
MACHTYPE=x86_64-pc-msys
MAILCHECK=60
MANPATH=/mingw64/local/man:/mingw64/share/man:/usr/local/man:/usr/share/man:/usr
/man:/share/man
MINGW_CHOST=x86_64-w64-mingw32
MINGW_MOUNT_POINT=/mingw64
MINGW_PACKAGE_PREFIX=mingw-w64-x86_64
MINGW_PREFIX=/mingw64
MSYS2_PATH=/usr/local/bin:/usr/bin:/bin
MSYS2_PS1='[\033[0;${TITLEPREFIX:$PWD/007}]\n[\033[32m]\u0@\h \[\033[35m\]$MSYS
TEM [\033[33m]\w[\033[36m]`__git_ps1`[\033[0m]\n$ '
MSYSTEM=MINGW64
MSYSTEM_CARCH=x86_64
MSYSTEM_CHOST=x86_64-w64-mingw32
MSYSTEM_PREFIX=/mingw64
NUMBER_OF_PROCESSORS=4
OPTERR=1
OPTIND=1
ORIGINAL_PATH='/mingw64/bin:/usr/bin:/c/Users/Alumno_ciclo/bin:/c/Program Files/
Common Files/Oracle/Java/javapath:/c/Windows/system32:/c/Windows:/c/Windows/Syst
em32/wbem:/c/Windows/System32/WindowsPowerShell/v1.0:/c/Windows/System32/OpenSSH
:/cmd:/c/Users/Alumno_ciclo/AppData/Local/Microsoft/WindowsApps:/c/Program Files
/JetBrains/IntelliJ IDEA 2024.2.3/bin:/c/Program Files/JetBrains/IntelliJ IDEA C
ommunity Edition 2024.2.2/bin:/c/Program Files/MySQL/MySQL Shell 8.0/bin:/c/Prog
ram Files/MySQL/MySQL Server 8.0/bin:/c/Users/Alumno_ciclo/AppData/Local/GitHubD
esktop/bin:/c/Users/Alumno_ciclo/AppData/Local/Programs/Microsoft VS Code/bin'
ORIGINAL_TEMP=/tmp
ORIGINAL_TMP=/tmp
OS=Windows_NT
OSTYPE=msys

```

- Muestra el contenido de la variable **PATH** (`echo %PATH%`) y de la variable **JAVA_HOME** (`echo %JAVA_HOME%`)

```

C:\\Users\\Alumno_ciclo>echo %path%
C:\\Program Files\\Common Files\\Oracle\\Java\\javapath;C:\\Windows\\system32;C:\\Windows;C:\\Windows\\System32\\Wbem;C:\\Windows\\Sy
stem32\\WindowsPowerShell\\v1.0;C:\\Windows\\System32\\OpenSSH\\;C:\\Program Files\\Git\\cmd;C:\\Users\\Alumno_ciclo\\AppData\\Local
\\Microsoft\\WindowsApps;C:\\Program Files\\JetBrains\\IntelliJ IDEA 2024.2.3\\bin;;C:\\Program Files\\JetBrains\\IntelliJ IDEA C
ommunity Edition 2024.2.2\\bin;;C:\\Program Files\\MySQL\\MySQL Shell 8.0\\bin\\;C:\\Program Files\\MySQL\\MySQL Server 8.0\\bin;C
:\\Users\\Alumno_ciclo\\AppData\\Local\\GitHubDesktop\\bin;C:\\Users\\Alumno_ciclo\\AppData\\Local\\Programs\\Microsoft VS Code\\bin'

C:\\Users\\Alumno_ciclo>

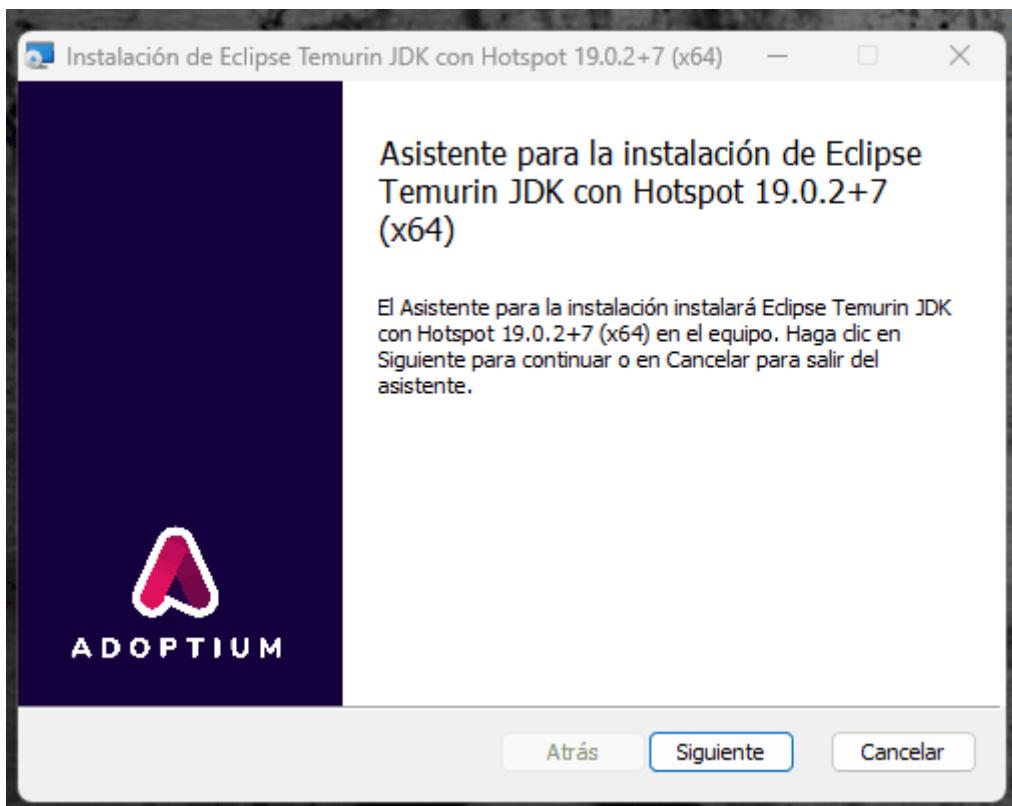
```

3. Instala el JDK 19 la implementación de Adoptium (Windows)

- Ves a la página de [Adoptium](#) y descarga la versión de Java 19 para Windows y la arquitectura de tu PC (x32/x64).

The screenshot shows the Adoptium website's latest releases page for Java 19 on Windows x64. The interface is dark-themed. At the top, it says "Latest Releases | Adoptium". Below that, it displays the message "instalación de paquetes están disponibles en la sección de documentación." A note below states "Usa la caja desplegable de abajo para listar la lista de las actuales liberaciones." A table allows filtering by "Sistema Operativo" (Windows), "Arquitectura" (x64), "Tipo de paquete" (Any), and "Version" (19). The table rows show the following data:

Sistema Operativo	Arquitectura	Tipo de paquete	Version
Windows	x64	Any	19
19.0.2+7 Temurin ✓	Windows	x64	JDK - 173 MB
			Checksum
			.msi
			JDK - 196 MB
			Checksum
.zip			
JRE - 33 MB			
Checksum			
.msi			
JRE - 45 MB			
Checksum			
.zip			



- Una vez instalado, muestra la versión de Java instalada y la ruta de instalación. (a través de comandos y adjunta una imagen) (`java -version` y `where java`)

```
Alumno_ciclo@PC-FP MINGW64 ~/Desktop
$ java -version
openjdk version "19.0.2" 2023-01-17
OpenJDK Runtime Environment Temurin-19.0.2+7 (build 19.0.2+7)
OpenJDK 64-Bit Server VM Temurin-19.0.2+7 (build 19.0.2+7, mixed mode, sharing)
h
Alumno_ciclo@PC-FP MINGW64 ~/Desktop
$ where java
C:\Program Files\Eclipse Adoptium\jdk-19.0.2.7-hotspot\bin\java.exe
C:\Program Files\Common Files\Oracle\Java\javapath\java.exe

Alumno_ciclo@PC-FP MINGW64 ~/Desktop
$
```

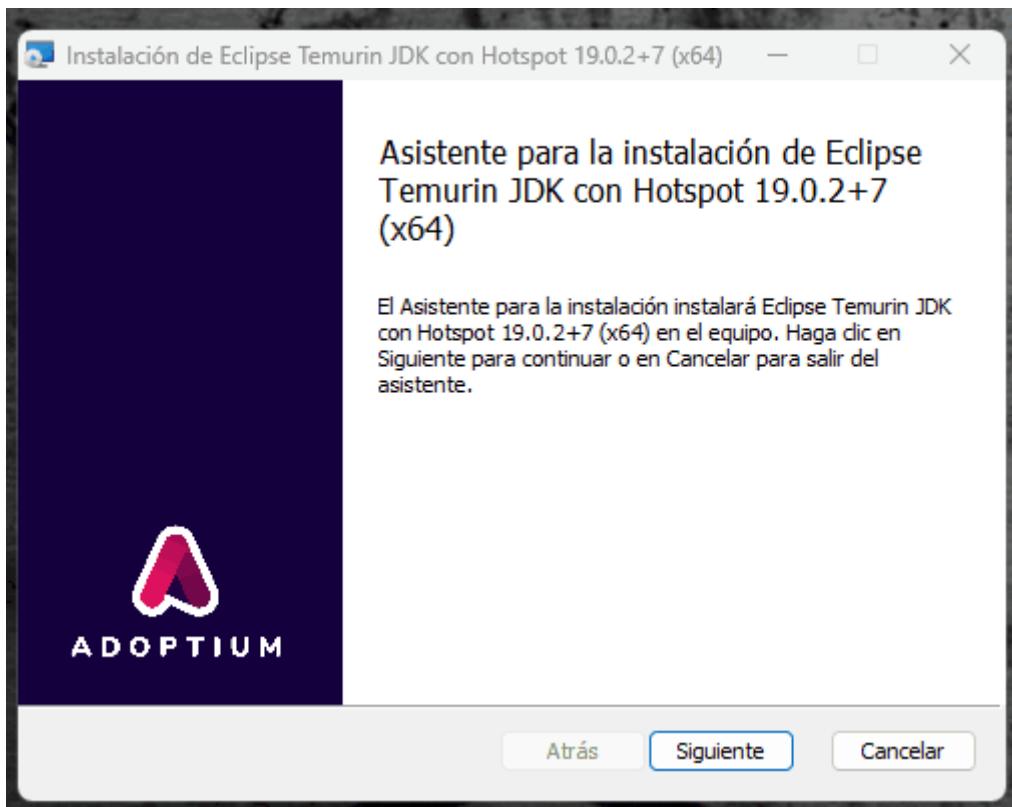
- ¿ La versión de Java que te muestra es la 19? ¿ Por qué?

Sí, por qué el asistente de instalación coloca la ruta en primer lugar dentro del "path" durante la instalación. Lo podemos ver directamente al observar que es la 1^a que devuelve el "where java", o podemos ir a la interfaz gráfica y observar el orden de la lista de valores de la variable (en editar variable).

4. Configura tu sistema para que utilice la versión de Java 19 como versión por defecto a nivel de usuario. (Si ya lo tienes explica por qué)

- ¿ Cómo has configurado tu sistema para que utilice la versión de Java 19 como versión por defecto?

Ya está por defecto al posicionarse la primera en la lista del "path", si no, bastaría con colocarla por encima de la ruta que ya existía. Lo he conseguido al leer todas las opciones del cuadro de instalación y observar que me preguntaba si directamente quería que el instalador la establecerla por defecto, señalando "Sí".



2. Utilización de SDKMan

5. Instala SDKMan en Windows. (Para ello puedes seguir la guía disponible [aqui](#))

- Instala SDKMan en Windows e explica los pasos que has seguido, adjunta una captura final de SDK funcionando.

Atendiendo al [contenido de apoyo para el SDK MAN](#); he buscado e instalado las librerías del protocolo Zip para mi terminal GitBash.

Aproximadamente 114.000 resultados

StackOverflow
<https://stackoverflow.com/questions/38782928/how-to-add-man-and-zip-to-git-bash-installation-on-windows>

How to add man and zip to "git bash" installation on Windows

Viewed 159k times

Question Answer

◀ 106 edited Jan 7, 2022 at 18:51

Here's another, slightly different, set of instructions to install `zip` for `git bash` on windows:

1. Navigate to this sourceforge page
2. Download `zip-3.0-bin.zip`
3. In the zipped file, in the `bin` folder, find the file `zip.exe`.
4. Extract the file `zip.exe` to your `mingw64` bin folder (for me: `C:\Program Files\Git\mingw64\bin`)
5. Navigate to this sourceforge page
6. Download `bzip2-1.0.5-bin.z...$]#]`

Ver más >

Content Under CC-BY-SA license

¿Te ha resultado útil?

Una vez habilitado el uso de Zip's en las terminales Bash, basta con utilizar el comando:

