

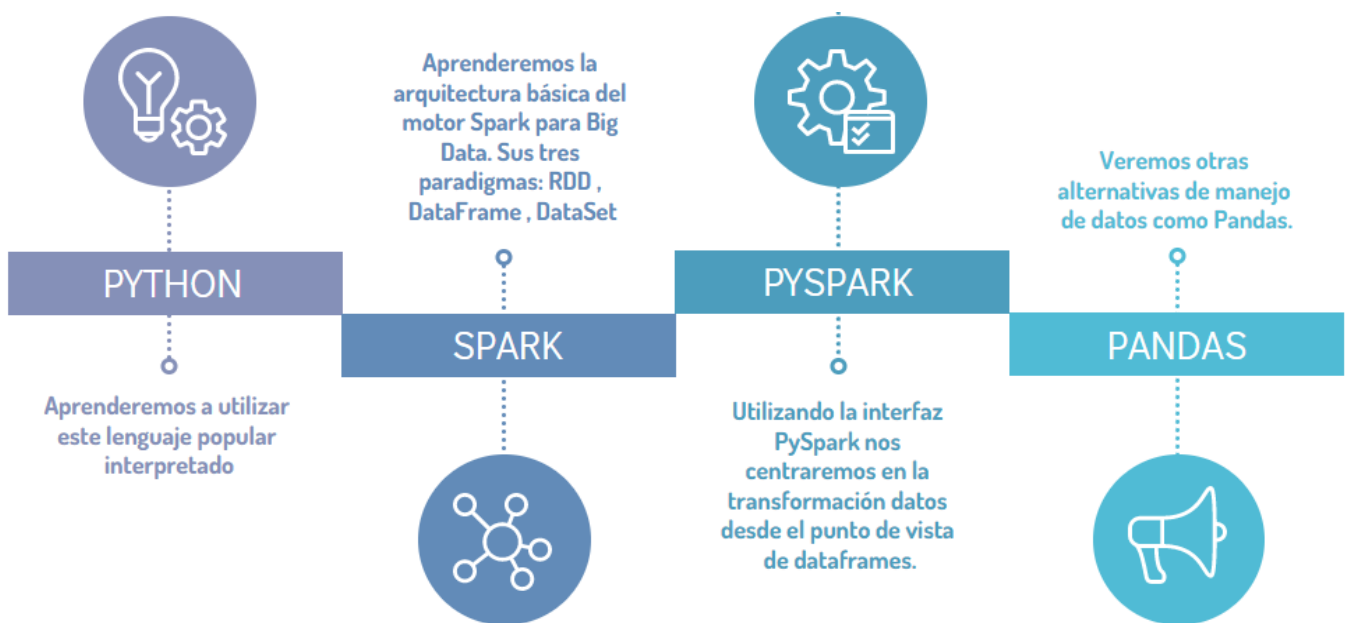
Fundamentos de manejo de datos con Python

- Ambientación de herramientas de trabajo

Autor: Marco Ycaza

Panorama general

Bienvenido al siguiente tutorial de fundamentos de manejo de datos , podrá observar a continuación la ruta propuesta de aprendizaje.



Pero para empezar con dicha ruta necesitamos tener ciertas herramientas.

A continuación listamos las siguientes:

N	Herramienta	Tipo	Versión recomendada	Usada por el autor
1	Git	Sistema de control de versiones (No es realmente necesario pero su terminal Git Bash es muy util)	última	v2.35.12
2	Java	lenguaje	8 en adelante	8_181

