.NET for Apache Spark

What is Apache Spark?

https://spark.apache.org/

Apache Spark™ is a general-purpose **distributed processing** engine for analytics over large data sets—typically, terabytes or petabytes of data.

Apache Spark can be used for processing batches of data, real-time streams, machine learning, and ad-hoc query.

Processing tasks are distributed over a cluster of nodes, and data is cached in-memory, to reduce computation time.

What is .NET For Apache Spark?

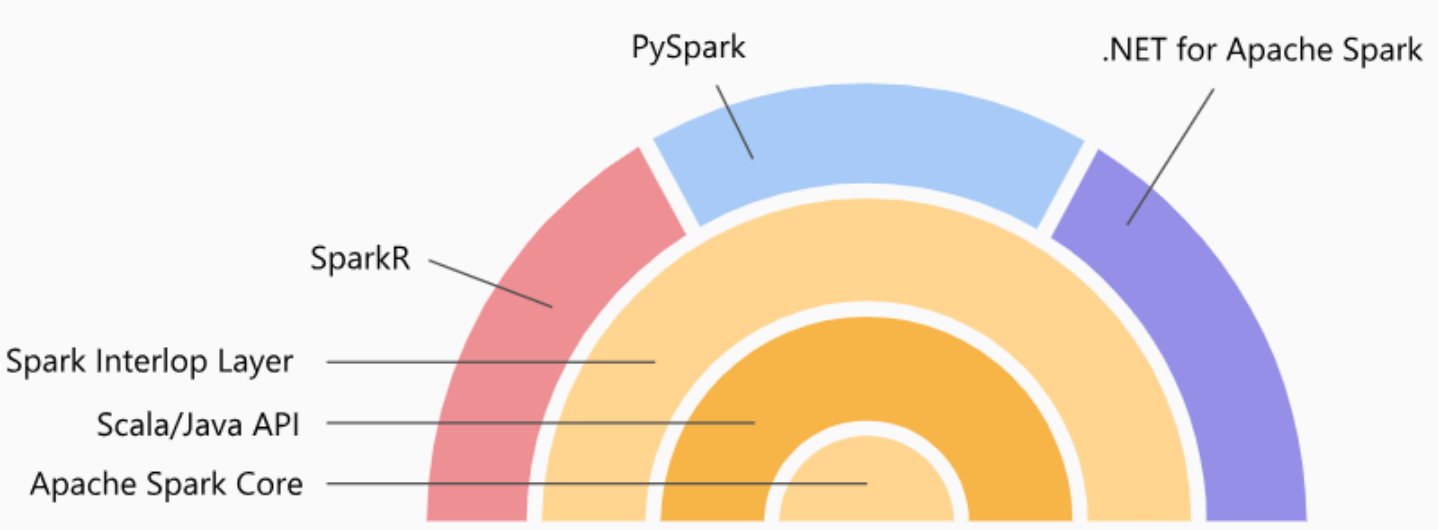
https://dotnet.microsoft.com/en-us/apps/data/spark

https://github.com/dotnet/spark

The .NET bindings for Spark are written on the Spark interop layer, designed to provide high performance bindings to multiple languages.

.NET for Apache Spark is compliant with .NET Standard—a formal specification of .NET APIs that are common across .NET implementations.

This means you can use .NET for Apache Spark anywhere you write .NET code.



.NET for Apache Spark C# Samples github repository

https://github.com/dotnet/spark/tree/main/examples/Microsoft.Spark.CSharp.Examples

There are three main types of samples/apps in the repo:

SQL/Batch: .NET for Apache Spark apps that analyze batch data, or data that has already been produced/stored.

https://github.com/dotnet/spark/tree/main/examples/Microsoft.Spark.CSharp.Examples/Sql/Batch

SQL/Streaming: .NET for Apache Spark apps that analyze structured streaming data, or data that is currently being produced live.

https://github.com/dotnet/spark/tree/main/examples/Microsoft.Spark.CSharp.Examples/Sql/Streaming

Machine Learning: .NET for Apache Spark apps infused with Machine Learning models based on ML.NET, an open source and cross-platform machine learning framework.

https://github.com/dotnet/spark/tree/main/examples/Microsoft.Spark.CSharp.Examples/MachineLearning

Tutorial: Get started with .NET for Apache Spark

https://learn.microsoft.com/es-es/previous-versions/dotnet/spark/tutorials/get-started?tabs=windows

1. Install .NET

To start building .NET apps, you need to download and install the .NET SDK (Software Development Kit).

Download and install the .NET Core SDK. Installing the SDK adds the dotnet toolchain to your PATH.

https://dotnet.microsoft.com/es-es/download/dotnet/3.1

Run the command to check the .Net Core SDK installation:

```
dotnet --version
```

IMPORTANT NOTE .NET for Apache Spark targets an out-of-support version of .NET (.NET Core 3.1).

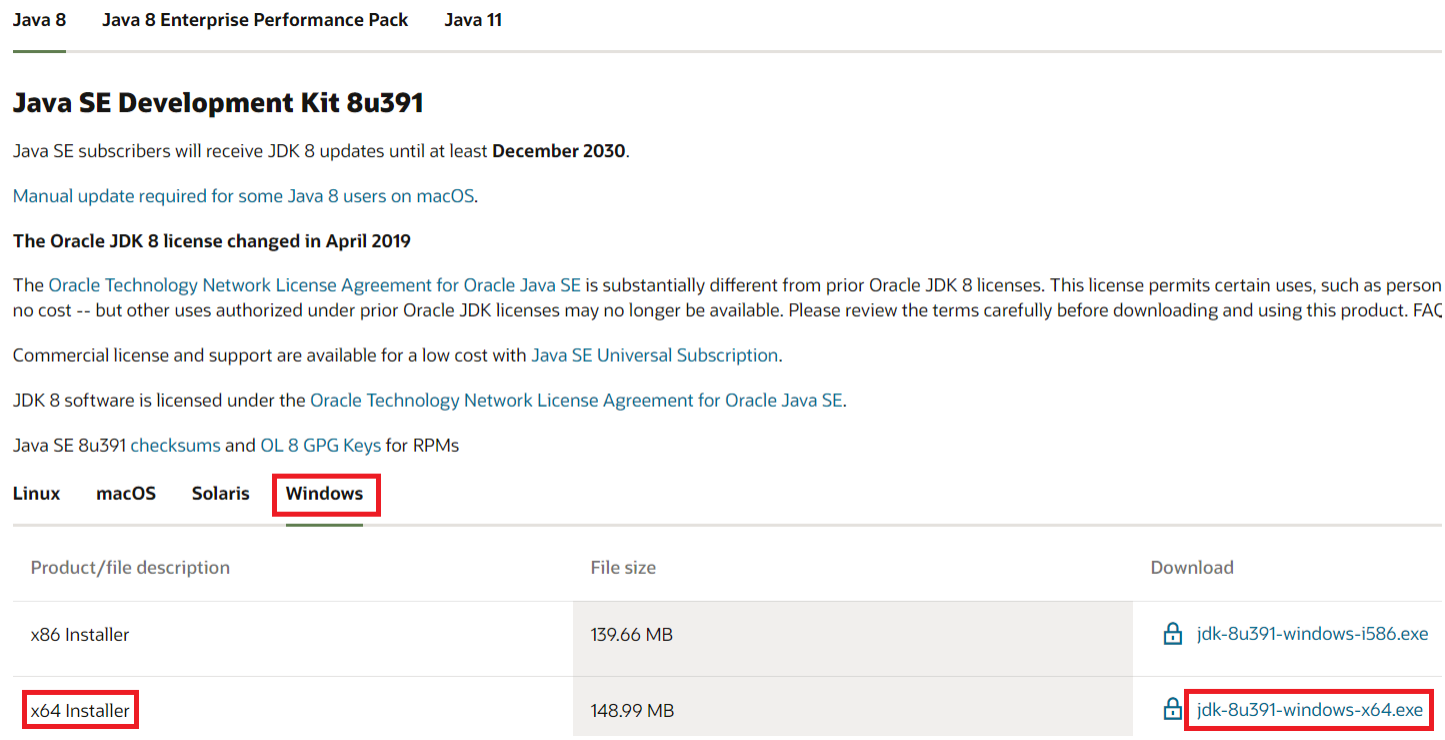
But I have .NET 7 installed in my laptop and .NET for Apache Spark is working fine.