```
curl -s "https://get.sdkman.io" | bash
```

```

MINGW64:/c/Users/Gonzalo/ + - X
Identity added: C:\Users\Gonzalo\.ssh\id_rsa (glavadoj01@suarezdefigueroa.es)

Gonzalo@GonzaPC MINGW64 ~/Desktop
$ sdk

NAME
  sdk – The command line interface (CLI) for SDKMAN!

SYNOPSIS
  sdk <subcommand> [candidate] [version]

DESCRIPTION
  SDKMAN! is a tool for managing parallel versions of multiple JVM related Software Development Kits on most Unix based systems. It provides a convenient Command Line Interface (CLI) and API for installing, switching, removing and listing Candidates.

SUBCOMMANDS & QUALIFIERS
  help      [subcommand]
  install   <candidate> [version] [path]
  uninstall <candidate> <version>
  list      [candidate]
  use       <candidate> <version>
  config    no qualifier
  default   <candidate> [version]
  home     <candidate> <version>
  env      [init|install|clear]
  current   [candidate]
  upgrade   [candidate]
  version   no qualifier
  offline   [enable|disable]
  selfupdate [force]
  update    no qualifier
  flush     [tmp|metadata|version]

EXAMPLES
  $ sdk install java 17.0.0-tem
  $ sdk help install

Gonzalo@GonzaPC MINGW64 ~/Desktop
$
```

- Muestra la versión de SDKMan instalada

```

Gonzalo@GonzaPC MINGW64 ~/Desktop
$ sdk version

SDKMAN!
script: 5.19.0
native: 0.5.0
```

- ¿ Dónde se ha instalado SDKMan? ¿ Por qué?

En el directorio raíz del usuario activo. Porque se instala solo en esta sesión y para este usuario. Y porque es la ruta que ofrece el

instalador por defecto y me parece un buen lugar.

Nombre	Fecha de modificación	Tipo	Tamaño
bin	07/02/2025 11:31	Carpeta de archivos	
candidates	07/02/2025 11:31	Carpeta de archivos	
contrib	07/02/2025 11:31	Carpeta de archivos	
etc	07/02/2025 11:31	Carpeta de archivos	
ext	07/02/2025 11:31	Carpeta de archivos	
libexec	07/02/2025 11:31	Carpeta de archivos	
src	07/02/2025 11:31	Carpeta vacía	
tmp	07/02/2025 11:31	Carpeta de archivos	
var	07/02/2025 11:33	Carpeta de archivos	

- Muestra las versiones de Java que tienes instaladas a través de SDKMan

Actualmente ninguna. Como ya se mencionó en el apartado 1, solo tengo 2, la que se instalo por primera vez con el IDE IntelliJ, y la que he instalado manualmente ahora mismo (adoptium). Ninguna de ellas mediante sdk man y por tanto, no estan en las carpetas "current" ni "candidate" del sdk.

Y tras aclaraciones del profesor: hasta que configuremos la integración (puntos siguientes de este trabajo); tienen rutas y procedimientos independientes (IDE y terminales Bash)

```
Gonzalo@GonzaPC MINGW64 ~/Desktop
$ sdk list java | grep install
Omit Identifier to install default version 21.0.6-tem:
    $ sdk install java
    $ sdk install java [TAB]
Or install a specific version by Identifier:
    $ sdk install java 21.0.6-tem
```

```
Gonzalo@GonzaPC MINGW64 ~/Desktop
$ |
```

- ¿Qué ventajas tiene instalar SDKMan?

Tiene la gran ventaja de que se configura una sola vez y tras esto, nos permite cambiar entre versiones Java con un comando sencillo. Centralizando todo el procedimiento (busqueda, actualización, instalación, cambio manual y automático, borrado) en entorno de consola sencillo y funcional.

- ¿ Instala la versión de Jara 8.0.302-zulu a través de SDKMan ?

Varias versiones requeridas no se encuentran disponibles actualmente. Por lo que se opta por instalar la versión más antigua disponible para cada build.

```
Gonzalo@GonzaPC MINGW64 ~/Desktop
$ sdk list java | grep zulu
Zulu          | 23.0.2.fx    | zulu           | 23.0.2.fx-zulu
               | 23.0.2        | zulu           | 23.0.2-zulu
               | 23.0.1.fx    | zulu           | 23.0.1.fx-zulu
               | 23.0.1        | zulu           | 23.0.1-zulu
               | 21.0.6.fx    | zulu           | 21.0.6.fx-zulu
               | 21.0.6        | zulu           | 21.0.6-zulu
               | 21.0.5.fx    | zulu           | 21.0.5.fx-zulu
               | 21.0.5        | zulu           | 21.0.5-zulu
               | 17.0.14.fx   | zulu           | 17.0.14.fx-zulu
               | 17.0.14       | zulu           | 17.0.14-zulu
               | 17.0.13.fx   | zulu           | 17.0.13.fx-zulu
               | 17.0.13       | zulu           | 17.0.13-zulu
               | 11.0.26.fx   | zulu           | 11.0.26.fx-zulu
               | 11.0.26       | zulu           | 11.0.26-zulu
               | 11.0.25.fx   | zulu           | 11.0.25.fx-zulu
               | 11.0.25       | zulu           | 11.0.25-zulu
               | 8.0.442.fx   | zulu           | 8.0.442.fx-zulu
               | 8.0.442       | zulu           | 8.0.442-zulu
               | 8.0.432.fx   | zulu           | 8.0.432.fx-zulu
               | 8.0.432       | zulu           | 8.0.432-zulu
               | 8.0.45        | zulu           | 8.0.45-zulu
               | 7.0.352       | zulu           | 7.0.352-zulu
               | 6.0.119       | zulu           | 6.0.119-zulu
```

```
Gonzalo@GonzaPC MINGW64 ~/Desktop
$ |
```

Se opta por la 8.0.45-zulu

- ¿ Instala la versión de Java 11.0.12-zulu a través de SDKMan ?

Se opta por la 11.0.25-zulu

- ¿ Instala la versión de Java 17.0.0-zulu a través de SDKMan ?

Se opta por la 17.0.13-zulu

```
Gonzalo@GonzaPC MINGW64 ~/Desktop
$ sdk install java 8.0.45-zulu

Downloading: java 8.0.45-zulu
In progress...
#####
100.0%


Installing: java 8.0.45-zulu
Done installing!


Setting java 8.0.45-zulu as default.

Gonzalo@GonzaPC MINGW64 ~/Desktop
$ sdk install java 11.0.25-zulu

Downloading: java 11.0.25-zulu
In progress...
#####
100.0%


Installing: java 11.0.25-zulu
Done installing!


Setting java 11.0.25-zulu as default.

Gonzalo@GonzaPC MINGW64 ~/Desktop
$ sdk install java 17.0.13-zulu

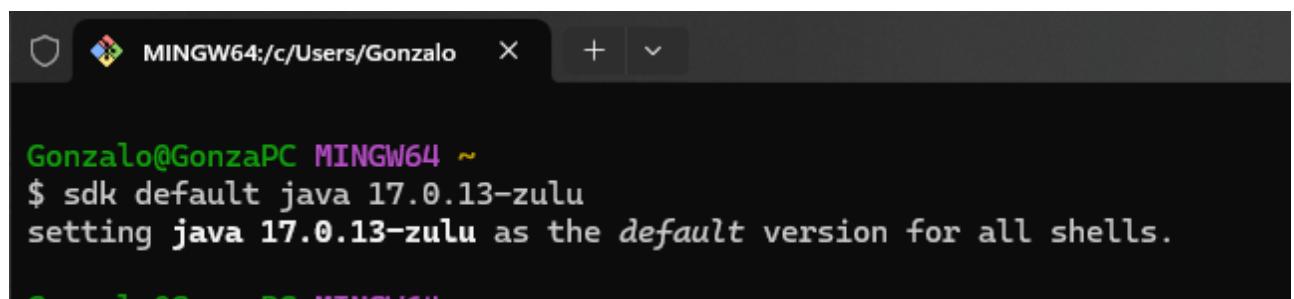
Downloading: java 17.0.13-zulu
In progress...
#####
100.0%


Installing: java 17.0.13-zulu
Done installing!


Setting java 17.0.13-zulu as default.