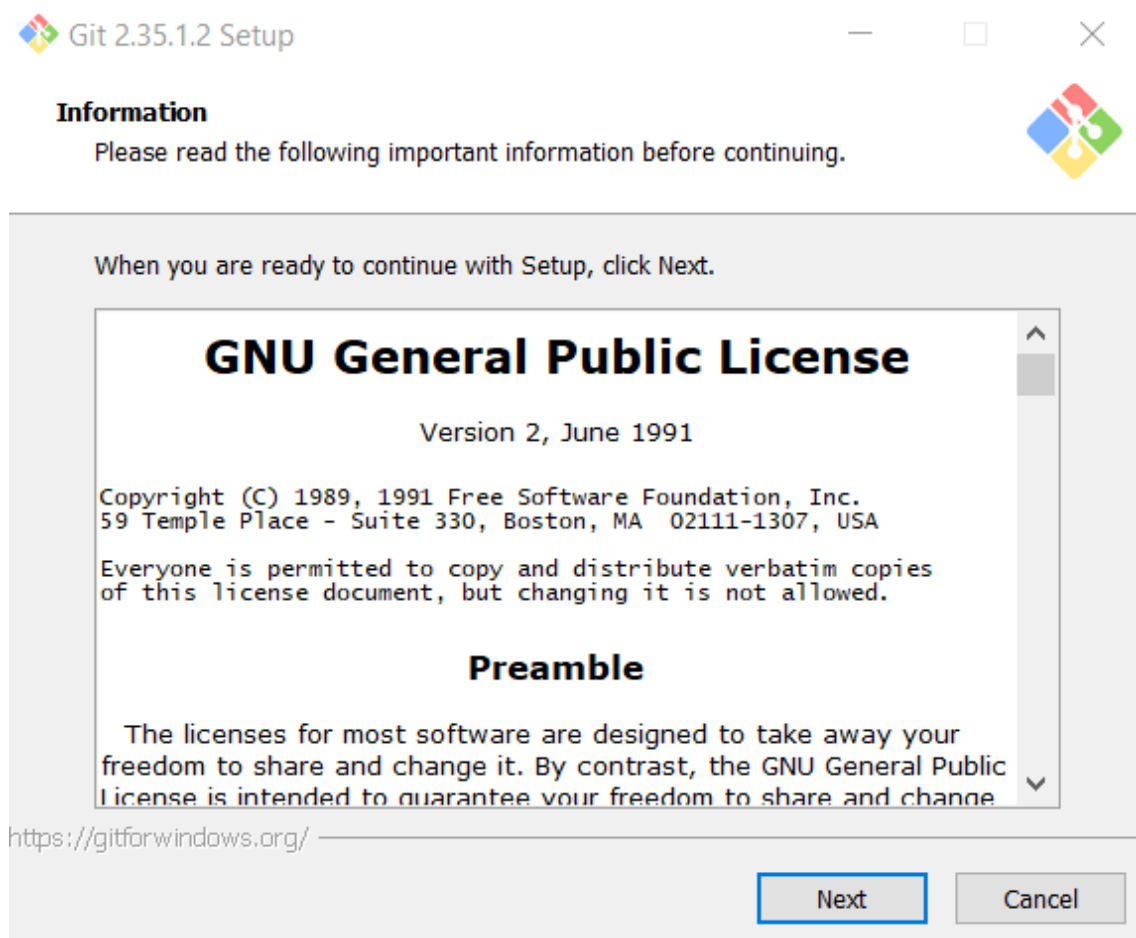
3	Python	lenguaje	3.5 en adelante	3.7.8
4	Spark	framework	2 en adelante	spark-2.4.4-bin-hadoop2.7
5	Hadoop	framework	+2.00	v2.7.0
6	Visual Studio Code	Editor de código	última	1.64.1
7	Notepad ++	Editor de texto	última	v8.2.1

Sería recomendable que usted instale las mismas versiones con las cuales fue desarrollado este tutorial para evitar diferencias no contempladas en este tutorial.

Instalaciones de herramientas:

Instalación de Git:

Visitar la siguiente página [Git - Downloading Package \(git-scm.com\)](https://git-scm.com/) y descargamos el instalador de nuestro sistema operativo , luego lo instalamos según la guía de auto ayuda que ofrece el instalador.



Instalación de Java:


Vamos a instalar el kit de desarrollo ya que este nos permitirá tener tanto el JRE (que incluye la virtual machine) y herramientas de compilación para desarrollo y testeo. De forma pura para

correr Spark necesitamos solo el JRE pero conviene tener el **JDK (1)** para posibles desarrollos.

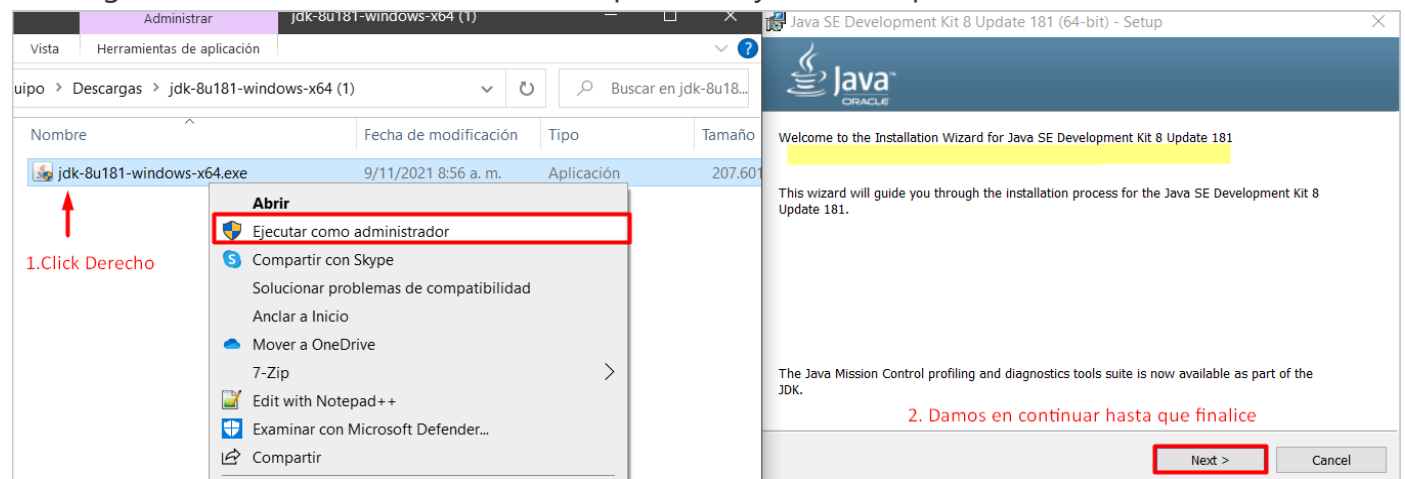
Proceda a descargar la siguiente versión a través de:

[Java Archive Downloads - Java SE 8 \(oracle.com\)](#)

Precisamente en el enlace del apartado de "Java SE Development Kit". Es probable que antes tengamos que crear una cuenta oracle.

Java SE Development Kit 8u181		
This software is licensed under the Oracle Binary Code License Agreement for Java SE Platform Products		
Product / File Description	File Size	Download
Windows x64	202.73 MB	 jdk-8u181-windows-x64.exe

Descargamos la versión de acuerdo a la arquitectura y sistema operativo de nuestro ordenador.



Instalación de Python:

Python es un lenguaje interpretado de simple sintaxis que nos permitirá usar el motor Spark de forma rápida. También será utilizado con Pandas al final del curso de fundamentos. El enlace de descarga es el siguiente:

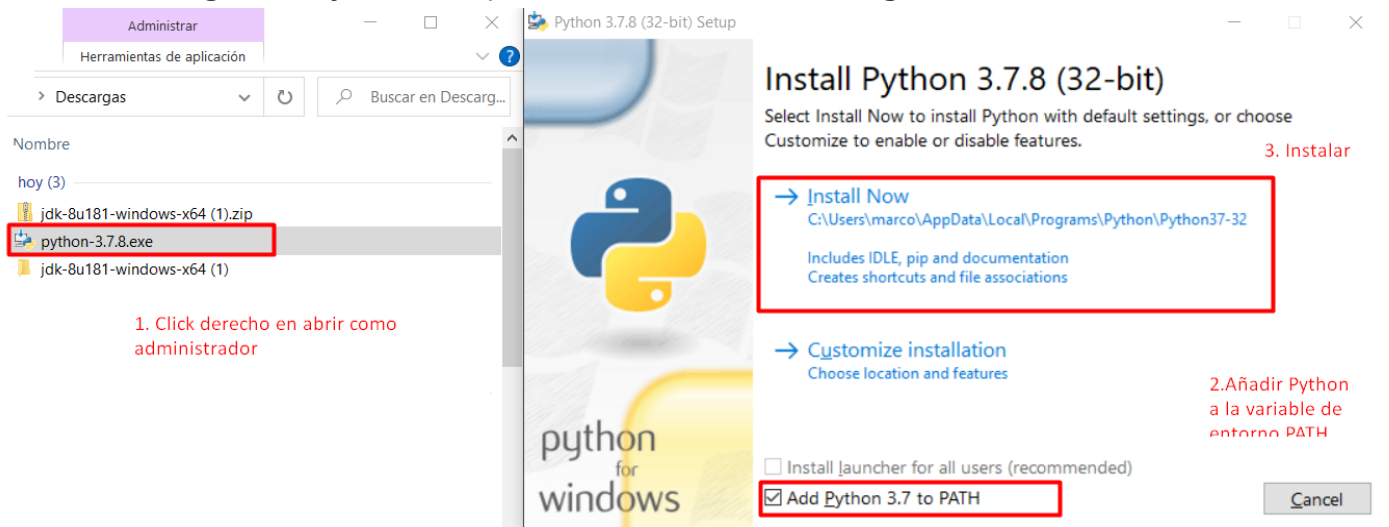
[Python Release Python 3.7.8 | Python.org](#)

Descargamos la versión de acuerdo a la arquitectura y sistema operativo de nuestro ordenador. En mi caso tengo Windows de 64 bits como SO.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		4d5b16e8c15be38eb0f4b8f04eb68cd0	23276116	SIG
XZ compressed source tarball	Source release		a224ef2249a18824f48fba9812f4006f	17399552	SIG
macOS 64-bit installer	macOS	for OS X 10.9 and later	2819435f3144fd973d3dea4ae6969f6d	29303677	SIG
Windows help file	Windows		65bb54986e5a921413e179d2211b9bfb	8186659	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	5ae191973e00ec490cf2a93126ce4d89	7536190	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	70b08ab8e75941da7f5bf2b9be58b945	26993432	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	b07dbb998a4a0372f6923185ebb6bf3e	1363056	SIG
Windows x86 embeddable zip file	Windows		5f0f83433bd57fa55182cb8ea42d43d6	6765162	SIG
Windows x86 executable installer	Windows		4a9244c57f61e3ad2803e900a2f75d77	25974352	SIG
Windows x86 web-based installer	Windows		642e566f4817f118abc38578f3cc4e69	1324944	SIG