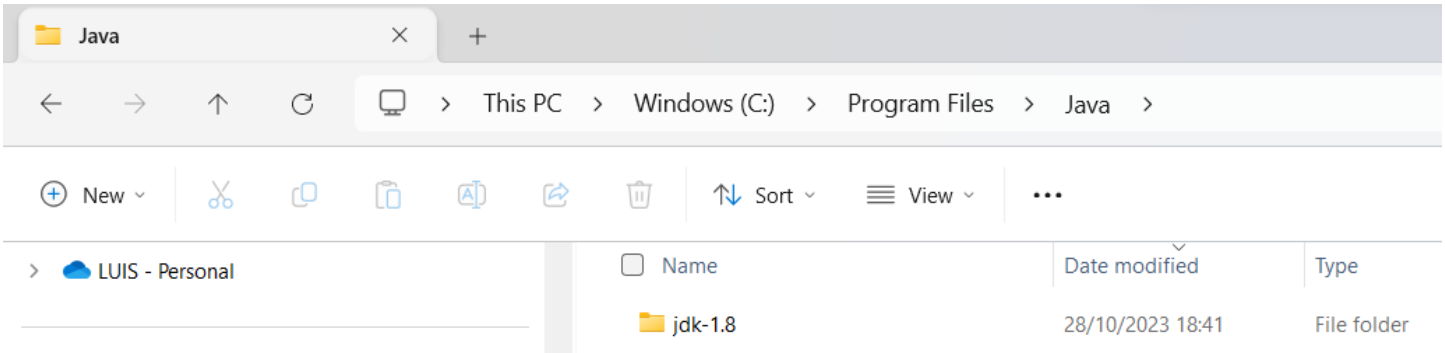
2. Install Java 8 (JDK)

Install Java 8.1 for Windows and macOS, or OpenJDK 8 for Ubuntu.

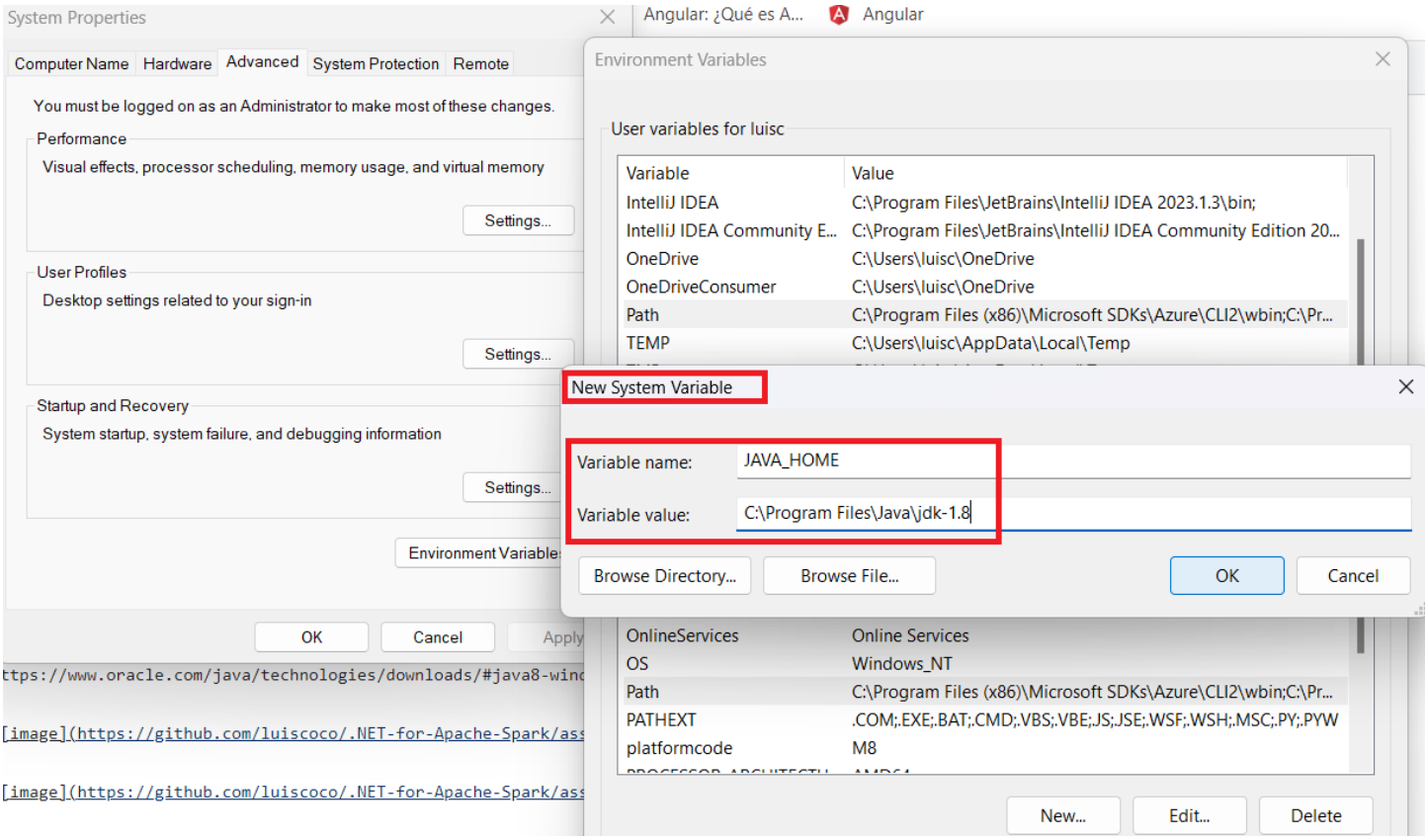
Select the appropriate version for your operating system. For example, select **jdk-8u201-windows-x64.exe** for a Windows x64 machine

https://www.oracle.com/java/technologies/downloads/#java8-windows

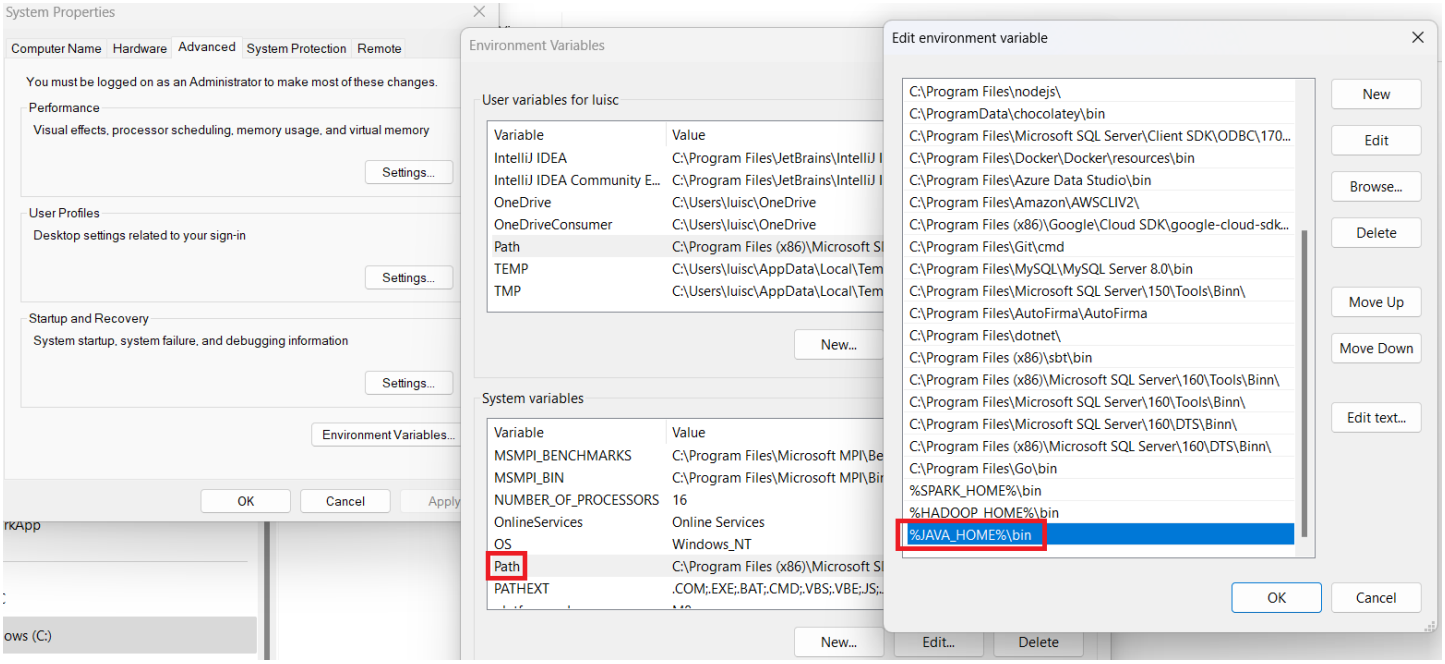




Now set the environmental variable JAVA_HOME poiting the bin folder in the Java installation



And add the JAVA_HOME to the PATH variable



Restart you laptop and check the Java 8 (JDK) installation is fine

Open a command prompt window an run the command

```
java -version
```

Administrator: Command Prompt

```
C:\>java -version
java version "1.8.0_391"
Java(TM) SE Runtime Environment (build 1.8.0_391-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.391-b13, mixed mode)
```

3. Install compression software

Apache Spark is downloaded as a compressed .tgz file.

Use an extraction program, like 7-Zip or WinZip, to extract the file.

4. Install Apache Spark

Download and install Apache Spark. You'll need to select from version 2.3.* or 2.4.0, 2.4.1, 2.4.3, 2.4.4, 2.4.5, 2.4.6, 2.4.7, 3.0.0, 3.0.1, 3.0.2, 3.1.1, 3.1.2, 3.2.0, or 3.2.1

(.NET for Apache Spark is not compatible with other versions of Apache Spark).

spark.apache.org/downloads.html

YouTube Maps Noticias Traducir RxJS v6.6.7 Angular: ¿Qué es A... Angular

```
groupId: org.apache.spark
artifactId: spark-core_2.12
version: 3.5.0
```

Installing with PyPi

PySpark is now available in pypi. To install just run `pip install pyspark`.

