



Data Growth Community

"Potenciando el crecimiento colectivo"

TEMA:

Guía para conectar PySpark con SQL Server en entorno Local





Gabriela Angel Chipana Perez

Analista de Datos

- Ciencia de la Computación - UNSA
- Analista de Datos - Colegio de Ingenieros
 - Desarrollador 9780Bitcoin



Gabriela Angel Chipana Perez

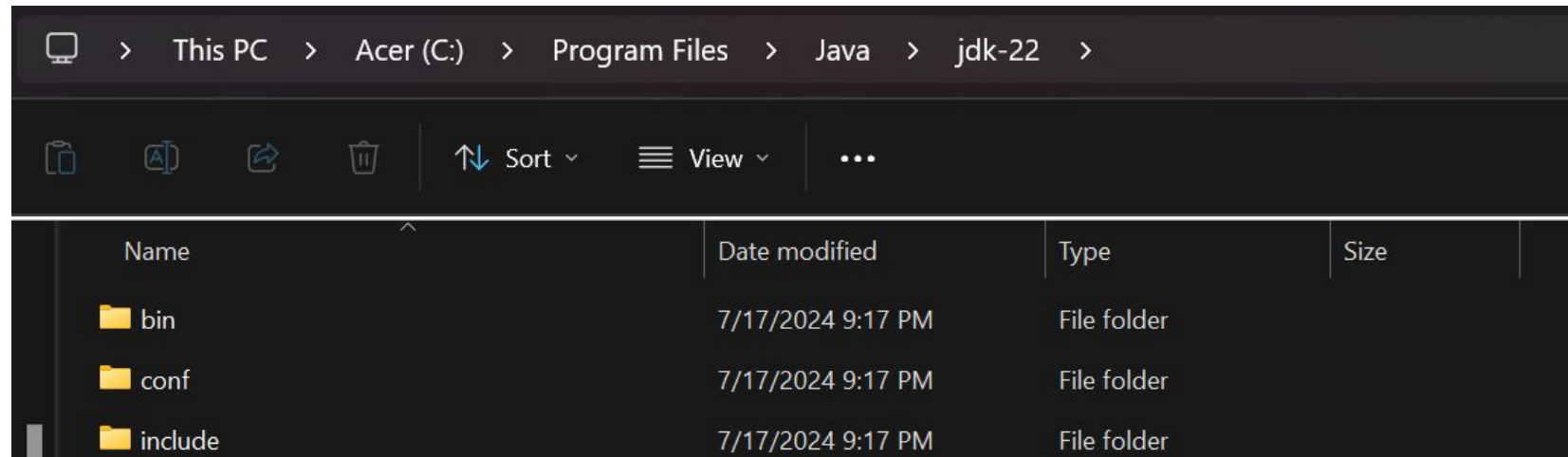


Data Growth
Community

1. Instalar y configurar variables de entorno para JDK
2. Para Windows: Instalar y descargar winutils.exe y hadoop.dll
3. Instalar y configurar variables de entorno para Spark
4. Instalar SQL Server y SQL Server Management Studio
5. Instalar el driver de SQL Server (mssql)

Java Development Kit

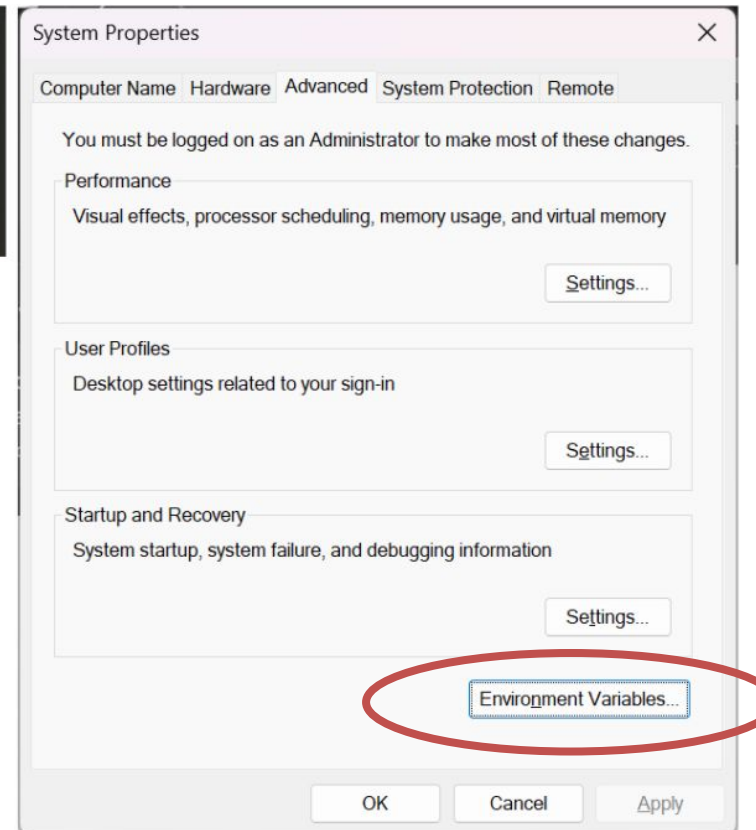
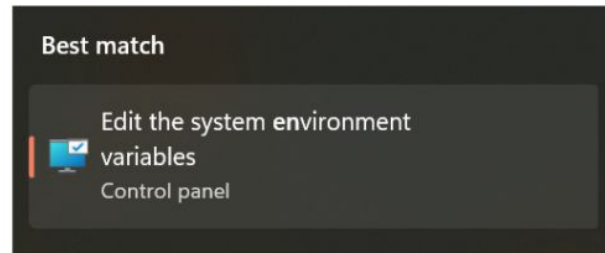
- Instalar
 - <https://www.oracle.com/java/technologies/downloads/#jdk22-linux>
- Agregar al path
 - Ubicar carpeta donde se instaló



