



Nombre completo	Secuencia	Fecha	Calificación
Ibarru Lima Julio Jair	GNM62	05/03/25	

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() {
```



INSTITUTO POLITÉCNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA DE INGENIERÍA Y CIENCIAS SOCIALES
Y ADMINISTRATIVAS



```
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.

En esta Práctica, aprendí a instalar y configurar Kotlin en Windows y OpenJDK, asegurando que estuviera listo para compilar y ejecutar programas, luego escribimos y guardamos nuestro primer código en Kotlin, lo cual fue fácil para el final de la práctica, fue divertido.



- Set-ExecutionPolicy RemoteSigned

```

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

Instale la versin ms reciente de PowerShell para obtener nuevas caractersticas y mejoras. https://aka.ms/PSWindows

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

Cambio de directiva de ejecucin
La directiva de ejecucin te ayuda a protegerte de scripts en los que no confas. Si cambias dicha directiva, podras
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 ejecucin?
[S] S [O] S a todo [N] No [T] No a todo [U] Suspender [?] Ayuda (el valor predeterminado es "N"): 0

```

- iwr -useb get.scoop.sh | iex

```

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

Instale la versin ms reciente de PowerShell para obtener nuevas caractersticas y mejoras. https://aka.ms/PSWindows

PS C:\Users\ibarr> 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\ibarr>

```

- scoop install git

```

Windows PowerShell
PS C:\Users\ibarr> 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\ibarr\scoop\shims\7zfm.exe a GUI binary.
Creating shim for '7zG'.
Making C:\Users\ibarr\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\ibarr\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) [=====] 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\ibarr\scoop\shims\gitk.exe a GUI binary.
Creating shim for 'git-gui'.
Making C:\Users\ibarr\scoop\shims\git-gui.exe a GUI binary.
Creating shim for 'scalar'.
Creating shim for 'tig'.
Creating shim for 'git-bash'.
Making C:\Users\ibarr\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\ibarr\scoop\apps\git\current\install-context.reg'
To create file-associations for .git* and .sh files, run
'C:\Users\ibarr\scoop\apps\git\current\install-file-associations.reg'

```

- scoop update

```
PS C:\Users\ibarr> 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\ibarr>
```

- Scoop bucket and java

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

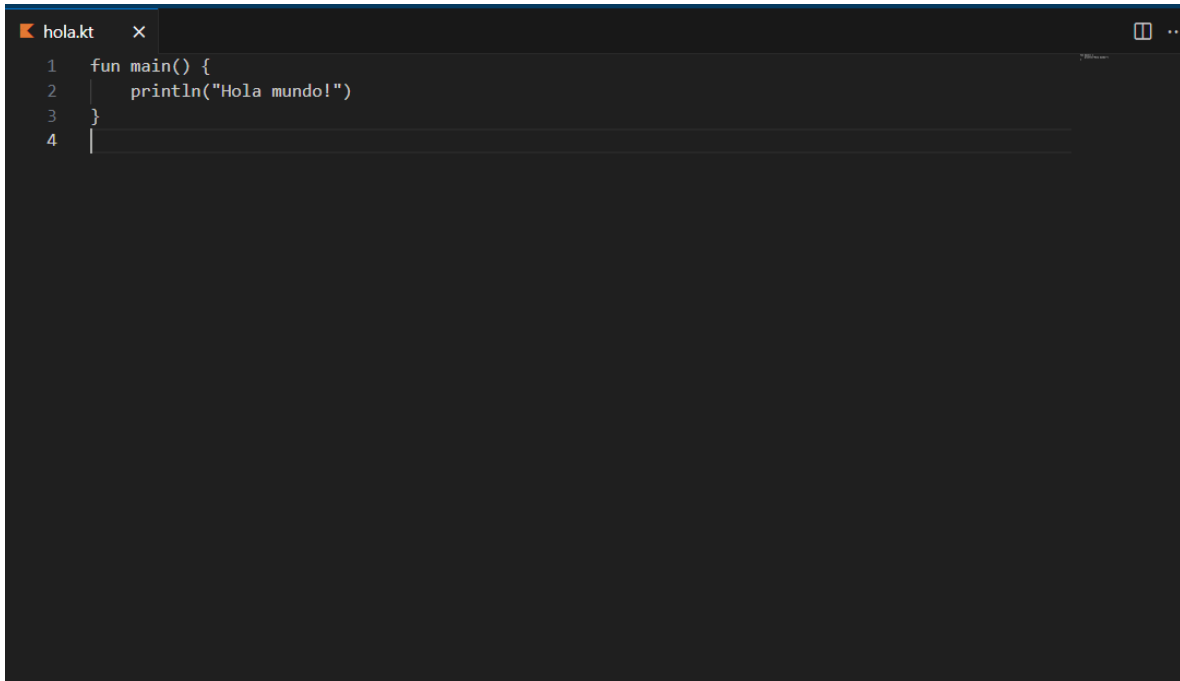
- scoop install java/openjdk

```
The java bucket was added successfully.
PS C:\Users\ibarr> 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\ibarr>
```

- scoop install kotlin

```
PS C:\Users\ibarr> 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 'kotlinx'.
Creating shim for 'kotlinx-jvm'.
'kotlin' (2.1.10) was installed successfully!
PS C:\Users\ibarr>
```

- Código



```
hol.kt x
1 fun main() {
2     println("Hola mundo!")
3 }
4 |
```

- HOLA MUNDO EN KOTLIN

```
PS C:\Users\ibarr\OneDrive\Escritorio\PracticasDM> kotlinc hola.kt
PS C:\Users\ibarr\OneDrive\Escritorio\PracticasDM> java HolaKt
Hola mundo!
PS C:\Users\ibarr\OneDrive\Escritorio\PracticasDM>
```