Gonzalo@GonzaPC MINGW64 ~/Desktop
```

6. Configura tu sistema para que utilice la versión de Java 17.0.0 como versión por defecto a nivel de usuario. (Para que las aplicaciones que ejecutes utilicen esta versión de Java)



```
Gonzalo@GonzaPC MINGW64 ~
$ sdk default java 17.0.13-zulu
setting java 17.0.13-zulu as the default version for all shells.
```

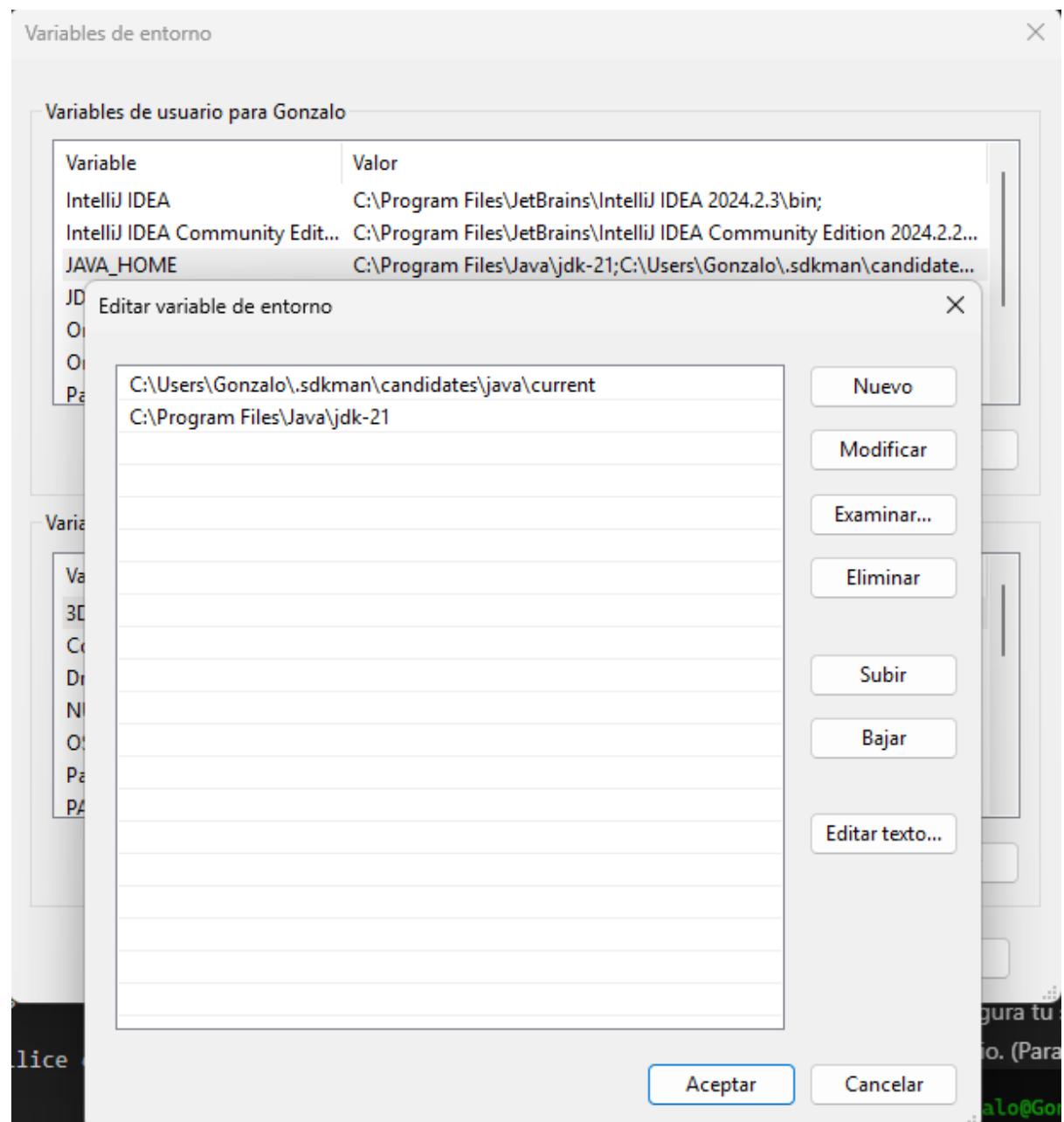
Hay que usar una terminal con permisos de administrador

- ¿Qué tienes hacer o comando tienes que utilizar (SDKMAN) para que una aplicación ejecutada desde la interfaz (Windows o Linux) utilice esa versión de Java?

```
sdk use java <IdVersión>
```

- ¿Qué variable de Entorno tienes que modificar para que una aplicación ejecutada desde la interfaz (Windows o Linux) utilice esa versión de Java? Hay que modificar la variable "\$Java_Home" para que apunte al directorio:

```
"C:\Users\Gonzalo\.sdkman\candidates\java\current"
```



7. Si necesitas compilar una aplicación de Java desde la terminal, fuera del IDE, y necesita compilarse con la versión de Java 8, ¿ Cómo lo harías?

Con SDK Man y el comando "javac"

- ¿ Qué comando de SDKMAN tienes que utilizar para que a nivel de la terminal actual use la versión de Java 8?

sdk use java 8.0.45-zulu

- ¿ Qué comando utilizas para compilar una aplicación de Java ?

javac RutaMiPrograma\MiPrograma.java

8. Un proyecto en el que estas trabajando, necesita la versión de Java 11, pero requieres compilarlo con esa versión, pero no quieres tener siempre que recordar esto, y quieres que se active automáticamente esa versión una vez accedas al directorio del proyecto.

- ¿ Cómo puedes realizar esto con SDKMAN ? (indica los comandos que tienes que utilizar y la configuración de la herramienta)

1º - Activamos la versión de Java que buscamos:

```
 sdk use java 11.0.25-zulu
```

2º - Hay que marcar el directorio del proyecto objetivo como "Monitoreado por SDK Man"

Para ello, utilizamos el comando "sdk env init" en el directorio en cuestión.

The screenshot shows a terminal window titled 'MINGW64:/c/Users/Gonzalo/'. The left sidebar lists files: '.idea', 'nexus', '.sdkmanrc', and 'softmar.iml'. The terminal window displays the following command sequence:

```
Gonzalo@GonzaPC MINGW64 ~  
$ cd OneDrive/Carpeta_Git/softmar  
Gonzalo@GonzaPC MINGW64 ~/OneDrive/Carpeta_Git/softmar  
$ sdk use java 11.0.25-zulu  
Using java version 11.0.25-zulu in this shell.  
Gonzalo@GonzaPC MINGW64 ~/OneDrive/Carpeta_Git/softmar  
$ sdk env init  
.sdkmanrc created.  
Gonzalo@GonzaPC MINGW64 ~/OneDrive/Carpeta_Git/softmar  
$ |
```

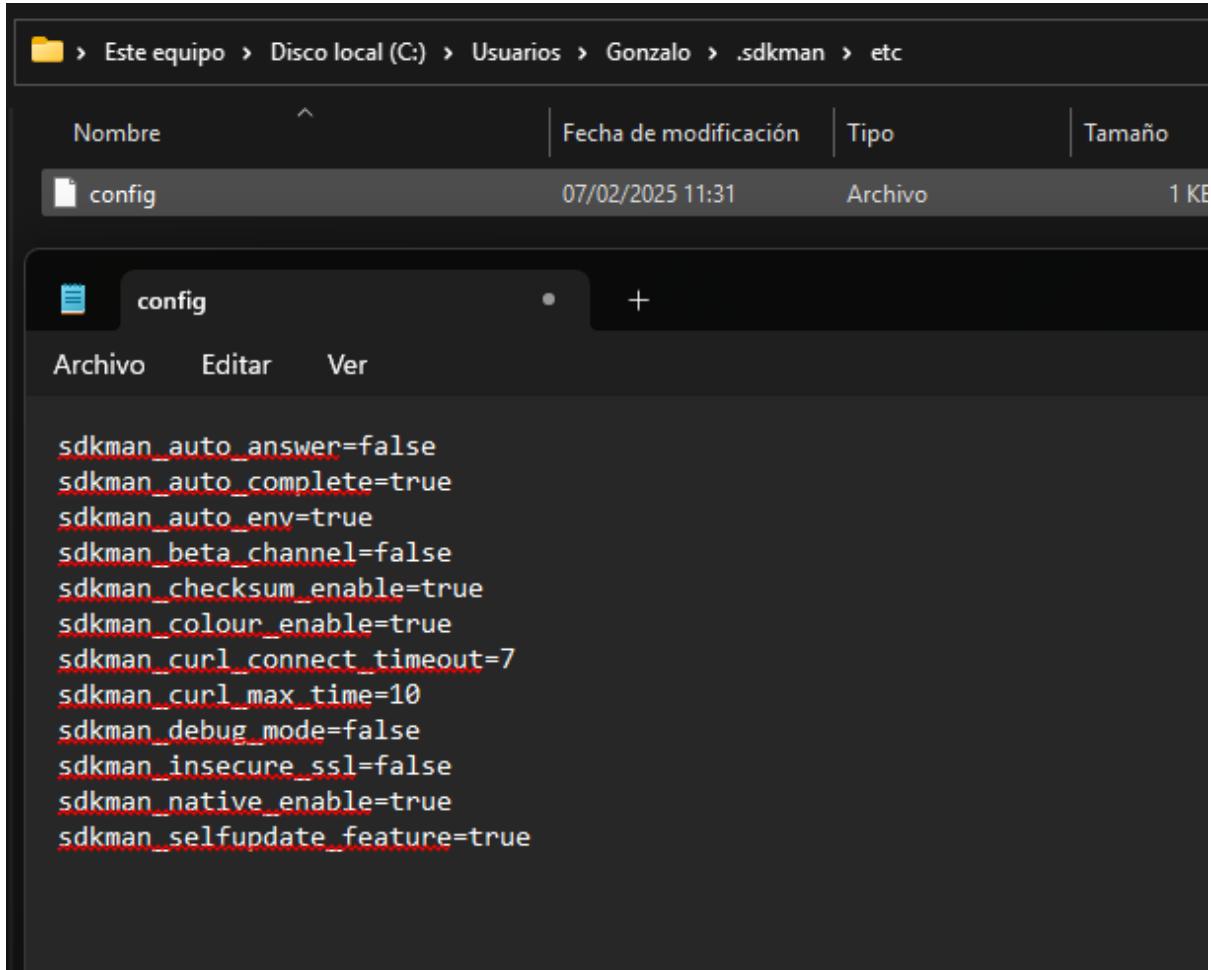
3º - Hay que activar la funcionalidad para el autocambio de la version de java mediante sdk env

How

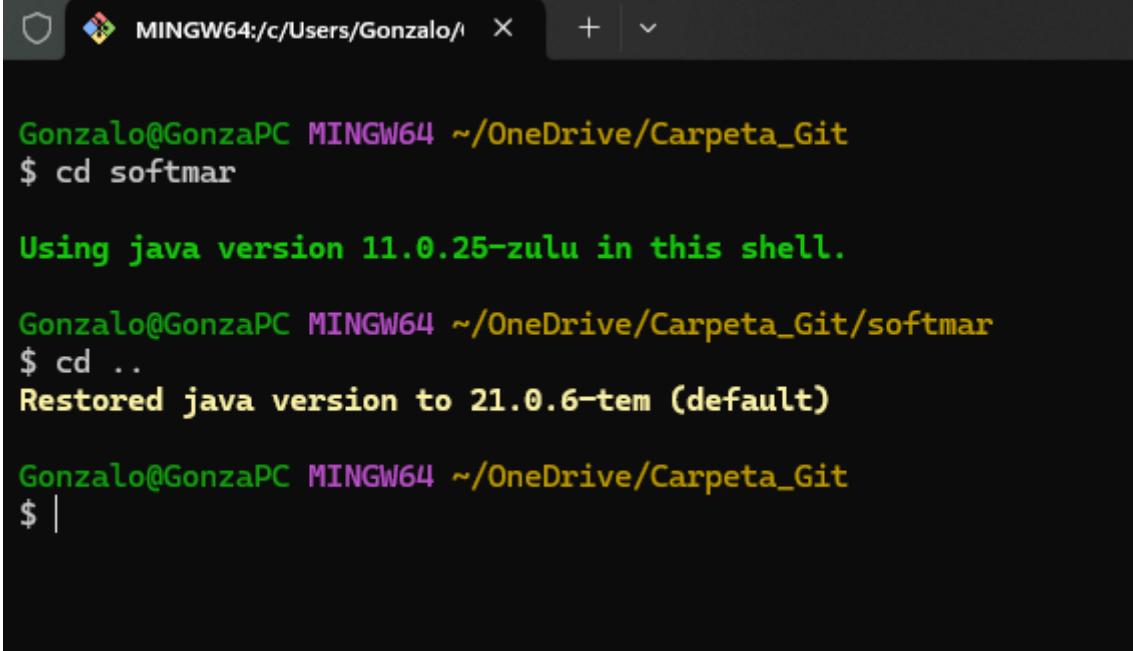
First of all, we need to enable the `auto_env` function, which will make the exchange happen whenever the directory is accessed via the terminal

To do this, edit the config file inside `~/.sdkman/etc/config` and change the value for `auto_env` to `true`, or, use this command where `$HOME` is your home folder