- Editar variables de entorno

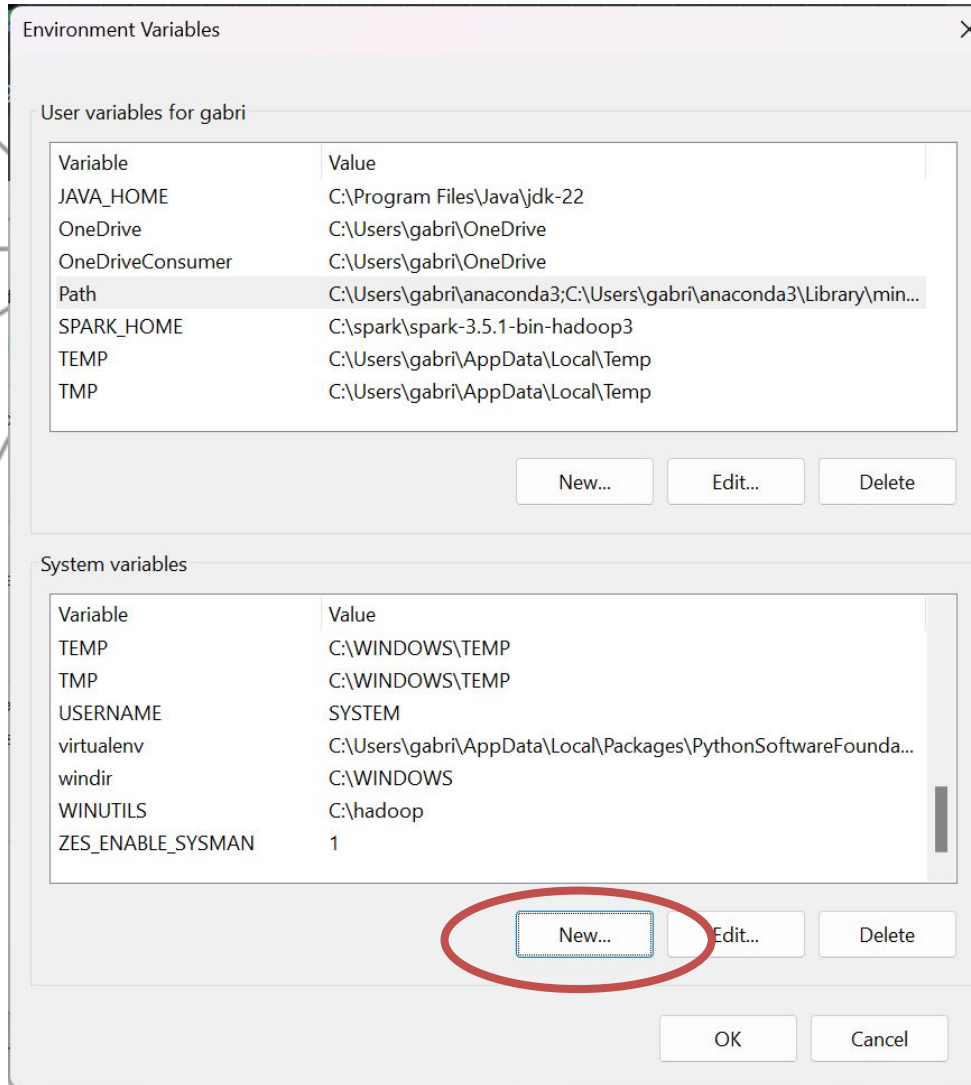
Java Development Kit

- Instalar
<https://www.oracle.com/java/technologies/downloads/#jdk22-windows>
- Agregar al path
 - Ubicar carpeta donde se instaló: **C:\Program Files\Java\jdk-22**
 - Editar variables de entorno

