



Índice

1. Introducción	4
2. Instalación y Configuración en Windows 10	5
2.1. Requisitos	5
2.2. Java JDK	6
2.2.1. Descarga	6
2.2.2. Instalacion	7
2.2.3. Variables de entorno	8
2.3. Anaconda	13
2.3.1. Descarga	13
2.3.2. Instalacion	15
2.4. Apache Spark	18
2.4.1. Descarga	18
2.4.2. Instalación	19
2.4.3. Variables de entorno	20
2.4.4. Scala y Python desde consola	21
2.4.5. Ejecutando Spark-shell y pyspark desde consola	21
2.4.6. Jupyter Notebook desde consola con Pyspark	22
2.4.7. Pyspark en anaconda	23
2.4.8. Scala desde Jupyter Notebook	23
2.4.9. R desde Jupyter Notebook	25
2.5. Apache Kafka	26
2.5.1. Descarga	26
2.5.2. Instalación	27
2.5.3. Variables de entorno	27
2.5.4. Verificación de Funcionamiento	31
3. Instalación y Configuración en Ubuntu	34
3.1. Requisitos	34
3.2. Java JDK	35
3.3. Anaconda	35
3.3.1. Instalación	35
4. Apache Spark	36
4.1. Tipos de administradores de clústers	36
5. Servicios en la nube	36

5.1.	Databricks	36
5.1.1.	Registro	36
5.1.2.	Creacion de Cluster	38
5.1.3.	Creacion y carga de Notebooks	42
5.2.	Google Cloud	46
5.2.1.	Registro	46
5.2.2.	Storage	49
5.2.3.	Dataproc	54
6.	Optimización	69
6.1.	Elegir un tipo de compresión	69

1. Introducción

La finalidad de este documento sera la instalación y configuración de todas las herramientas necesarias para la programación en Apache Spark tanto en Python, como en Scala.

Como extra, tendremos una sección donde podremos ver como trabajar con Apache Spark por medio del procesamiento en la nube.

2. Instalación y Configuración en Windows 10

2.1. Requisitos

Sistema con Windows 10 de 64bits

Descompresor de archivos, nosotros utilizaremos 7-Zip

2.2. Java JDK

El Java JDK es el Java Development Kit, que traducido al español significa, Herramientas de desarrollo para Java. Aquí nos encontraremos con el compilador javac que es el encargado de convertir nuestro código fuente (.java) en bytecode (.class), el cual posteriormente sera interpretado y ejecutado en la JVM, Java Virtual Machine por sus siglas en inglés, que nuevamente en español significa, La Maquina Virtual de Java.

Puede que nos suene mas Java JRE, este es el Java Runtime Environment, que en español significa, Entorno de Ejecución de Java. En palabras del propio portal de Java es la implementación de la Máquina virtual de Java que realmente ejecuta los programas de Java, esto quiere decir que aquí encontraremos todo lo necesario para ejecutar nuestras aplicaciones escritas en Java.

Normalmente el JRE esta destinado a usuarios finales que no requieren el JDK, pues a diferencia de este, no contiene los programas necesarios para crear aplicaciones en el lenguaje Java, es así, que el JRE se puede instalar sin necesidad de instalar el JDK, pero al instalar el JDK, este siempre cuenta en su interior con el JRE.

2.2.1. Descarga

Si vamos a la web oficial de Oracle para la descarga de Java ([pagina oficial](#)) e intentamos descargarlo, nos obligara a crearnos una cuenta. Para evitar esto descargaremos la versión libre de Java, OpenJDK. Accederemos a través de la siguiente dirección:

<https://openjdk.java.net/>

Como podemos ver en las siguientes imágenes, en el segundo párrafo de la pagina principal, donde empieza con Download, haremos click en [jdk.java.net/16](https://openjdk.java.net/16) y esto nos lleva a la pagina de descargas. Una vez aquí, en la columna de la izquierda tenemos que seleccionar Java SE 15 ya que nuestra instalación no funcionará con las versiones 16 o posteriores. En nuestro caso seleccionaremos la versión de Windows/x64.

(a) Pagina principal

The main OpenJDK website features a large orange "OpenJDK" logo. On the left sidebar, there's a navigation menu with links like "Workshop", "OpenJDK FAQ", "Contributing", "Sponsoring", "Developers' Guide", "Vulnerabilities", "Mailing Lists", "IRC - Wiki", "Bylaws - Census", "Legal", "JEP Process", "Source code", "Mercurial", "GitHub", "Groups", "IDE Tooling & Support", "Internationalization", "IMX", "Members", "Networking", "Porters", "Quality", "Security", "Services", "Sound", "Swing", "Vulnerability", "Web", and "Projects". The main content area has sections for "What is this?", "Download", "Learn how to use the JDK", and "Hack on the JDK itself".

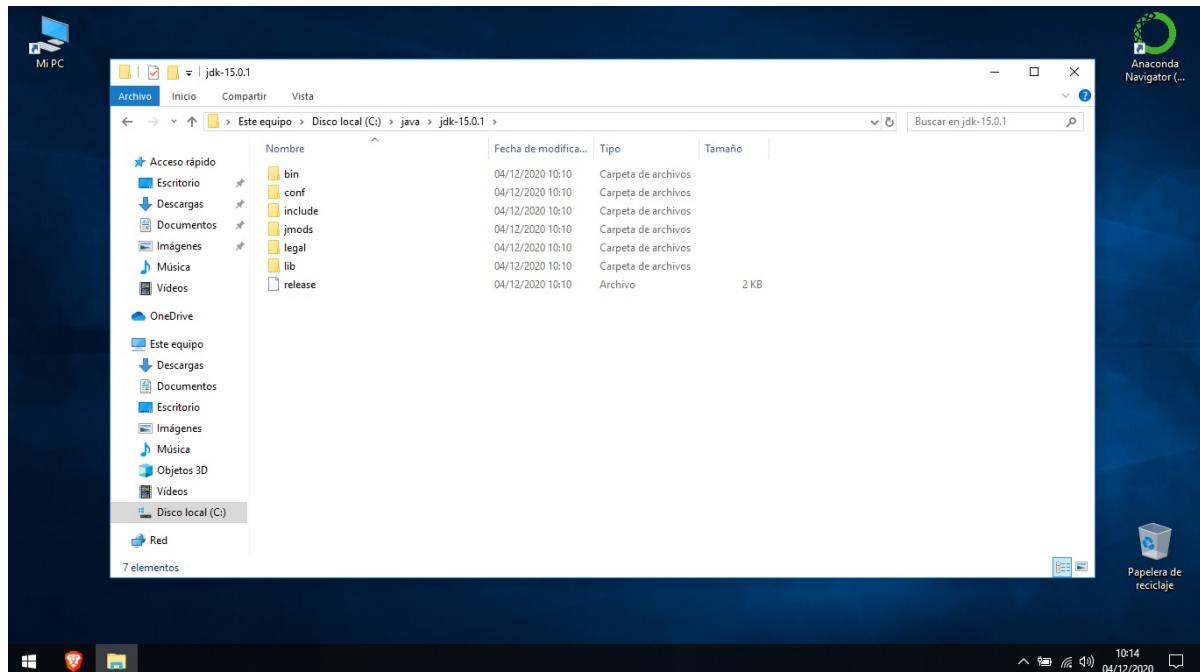
(b) Pagina de descargas

The download page for Java Platform, Standard Edition 15 Reference Implementations shows a sidebar with "Reference Implementations" for Java SE 15, Java SE 14, Java SE 13, Java SE 12, Java SE 11, Java SE 10, Java SE 9, Java SE 8, Java SE 7, and Java SE 6. The "Java SE 15" link is highlighted. Below the sidebar, there's a section titled "RI Binary (build 15+36) under the GNU General Public License version 2" with two options: "Oracle Linux 7.5 x64 Java Development Kit (sha256) 187 MB" and "Windows 10 x64 Java Development Kit (sha256) 187 MB". The "Windows 10 x64 Java Development Kit" link is also highlighted.

2.2.2. Instalacion

Una vez tengamos descargado nuestro OpenJDK, en nuestra carpeta de Descargas veremos que se trata de un archivo zip. Lo que primero que debemos hacer para instalarlo sera crear una carpeta en la raiz de nuestro disco duro (C:/) que se llame 'java'. Lo siguiente sera descomprimir el archivo descargado dentro de esa carpeta de manera que la ruta al contenido de Java JDK quedara en *C:/java/jdk-15.0.1*

En la siguiente imagen podemos ver como debería quedarnos:



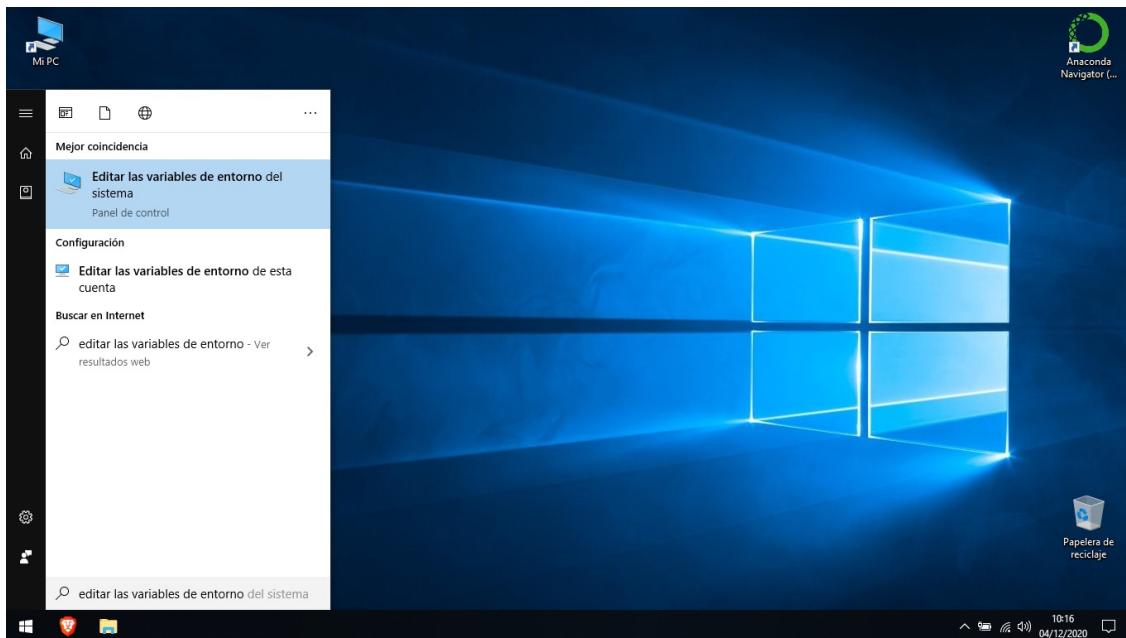
2.2.3. Variables de entorno

Seguramente muchos nunca hayan escuchado este termino. Para que se entienda de una manera sencilla, una variable de entorno no es mas que, una palabra, o un texto facilmente recordable que nos permitirá acceder a rutas mas complejas de forma mas sencilla. En el caso de Java por ejemeplo, sera mucho mas sencillo recordar 'java' que la ruta a la carpeta donde lo hemos instalado (*C:/java/jdk-15.0.1*). Ademas, las variables de entorno facilitan al resto de programas con dependencias externas conocer la dirección donde se encuentran estas.

