

Nombre completo	Secuencia	Fecha	Calificación
Pérez Galeana Jemmy Alondra	6NM61	5/03/2025	

## PRÁCTICA 0. CONFIGURACIÓN DE KOTLIN

### I. OBJETIVO DE LA PRÁCTICA.

Al término de la práctica, el alumno tendrá un entorno de desarrollo de aplicaciones para consola con el lenguaje de programación Kotlin para poder poner en práctica las prácticas iniciales en esta tecnología.

### II. MATERIAL Y EQUIPO.

- Equipo de cómputo con sistema operativo Linux / Windows.

### III. INTRODUCCIÓN.

Kotlin es un lenguaje moderno y versátil que permite desarrollar aplicaciones de manera eficiente. En esta práctica, el alumno configurará un entorno de desarrollo para ejecutar programas de consola en Kotlin. Esto le permitirá familiarizarse con la sintaxis del lenguaje y realizar sus primeras implementaciones. Además, conocerá las herramientas necesarias para compilar y ejecutar código desde la terminal. Con esta base, podrá avanzar en el aprendizaje y aplicación de Kotlin en distintos proyectos..

### IV. DESARROLLO DE LA PRÁCTICA.

1. Desde la terminal de tu sistema operativo ejecuta los siguientes comandos:

- a. WINDOWS

*Ejecutar en PowerShell como administrador*  
Set-ExecutionPolicy RemoteSigned

*Ejecutar en PowerShell como usuario*

```
iwr -useb get.scoop.sh | iex
scoop install git
scoop update
scoop bucket add java
scoop install java/openjdk
scoop install kotlin
```

- b. LINUX DEBIAN

```
sudo apt install openjdk-17-jdk
sudo apt install kotlin
```

2. Compilación y ejecución:

- a. Capture el siguiente código en un archivo de texto plano, guárdelo como hola.kt

```
fun main() {
```



```
println("Hola mundo!")
```

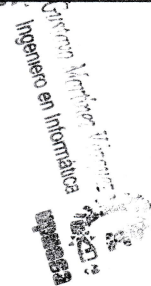
```
}
```

- b. Compile desde la terminal el código anterior para JVM y ejecútelo a partir del bytecode generado:

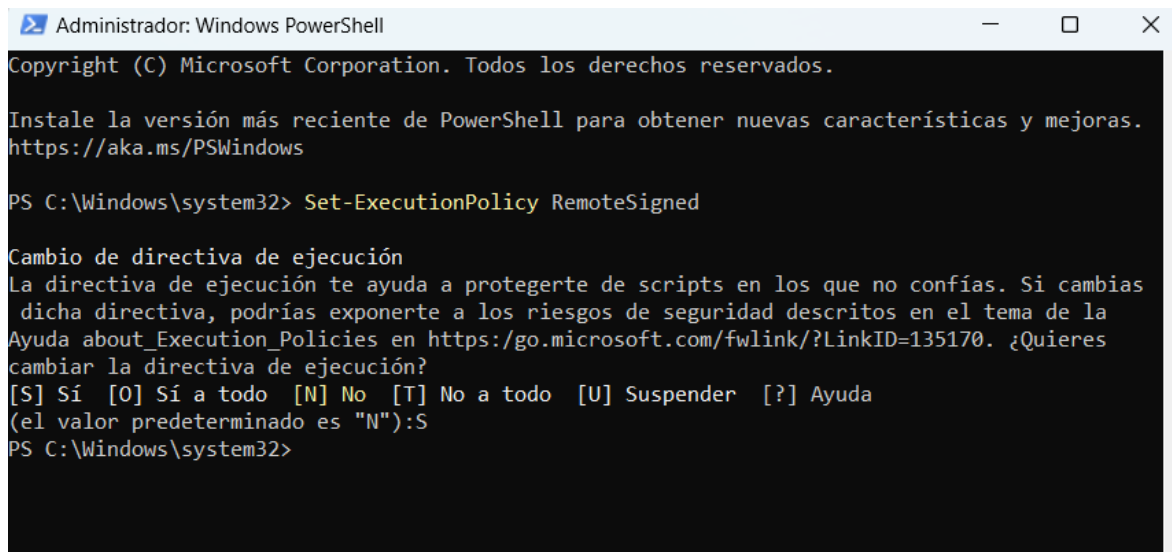
```
kotlinc hola.kt  
java HolaKT
```

3. Conclusiones. Escriba en los comentarios su opinión sobre el lenguaje Kotlin y su relación con la tecnología Java, así como si tuvo alguna dificultad para desarrollar la práctica y cómo solucionó la situación.

Kotlin se me hace muy interesante porque a través de él se pueden realizar aplicaciones móviles. La relación que tiene con Java es que se pueden usar sus bibliotecas y se ejecuta en la JVM. Una de las dificultades que tuve fue al momento de compilar el código, que algunas instrucciones no funcionaron en mi computadora.