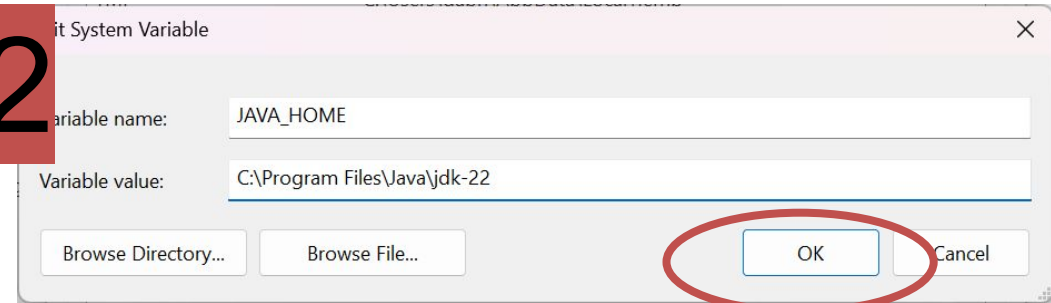


Java Development Kit

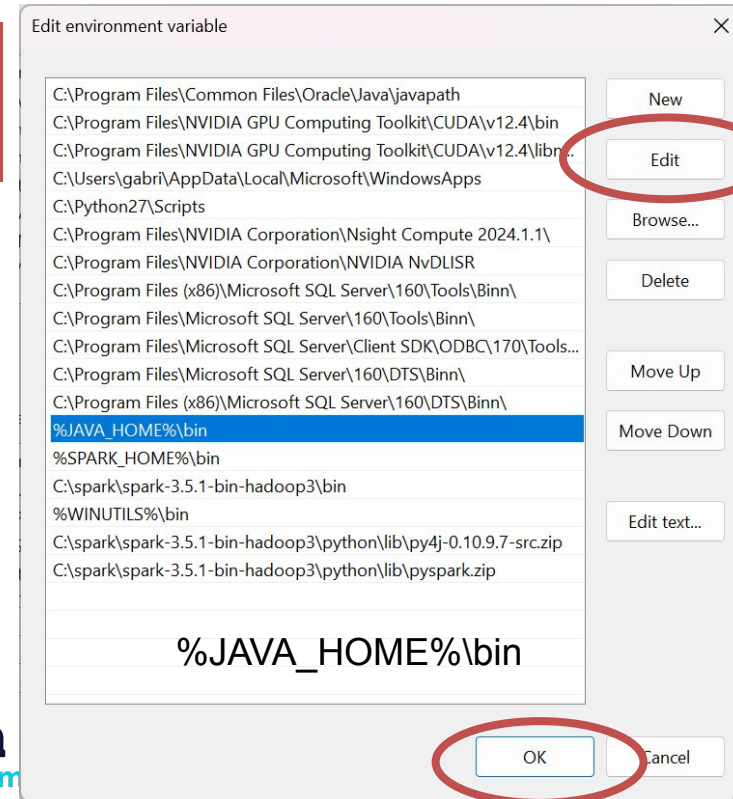
1



2



3

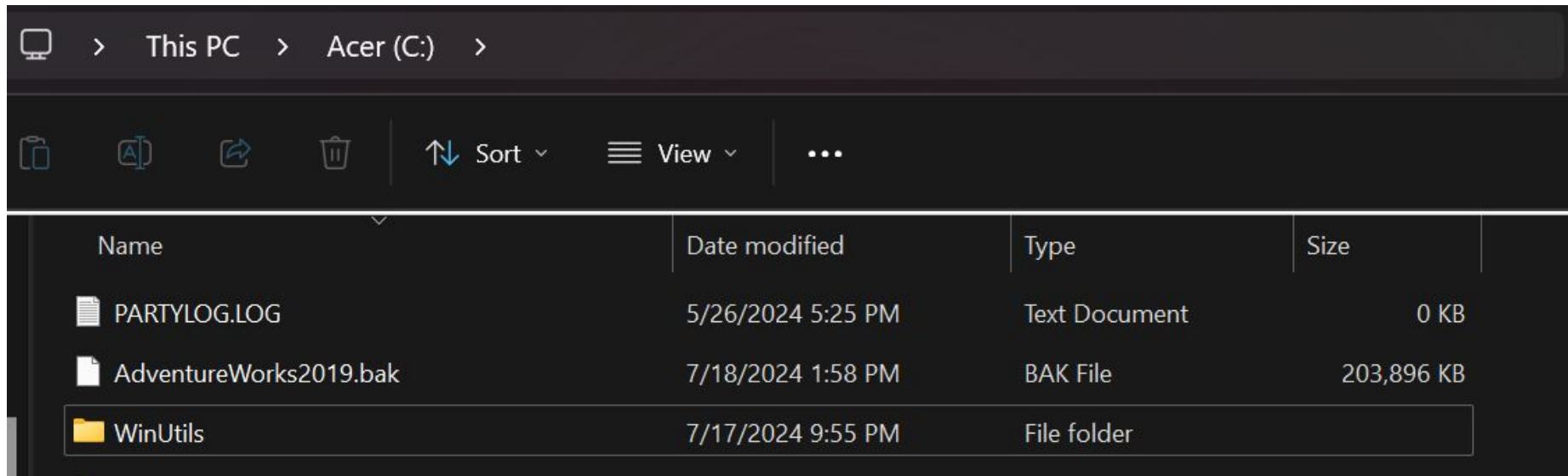


4

Wintutls.exe y hadoop.dll