Una vez descargado el ejecutable , proceder a instalarlo de la siguiente manera:



instalamos el paquete ipython a través de pip:

```
C:\Users\marco>pip install ipython
```

Instalación de Spark con Hadoop:

Dentro de la página oficial de Apache Spark , vamos buscando el apartado de releases antiguos encontramos el siguiente recurso:


[Index of /dist/spark/spark-2.4.4 \(apache.org\)](https://index.of/dist/spark/spark-2.4.4)

Index of /dist/spark/spark-2.4.4




Name	Last modified	Size	Description
 Parent Directory	-	-	-
 spark-2.4.4-bin-hadoop2.7.tgz	2019-08-27 22:01	219M	
 spark-2.4.4-bin-hadoop2.6.tgz	2019-08-27 22:01	218M	
 pyspark-2.4.4.tar.gz	2019-08-27 22:01	206M	

Descargamos la versión más actual , aunque para efectos del tutorial se sugiere la misma versión que usa el autor.

Una vez descargado , podremos notar que el archivo tiene un formato de compresión tipo UNIX , con una herramienta como 7Zip podremos descromprimirlo en dos intentos. El primer intento nos devolverá un archivo .tar y el segundo intento (sobre el .tar) nos devolverá la carpeta sin comprimir.

Nombre	Fecha de modificación	Tipo	Tamaño
hoy (5)			
 spark-2.4.4-bin-hadoop2.7.tgz	9/02/2022 8:31 a. m.	Archivo TGZ	224.699 KB

Luego , copiar dicha carpeta dentro de un folder llamado "Spark" (2) en nuestro disco principal.


 > Este equipo > Windows-SSD (C:) > Spark				
Nombre	Fecha de modificación	Tipo	Tamaño	
 spark-2.4.4-bin-hadoop2.7	7/02/2022 10:41 p. m.	Carpeta de archivos		
 spark-3.0.0-bin-hadoop2.7	7/02/2022 10:32 p. m.	Carpeta de archivos		

Descarga de utilitario Hadoop:

Acceder a la siguiente ruta y busar el archivo "winutils.exe" desde el folder bin del hadoop-2.7.1 :

[winutils/winutils.exe at master · steveloughran/winutils \(github.com\)](https://github.com/steveloughran/winutils).

master winutils / hadoop-2.7.1 / bin / **winutils.exe** Go to file ...

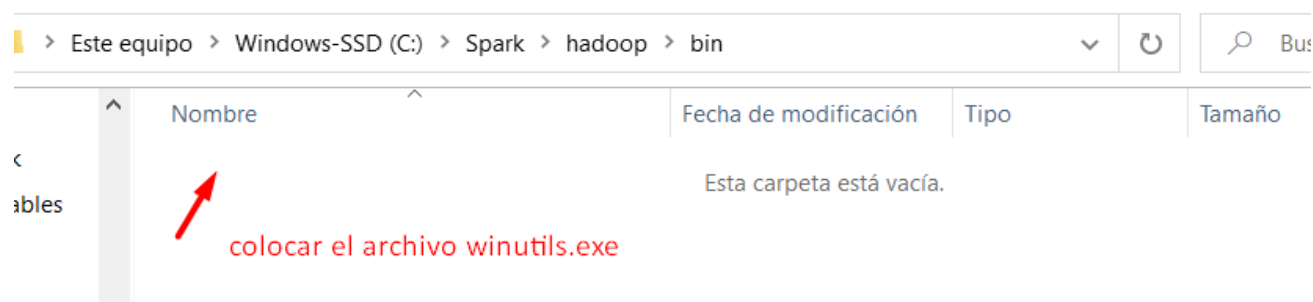
 steveloughran add 2.6.4 and 2.7.1 windows binaries Latest commit 7665f01 on 12 Feb 2016 History

1 contributor

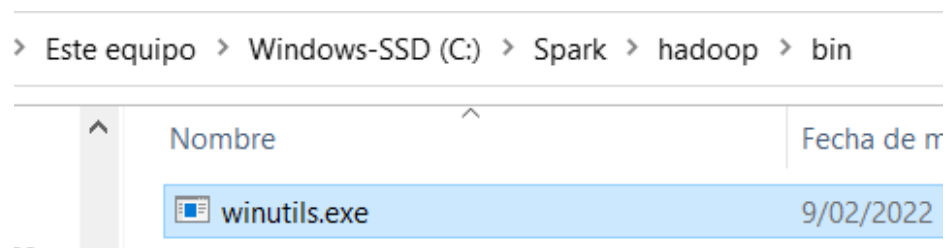
107 KB **Download** 📄 🗑️

[View raw](#)