1. Desde la terminal se ejecutaron los siguientes comandos:
  - a) Set-ExecutionPolicy RemoteSigned



```
Administrador: Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

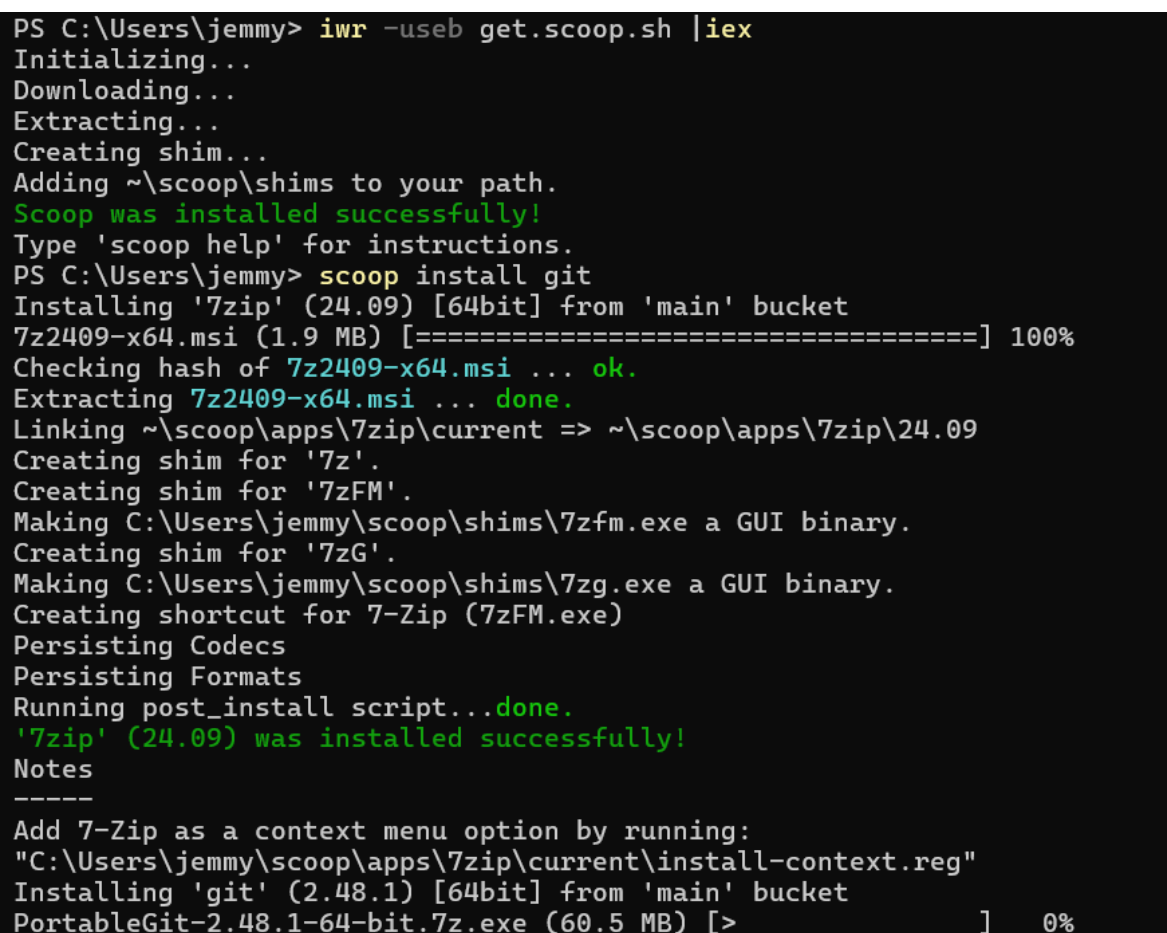
Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras.
https://aka.ms/PSWindows

PS C:\Windows\system32> Set-ExecutionPolicy RemoteSigned

Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias
dicha directiva, podrías exponerte a los riesgos de seguridad descritos en el tema de la
Ayuda about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkID=135170. ¿Quieres
cambiar la directiva de ejecución?
[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda
(el valor predeterminado es "N"):S
PS C:\Windows\system32>
```

Los comandos siguientes se ejecutaron en PowerShell como usuario:

- b) iwr -useb get.scoop.sh | iex
- c) scoop install git



```
PS C:\Users\jemmy> iwr -useb get.scoop.sh | iex
Initializing...
Downloading...
Extracting...
Creating shim...
Adding ~\scoop\shims to your path.
Scoop was installed successfully!
Type 'scoop help' for instructions.
PS C:\Users\jemmy> scoop install git
Installing '7zip' (24.09) [64bit] from 'main' bucket
7z2409-x64.msi (1.9 MB) [=====] 100%
Checking hash of 7z2409-x64.msi ... ok.
Extracting 7z2409-x64.msi ... done.
Linking ~\scoop\apps\7zip\current => ~\scoop\apps\7zip\24.09
Creating shim for '7z'.
Creating shim for '7zFM'.
Making C:\Users\jemmy\scoop\shims\7zfm.exe a GUI binary.
Creating shim for '7zG'.
Making C:\Users\jemmy\scoop\shims\7zg.exe a GUI binary.
Creating shortcut for 7-Zip (7zFM.exe)
Persisting Codecs
Persisting Formats
Running post_install script...done.
'7zip' (24.09) was installed successfully!
Notes
-----
Add 7-Zip as a context menu option by running:
"C:\Users\jemmy\scoop\apps\7zip\current\install-context.reg"
Installing 'git' (2.48.1) [64bit] from 'main' bucket
PortableGit-2.48.1-64-bit.7z.exe (60.5 MB) [>] 0%
```

```
PortableGit-2.48.1-64-bit.7z.exe (60.5 MB) [=====] 100%
Checking hash of PortableGit-2.48.1-64-bit.7z.exe ... ok.
Extracting PortableGit-2.48.1-64-bit.7z.exe ... done.
Linking ~\scoop\apps\git\current => ~\scoop\apps\git\2.48.1
Creating shim for 'sh'.
Creating shim for 'bash'.
Creating shim for 'git'.
Creating shim for 'gitk'.
Making C:\Users\jemmy\scoop\shims\gitk.exe a GUI binary.
Creating shim for 'git-gui'.
Making C:\Users\jemmy\scoop\shims\git-gui.exe a GUI binary.
Creating shim for 'scalar'.
Creating shim for 'tig'.
Creating shim for 'git-bash'.
Making C:\Users\jemmy\scoop\shims\git-bash.exe a GUI binary.
Creating shortcut for Git Bash (git-bash.exe)
Creating shortcut for Git GUI (git-gui.exe)
Running post_install script...done.
'git' (2.48.1) was installed successfully!
Notes
-----
Set Git Credential Manager Core by running: "git config --global
credential.helper manager"

To add context menu entries, run
'C:\Users\jemmy\scoop\apps\git\current\install-context.reg'

To create file-associations for .git* and .sh files, run
'C:\Users\jemmy\scoop\apps\git\current\install-file-associations.reg'
PS C:\Users\jemmy> |
```

d) scoop update git

```
PS C:\Users\jemmy> scoop update git
git: 2.48.1 (latest version)
Latest versions for all apps are installed! For more information try 's
coop status'
PS C:\Users\jemmy> scoop update
Updating Scoop...
Updating Buckets...
INFO Converting 'main' bucket to git repo...
The main bucket was removed successfully.
Checking repo... OK
The main bucket was added successfully.
Scoop was updated successfully!
PS C:\Users\jemmy> |
```

e) scoop bucket add java

```
PS C:\Users\jemmy> scoop bucket add java
Checking repo... OK
The java bucket was added successfully.
```

f) scoop install java/openjdk

```
The java bucket was added successfully.
PS C:\Users\jemmy> scoop install java/openjdk
Installing 'openjdk' (23.0.2-7) [64bit] from 'java' bucket
openjdk-23.0.2_windows-x64_bin.zip (199.5 MB) [=====] 100%
Checking hash of openjdk-23.0.2_windows-x64_bin.zip ... ok.
Extracting openjdk-23.0.2_windows-x64_bin.zip ... done.
Linking ~\scoop\apps\openjdk\current => ~\scoop\apps\openjdk\23.0.2-7
Adding ~\scoop\apps\openjdk\current\bin to your path.
'openjdk' (23.0.2-7) was installed successfully!
PS C:\Users\jemmy> |
```

g) scoop install kotlin

```
PS C:\Users\jemmy> scoop install kotlin
Installing 'kotlin' (2.1.10) [64bit] from 'main' bucket
kotlin-compiler-2.1.10.zip (82.4 MB) [=====] 100%
Checking hash of kotlin-compiler-2.1.10.zip ... ok.
Extracting kotlin-compiler-2.1.10.zip ... done.
Linking ~\scoop\apps\kotlin\current => ~\scoop\apps\kotlin\2.1.10
Creating shim for 'kapt'.
Creating shim for 'kotlin'.
Creating shim for 'kotlinc'.
Creating shim for 'kotlinc-js'.
Creating shim for 'kotlinc-jvm'.
'kotlin' (2.1.10) was installed successfully!
```

## 2. Compilación y ejecución

a) Se capturo el siguiente código en un archivo txt, pero con la .kt. El código es un “Hola mundo”.

Archivo	Editar	Ver
<pre>fun main() {     println("Hola mundo!") }</pre>		

b) Se realizo la compilación del código en la termina para JVM. Se tuvo que cambiar el comando java HolaKT por hola. Jar y por java -jar hola.jar

```
PS C:\Users\jemmy\OneDrive\Documentos\UPIICSA\SEXTO\PROGRAMACION MOVIL\PRIMER PARCIAL> kotlinc hola.kt -include-runtime -d hola.jar
PS C:\Users\jemmy\OneDrive\Documentos\UPIICSA\SEXTO\PROGRAMACION MOVIL\PRIMER PARCIAL> java -jar hola.jar
Hola mundo!
```