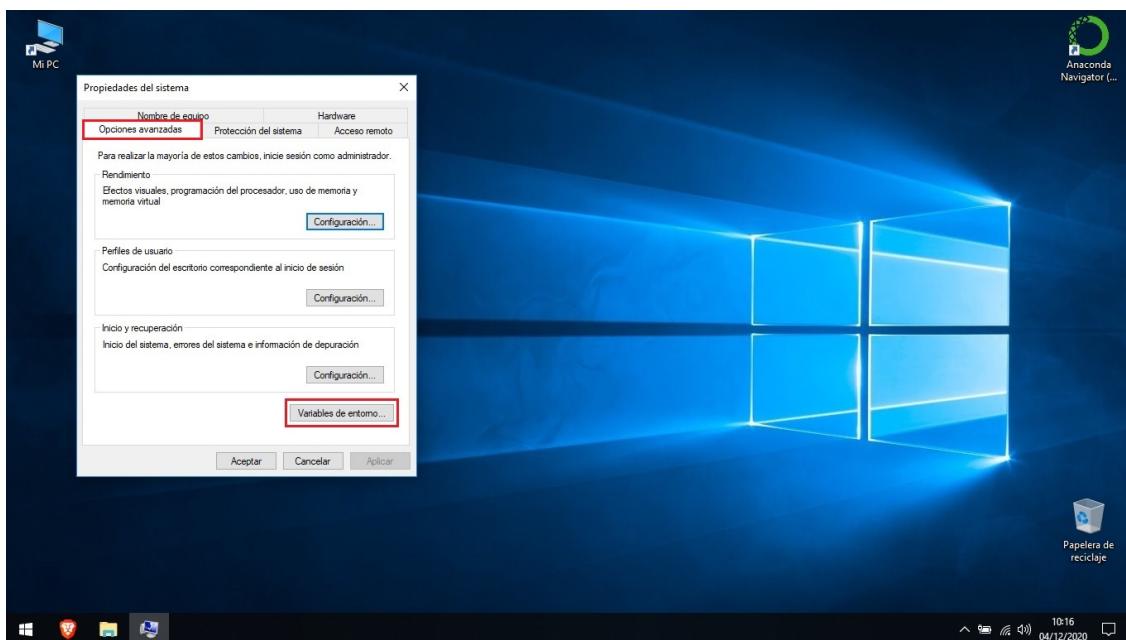
En este caso, deberemos crear la variable de entorno **JAVA_HOME** y actualizar la variable de entorno **PATH**. La primera se utilizará para que Java sepa dónde se encuentra la instalación de Java JDK y la segunda es para poder ejecutar los comandos de Java (como javac, java, etc) desde cualquier lugar, como desde la consola del sistema.

JAVA_HOME

Hacemos click en inicio y escribiremos 'Editar las variables de entorno del sistema' como en la siguiente imagen:

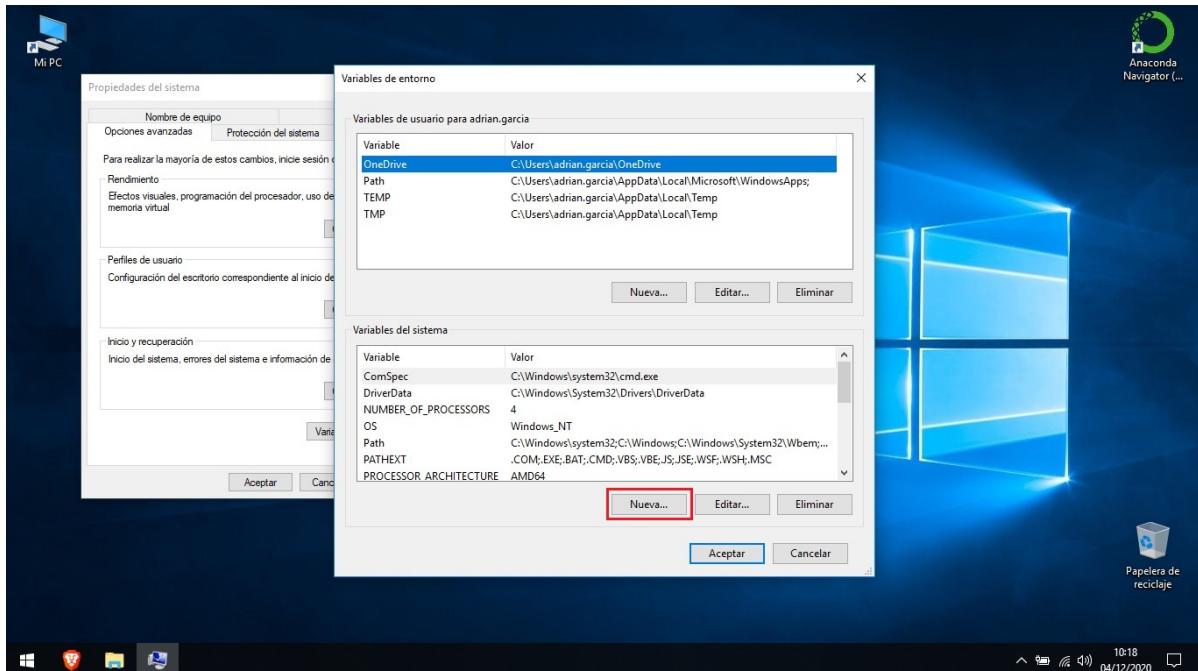


Al abrir el editor veremos que se abre la ventana de Propiedades del sistema. Hacemos click en la pestaña de 'Opciones avanzadas' y abajo hacemos click en 'Variables de entorno' como en la siguiente imagen:

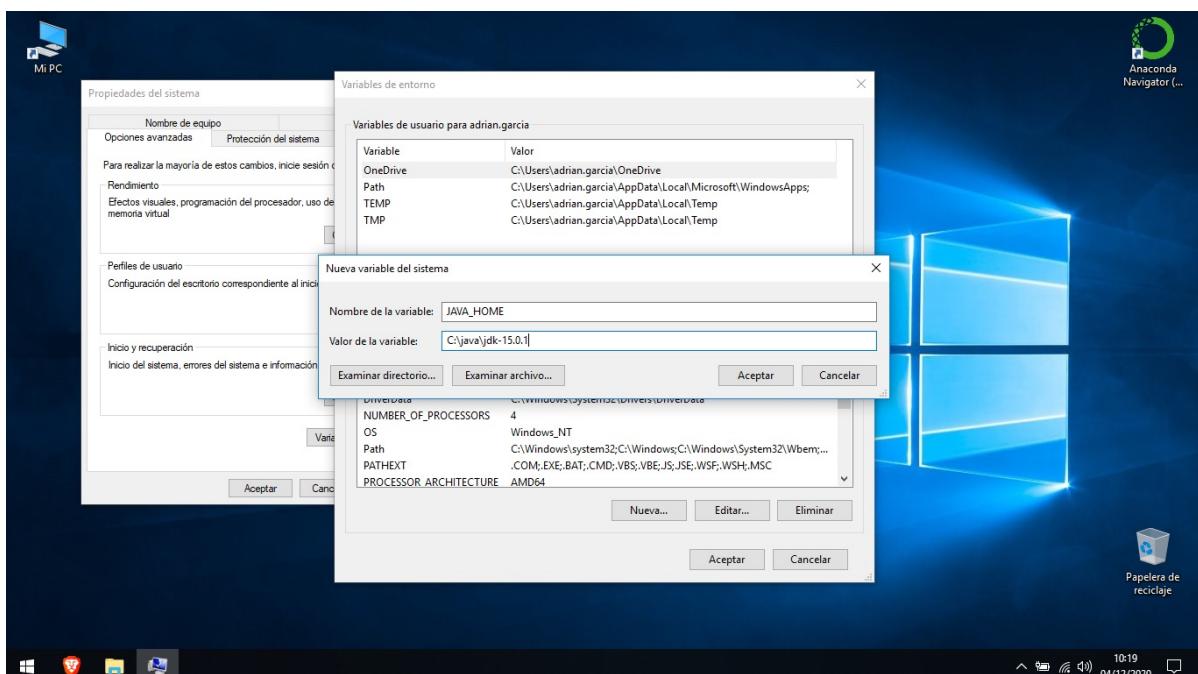


En la ventana que se nos ha abierto veremos dos recuadros. En uno pondrá *Variables de usuario* y en el de debajo *Variables del sistema*. Lo mas recomendable es crear las variables de entorno para el sistema, con el fin de que cualquier usuario tenga acceso a la ejecución de java.

Entonces en el grupo de Variables del sistema hacemos click en el botón 'Nueva'.

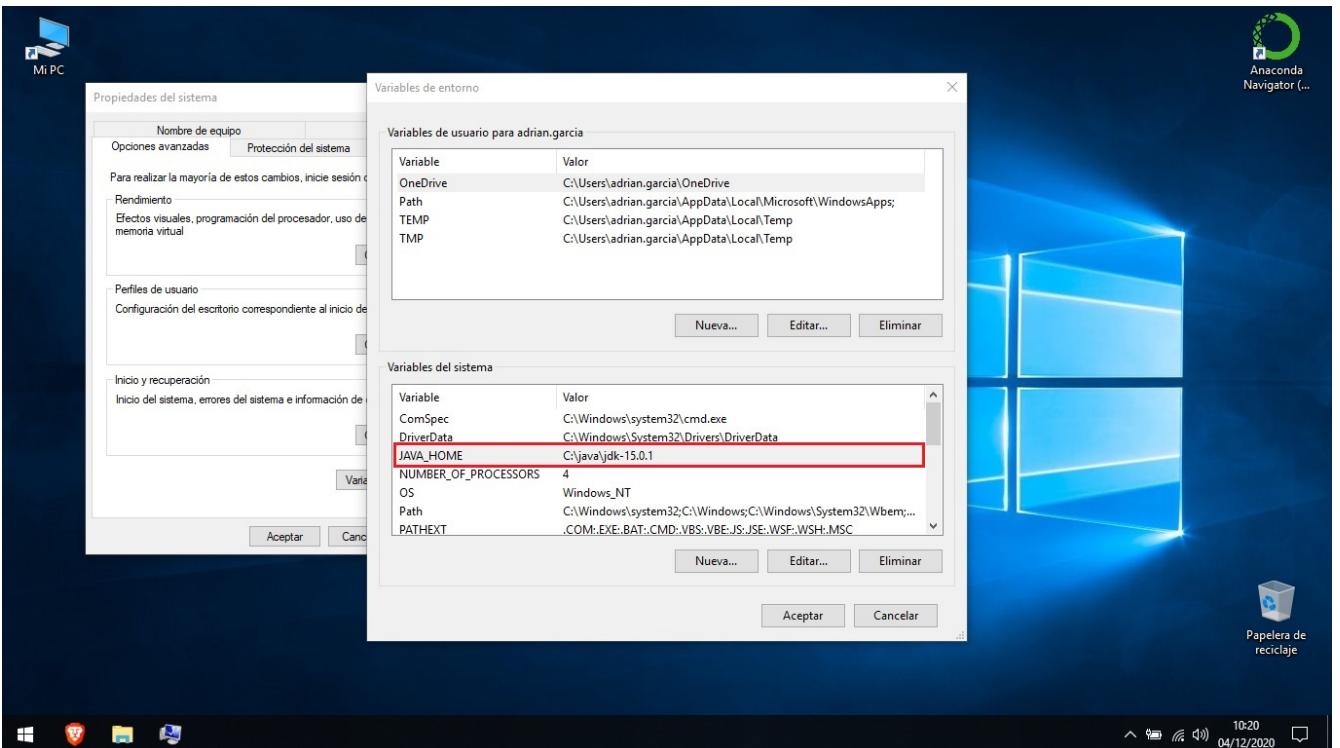


En el nombre escribiremos '`JAVA_HOME`', mientras que en valor la variable escribiremos la ruta donde se instaló el Java JDK, en nuestro caso será `C:/java/jdk-11.0.2`