Una vez descargado el archivo , creamos un folder dentro de la carpeta Spark llamado hadoop y un subfolder llamado bin:

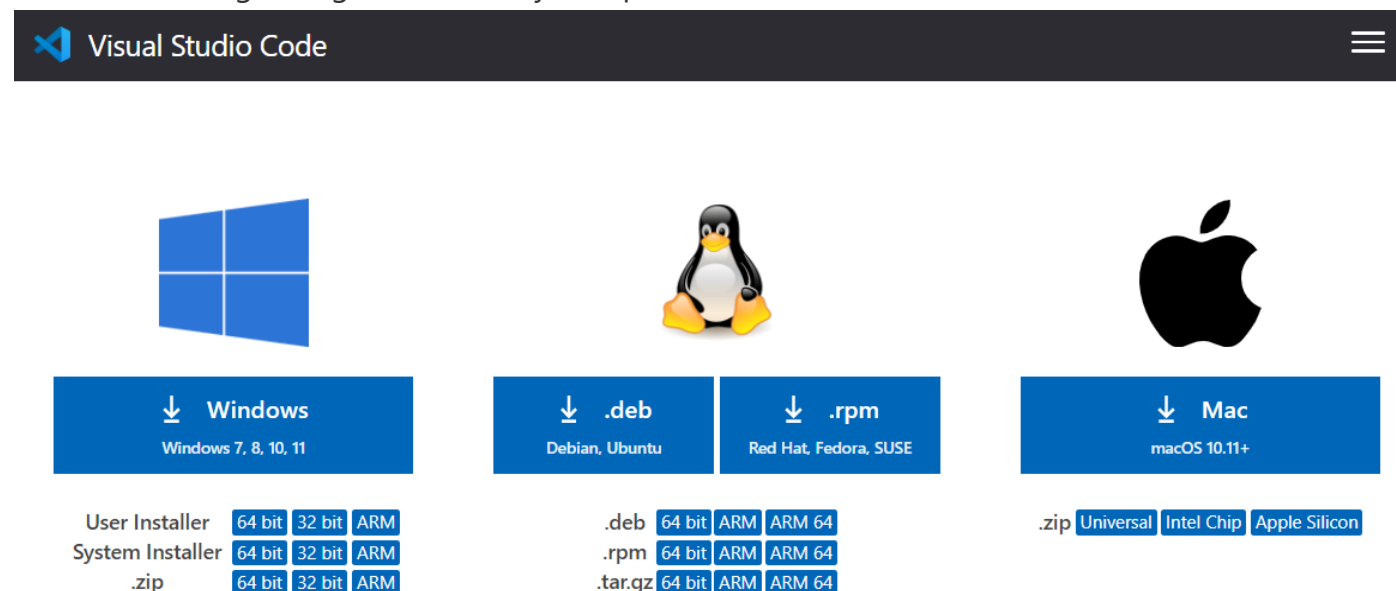


Nos quedaría lo siguiente



Instalación de Visual Studio Code:

Nos dirigimos a la siguiente ruta : [Download Visual Studio Code - Mac, Linux, Windows](#) y descargamos el instalador de nuestro sistema operativo , luego como cualquier editor de código lo instalamos según la guía de auto ayuda que ofrece el instalador.



Instalación de Notepad++:

Nos dirigimos a la siguiente ruta : [Notepad++ 8.3 \(Boycott Beijing 2022\)| Notepad++ \(notepad-plus-plus.org\)](#) y descargamos el instalador de nuestro sistema operativo , luego como cualquier editor de código lo instalamos según la guía de auto ayuda que ofrece el instalador.



Current Version 8.3

- Home
- Download
- News
- Online Help
- Resources
- RSS
- Donate
- Author

Notepad++ 8.3 (Boycott Beijing 2022)

Release Date: 2022-02-03

Download 64-bit x64



Configuración de variables de entorno:

presionamos las teclas **Windows + R** y escribimos **SystemPropertiesAdvanced**

Ejecutar



Escriba el nombre del programa, carpeta, documento o recurso de Internet que desea abrir con Windows.

Abrir:

SystemPropertiesAdvanced



Aceptar

Cancelar

Examinar...

Luego nos aparecerá la ventana de propiedades y damos click a "variables de entorno"

Nombre de equipo Hardware

Opciones avanzadas Protección del sistema Acceso remoto

Para realizar la mayoría de estos cambios, inicie sesión como administrador.