Convenience Docker Container Images

Spark Docker Container images are available from DockerHub, these images contain non-ASF software and may be subject to different license terms.

Release notes for stable releases

- Spark 3.5.0 (Sep 13 2023)
- Spark 3.4.1 (Jun 23 2023)
- Spark 3.3.3 (Aug 21 2023)
- Spark 3.2.4 (Apr 13 2023)

Archived releases

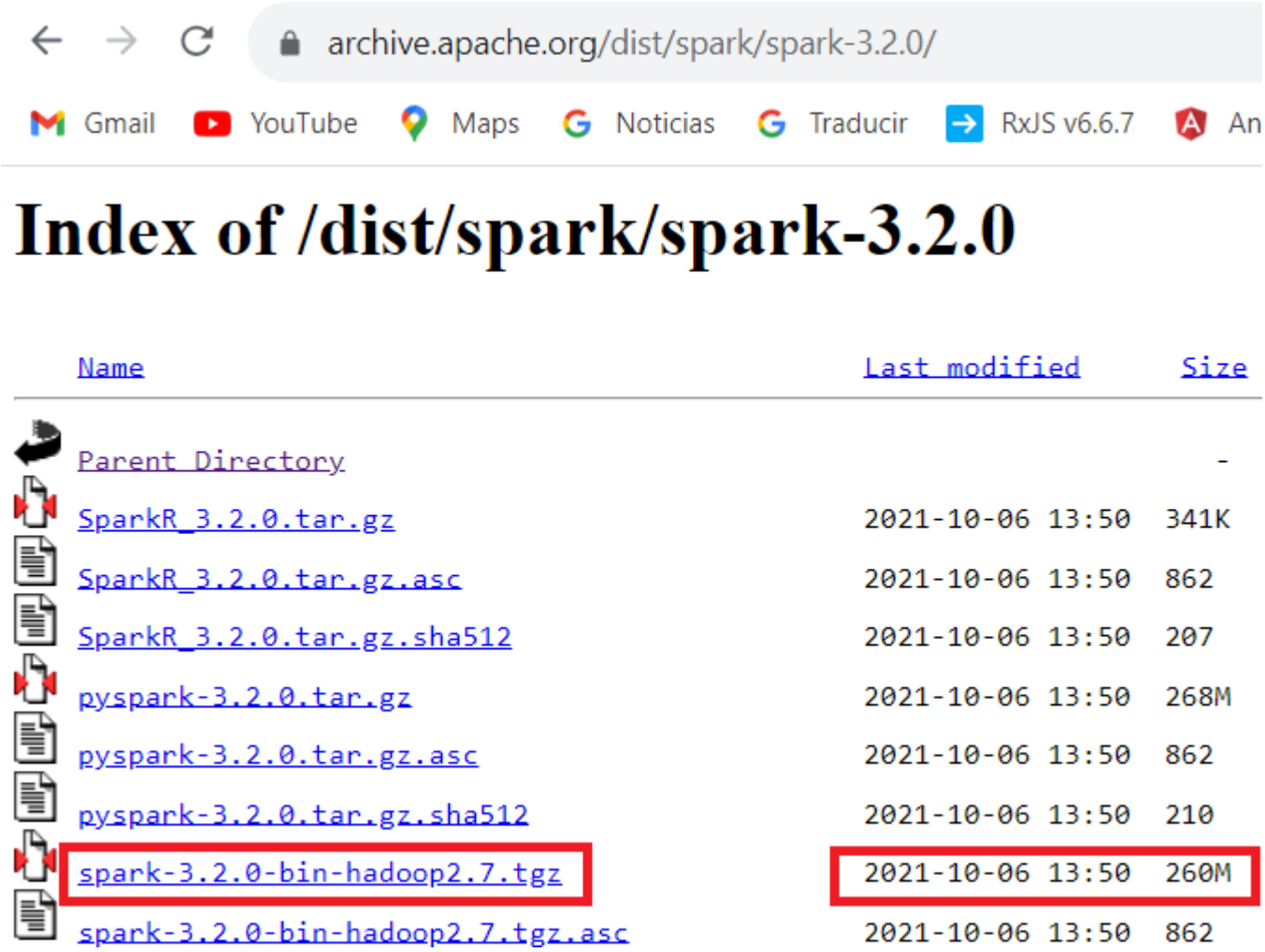
As new Spark releases come out for each development stream, previous ones will be archived, but they are still available at Spark release archives.