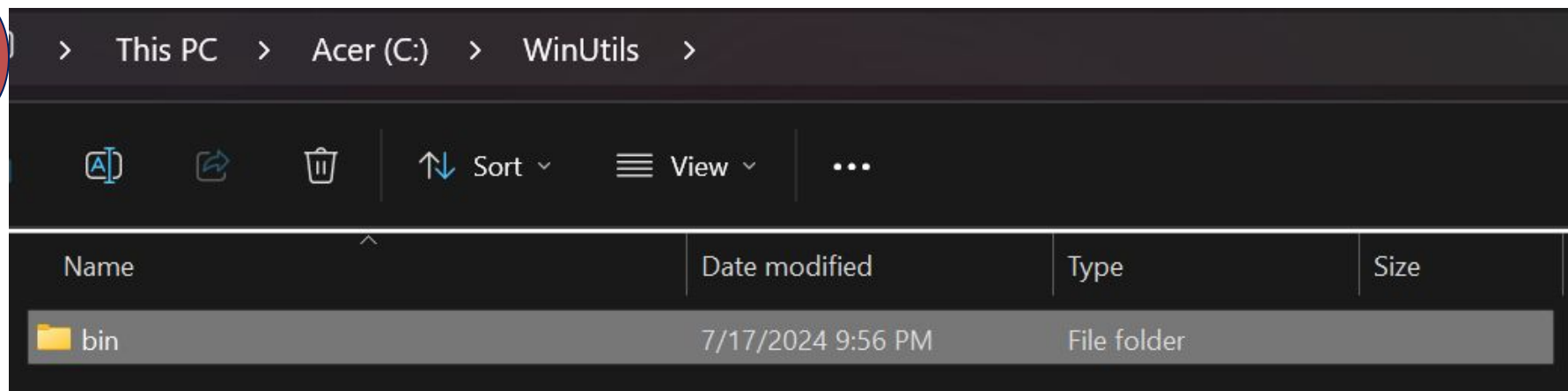
- <https://github.com/kontext-tech/winutils/blob/master/hadoop-3.3.0/bin/winutils.exe>
- <https://github.com/kontext-tech/winutils/blob/master/hadoop-3.3.0/bin/hadoop.dll>