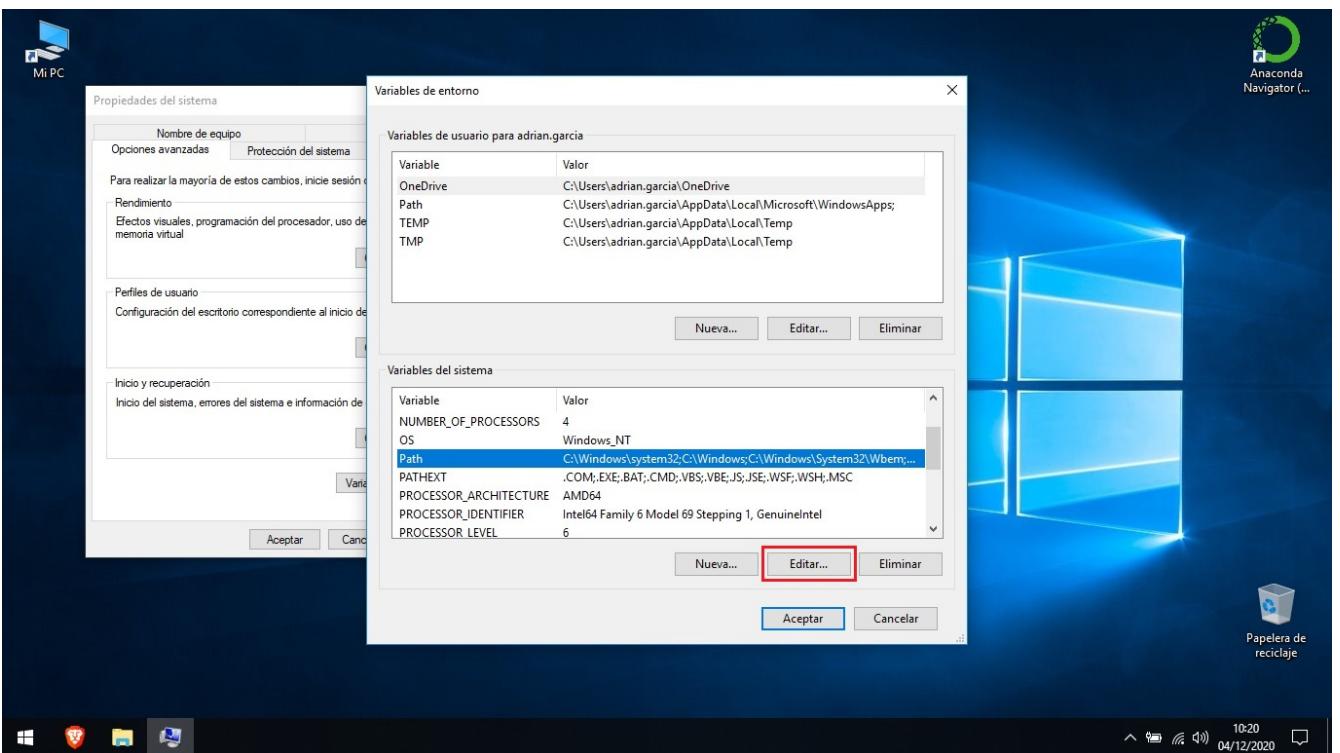
Hacemos clic en aceptar.

Debería quedarnos de la siguiente manera:

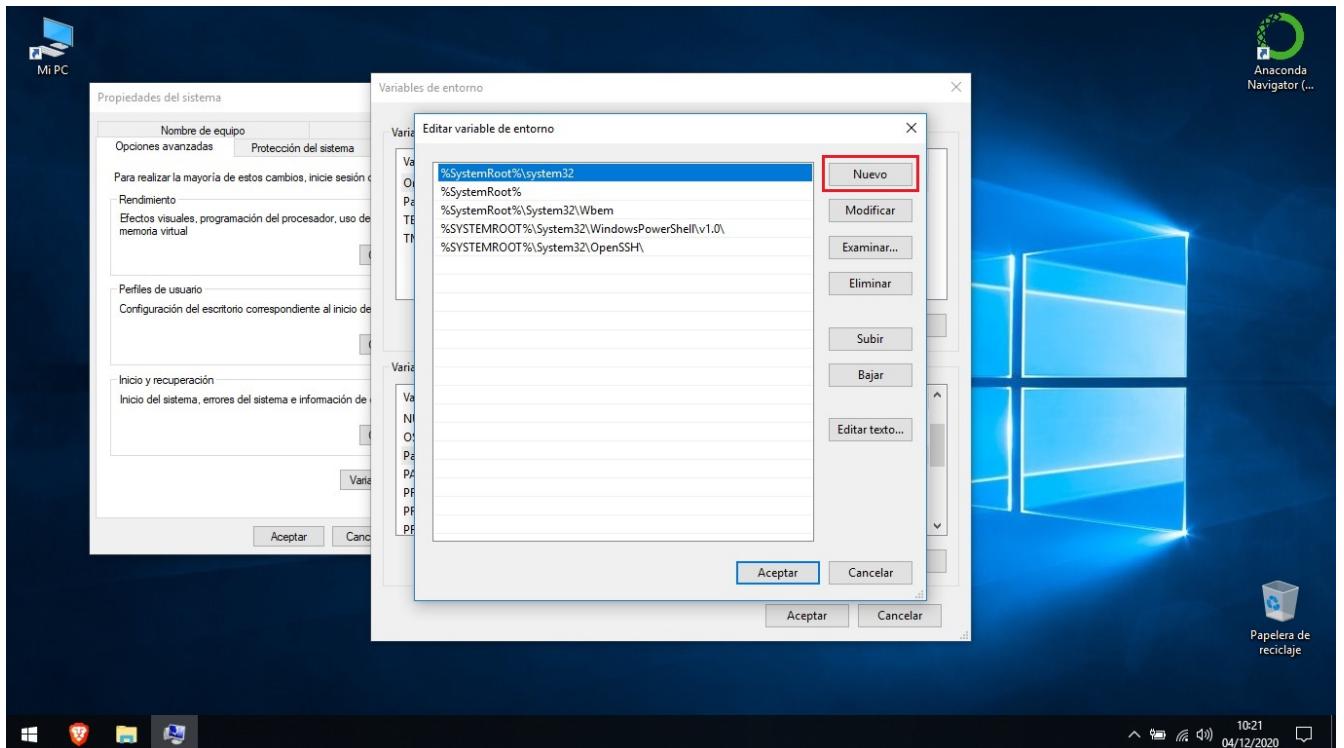


• PATH

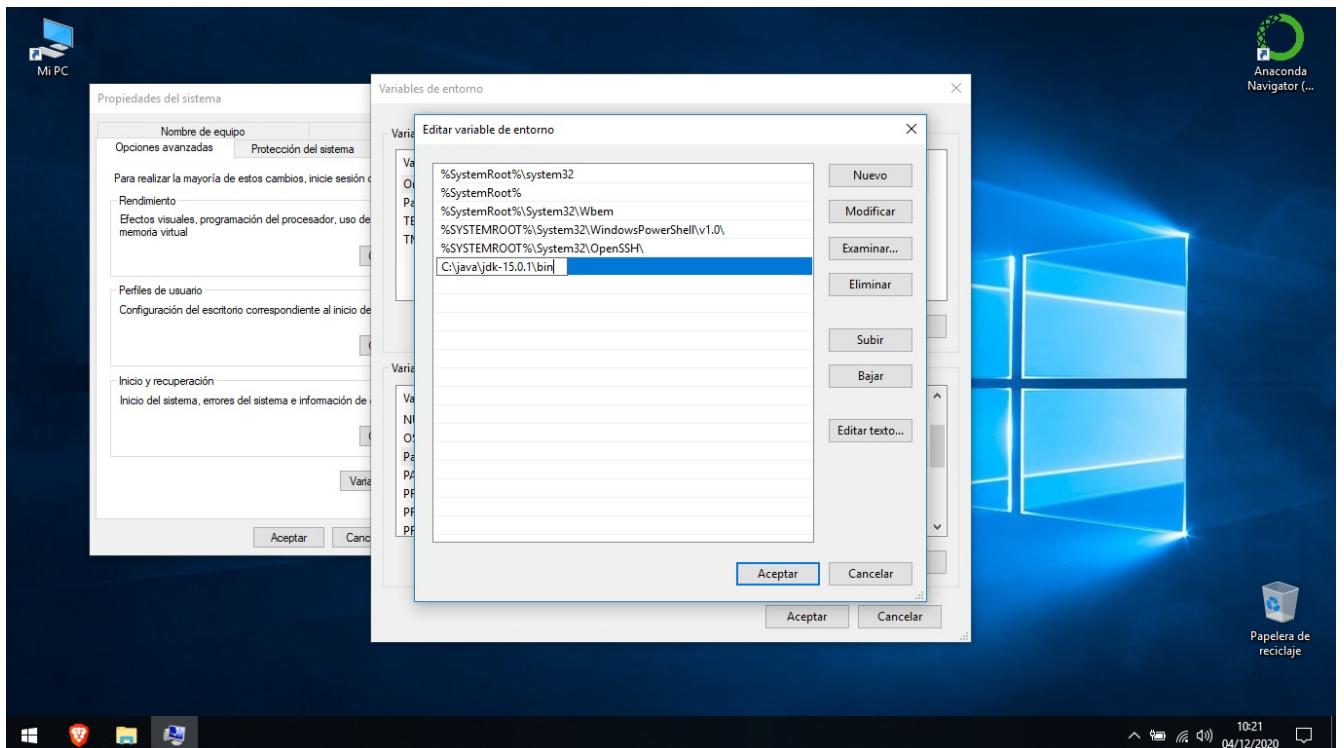
Si por alguna razón no tienes abierta la ventana de variables de entorno, repite los pasos anteriores. En este caso vamos a editar la variable Path que ya existe dentro de variables del sistema. La seleccionamos y le damos a editar:



Podrás ver todos los valores que tiene por defecto la variable Path. No los modifiques o elimines, solo haz click en 'Nuevo'

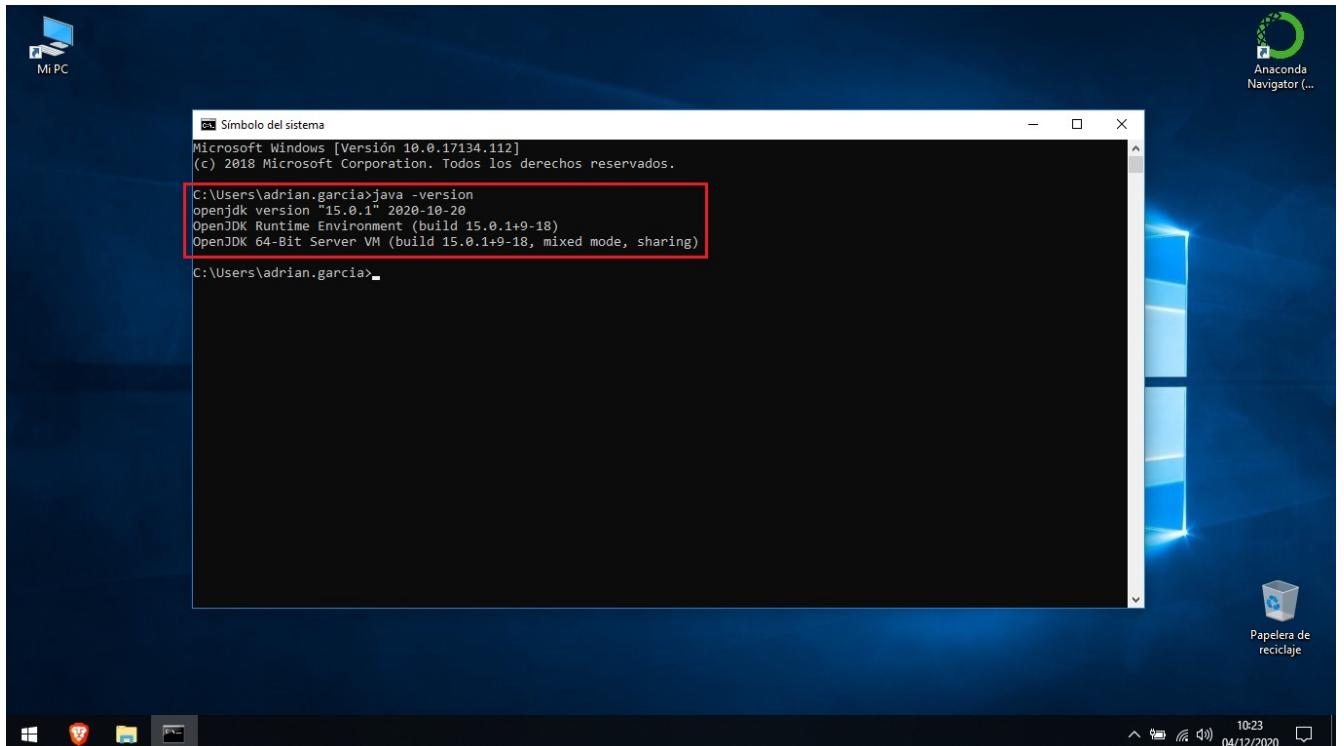


Escribiremos la ruta a la instalación de java, pero en este caso direccionandolo a la carpeta bin. Si has seguido todos los pasos hasta aquí, la ruta sera *C:/java/jdk-11.0.2/bin*



Aceptamos en la ventana de Path y volvemos a aceptar en la ventana de variables de entorno y en propiedades del sistema.

Ahora solo nos quedara comprobar que java esta correctamente instalado y las variables de entorno han sido correctamente configuradas. Para ello, como cuando abrimos el editor de variables de entorno, hacemos click en inicio y ahora escribiremos *cmd* abriendo el programa **Símbolo del sistema**. En la ventana de comandos escribiremos **java -version** y deberíamos obtener el siguiente resultado:



Como hemos podido comprobar, al introducir el termino *java*, gracias a las variables de entorno, windows sabe la ruta a la que se tiene que dirigir. Y con el argumento *-version* ejecuta la comprobación de la version instalada.

Con esto ya podemos decir que tenemos Java instalado y configurado en nuestro sistema.

2.3. Anaconda

Anaconda es una solución flexible de código abierto que proporciona las utilidades para crear, distribuir, instalar, actualizar y administrar software de manera multiplataforma. Además nos facilita la gestión de múltiples entornos de datos que se pueden mantener y ejecutar por separado sin interferencias entre sí.

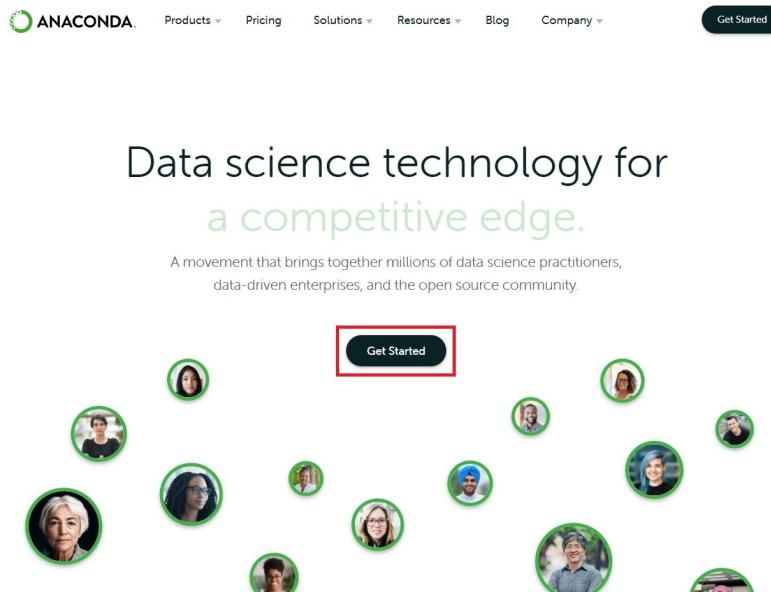
Nos va a servir para el procesamiento de datos a gran escala, el análisis predictivo y la informática científica, que tiene como objetivo simplificar la gestión de empaquetado y distribución. Esta es quizás la Suite más completa para la Ciencia de datos con Python y que nos brinda una gran cantidad de funcionalidades que nos van a permitir desarrollar aplicaciones de una manera más eficiente, rápida y sencilla.

2.3.1. Descarga

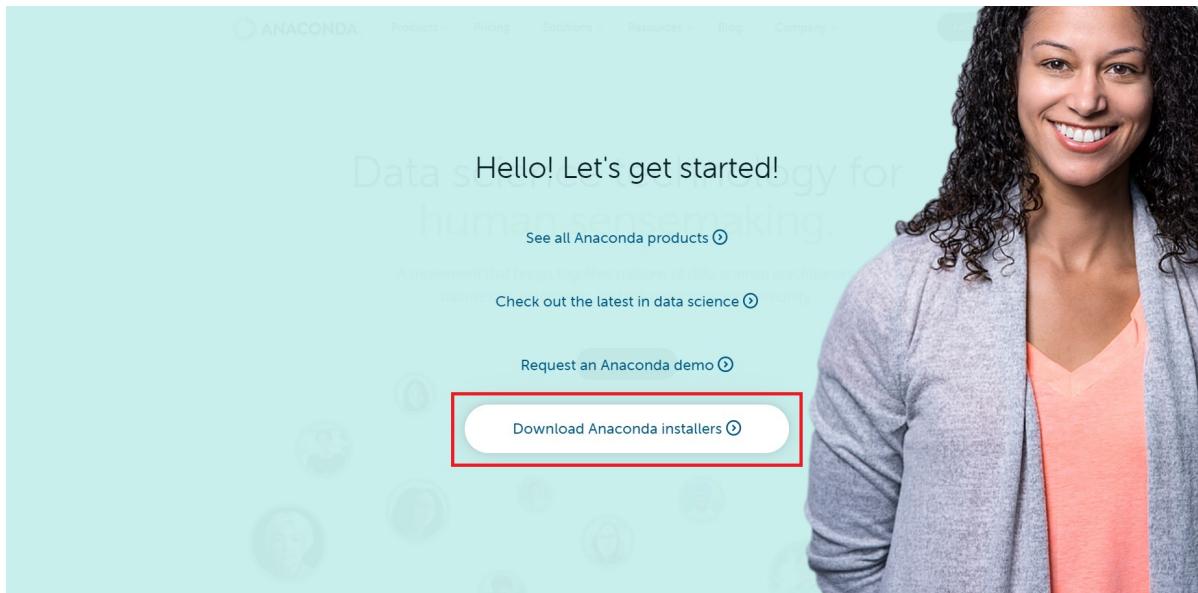
El primero paso será ir a la web oficial de Anaconda:

<https://www.anaconda.com/>

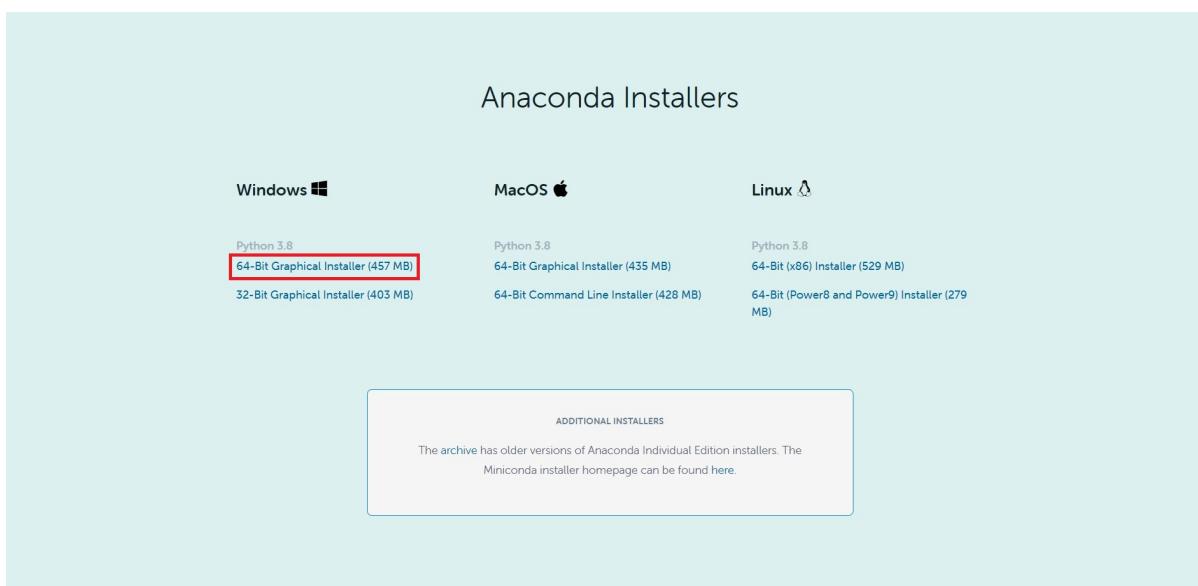
En la web principal haremos click en *Get Started*:



En la ventana emergente seleccionamos *Download Anaconda Installers*

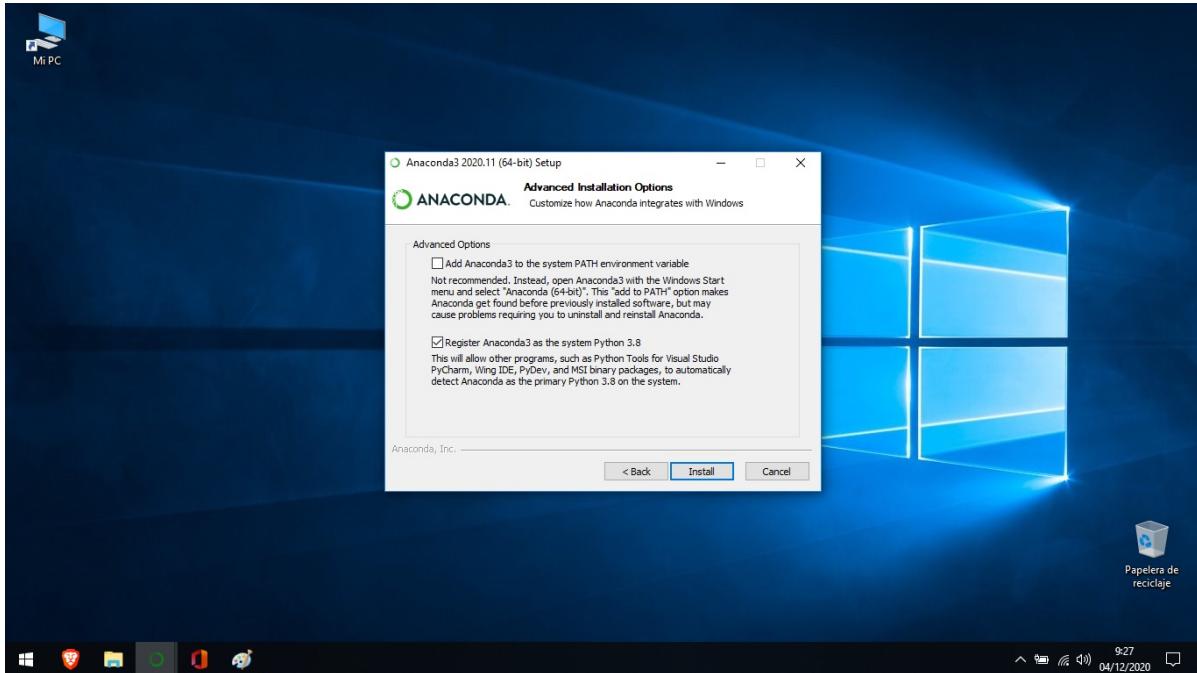


Y seleccionamos la versión compatible con nuestro sistema, en este caso Windows 10 de 64 bits:

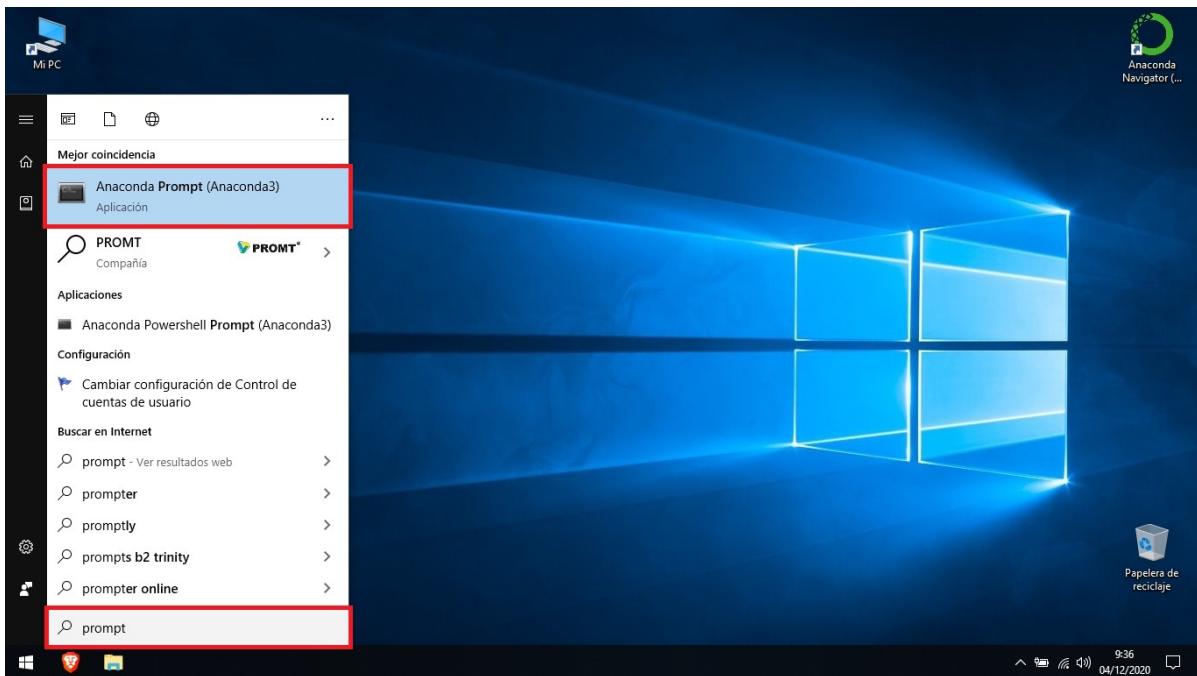


2.3.2. Instalacion

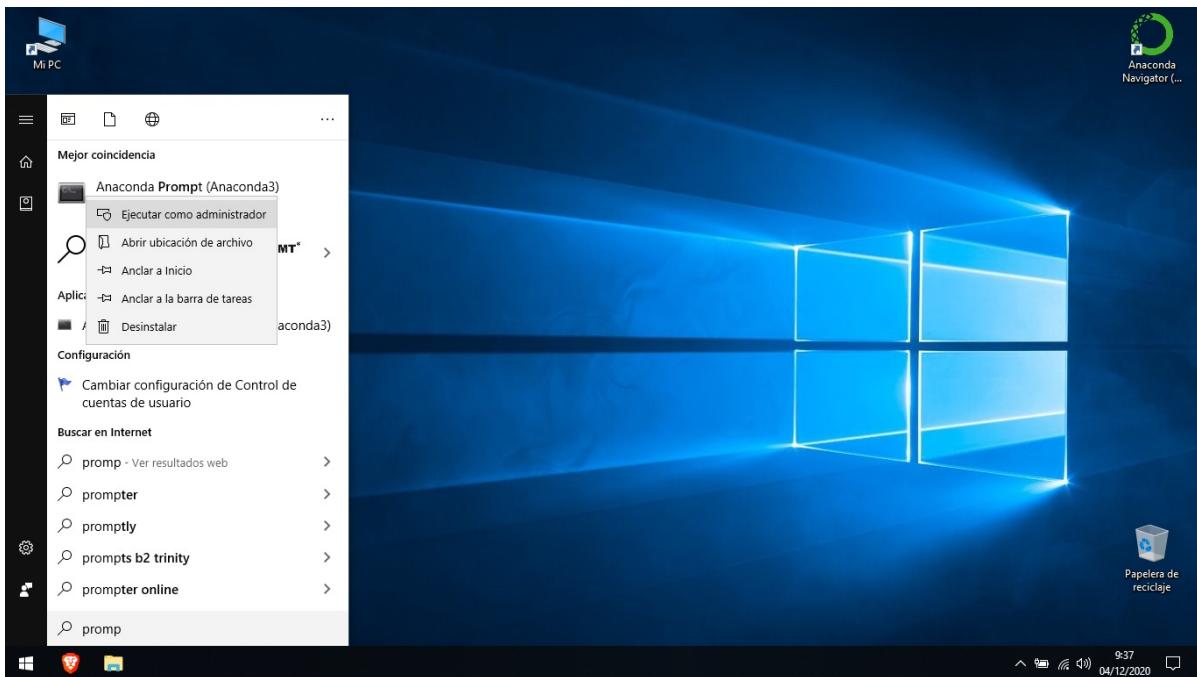
Vamos a nuestra carpeta de descargas e iniciamos el instalador. Durante la instalación dejaremos todos los valores por defecto.



Anaconda durante su instalación también nos instalará Python. Para comprobar que ha sido correctamente instalado, vamos a inicio de windows y escribimos *Prompt* y ejecutamos *Anaconda Prompt (Anaconda 3)* como se muestra en la siguiente imagen:

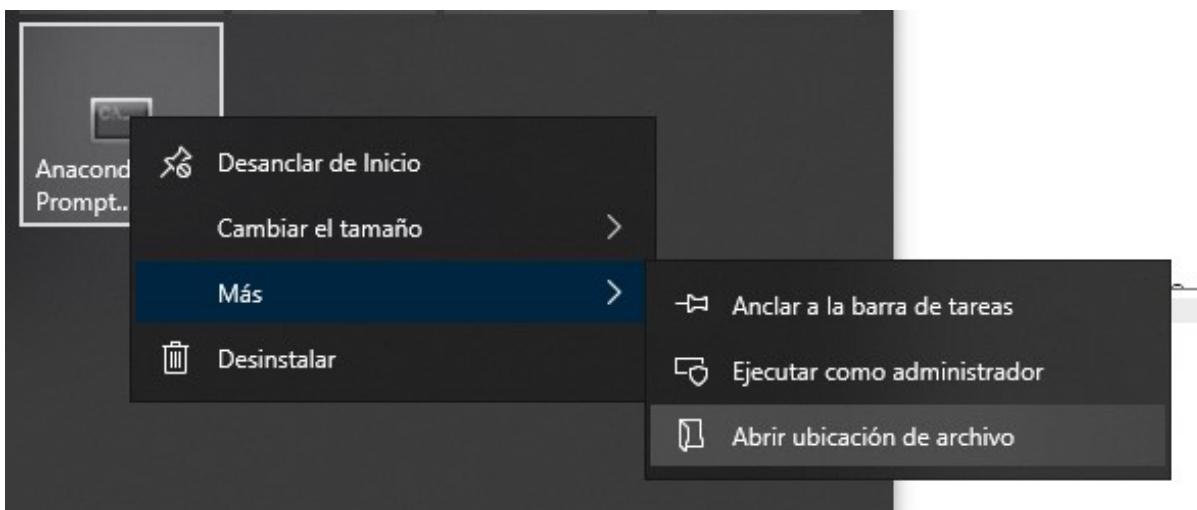


Haremos click derecho y ejecutaremos como administrador. Esto hará que salga una pantalla emergente de windows que nos preguntara si estamos seguros de que queremos permitir que este programa haga modificaciones en el equipo, aceptamos.