Editing .NET-for-Apache-Spark/ | Index of /dist/spark

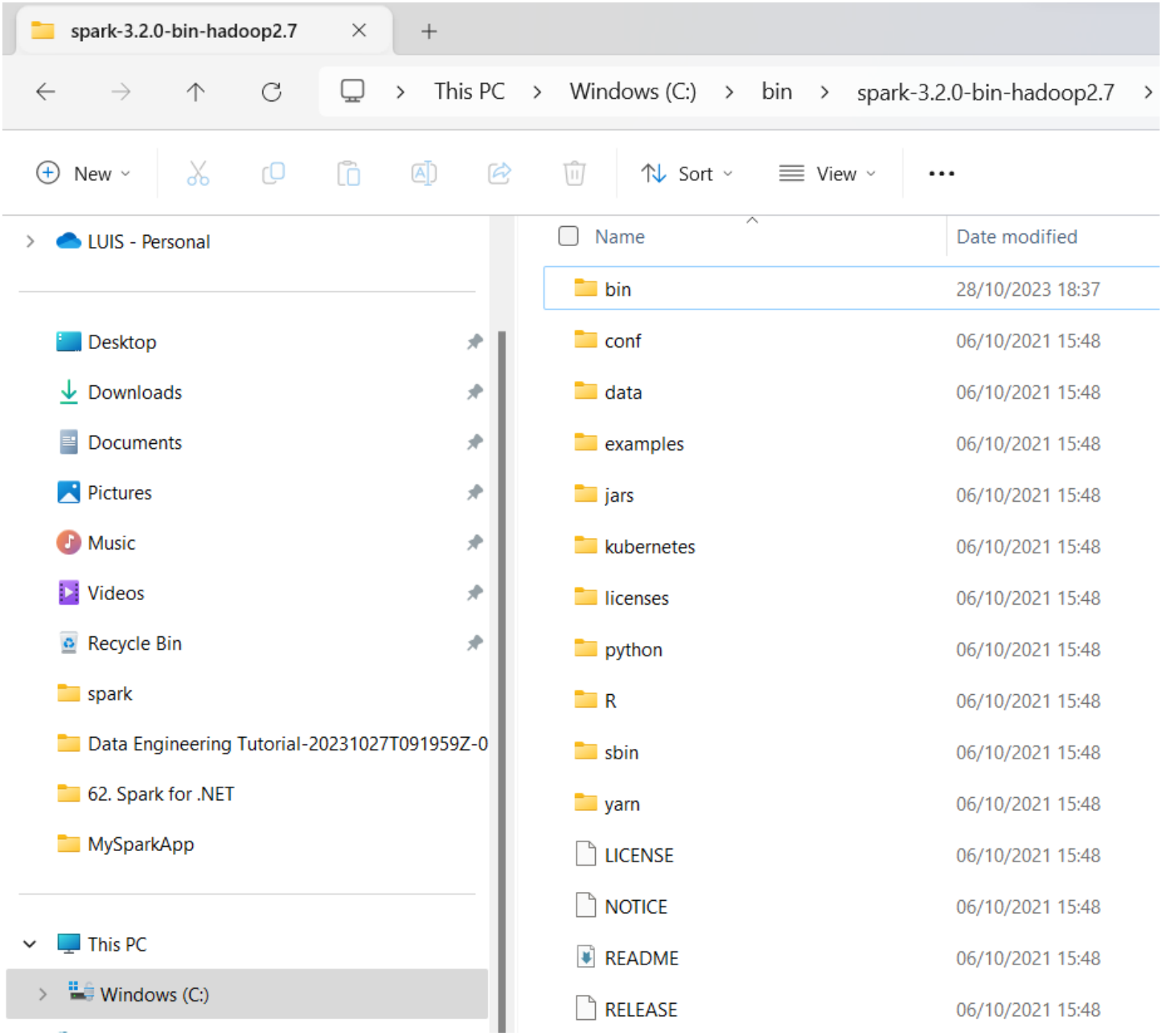
archive.apache.org/dist/spark/

Gmail YouTube Maps Noticias Traducir

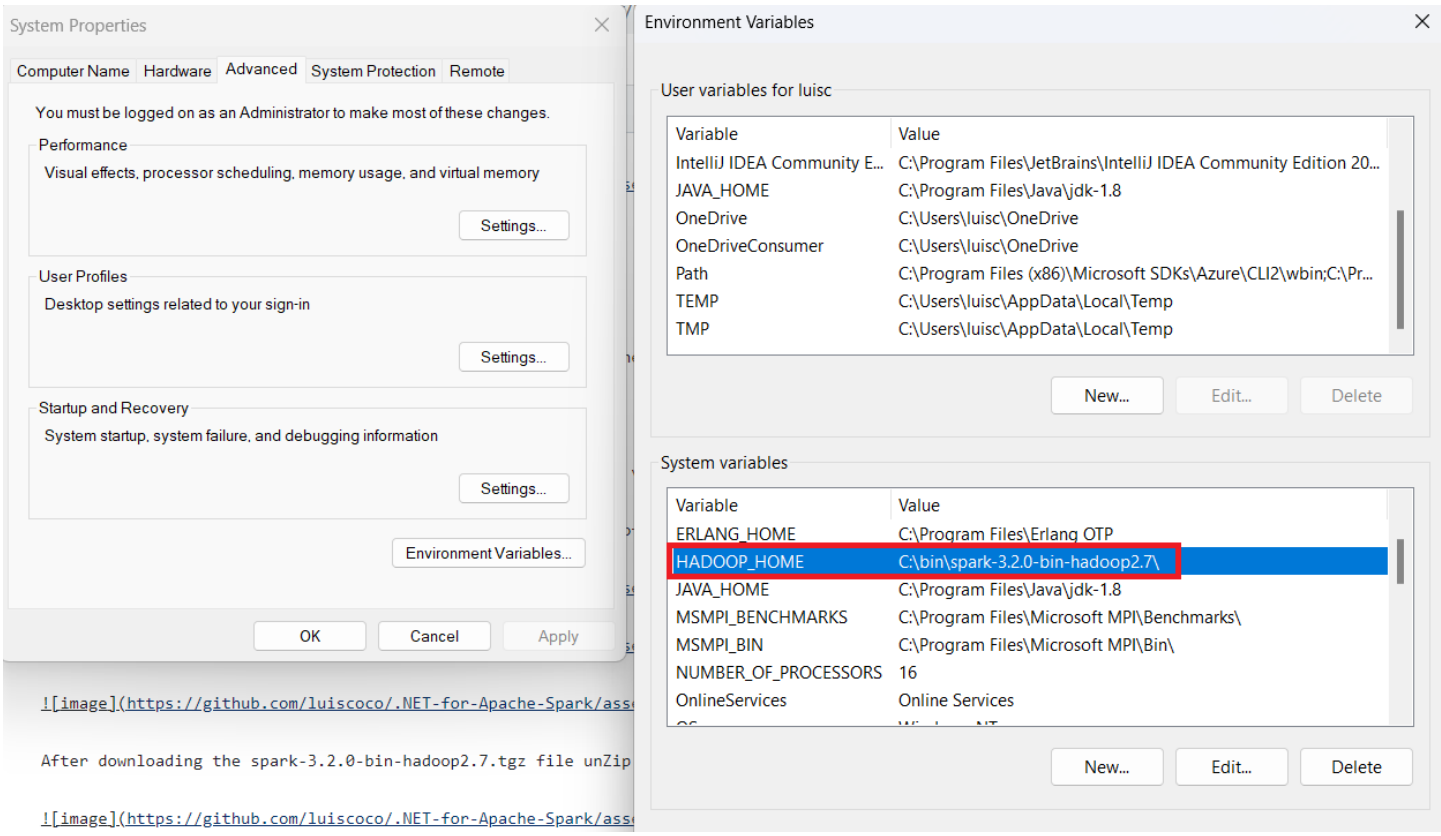
spark-2.4.4/	2019-08-31 05:16	-
spark-2.4.5/	2020-02-06 18:37	-
spark-2.4.6/	2020-06-05 18:02	-
spark-2.4.7/	2020-11-05 18:45	-
spark-2.4.8/	2022-06-17 11:13	-
spark-3.0.0-preview/	2019-11-06 23:15	-
spark-3.0.0-preview2/	2019-12-22 18:53	-
spark-3.0.0/	2020-06-16 09:19	-
spark-3.0.1/	2020-11-05 18:46	-
spark-3.0.2/	2021-02-19 17:24	-
spark-3.0.3/	2022-06-17 11:12	-
spark-3.1.1/	2021-03-02 11:01	-
spark-3.1.2/	2022-06-17 11:12	-
spark-3.1.3/	2022-06-17 11:12	-
spark-3.2.0/	2021-10-13 09:09	-