```
sed -i 's/sdkman_auto_env=false/sdkman_auto_env=true/' "$HOME/.sdkman/etc/config"
```



- Haz una captura de pantalla entrando y saliendo del directorio del proyecto, para ver como se activa y desactiva una versión y otra de Java.



```
Gonzalo@GonzaPC MINGW64 ~/OneDrive/Carpeta_Git
$ cd softmar

Using java version 11.0.25-zulu in this shell.

Gonzalo@GonzaPC MINGW64 ~/OneDrive/Carpeta_Git/softmar
$ cd ..
Restored java version to 21.0.6-tem (default)

Gonzalo@GonzaPC MINGW64 ~/OneDrive/Carpeta_Git
$ |
```

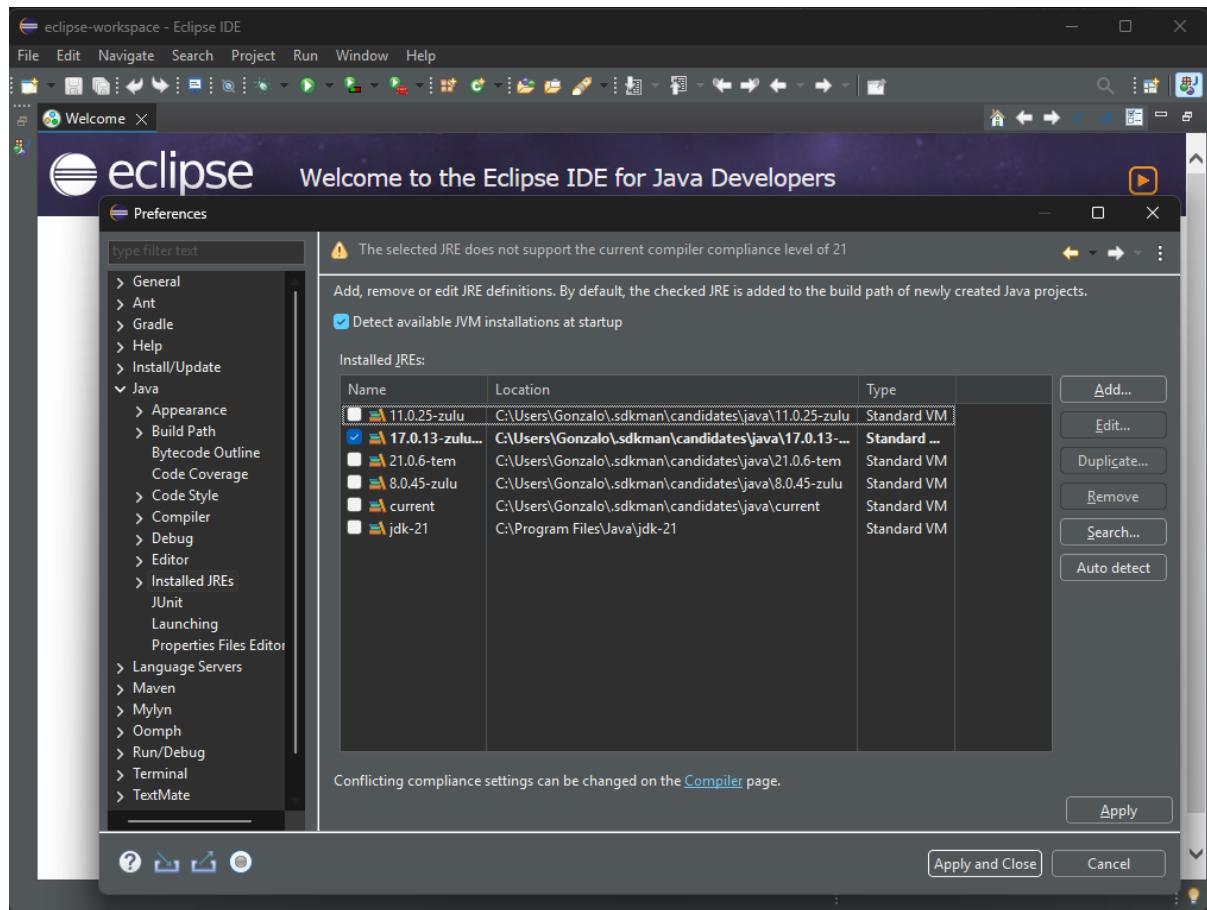
9. Ahora en Eclipse, configura el JDK 17 descargado con SDKMAN, como JDK por defecto.

- ¿ Cómo has configurado Eclipse para que utilice el JDK 17 descargado con SDKMAN? (Muestra una captura de pantalla)

Pues lo 1º de todo, descargarando e instalando Eclipse. Tras ello, acudimos de nuevo a la documentación de referencia para sdk man

Contenido de apoyo para el SDK MAN

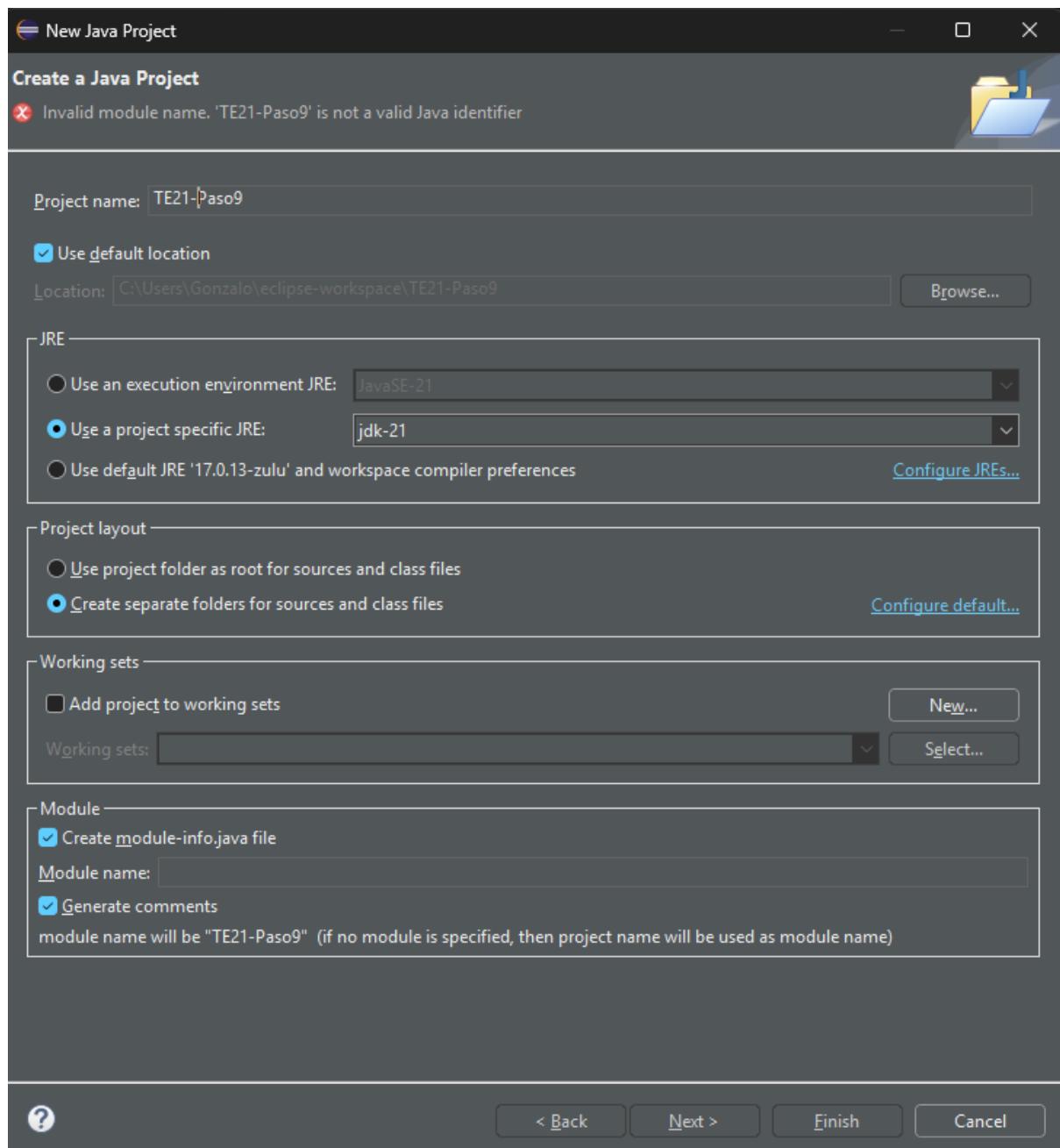
En la documentación podemos ver que podemos seleccionar la JDK por defecto desde el menú de preferencias, seleccionando pues la versión 17 alojada en el directorio .sdk/candidates/java



- Inicia un nuevo proyecto (TE21-Paso9) en Eclipse y muestra la versión de Java que aparece por defecto para el Workspace. (Muestra una captura de pantalla)

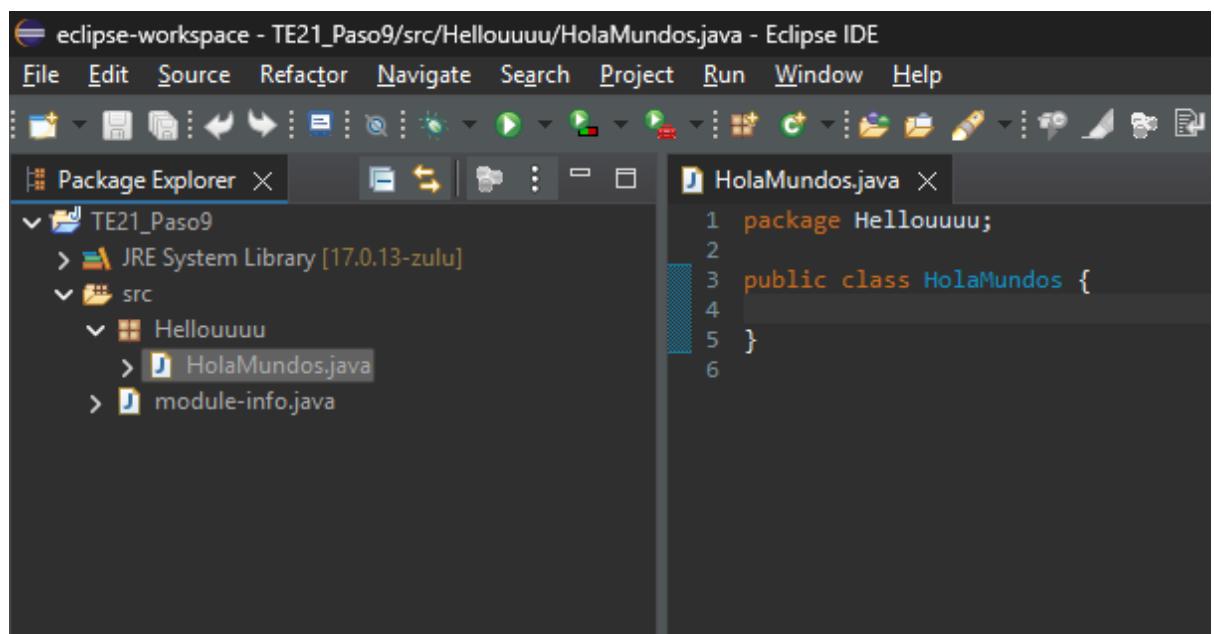
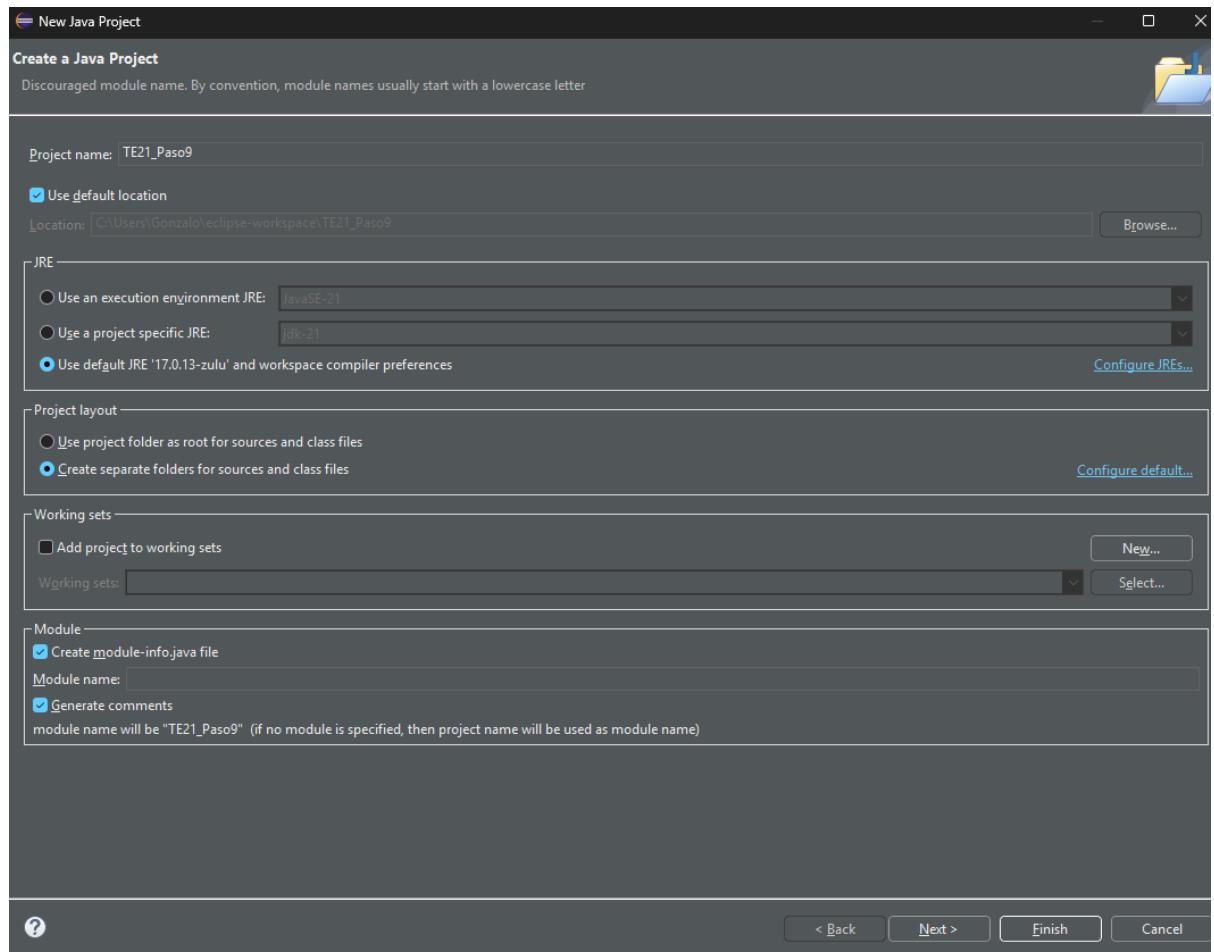
Pues en este paso nos encontramos con un problema de nomenclatura: El guión en el nombre del proyecto.