1

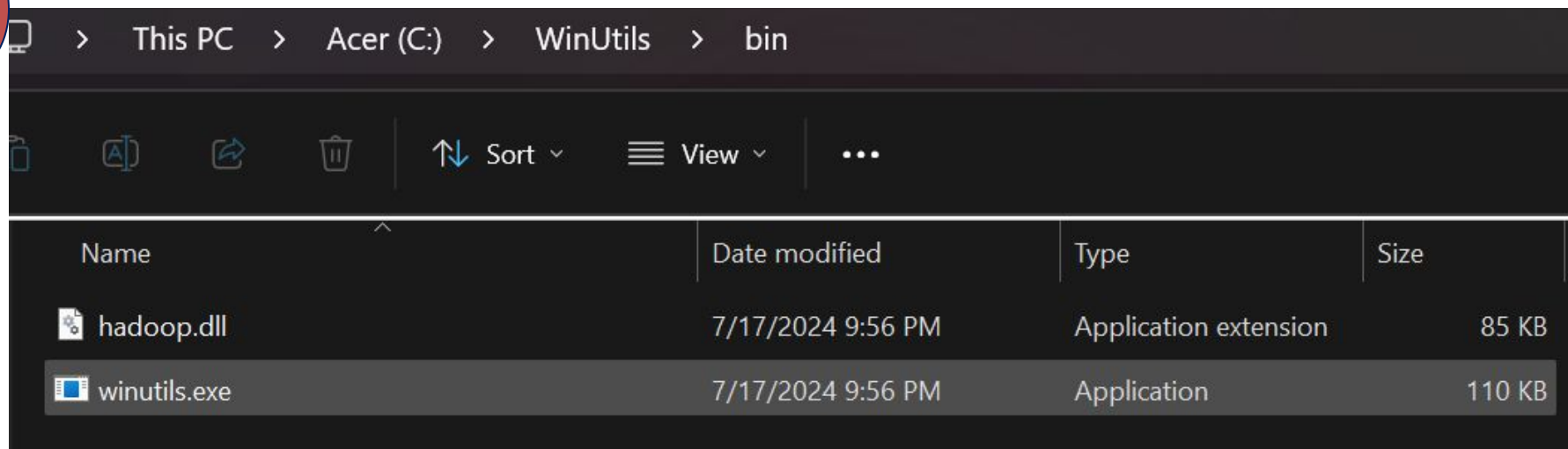


Wintutils.exe y hadoop.dll

2

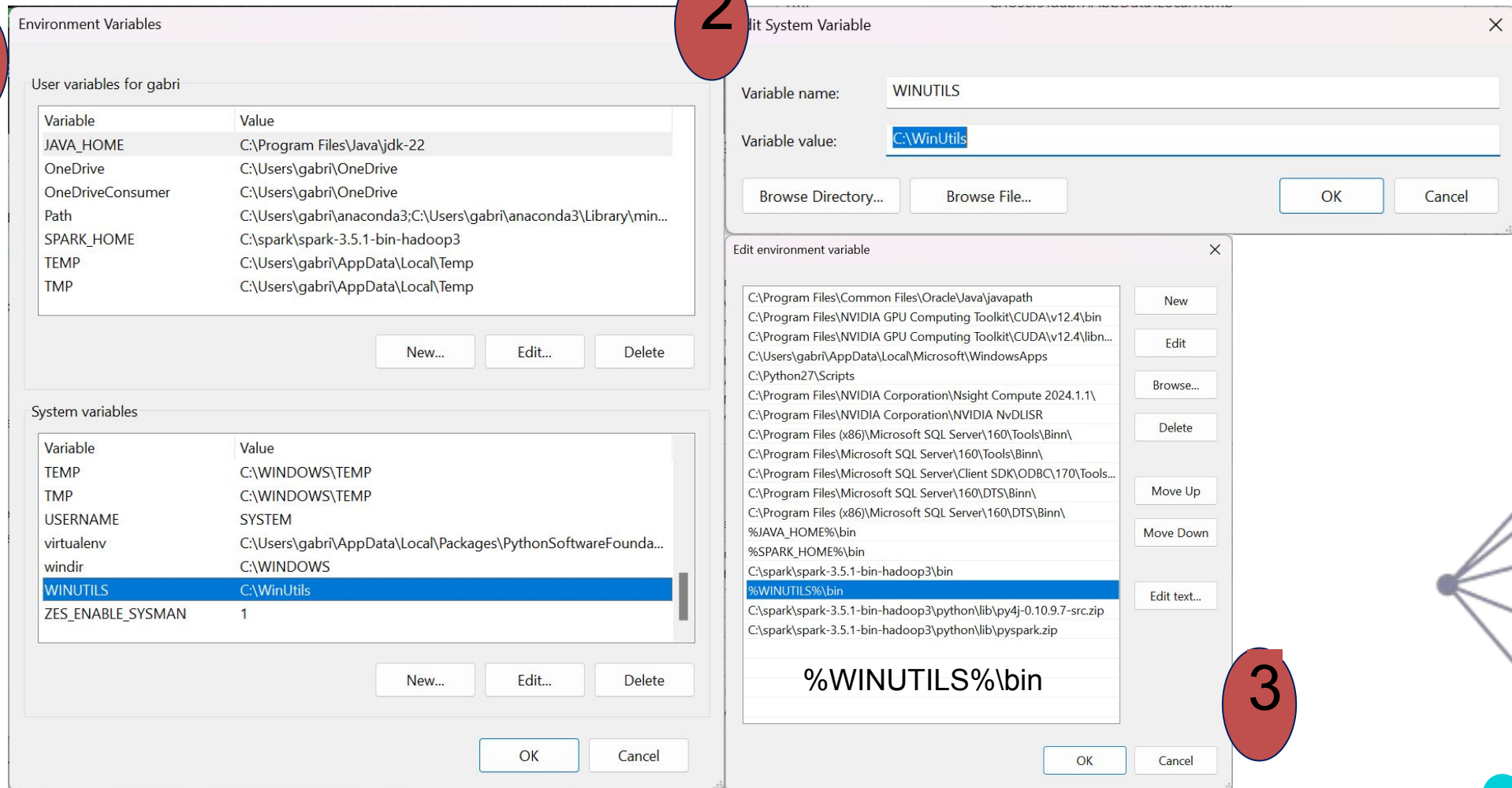


3



Wintutls.exe y hadoop.dll

- Agregar al path



SPARK

<https://www.apache.org/dyn/closer.lua/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz>

Creamos una carpeta en `C:\spark` y extraemos
Agregamos a las variables de entorno `SPARK_HOME` y al path `%SPARK_HOME%\bin`

`C:\spark\spark-3.5.1-bin-hadoop3\python\lib\py4j-0.10.9.7-src.zip`

`C:\spark\spark-3.5.1-bin-hadoop3\python\lib\pyspark.zip`

Abrimos SQL Server Configuration

SPARK

<https://www.apache.org/dyn/closer.lua/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz>

Creamos una carpeta en `C:\spark` y extraemos
Agregamos a las variables de entorno `SPARK_HOME` y al path `%SPARK_HOME%\bin`