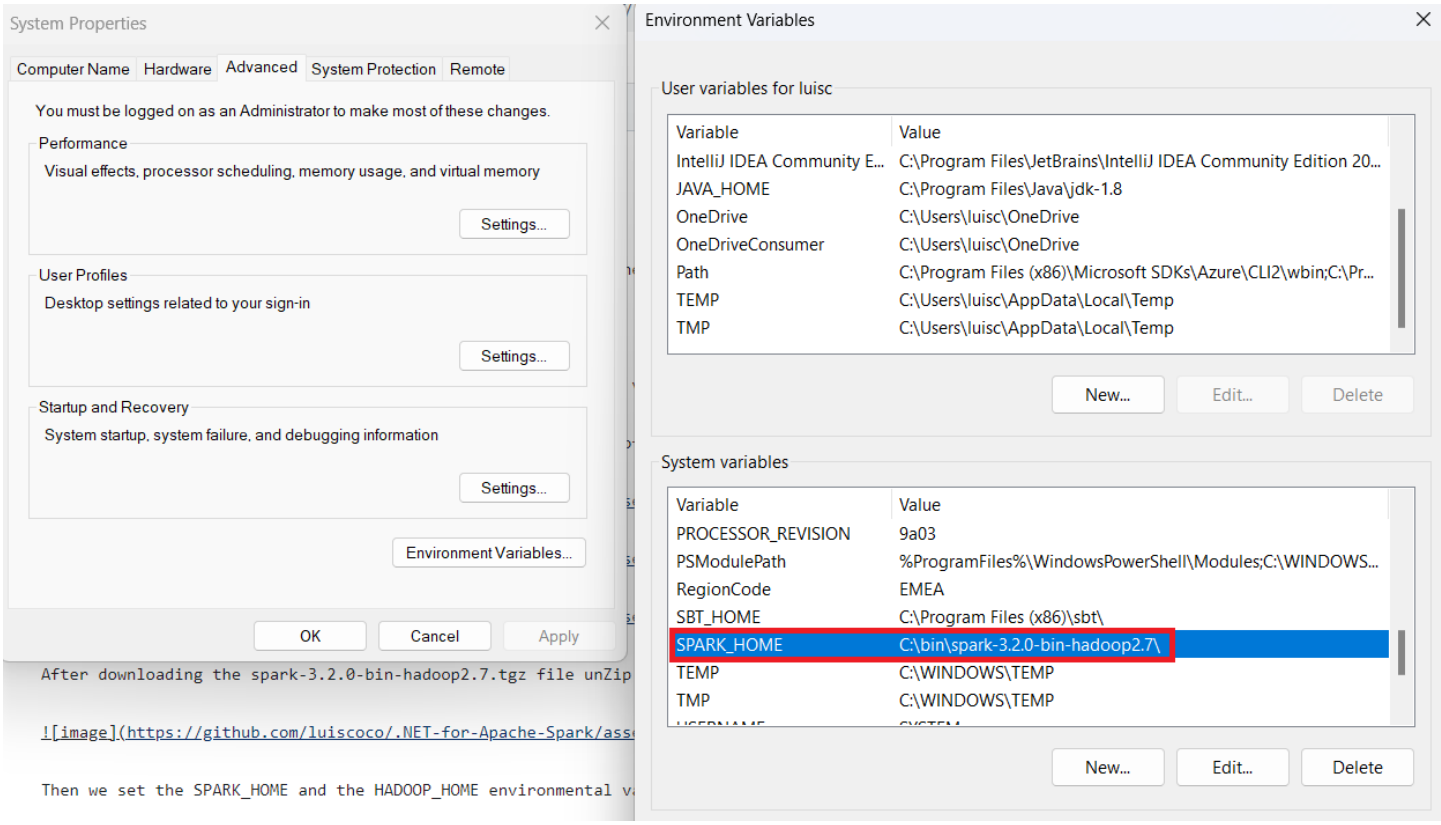


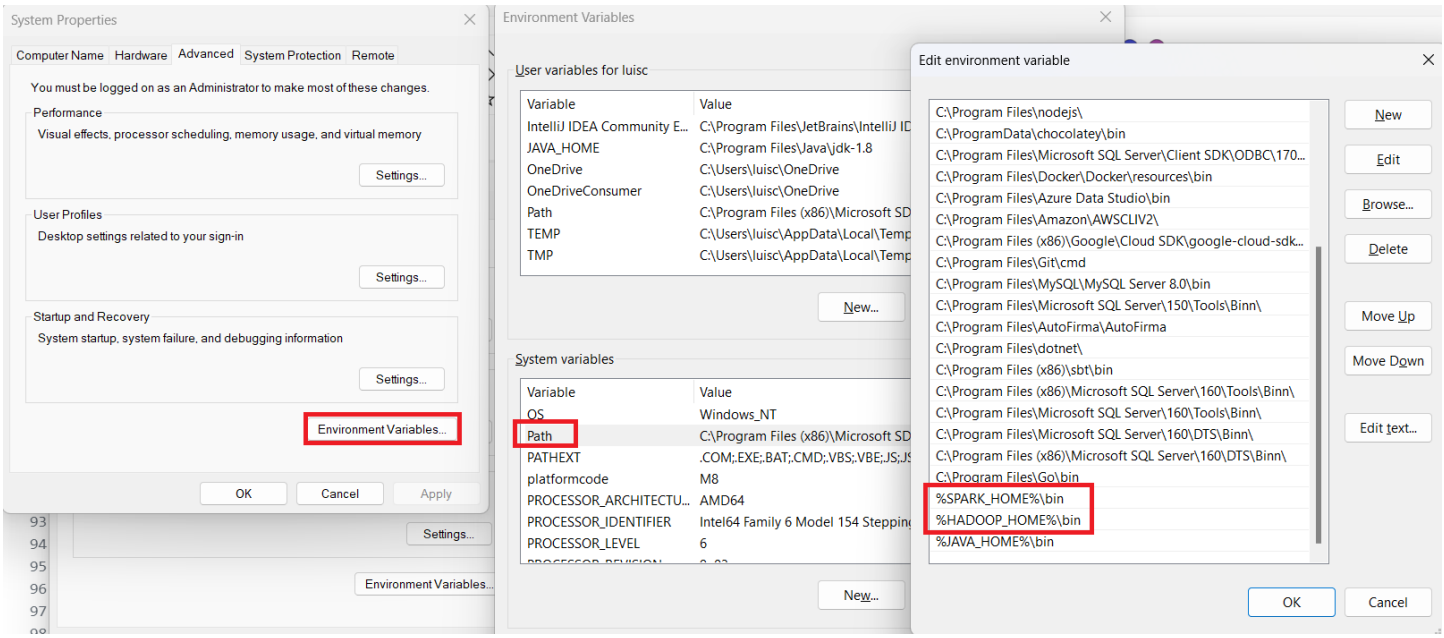
After downloading the spark-3.2.0-bin-hadoop2.7.tgz file unZip it and place it in the following path: C:\bin\spark-3.2.0-bin-hadoop2.7



Then we set the SPARK_HOME and the HADOOP_HOME environmental variables and we add them to the PATH variable


System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply
Environment Variables
User variables for luisc
Variable
Value
IntelliJ IDEA Community E...
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 20...
JAVA_HOME
C:\Program Files\Java\jdk-1.8
OneDrive
C:\Users\luisc\OneDrive
OneDriveConsumer
C:\Users\luisc\OneDrive
Path
C:\Program Files (x86)\Microsoft SDKs\Azure\CLI2\wbin;C:\Pr...
TEMP
C:\Users\luisc\AppData\Local\Temp
TMP
C:\Users\luisc\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
ERLANG_HOME
C:\Program Files\Erlang OTP
HADOOP_HOME
C:\bin\spark-3.2.0-bin-hadoop2.7\
JAVA_HOME
C:\Program Files\Java\jdk-1.8
MSMPI_BENCHMARKS
C:\Program Files\Microsoft MPI\Benchmarks\
MSMPI_BIN
C:\Program Files\Microsoft MPI\Bin\
NUMBER_OF_PROCESSORS
16
OnlineServices
Online Services
New...
Edit...
Delete
![image](https://github.com/luiscoco/.NET-for-Apache-Spark/ass
After downloading the spark-3.2.0-bin-hadoop2.7.tgz file unZip
![image](https://github.com/luiscoco/.NET-for-Apache-Spark/ass

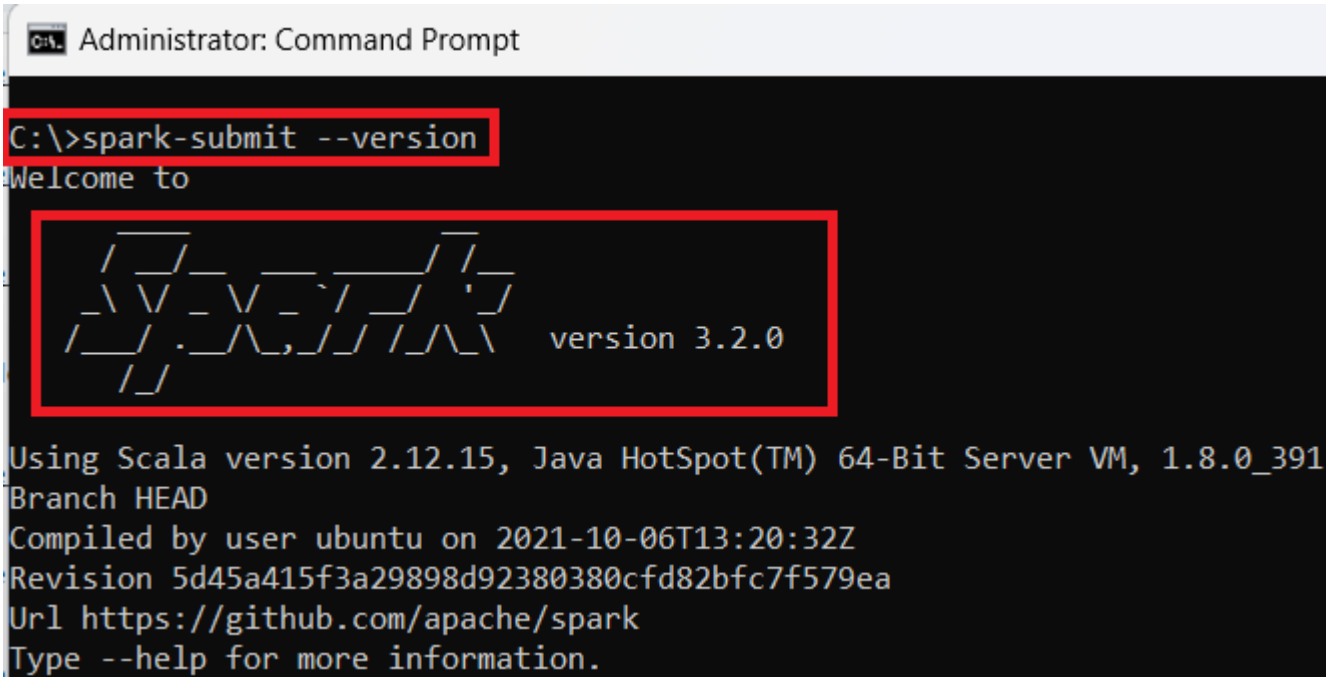

System Properties
Computer Name
Hardware
Advanced
System Protection
Remote
You must be logged on as an Administrator to make most of these changes.
Performance
Visual effects, processor scheduling, memory usage, and virtual memory
Settings...
User Profiles
Desktop settings related to your sign-in
Settings...
Startup and Recovery
System startup, system failure, and debugging information
Settings...
Environment Variables...
OK
Cancel
Apply
Environment Variables
User variables for luisc
Variable
Value
IntelliJ IDEA Community E...
C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 20...
JAVA_HOME
C:\Program Files\Java\jdk-1.8
OneDrive
C:\Users\luisc\OneDrive
OneDriveConsumer
C:\Users\luisc\OneDrive
Path
C:\Program Files (x86)\Microsoft SDKs\Azure\CLI2\wbin;C:\Pr...
TEMP
C:\Users\luisc\AppData\Local\Temp
TMP
C:\Users\luisc\AppData\Local\Temp
New...
Edit...
Delete
System variables
Variable
Value
PROCESSOR_REVISION
9a03
PSModulePath
%ProgramFiles%\WindowsPowerShell\Modules;C:\WINDOWS...
RegionCode
EMEA
SBT_HOME
C:\Program Files (x86)\sbt\
SPARK_HOME
C:\bin\spark-3.2.0-bin-hadoop2.7\
TEMP
C:\WINDOWS\TEMP
TMP
C:\WINDOWS\TEMP
New...
Edit...
Delete
After downloading the spark-3.2.0-bin-hadoop2.7.tgz file unZip
![image](https://github.com/luiscoco/.NET-for-Apache-Spark/ass
Then we set the SPARK_HOME and the HADOOP_HOME environmental v



Now we restart our computer.

We open a new command prompt and run the following command to confirm Spark is installed fine

```
spark-submit --verion
```



5. Install Microsoft.Spark.Worker

Download the Microsoft.Spark.Worker release from the .NET for Apache Spark GitHub.

For example if you're on a Windows machine and plan to use .NET Core, download the Windows x64 netcoreapp3.1 release.