Rendimiento

Efectos visuales, programación del procesador, uso de memoria y memoria virtual

Configuración...

Perfiles de usuario

Configuración del escritorio correspondiente al inicio de sesión

Configuración...

Inicio y recuperación

Inicio del sistema, errores del sistema e información de depuración

Configuración...

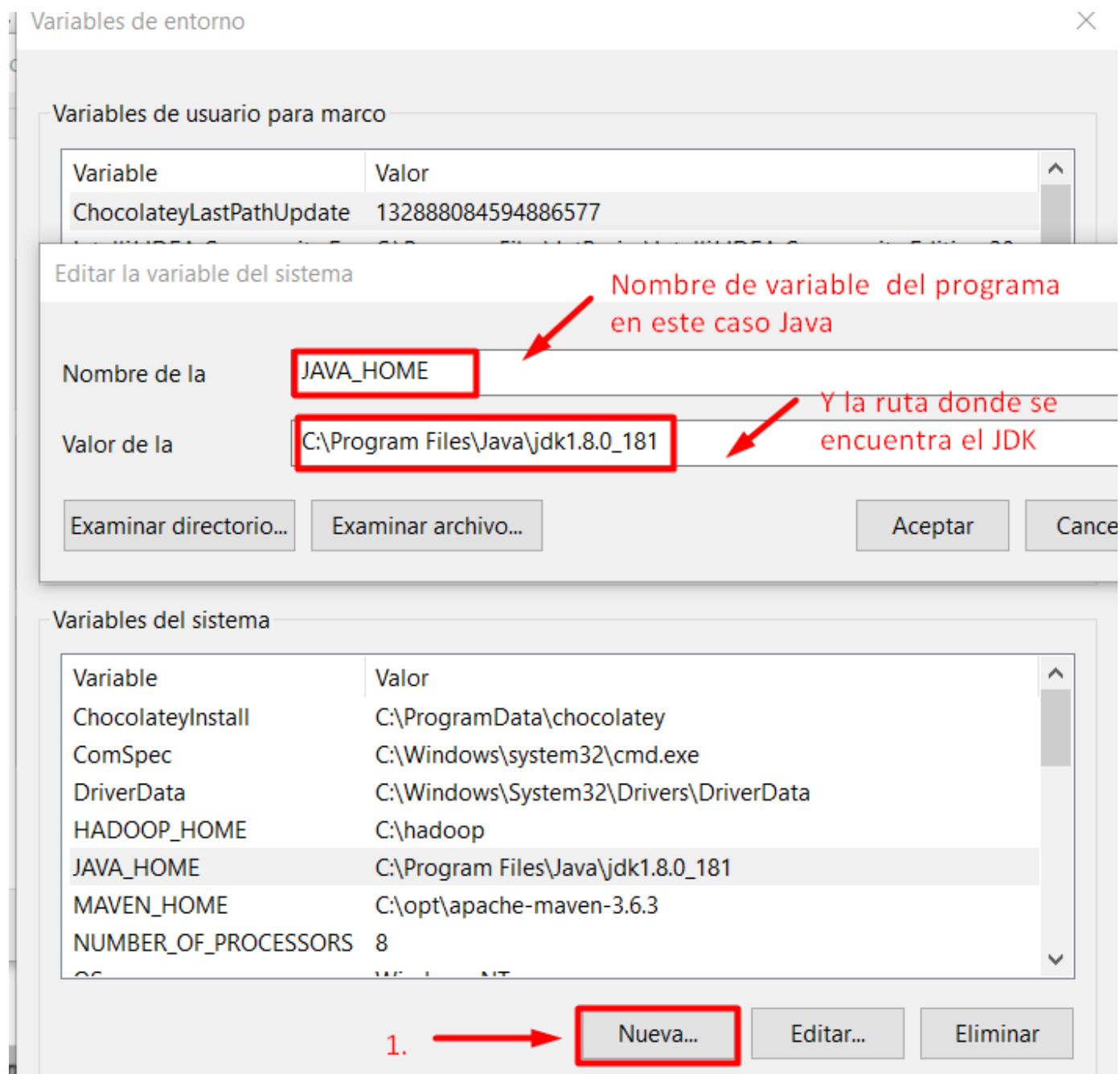
Variables de entorno...