Eclipse no nos permite crear un proyecto con ese nombre. Por lo que se opta por sustituirlo por una "barra baja"; quedando el nombre como: "TE21_Paso9"

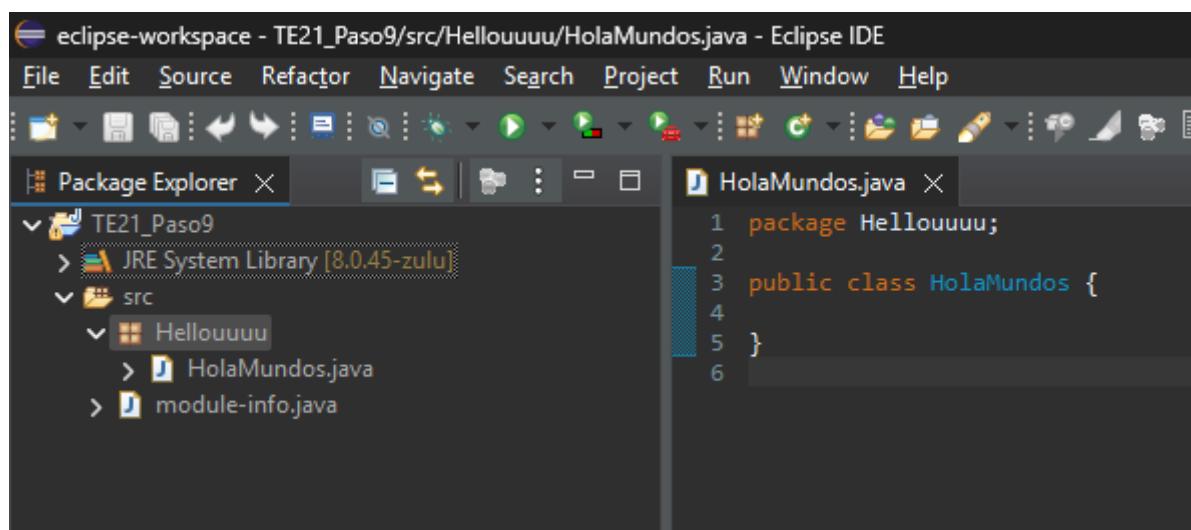
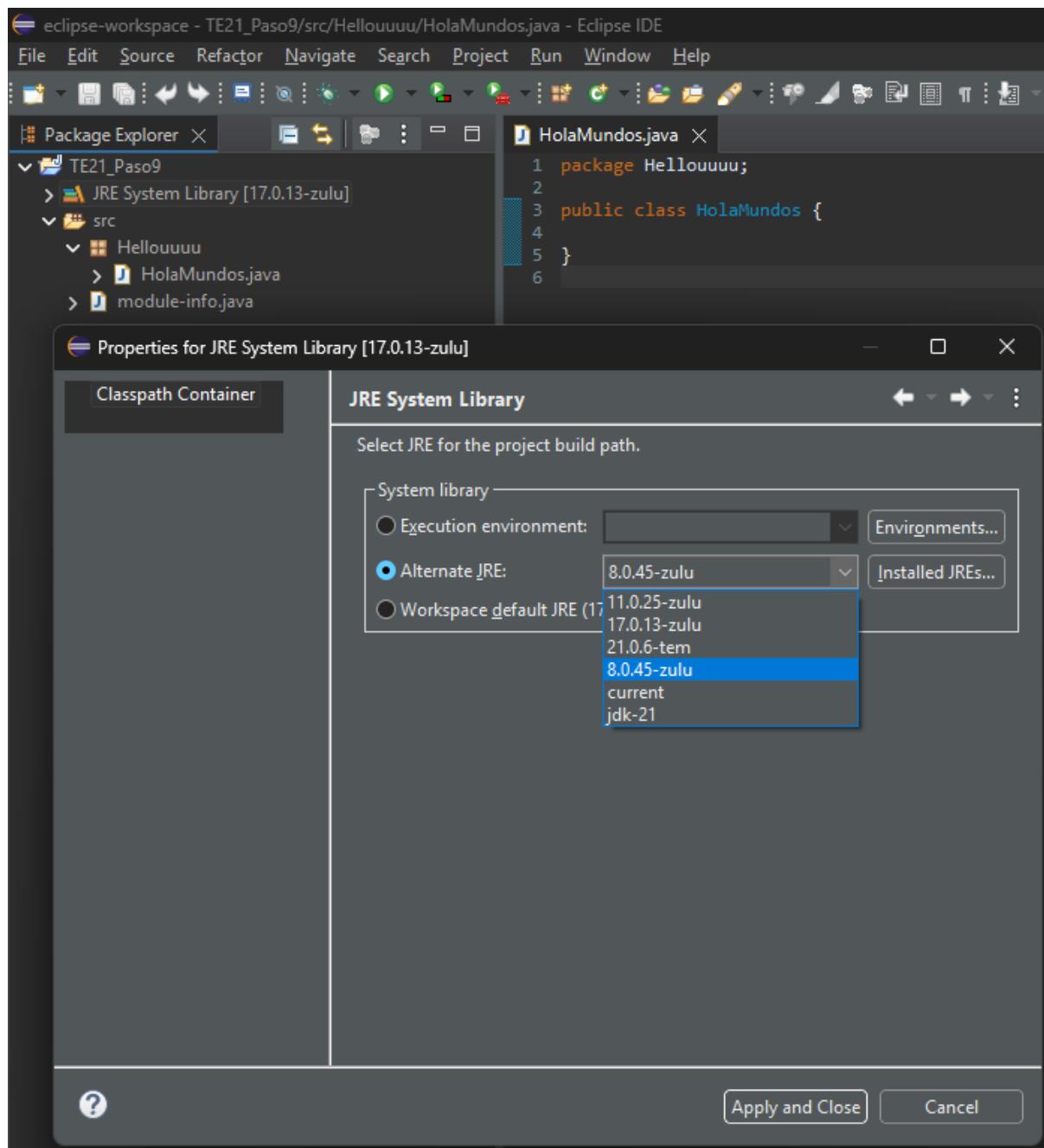


Ademas, tambien vemos, que nos pregunta que JDK queremos utilizar, por lo que seleccionamos la opción:
"Use default JRE zulu-17"
Que es la que acabamos de configurar en el paso previo.

Curiosamente, la opción intermedia que detecta con una jdk-21; hace referencia, a la JDK que se encuentra en "C:\Program Files\Java\" que viene a ser la que se intala por defecto con IntelliJ. A pesar de que actualmente, no se encuentra en el PATH ni en ninguna variable de entorno.



- Cambia la versión de Java del proyecto para que utilice la versión de Java 8. (Muestra una captura de pantalla)



3. Utilización de JetBrains IntelliJ IDEA y Eclipse

10. Crea un nuevo proyecto en IntelliJ IDEA (TE21-Paso10) y configura en ese directorio, con SDKMAN para que utilice la versión de Java 11.

The screenshot shows the IntelliJ IDEA interface. On the left, the Project tool window displays a project named 'TE21-Paso10' located at 'C:\Users\Gonzalo\OneDrive\IntelliJ Proyectos'. Inside the project, there is an '.idea' folder and a 'TE21-Paso10' directory, which contains an '.sdkmanrc' file. This file is selected and shown in the code editor on the right. The content of the file is:

```
# Enable auto-env through the sdkman_auto_env config
# Add key=value pairs of SDKs to use below
java=11.0.25-zulu
```

Below the code editor is a terminal window titled 'Local'. It shows the following command history:

```
Gonzalo@GonzaPC MINGW64 ~/OneDrive/IntelliJ Proyectos/TE21-Paso10 (main)
$ sdk use java 11.0.25-zulu

Using java version 11.0.25-zulu in this shell.

Gonzalo@GonzaPC MINGW64 ~/OneDrive/IntelliJ Proyectos/TE21-Paso10 (main)
$ sdk env init
.sdkmanrc created.

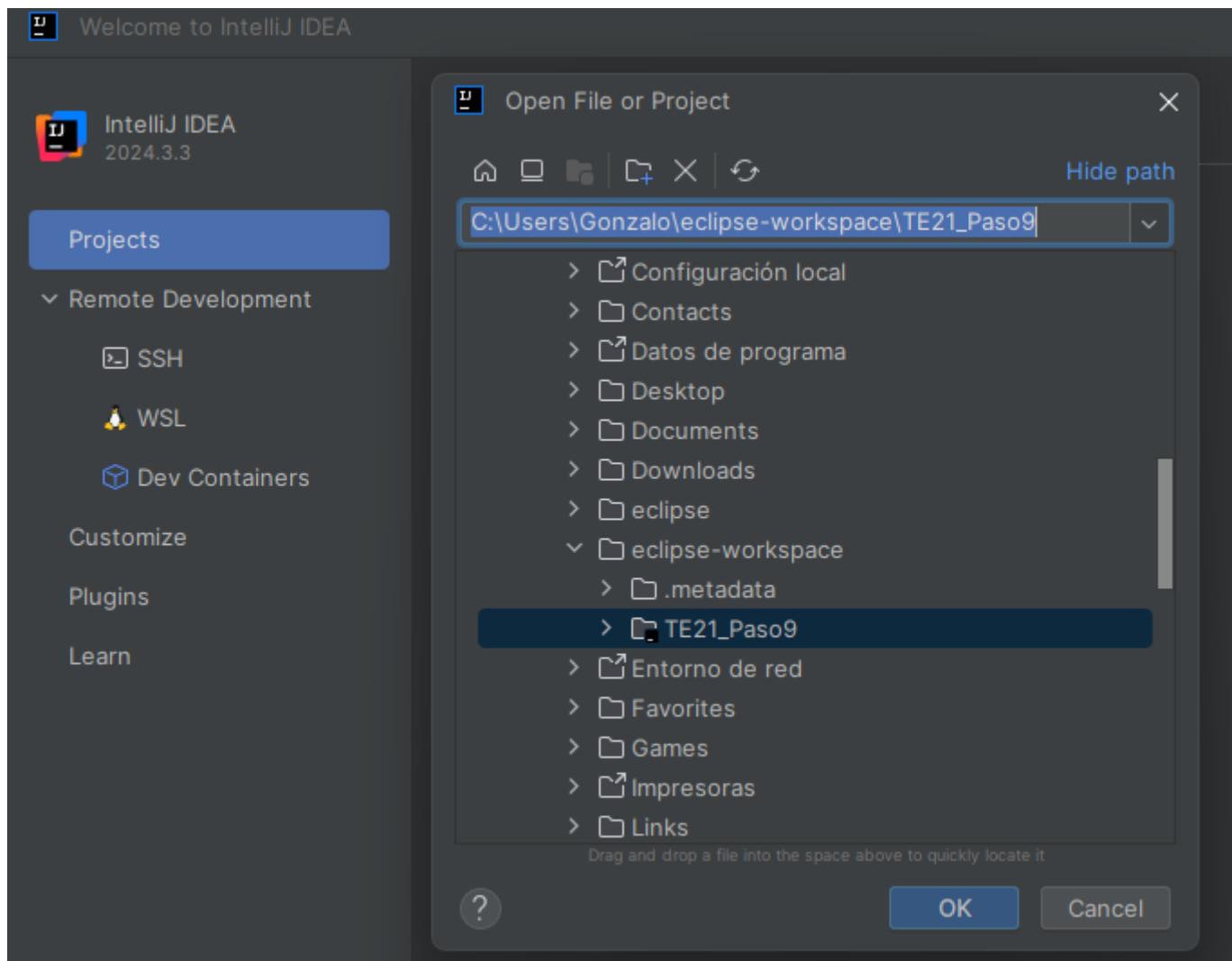
Gonzalo@GonzaPC MINGW64 ~/OneDrive/IntelliJ Proyectos/TE21-Paso10 (main)
$
```

- Ahora al abrir IntelliJ IDEA, debe activar esa versión automáticamente, pues detectar la configuración.
(Incluye una captura de pantalla o GIF de la configuración)

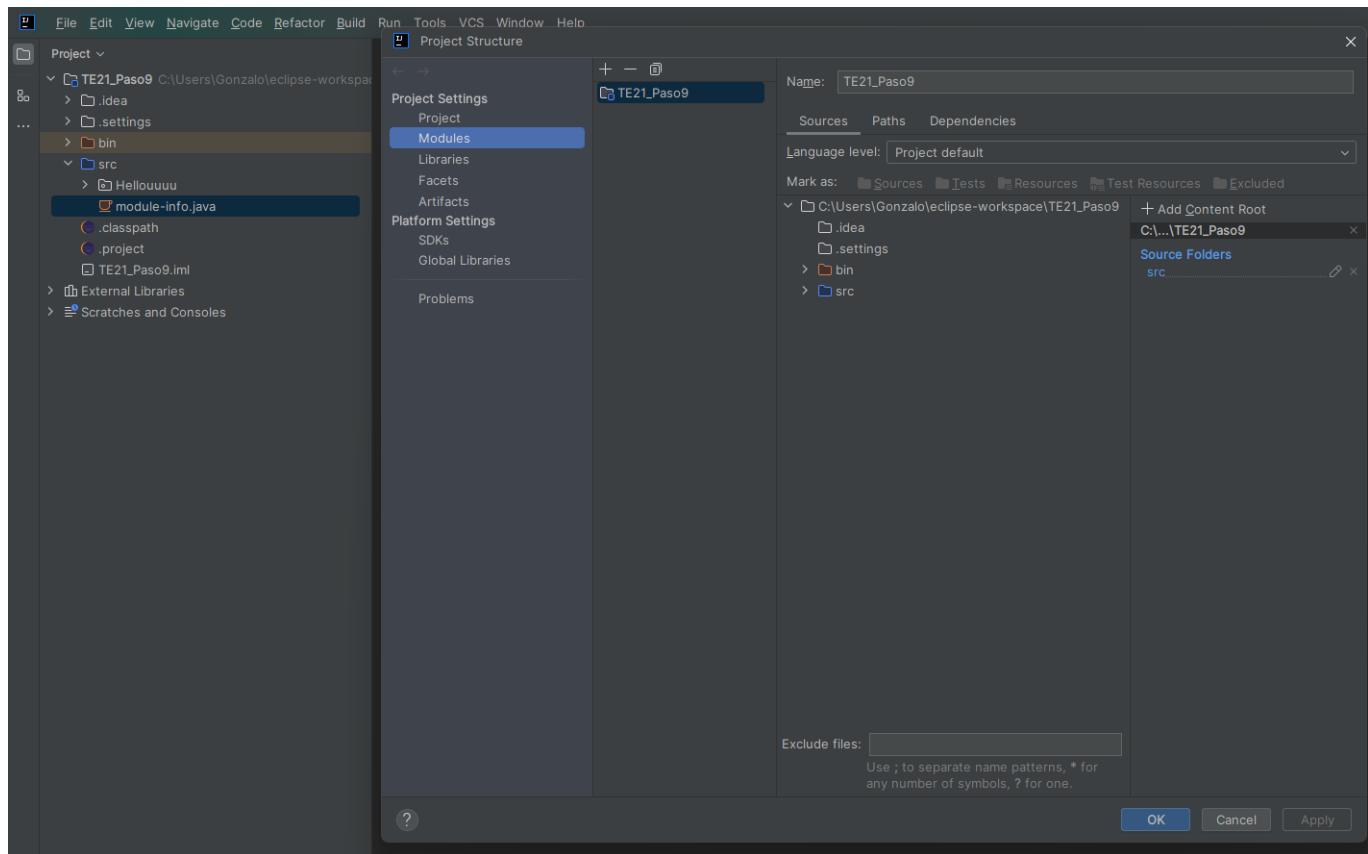


11. Importar el proyecto TE21-Paso9 en IntelliJ IDEA que has creado en Eclipse.

Para realizar este paso, ademas de abrir el proyecto desde IntellIJ, debemos de referencias la dependencia a un modulo externo.

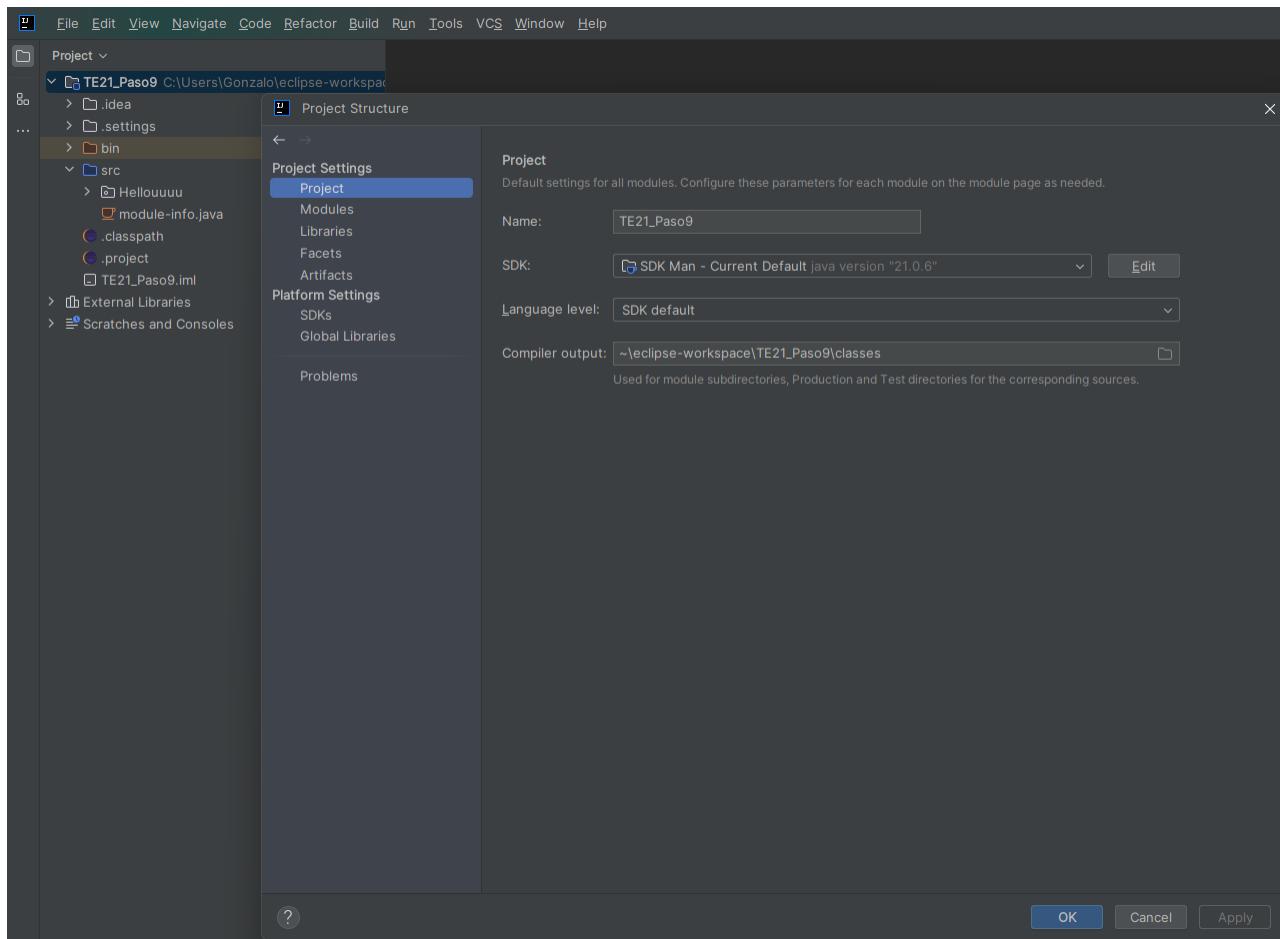


Ademas, se puede concretar y señalar que proviene de Eclipse.



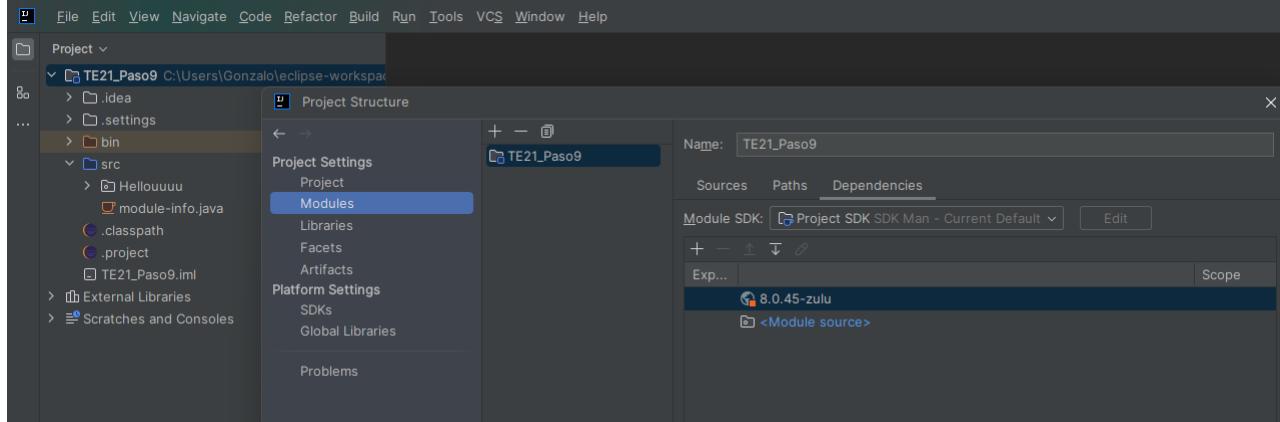
- Revisa la configuración de la versión de Java que utiliza el proyecto ¿Es la misma que utiliza Eclipse?. (Muestra una captura de pantalla)