.NET for Apache Spark v2.1.1 Latest

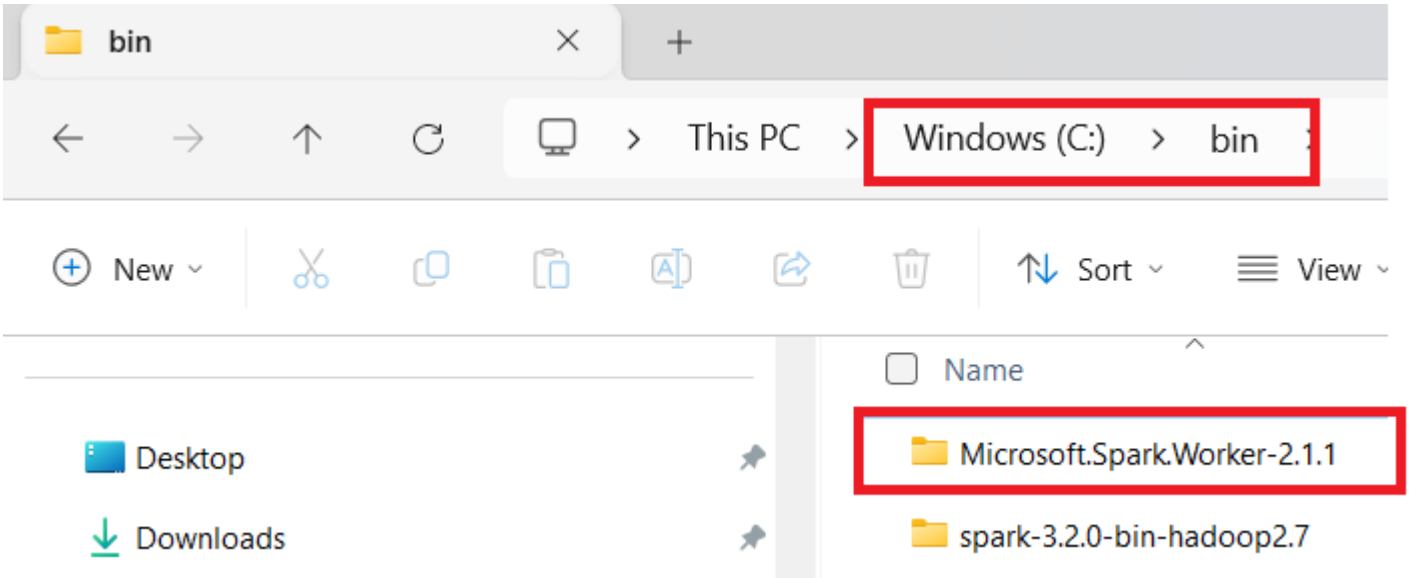
We are excited to announce the release of **.NET for Apache Spark v2.1.1**! Thank you for trying it out and we look forward to your feedback!

Please checkout the Release Notes.

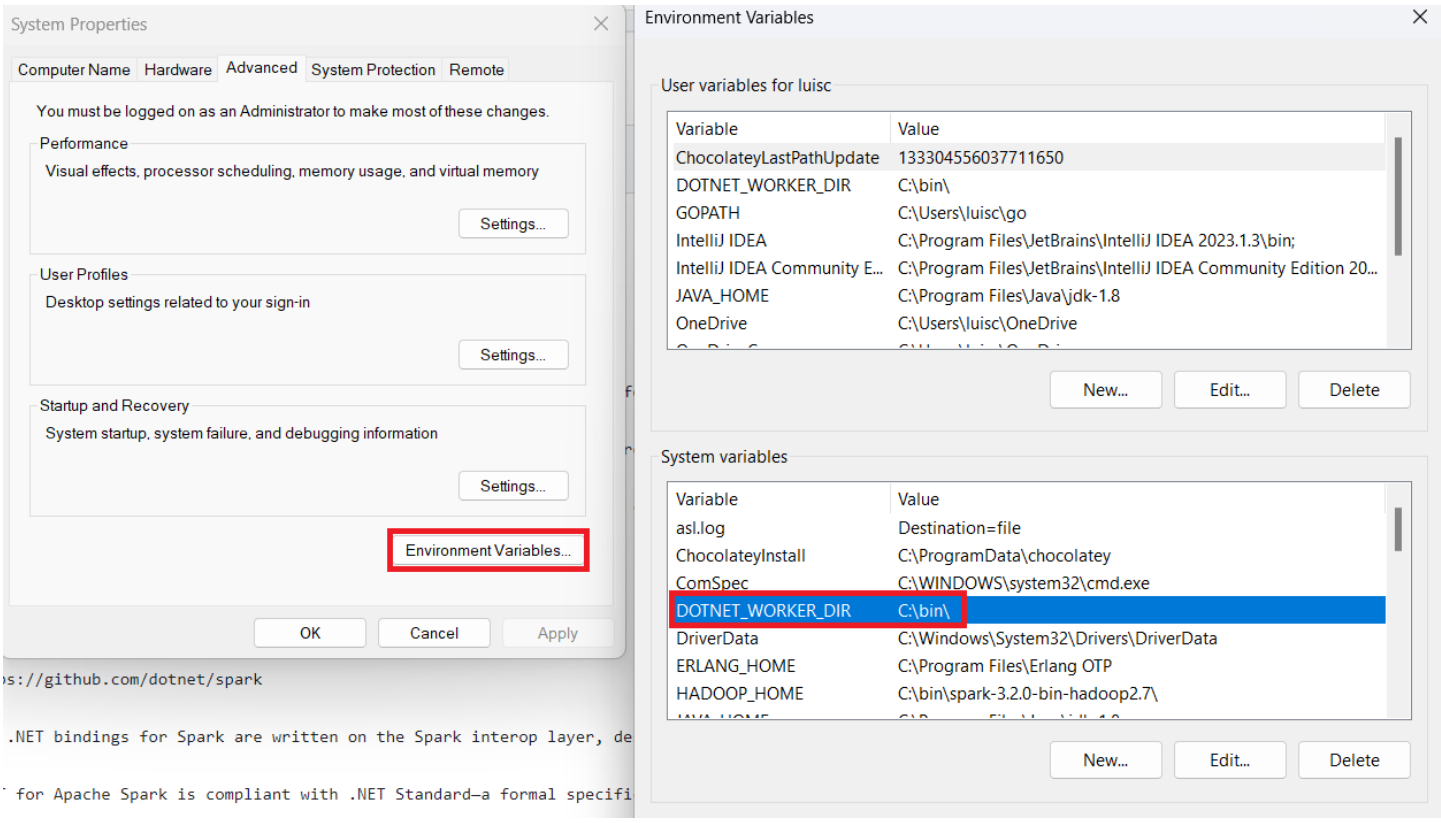
Download `Microsoft.Spark.Worker`:

	netcoreapp3.1	net461
Linux	x64 (tar.gz | zip)	N/A
Windows	x64 (zip)	x64 (zip)
MacOS	x64 (zip)	N/A

After downloading the compressed file "Microsoft.Spark.Worker.netcoreapp3.1.win-x64-2.1.1" place it in C:\bin\



6. Create a new environmental variable DOTNET_WORKER_DIR to store the Microsoft.Spark.Worker installation folder



7. Install WinUtils (Windows only)

.NET for Apache Spark requires WinUtils to be installed alongside Apache Spark.

Download winutils.exe

https://github.com/steveloughran/winutils/tree/master

Then, copy WinUtils into C:\bin\spark-3.0.1-bin-hadoop2.7\bin

> Windows (C:) > bin > spark-3.2.0-bin-hadoop2.7 > bin

Sort View ...

Name	Date modified	Type	Size
pyspark2	06/10/2021 15:48	Windows Comma...	2 KB
run-example	06/10/2021 15:48	File	2 KB
run-example	06/10/2021 15:48	Windows Comma...	2 KB
spark-class	06/10/2021 15:48	File	4 KB
spark-class	06/10/2021 15:48	Windows Comma...	2 KB
spark-class2	06/10/2021 15:48	Windows Comma...	3 KB
sparkR	06/10/2021 15:48	File	2 KB
sparkR	06/10/2021 15:48	Windows Comma...	2 KB
sparkR2	06/10/2021 15:48	Windows Comma...	2 KB
spark-shell	06/10/2021 15:48	File	4 KB
spark-shell	06/10/2021 15:48	Windows Comma...	2 KB
spark-shell2	06/10/2021 15:48	Windows Comma...	2 KB
spark-sql	06/10/2021 15:48	File	2 KB
spark-sql	06/10/2021 15:48	Windows Comma...	2 KB
spark-sql2	06/10/2021 15:48	Windows Comma...	2 KB
spark-submit	06/10/2021 15:48	File	2 KB
spark-submit	06/10/2021 15:48	Windows Comma...	2 KB
spark-submit2	06/10/2021 15:48	Windows Comma...	2 KB
winutils	10/11/2022 12:09	Application	107 KB