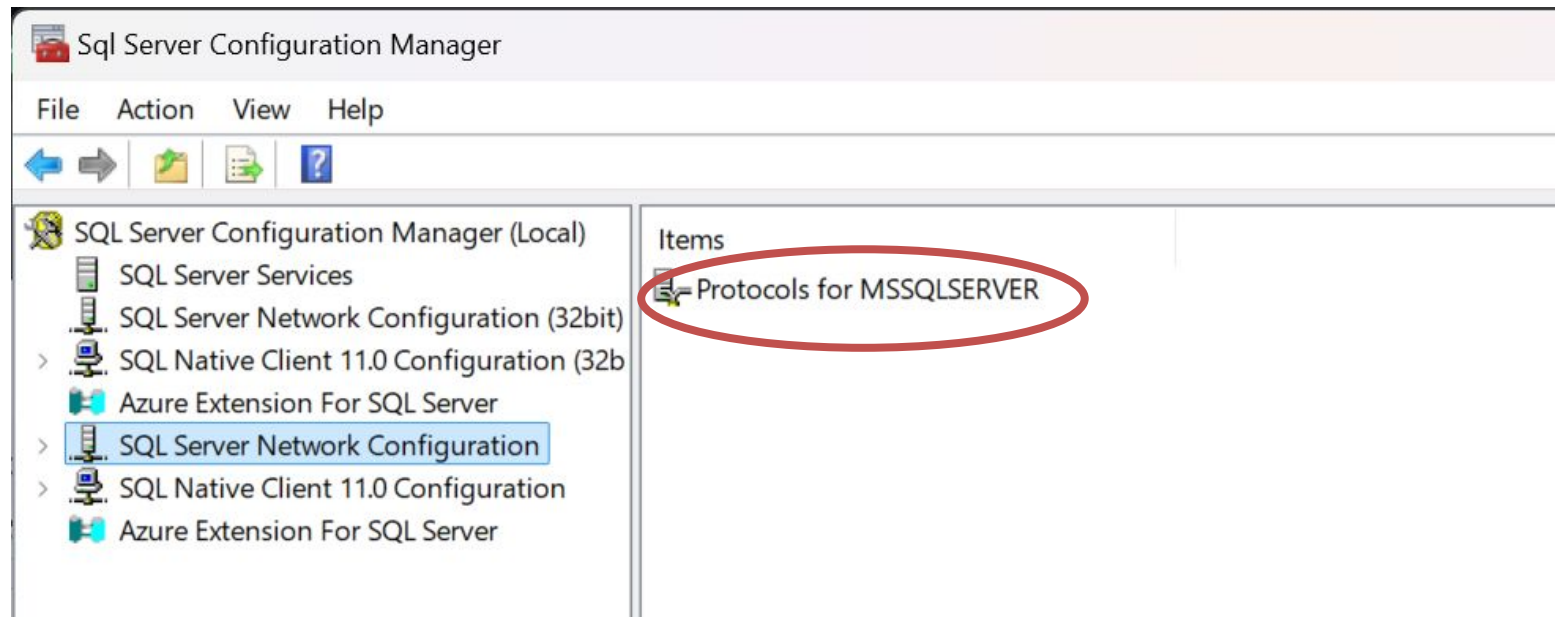
`C:\spark\spark-3.5.1-bin-hadoop3\python\lib\py4j-0.10.9.7-src.zip`

`C:\spark\spark-3.5.1-bin-hadoop3\python\lib\pyspark.zip`

SQL Server y SQL Server Management Studio

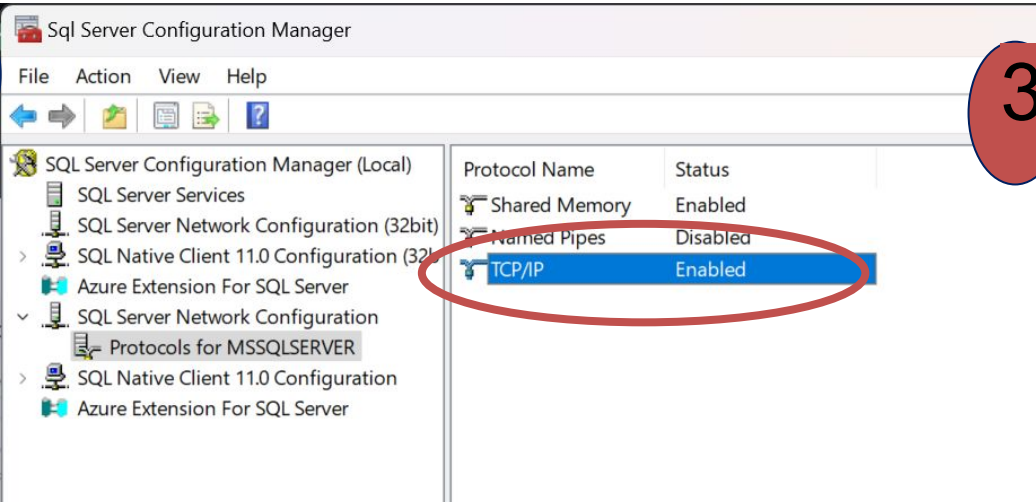
- SQL Server: <https://www.microsoft.com/es-es/sql-server/sql-server-downloads>
- SQL Server Management Studio:
<https://learn.microsoft.com/es-es/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16#download-ssms>