Aceptar Cancelar Aplicar

Una vez hecho ello , damos click en Nueva y creamos la siguiente variable:

llave : **JAVA_HOME**

valor : **C:\Program Files\Java\jdk1.8.0_181**



De la misma manera tenemos que crear variables de entorno para Spark siendo:

llave : SPARK_HOME

valor: C:\Spark\spark-2.4.4-bin-hadoop2.7

Y estas otras:

Llave	Valor
JAVA_HOME	C:\Progra~1\Java\jdk1.8.0_181
SPARK_HOME	C:\Spark\spark-2.4.4-bin-hadoop2.7
HADOOP_HOME	C:\Spark\hadoop

PYTHONPATH (3)	%SPARK_HOME%/python;%SPARK_HOME%/python/lib/py4j-0.10.7-src.zip;%PYTHONPATH%
PYSPARK_DRIVER_PYTHON	ipython

Por último , en la misma ventana de variables de entorno , ubicamos la variable Path que sirve para que el sistema operativo pueda buscar los ejecutables cuando uno los invoque por nombre a través del terminal de comandos. Por ejemplo cuando uno ejecute en la terminal "pyspark" va a revisar la variable path y va encontrar que existe dicho ejecutable bajo la ruta %SPARK_HOME%\bin.

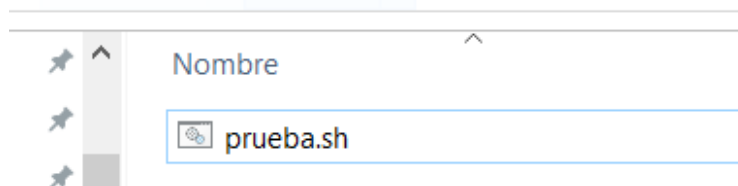
Checkpoint:

Para revisar que hemos instalado y configurado las herramientas , para ello creamos un archivo en el escritorio llamado [prueba.sh](#) que deberá tener el siguiente contenido:

```
#!/bin/bash
printf "Java se encuentra en: %s \n" "$JAVA_HOME"
printf "Spark se encuentra en: %s \n" "$SPARK_HOME"
printf "Hadoop encuentra su binario en: %s \n" "$HADOOP_HOME"
IFS=':' read -r -a array <<< "$PATH"
printf "#####\n"
printf "#####EN LA VARIABLE PATH #####\n"
printf "#####\n"
for element in "${array[@]}"
do
    if [[ "$element" == */bin ]]; then
        printf "%s \n" "$element"
    fi
done
trap 'sleep infinity' EXIT
```

Tendríamos un archivo así , luego damos doble click:

> Este equipo > Escritorio > prueba_variables



Y en resultado deberíamos ver que hay rutas para java , spark y hadoop. De la misma forma deberían tener sus respectivos "bin" en la variable "Path"

```
Java se encuentra en: C:\Progra~1\Java\jdk1.8.0_181
Spark se encuentra en: C:\apachetools\spark-2.4.4-bin-hadoop2.7
Hadoop encuentra su binario en: C:\apachetools\hadoop
```

variables básicas de entorno

```
#####EN LA VARIABLE PATH #####
#####
```

```
/c/Users/marco/bin
```

```
/mingw64/bin
```

```
/usr/local/bin
```

```
/usr/bin
```

```
/bin
```

```
/mingw64/bin
```

```
/usr/bin
```

```
/c/Users/marco/bin
```

```
/c/Program Files/Azure Data Studio/bin
```

```
/c/Progra~1/Java/jdk1.8.0_181/bin
```

```
/c/apachetools/spark-2.4.4-bin-hadoop2.7/bin
```

```
/c/apachetools/hadoop/bin
```

```
/c/Program Files/JetBrains/IntelliJ IDEA Community Edition 2021.3.1/bin
```

```
/c/Program Files/Azure Data Studio/bin
```

```
/c/Users/marco/AppData/Local/Coursier/data/bin
```

```
/c/Users/marco/AppData/Local/Coursier/cache/arc/https/github.com/AdoptOpenJDK/op
enjdk11-binaries/releases/download/jdk-11%252B28/OpenJDK11-jdk_x64_windows_hotsp
ot_11_28.zip/jdk-11+28/bin
```

```
/c/Users/marco/AppData/Local/Programs/Microsoft VS Code/bin
```

En el video se llama a la carpeta "apachetools", en el documento está como "Spark" este nombre es arbitrario