8. Create a .NET console application

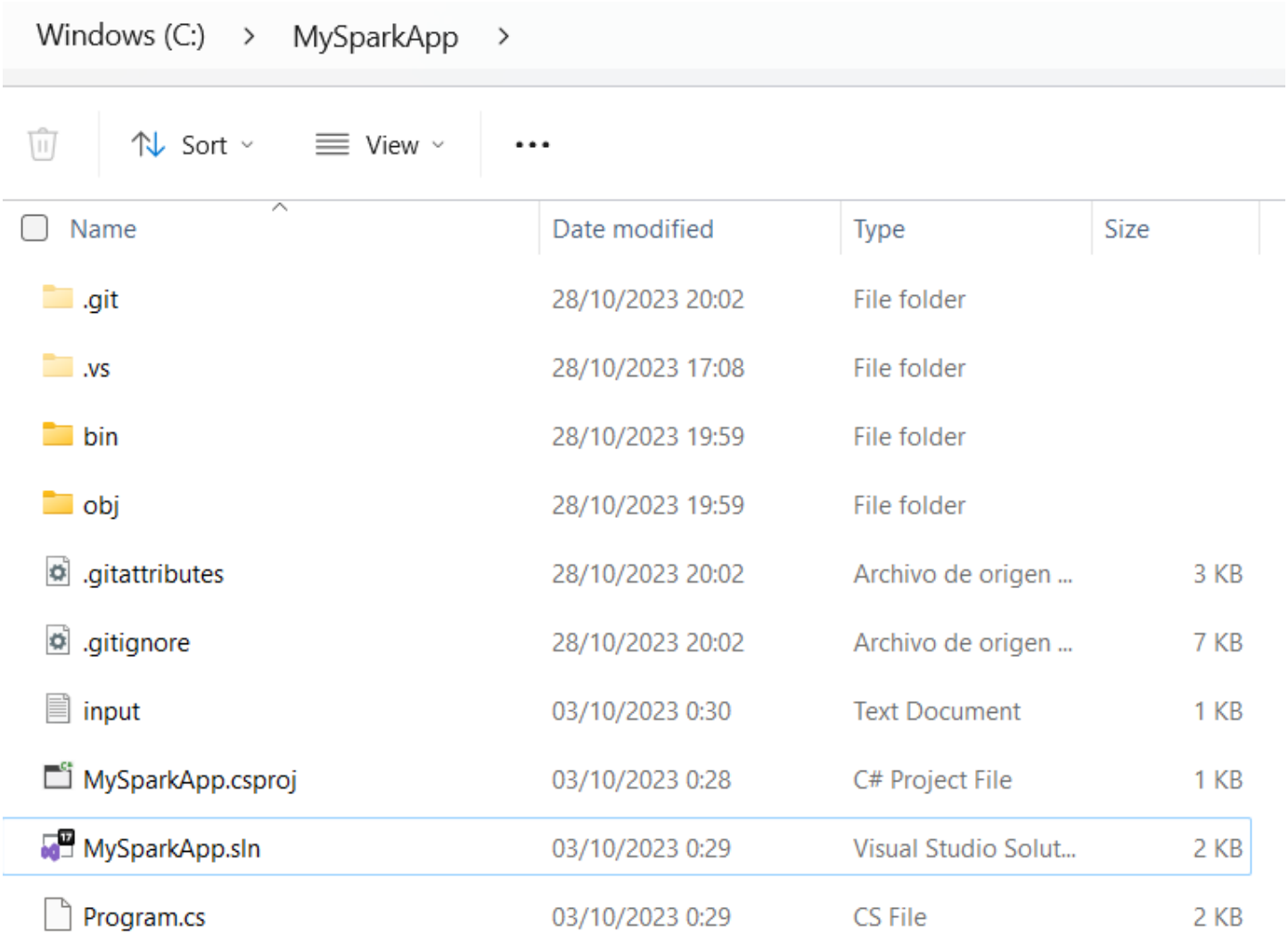
Open a command prompt window and run the following command to create a console .NET application

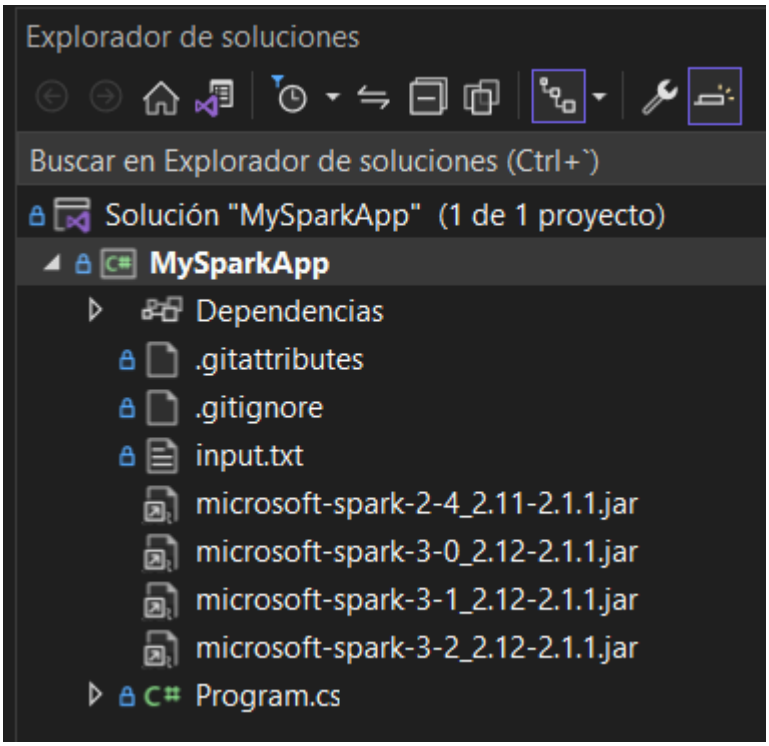
```
dotnet new console -o MySparkApp
cd MySparkApp
```

The dotnet command creates a new application of type console for you.

The -o parameter creates a directory named MySparkApp where your app is stored and populates it with the required files.

The cd MySparkApp command changes the directory to the app directory you created.



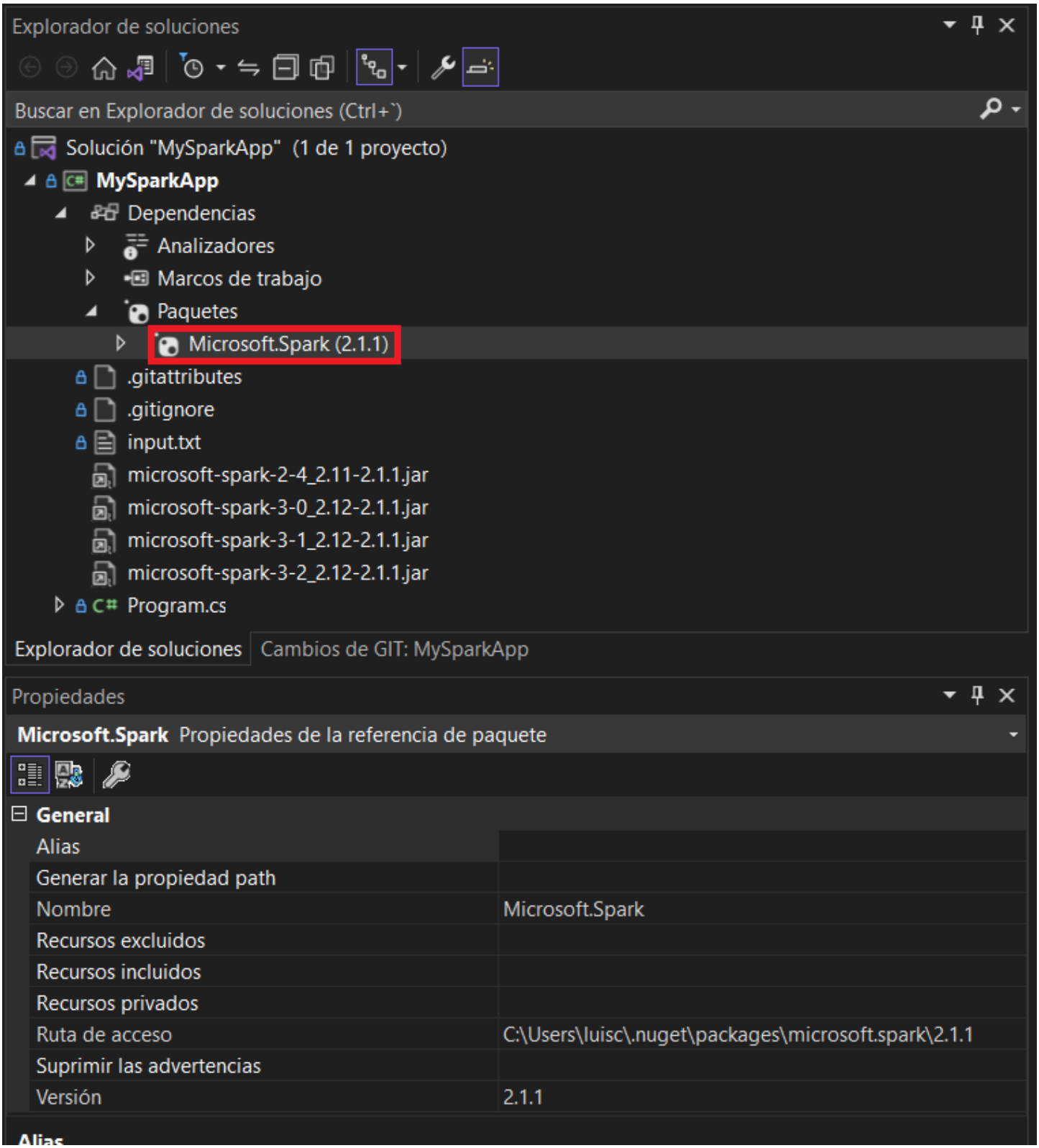


9. Install NuGet package

To use .NET for Apache Spark in an app, install the Microsoft.Spark package.