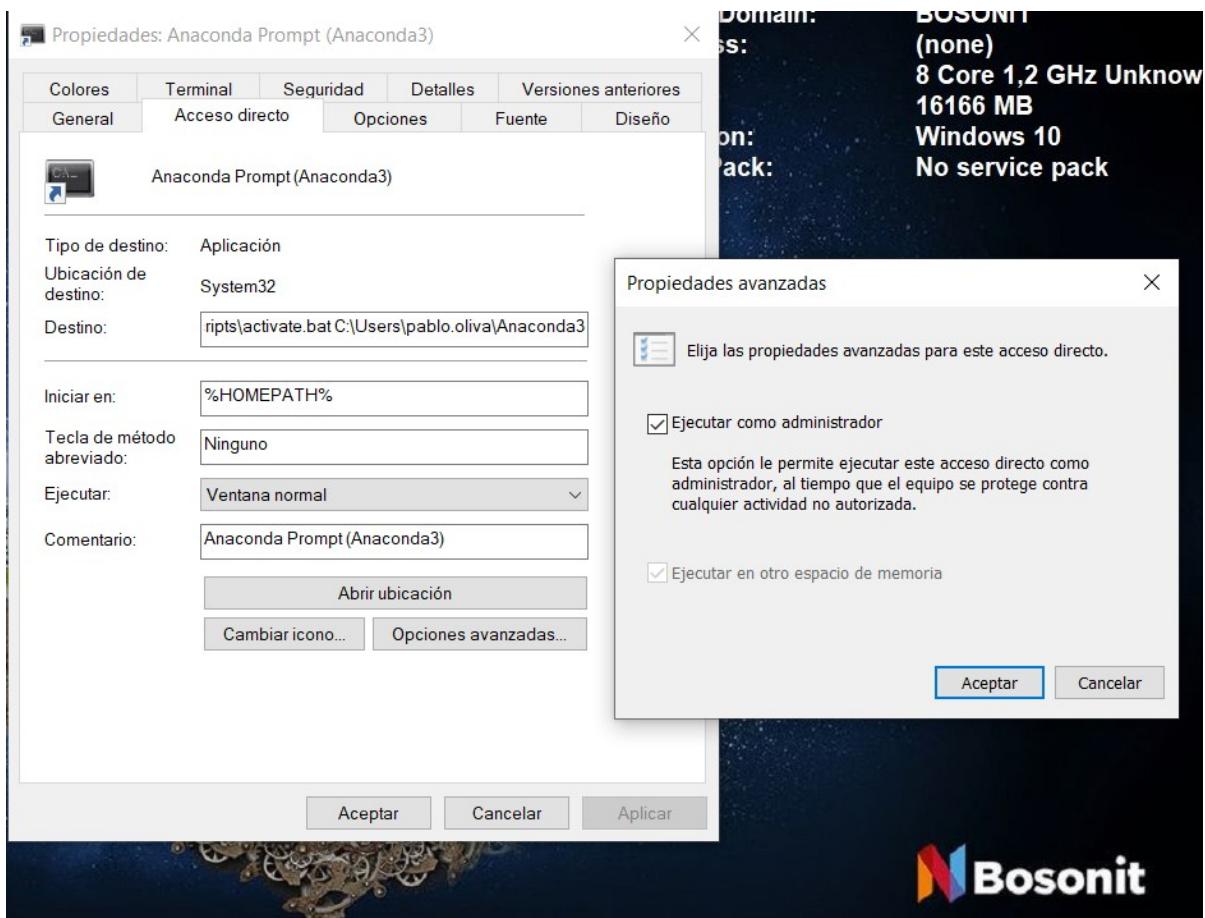


NOTA: Si no ejecutamos la consola de Anaconda como administrador, al intentar instalar cualquier librería o paquete extra de python, nos dará error sin especificarnos la razón. Por eso es buena práctica acostumbrarse a hacerlo siempre como administrador.

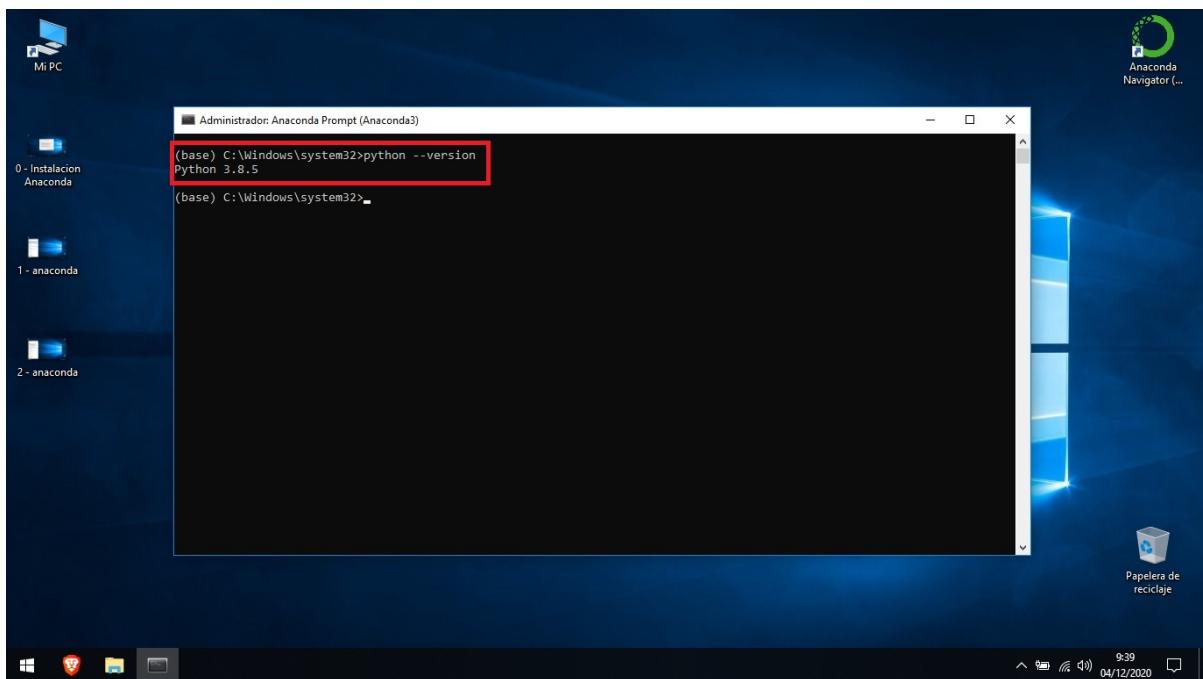
Podemos configurar que por defecto la consola de Anaconda se ejecute como administrador haciendo click derecho sobre el ícono de *Anaconda Promt* y pulsando en Más y después en Abrir ubicación de archivo.



En esta carpeta hacemos click derecho sobre el ícono de acceso directo de *Anaconda Prompt (Anaconda3)* y pulsamos en Propiedades. En la ventana emergente, en la pestaña Acceso directo pulsamos en Opciones avanzadas y por último dejamos seleccionada la opción Ejecutar como administrador.



Al ejecutar esta consola veremos que se trata de una similar a la de windows, aunque con la diferencia de que se trata de la propia de Anaconda. Ahora, para asegurarnos de que la instalación de Python ha sido correcta, escribiremos **python --version** y si todo ha ido bien obtendremos un mensaje con la versión de Python instalado como en la siguiente imagen:



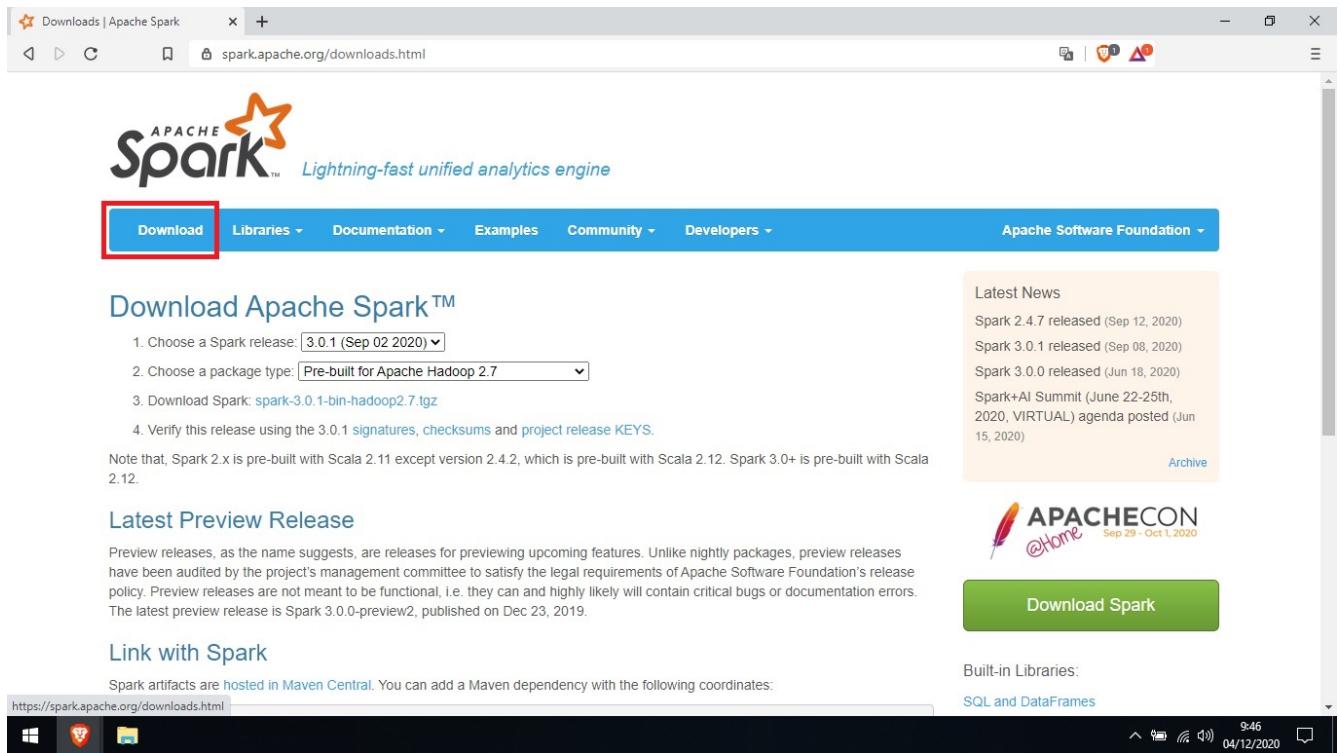
2.4. Apache Spark

2.4.1. Descarga

Para instalar Spark vamos a ir a la pagina <https://spark.apache.org/>

Le damos a “Download Spark”

Y seleccionamos la version que aparece en la foto:



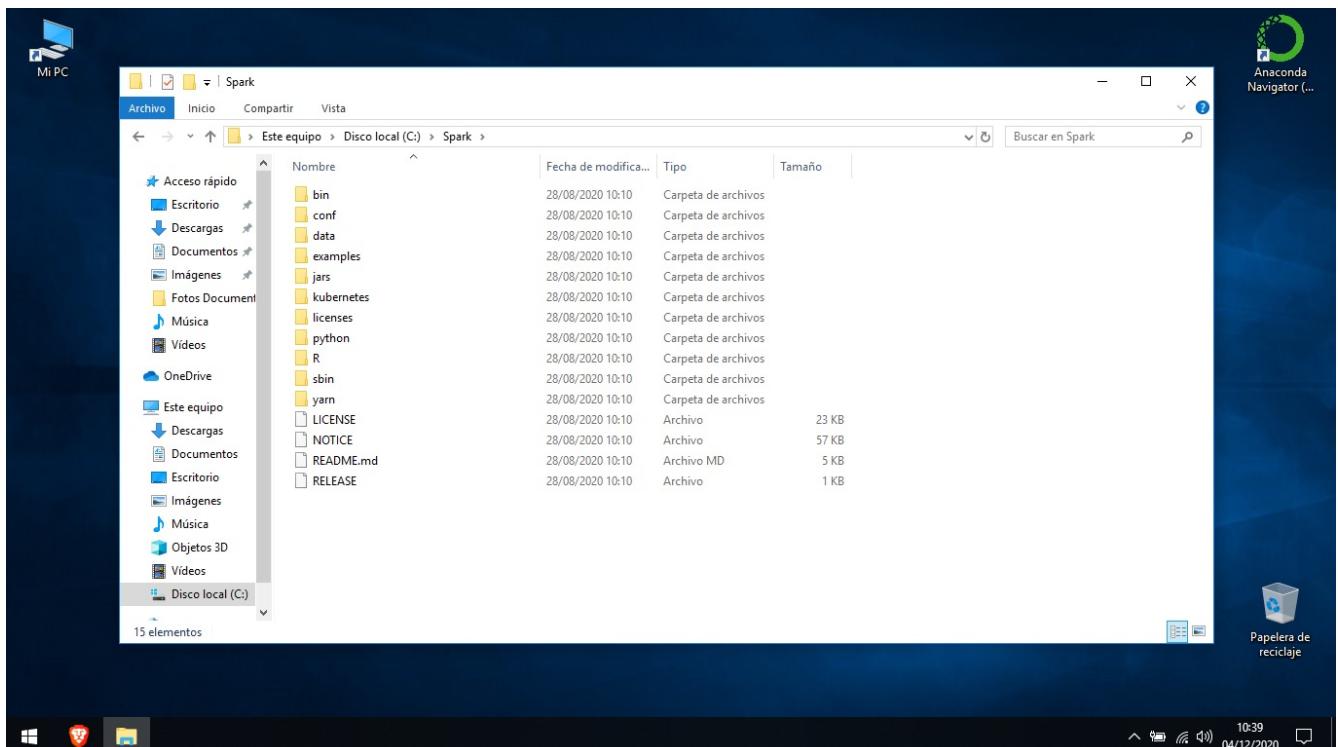
Nos descargara un archivo comprimido con extension “.tgz”

Si ya disponemos de un descomprimidor instalado lo utilizamos, si no tendremos que descargar uno. En mi caso utilice 7-Zip por ser software libre. La pagina web es <https://www.7-zip.org/>

2.4.2. Instalación

Ahora igual que tuvimos que hacer con Java JDK, tendremos que crear en la raíz del sistema una carpeta Spark de forma que quede: C:/Spark

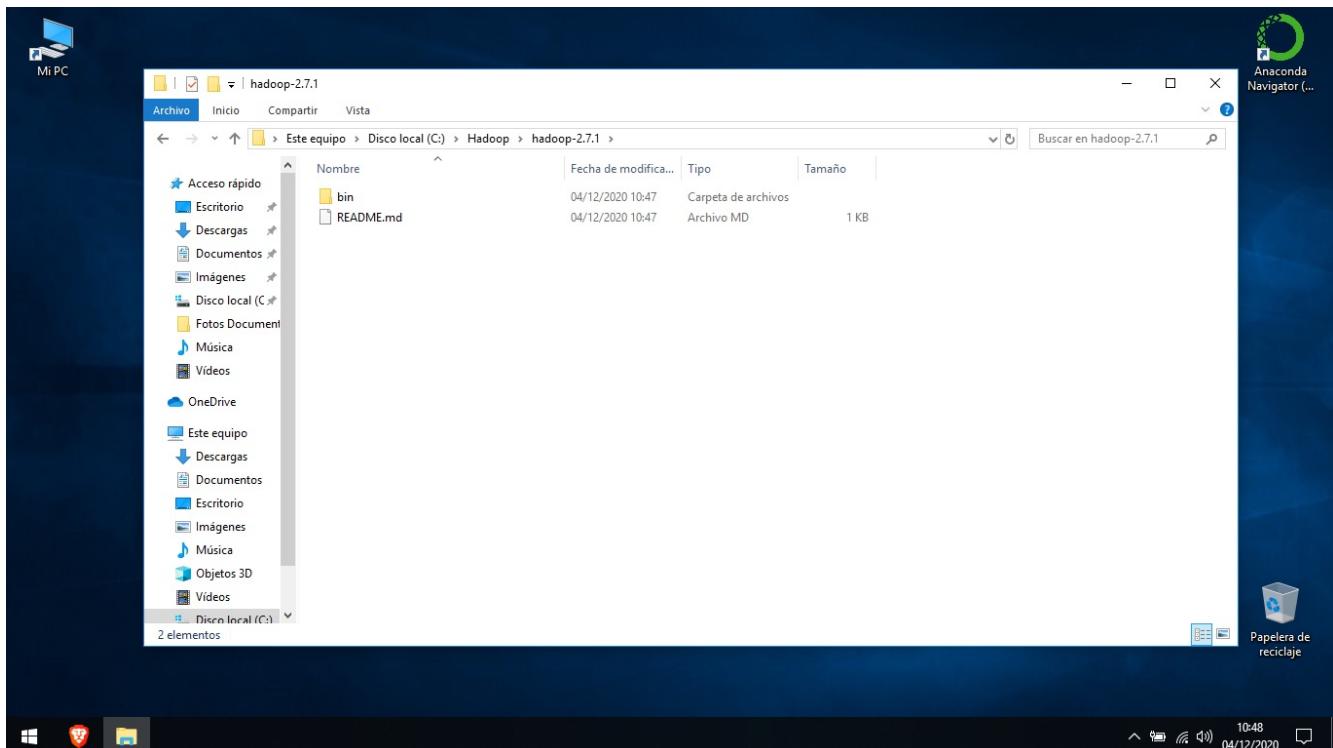
En esta carpeta sera donde descomprimiremos el archivo de Apache Spark recién descargado. Quedara de la siguiente forma:



Ahora algo que también necesitaremos sera Winutils. Antes de que te preguntes qué es eso de Winutils, déjame decirte que son un conjunto de herramientas necesarias para que la instalación de Hadoop pueda funcionar en Windows.

Si recuerdas, en el momento de la descarga de Apache Spark indicamos que queríamos el paquete con Apache Hadoop 2.7 y una de las condiciones que existen para que Hadoop funcione en ordenadores con Windows es la presencia de Winutils en el directorio bin de su instalación. Esto se indica en la [documentación oficial de Apache Hadoop](#).

Una vez descargado el [repositorio de GitHub](#) buscamos la carpeta con la versión de las winutils para nuestra versión de Hadoop y la copiaremos en una carpeta a la que llamarémos hadoop-2.7.1 como hicimos con Spark:



2.4.3. Variables de entorno

En concreto vamos a crear cuatro variables de entorno y modificar la variable Path.

- SPARK_HOME: Ruta al directorio donde hemos descomprimido el paquete de Apache Spark.
- HADOOP_HOME: Apunta al directorio donde hemos copiado la carpeta con el archivo Winutils.
- JAVA_HOME: Es el directorio donde se ha instalado el JDK de Java
- PYTHON_HOME: Es el directorio donde se ha instalado Anaconda3 y se encuentra el ejecutable python.exe. En nuestro caso la ruta ha sido *C:/Users/usuario/Anaconda3/python.exe*
- PATH: Aquí añadiremos dos nuevas rutas. El directorio bin de la carpeta de Apache Spark y el directorio bin de la carpeta JDK de Java.

2.4.4. Scala y Python desde consola

2.4.5. Ejecutando Spark-shell y pyspark desde consola



Mi PC



Anaconda
Navigator ...

Administrador: Anaconda Prompt (Anaconda3) - pyspark

```
(base) C:\Windows\system32>pyspark
Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.unsafe.Platform (file:/C:/Spark/jars/spark-unsafe_2.12-3.0.1.jar)
to constructor java.nio.DirectByteBuffer(long,int)
WARNING: Please consider reporting this to the maintainers of org.apache.spark.unsafe.Platform
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/12/04 11:06:12 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to

    / \ / \ / \ / \ / \
   / \ \ / \ / \ / \ / \ / \
  / \ / \ / \ / \ / \ / \ / \
 / \ / \ / \ / \ / \ / \ / \ / \
version 3.0.1

Using Python version 3.8.5 (default, Sep 3 2020 21:29:08)
SparkSession available as 'spark'.
>>>
>>> -
```