1 Abrir Sql Server Configuration Manager



SQL Server y SQL Server Management Studio

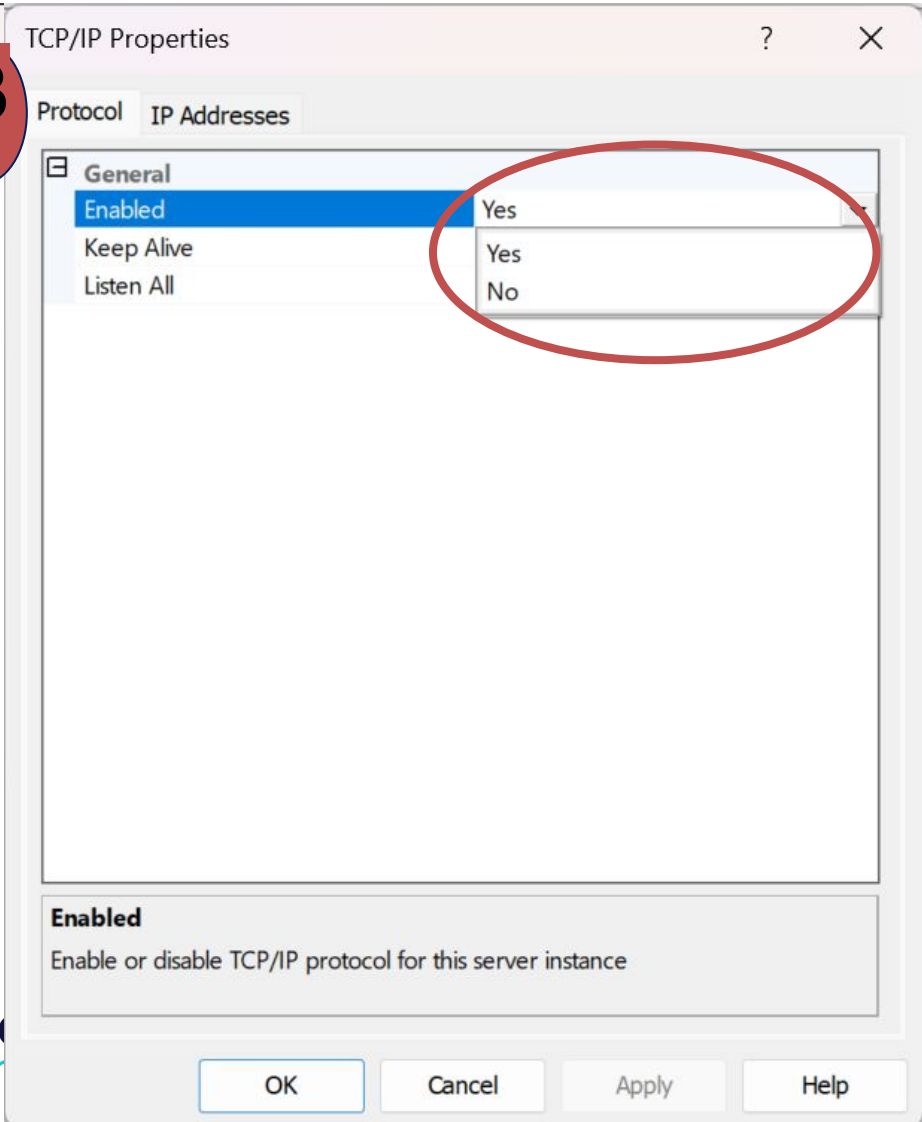
2



SQL Server Configuration Manager (Local)

Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Disabled
TCP/IP	Enabled

3



TCP/IP Properties

Protocol IP Addresses

General

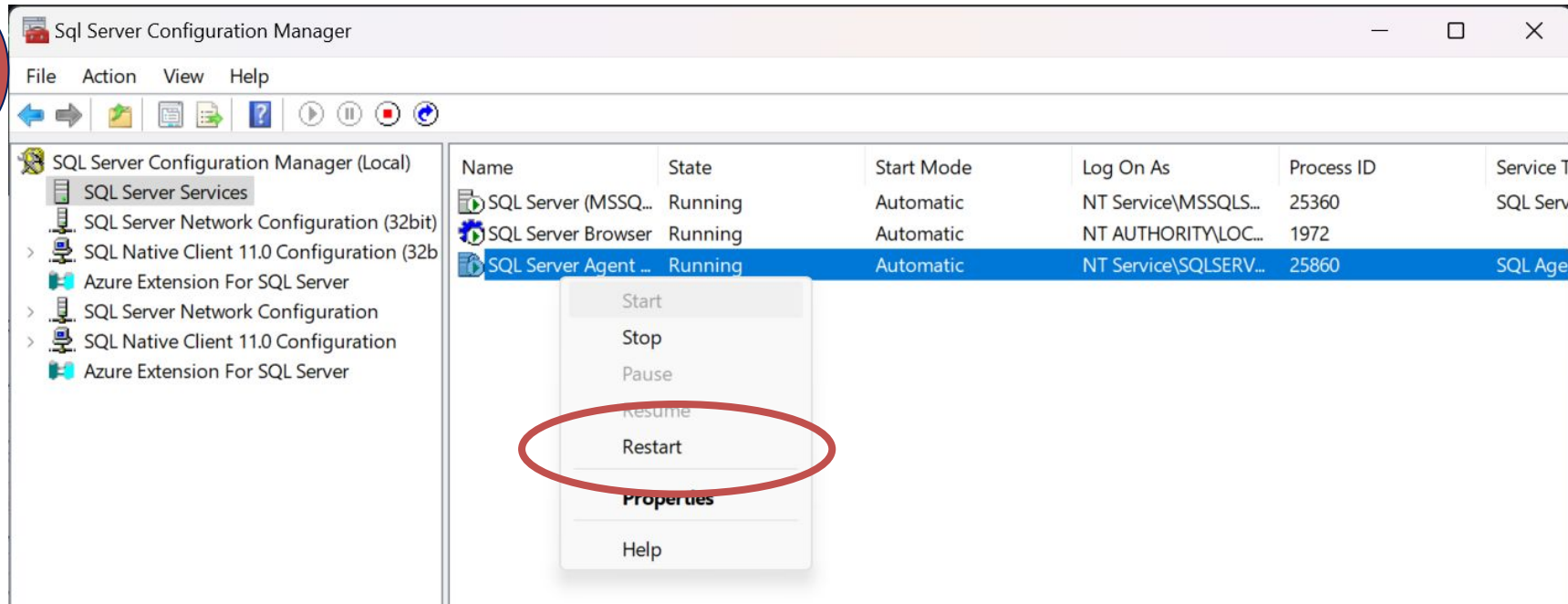
Enabled	Yes
Keep Alive	Yes
Listen All	No

Enabled
Enable or disable TCP/IP protocol for this server instance

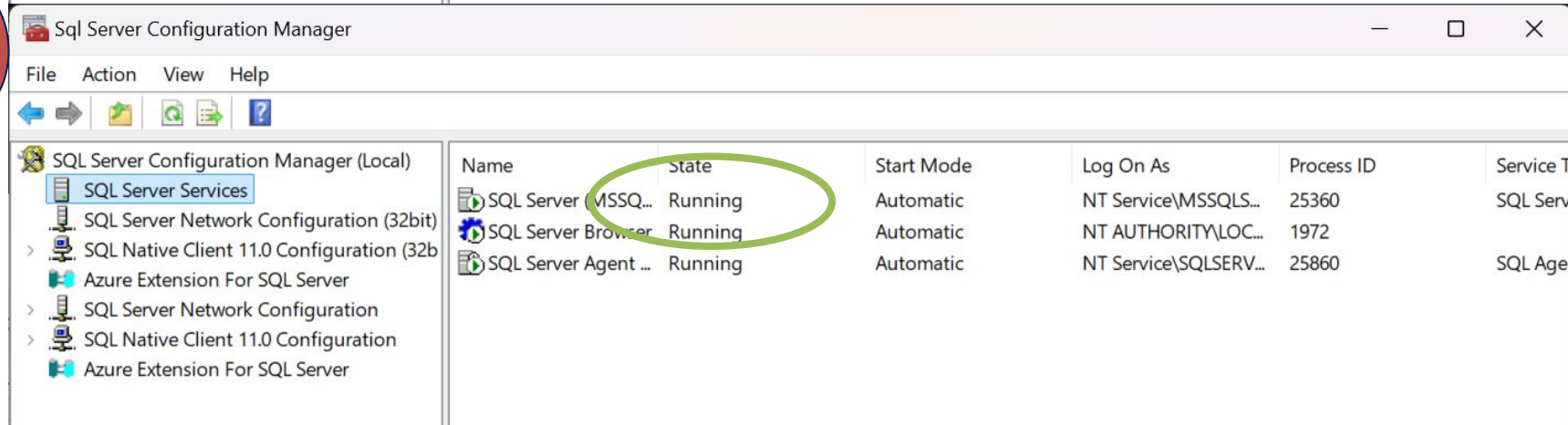
OK Cancel Apply Help

SQL Server y SQL Server Management Studio

4



5



Driver JDBC para MSSQL (Microsoft SQL Server)

<https://learn.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver16>

Extraemos en una carpeta y anotamos el path al archivo:
"enu\jars\mssql-jdbc-12.6.3.jre8.jar"

Python

```
try:
    spark = SparkSession.builder \
        .appName(appName) \
        .config("spark.jars", "C:\\JDBC_sql_server\\sqljdbc_12.6\\enu\\mssql-jdbc-12.6.3.jre8.jar") \
        .getOrCreate()
except Exception as e:
    print(e)
```



Data Growth
Community



Data Growth
Community

"Potenciando el crecimiento
colectivo"

Síguenos en:



datagrowthcommunity



Datagrowth.community



datagrowthcommunity

¡Gracias!