Estos son los bins a través de los cuales windows encontrará los ejecutables

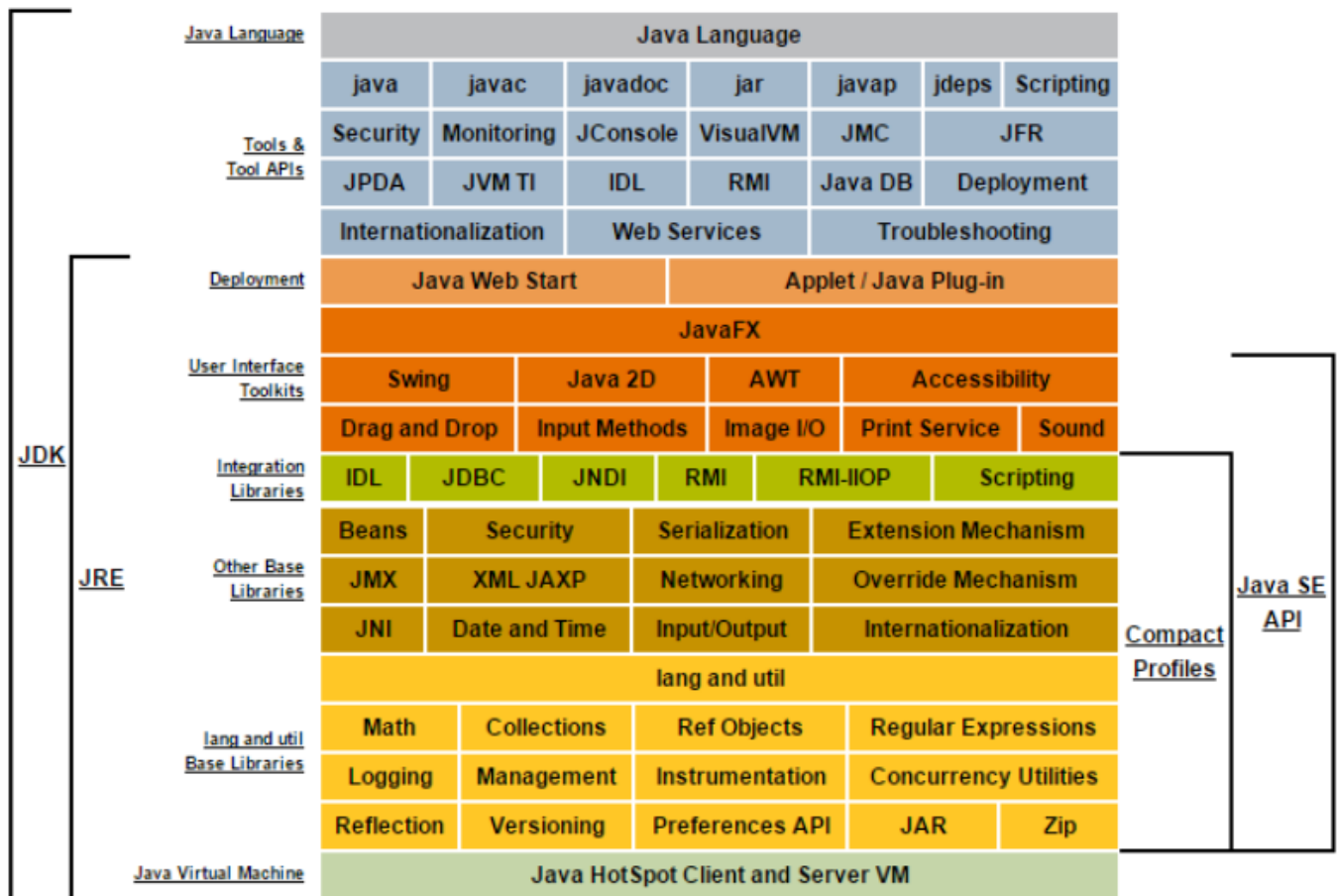
Luego forzamos el cierre del script.

Si seguir el documento se hace pesado , puedes seguir el siguiente tutorial que he preparado para facilitar tu configuración. Da click en el enlace -> [Video Tutorial](#)

ANEXO

(1) Diagrama conceptual del JDK

Description of Java Conceptual Diagram



(2) Puede ser cualquier nombre arbitrario , o de plano copiar la carpeta de spark con hadoop directamente a la raíz. Sin embargo , se tendrá que tener cuidado de poner la ruta correcta al crear las variables de entorno.

(3) Para poder usar pyspark en un kernel de python se tiene que tener acceso a la máquina virtual de java (JVM) para ello debemos definir la variable PYTHONPATH la cual nos permite adicionar directorios externos en donde se buscara módulos y paquetes. Justamente el módulo externo que queremos que python pueda acceder es Py4J. Para poder configurar correctamente esta variable revisemos la ruta %SPARK_HOME%/python/lib y veamos qué version de Py4J tenemos:

```
Directorio de C:\spark\spark-2.4.4-bin-hadoop2.7\python\lib
09/02/2022 10:37 a. m. <DIR> .
09/02/2022 10:37 a. m. <DIR> ..
29/12/2021 06:19 p. m. 42.437 py4j-0.10.7-src.zip
29/12/2021 06:19 p. m. 1.445 PY4J_LICENSE.txt
29/12/2021 06:19 p. m. 591.770 pyspark.zip
3 archivos 635.652 bytes
2 dirs 135.864.016.896 bytes libres
```

Debería salir 0.10.7 si es que has descargado spark 2.4.4 con hadoop 2.7