No, no es la misma configuración. La que se ha activado es la version DEFAULT del SDKMAN. Que es la que esta seleccionada como predeterminada para IntelliJ IJ



Explica según tu opinión y en base a la configuración aplicada al proyecto de Eclipse realizada en el paso 9, si debe ser la misma versión de JDK en ambos proyectos o si esto depende de otras configuraciones externas al proyecto.

En mi opinión, el proceso es correcto. Ya que a cada IDE le hemos dado unas configuraciones por defecto. Salvo que se encuentren un archivo .sdkmanrc que les indique lo contrario. Y en concreto, para el proyecto eclipse, no creamos dicho archivo de configuración.



Ademas, podemos observar en las dependencias del Modulo importado de Eclipse, se indica que este a su vez depende de la SDK 8-zulu.

12. Crea un nuevo proyecto en IntelliJ IDEA (TE21-Paso12) que se guarde en la carpeta TE21-Paso12.

- Configura el proyecto para que utilice la versión de Java 17 descargada con SDKMAN. (Muestra una captura de pantalla de la configuración del fichero .sdkmanrc)

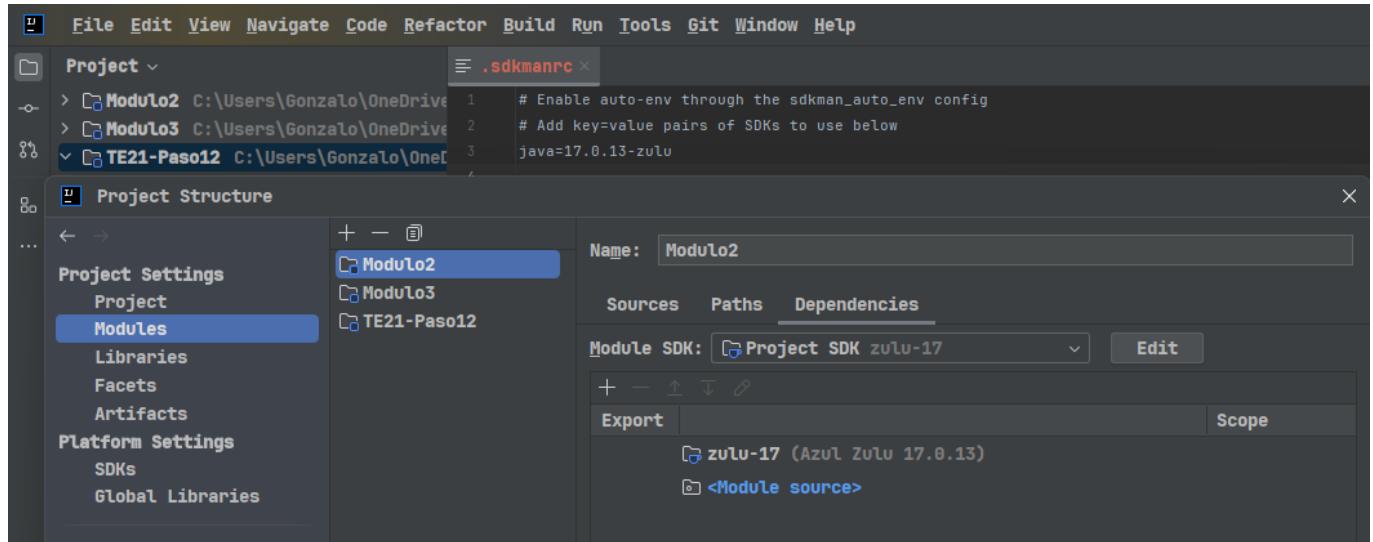
The screenshot shows the IntelliJ IDEA interface. In the top navigation bar, the project name is 'TE21-Paso12'. The left sidebar shows the project structure with 'src' and 'Main' modules, and files like '.gitignore' and '.sdkmanrc'. The right panel displays the contents of the '.sdkmanrc' file:

```
# Enable auto-env through the sdkman_auto_env config
# Add key=value pairs of SDKs to use below
java=17.0.13-zulu
```

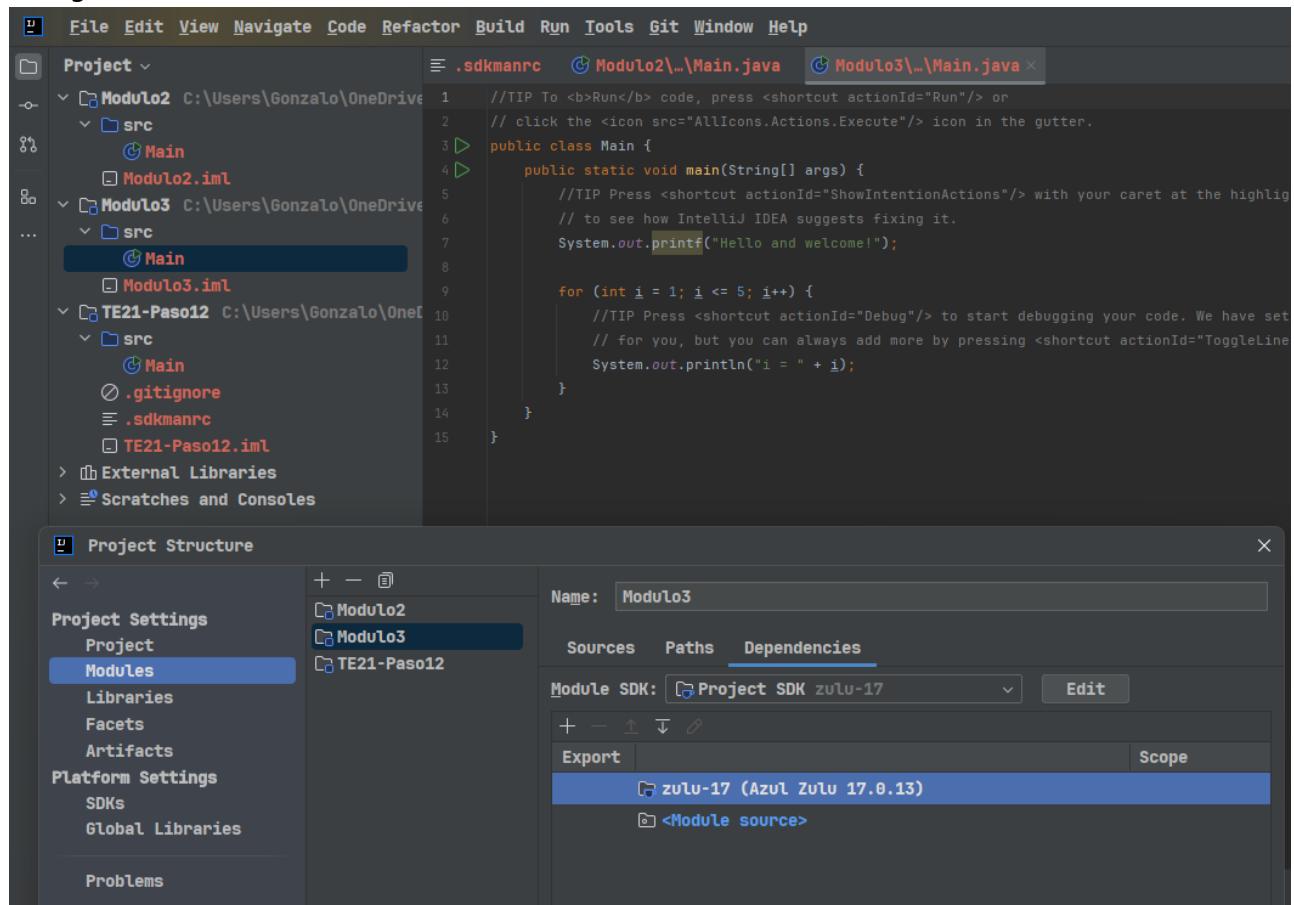
Below the editor, there is a terminal window titled 'Local'. It shows the command \$ sdk env init followed by the message '.sdkmanrc created.'.

- Agrega otro módulo al proyecto, que se guarde en la carpeta Modulo2.
- Agrega otro módulo al proyecto, que se guarde en la carpeta Modulo3.

(Muestra una captura de pantalla de la estructura del proyecto en IntelliJ IDEA)



- Vincula el proyecto principal, con los módulos 2 y 3. (Muestra una captura de pantalla de la configuración de los módulos)



- En el módulo 2, crea una clase que se llame **Utilidades** y que tenga un método que se llame **calculadora** y que tenga los métodos de suma, resta, multiplicación y división.

(Muestra el código de la clase **Utilidades** con un bloque de código)

```
Gonzalo@GonzaPC MINGW64 ~/OneDrive/IntelliJ Proyectos/TE21-Paso12 (main)
$ cat -n Modulo2/src/utilidades/Utilidades.java
 1 package utilidades;
 2
 3 public class Utilidades {
 4     public static double calculadora(double a, double b, char op) {
 5         if (op == '+') {
 6             return a + b;
 7         }
 8         else if (op == '-') {
 9             return a - b;
10         }
11         else if (op == '*') {
12             return a * b;
13         }
14         else if (op == '/') {
15             return a / b;
16         }
17         else if (op == '%') {
18             return a % b;
19         }
20         System.out.println("Operación Incorrecta");
21         return 0;
22     }
23 }
```

```
Gonzalo@GonzaPC MINGW64 ~/OneDrive/IntelliJ Proyectos/TE21-Paso12 (main)
$
```

14. En el módulo 3, crea una clase llamada **Conversor** que tenga un método que se llame **Texto_to_Uppercase** que convierta un texto a mayúsculas, y otro método que se llame **Texto_to_Lowercase** que convierta un texto a minúsculas.

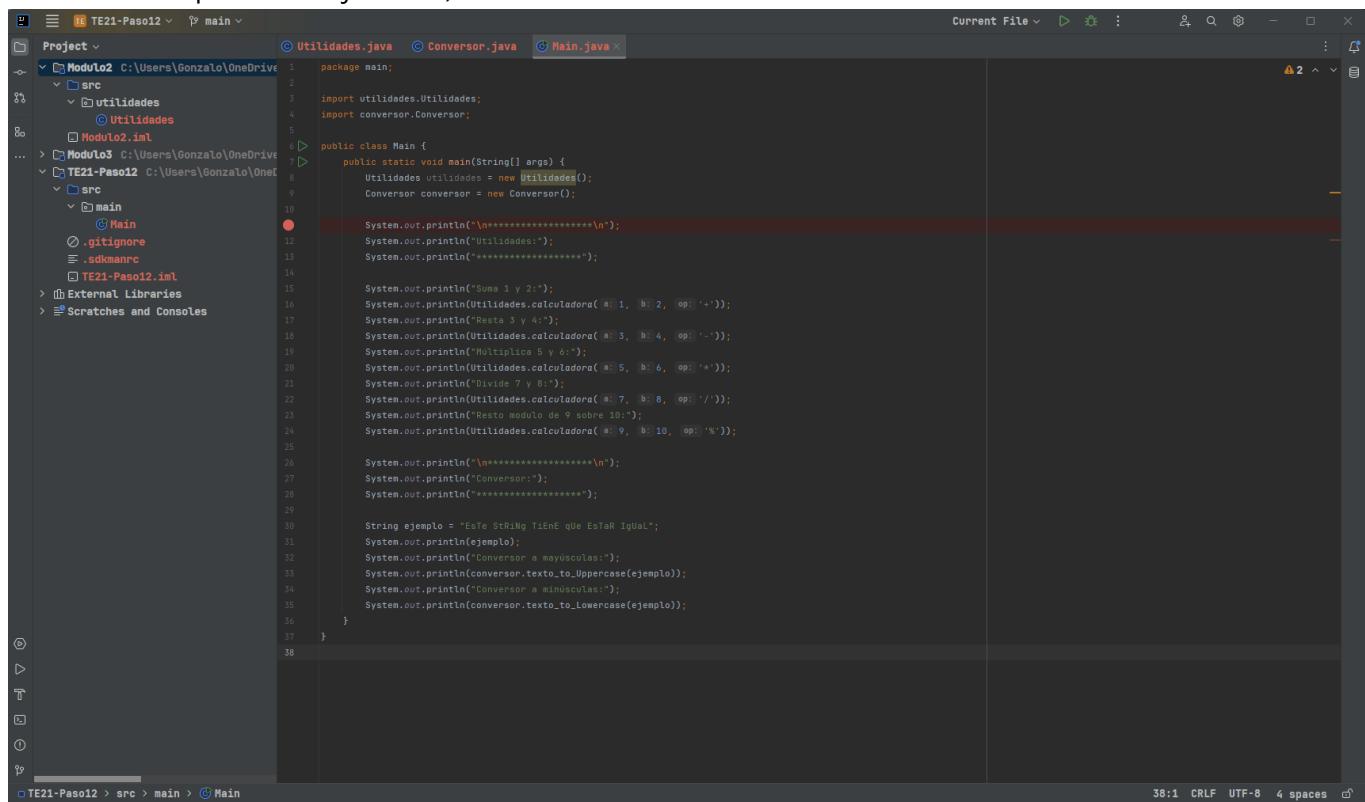
(Muestra el código de la clase **Conversor** con un bloque de código)

```
Gonzalo@GonzaPC MINGW64 ~/OneDrive/IntelliJ Proyectos/TE21-Paso12 (main)
$ cat -n Modulo3/src/conversor/Conversor.java
 1 package conversor;
 2
 3 public class Conversor {
 4     public String texto_to_Uppercase(String texto) {
 5         return texto.toUpperCase();
 6     }
 7
 8     public String texto_to_Lowercase(String texto) {
 9         return texto.toLowerCase();
10    }
11 }

Gonzalo@GonzaPC MINGW64 ~/OneDrive/IntelliJ Proyectos/TE21-Paso12 (main)
$
```

15. En el módulo principal, crea una clase llamada **Principal** que tenga un método **main** que instancie las clases **Utilidades** y **Conversor** y que muestre por consola el resultado de las operaciones de la clase **Utilidades** y el resultado de las operaciones de la clase **Conversor**.

(Muestra un gif donde se muestre la ejecución del programa, en depuración y se visualice que no existen errores de compilación ni ejecución).



The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows two modules: **Modulo2** and **Modulo3**. **Modulo3** contains a **src** directory with **Utilidades** and **Conversor** packages.
- Code Editor:** The file **Main.java** is open, showing Java code that instantiates **Utilidades** and **Conversor** classes and prints their results to the console.
- Status Bar:** Shows the file is 38:1, uses CRLF line endings, is in UTF-8 encoding, and has 4 spaces for indentation.

```
package main;
import utilidades.Utilidades;
import conversor.Conversor;

public class Main {
    public static void main(String[] args) {
        Utilidades utilidades = new Utilidades();
        Conversor conversor = new Conversor();

        System.out.println("\n*****\n");
        System.out.println("Utilidades:");
        System.out.println("*****");

        System.out.println("Suma 1 y 2:");
        System.out.println(utilidades.calculadora( a: 1, b: 2, op: '+' ));
        System.out.println("Resta 3 y 4:");
        System.out.println(utilidades.calculadora( a: 3, b: 4, op: '-' ));
        System.out.println("Multiplica 5 y 6:");
        System.out.println(utilidades.calculadora( a: 5, b: 6, op: '*' ));
        System.out.println("Divide 7 y 8:");
        System.out.println(utilidades.calculadora( a: 7, b: 8, op: '/' ));
        System.out.println("Resto modulo de 9 sobre 10:");
        System.out.println(utilidades.calculadora( a: 9, b: 10, op: '%' ));

        System.out.println("\n*****\n");
        System.out.println("Conversor:");
        System.out.println("*****");

        String ejemplo = "Este String Tiene que Estar Igual";
        System.out.println(ejemplo);
        System.out.println("Conversor a mayusculas:");
        System.out.println(conversor.texto_to_Uppercase(ejemplo));
        System.out.println("Conversor a minusculas:");
        System.out.println(conversor.texto_to_Lowercase(ejemplo));
    }
}
```