In your command prompt or terminal, run the following command:

```
dotnet add package Microsoft.Spark
```



10. Write your app

Open Program.cs in Visual Studio Code, or any text editor, and replace all of the code with the following:

```
using Microsoft.Spark.Sql;
using static Microsoft.Spark.Sql.Functions;

namespace MySparkApp
```

```
{
    class Program
    {
        static void Main(string[] args)
        {
            // Create Spark session
            SparkSession spark =
                SparkSession
                    .Builder()
                    .AppName("word_count_sample")
                    .GetOrCreate();

            // Create initial DataFrame
            string filePath = args[0];
            DataFrame dataFrame = spark.Read().Text(filePath);

            //Count words
            DataFrame words =
                dataFrame
                    .Select(Split(Col("value")," ").Alias("words"))
                    .Select(Explode(Col("words")).Alias("word"))
                    .GroupBy("word")
                    .Count()
                    .OrderBy(Col("count").Desc());

            // Display results
            words.Show();

            // Stop Spark session
            spark.Stop();
        }
    }
}
```

SparkSession is the entrypoint of Apache Spark applications, which manages the context and information of your application.

Using the **Text method**, the text data from the file specified by the filePath is read into a DataFrame.

A **DataFrame** is a way of organizing data into a set of named columns. Then, a series of **transformations** is applied to split the sentences in the file, group each of the words, count them and order them in descending order. The result of these operations is stored in another DataFrame.

Note that at this point, no operations have taken place because .NET for Apache Spark **lazily evaluates the data**.

It's not until the **Show** method is called to display the contents of the words transformed DataFrame to the console that the **operations** defined in the lines above **execute**.

Once you no longer need the Spark session, use the **Stop** method to stop your session.

11. Create data file

Your app processes a file containing lines of text.

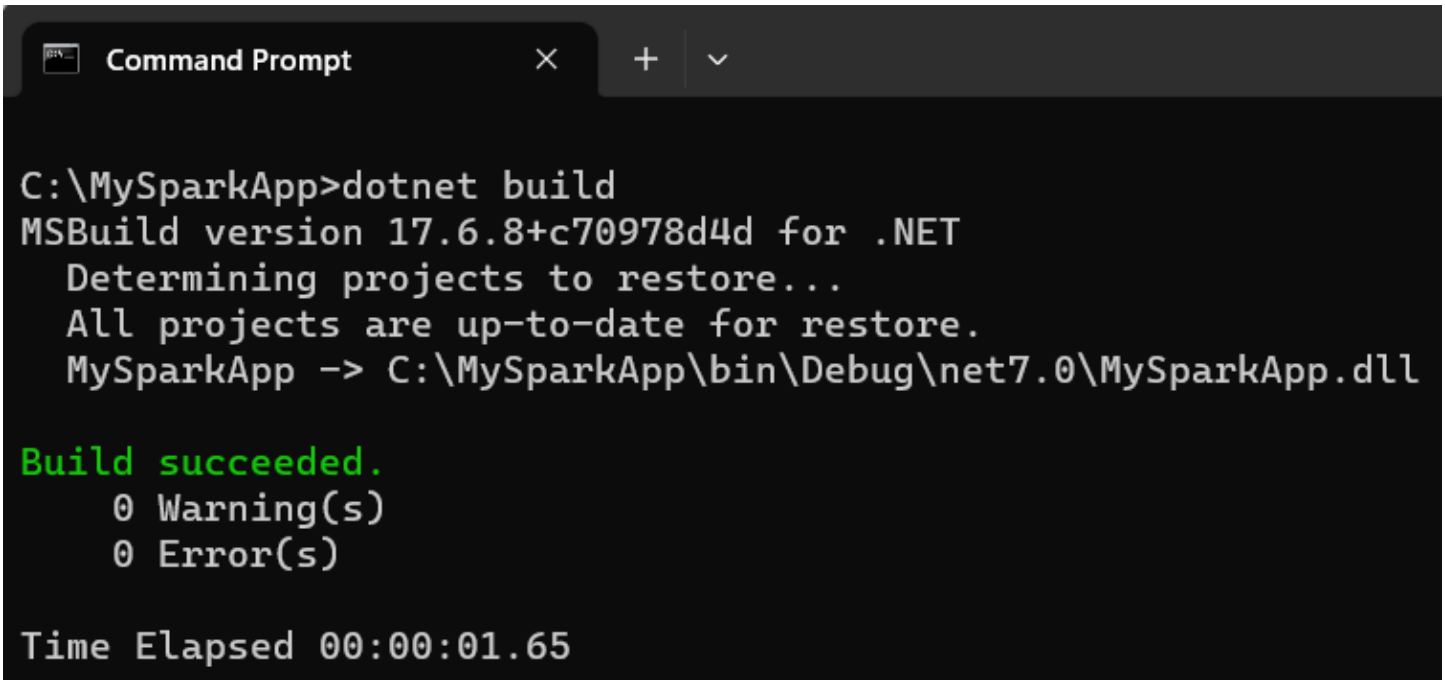
Create a file called **input.txt** file in your **MySparkApp directory**, containing the following text:

```
Hello World
This .NET app uses .NET for Apache Spark
This .NET app counts words with Apache Spark
```

12. Run your .NET for Apache Spark app

Run the following command to build your application:

```
dotnet build
```



Navigate to your build output directory and use the spark-submit command to submit your application to run on Apache Spark.

Make sure to replace with the version of your .NET worker and <path-of-input.txt> with the path of your input.txt file is stored.

```
C:\MySparkApp\bin\Debug\net7.0>spark-submit ^
More? --class org.apache.spark.deploy.dotnet.DotnetRunner ^
More? --master local ^
More? microsoft-spark-3-2_2.12-2.1.1.jar ^
More? dotnet MySparkApp.dll C:\MySparkApp\input.txt
```

spark-submit is a command used to submit Spark applications to a cluster

org.apache.spark.deploy.dotnet package. This class is responsible for **running .NET Spark applications**

--master local specifies the **Spark master URL**. In this case, it's set to "local," which means the Spark application will run locally on the machine. It won't be submitted to a Spark cluster.

microsoft-spark-3-2_2.12-2.1.1.jar specifies the JAR file containing the Microsoft Spark assembly. This JAR file is required for running .NET Spark applications. The version numbers and other details in the filename may vary based on your specific setup.

In the last line we run the **dotnet** command to run the **MySparkApp.dll** assembly, and we send as argument the text file path **C:\MySparkApp\input.txt**

13. Output after running the application

```
Command Prompt

C:\MySparkApp\bin\Debug\net7.0>spark-submit ^
More? --class org.apache.spark.deploy.dotnet.DotnetRunner ^
More? --master local ^
More? microsoft-spark-3-2_2.12-2.1.1.jar ^
More? dotnet MySparkApp.dll C:\MySparkApp\input.txt
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
23/10/28 21:49:54 INFO DotnetRunner: Starting DotnetBackend with dotnet.
23/10/28 21:49:54 INFO DotnetBackend: The number of DotnetBackend threads is set to 10.
23/10/28 21:49:55 INFO DotnetRunner: Port number used by DotnetBackend is 51399
23/10/28 21:49:55 INFO DotnetRunner: Adding key=spark.jars and value=file:/C:/MySparkApp/bin/Debug/net7.0/microsoft-spar
k-3-2_2.12-2.1.1.jar to environment
23/10/28 21:49:55 INFO DotnetRunner: Adding key=spark.app.name and value=org.apache.spark.deploy.dotnet.DotnetRunner to
environment
23/10/28 21:49:55 INFO DotnetRunner: Adding key=spark.submit.pyFiles and value= to environment
23/10/28 21:49:55 INFO DotnetRunner: Adding key=spark.submit.deployMode and value=client to environment
23/10/28 21:49:55 INFO DotnetRunner: Adding key=spark.master and value=local to environment
[2023-10-28T19:49:56.0069209Z] [LUISCOCOENRIQUE] [Info] [ConfigurationService] Using port 51399 for connection.
[2023-10-28T19:49:56.0139918Z] [LUISCOCOENRIQUE] [Info] [JvmBridge] JvMBridge port is 51399
[2023-10-28T19:49:56.0159726Z] [LUISCOCOENRIQUE] [Info] [JvmBridge] The number of JVM backend thread is set to 10. The m
```

```
Command Prompt

23/10/28 21:50:09 INFO DAGScheduler: ResultStage 2 (showString at NativeMethodAccessorImpl.java:0) finished in 0,332 s
23/10/28 21:50:09 INFO DAGScheduler: Job 1 is finished. Cancelling potential speculative or zombie tasks for this job
23/10/28 21:50:09 INFO TaskSchedulerImpl: Killing all running tasks in stage 2: Stage finished
23/10/28 21:50:09 INFO DAGScheduler: Job 1 finished: showString at NativeMethodAccessorImpl.java:0, took 0,369996 s
23/10/28 21:50:09 INFO CodeGenerator: Code generated in 15.1312 ms
23/10/28 21:50:09 INFO CodeGenerator: Code generated in 18.5671 ms
+------+-----+
|  word|count|
+------+-----+
|  .NET|    3|
|Apache|    2|
|  This|    2|
| Spark|    2|
|   app|    2|
| World|    1|
|   for|    1|
|counts|    1|
| words|    1|
|  with|    1|
|  uses|    1|
| Hello|    1|
+------+-----+

23/10/28 21:50:09 INFO SparkUI: Stopped Spark web UI at http://host.docker.internal:4040
23/10/28 21:50:09 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
23/10/28 21:50:09 INFO MemoryStore: MemoryStore cleared
23/10/28 21:50:09 INFO BlockManager: BlockManager stopped
```