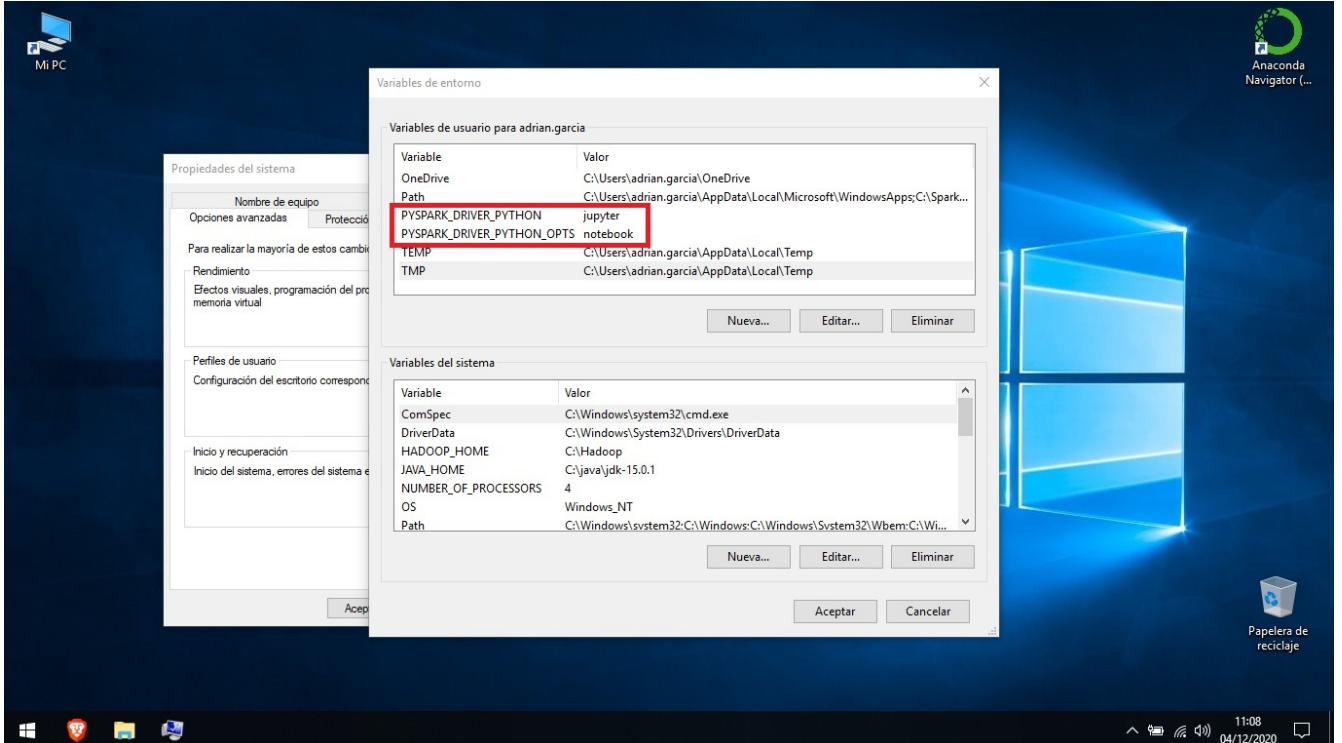
Papelera de reciclaje



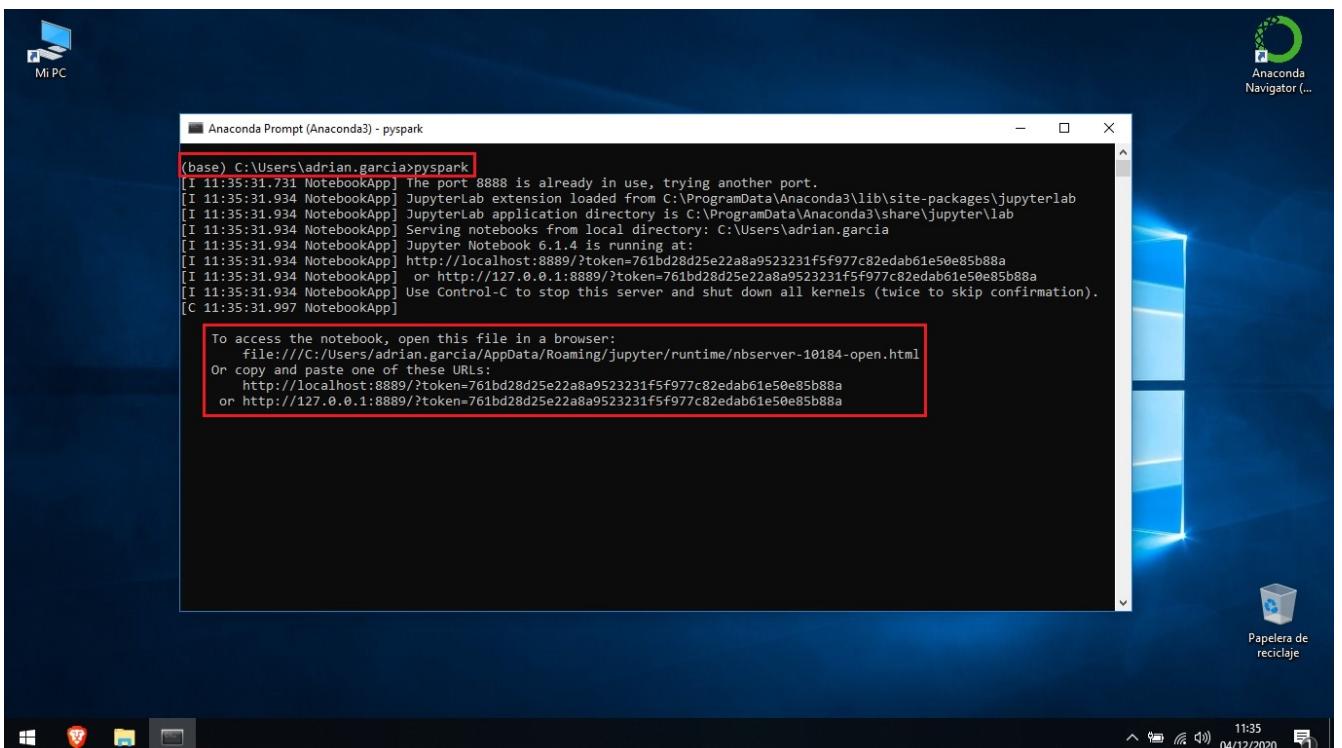
Ejecutar en la consola de anaconda pip install pyspark

2.4.6. Jupyter Notebook desde consola con Pyspark

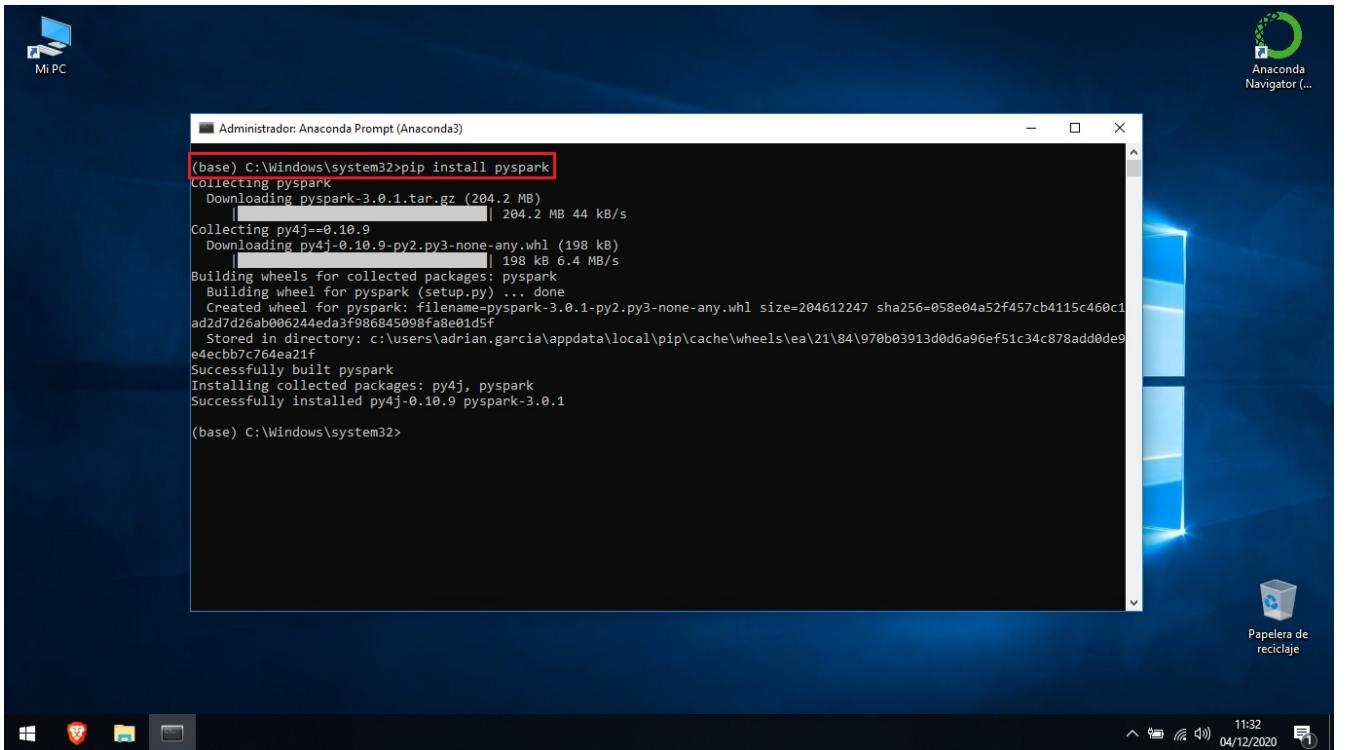
Añadimos las variables de entorno de jupyter y notebook



Hacemos pyspark en la consola de anaconda. Nos abrirá un notebook y en la consola veremos:



2.4.7. Pyspark en anaconda



2.4.8. Scala desde Jupyter Notebook

Abrimos la consola de anaconda

Ejecutamos el siguiente código que instalará el intérprete de Scala:

```
pip install spylon-kernel
```

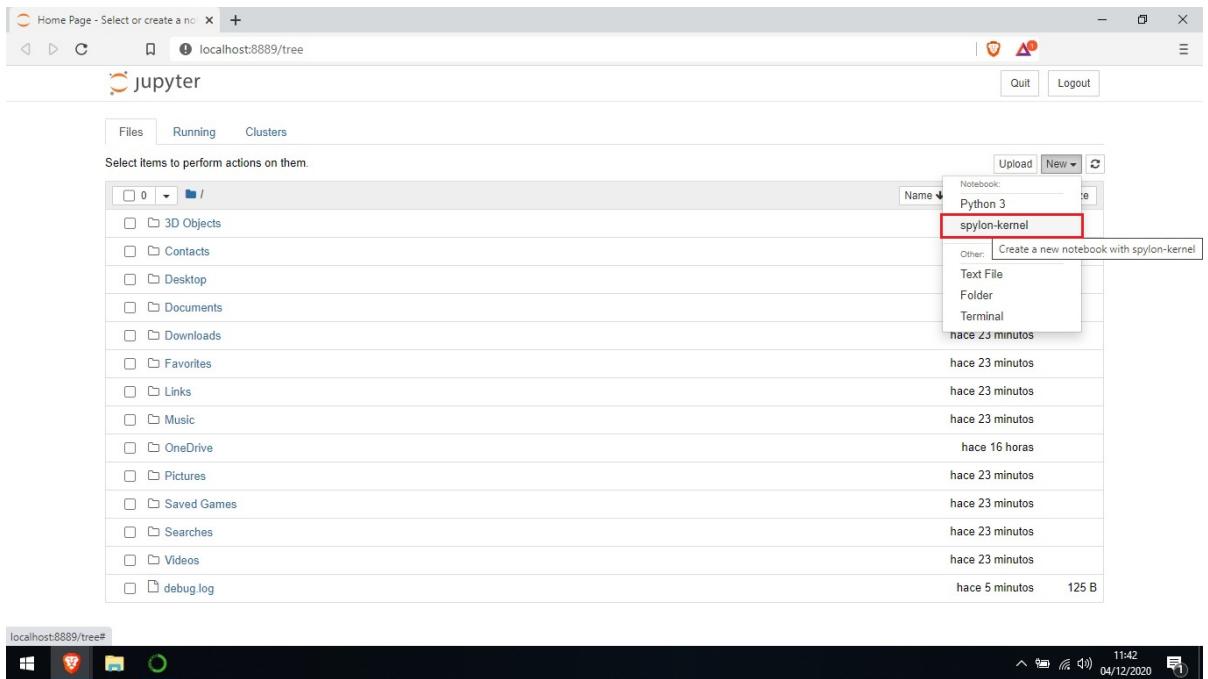
Añadimos que nos permita seleccionar el kernel de scala desde el notebook. Para ello ejecutamos

```
python -m spylon\_kernel install
```

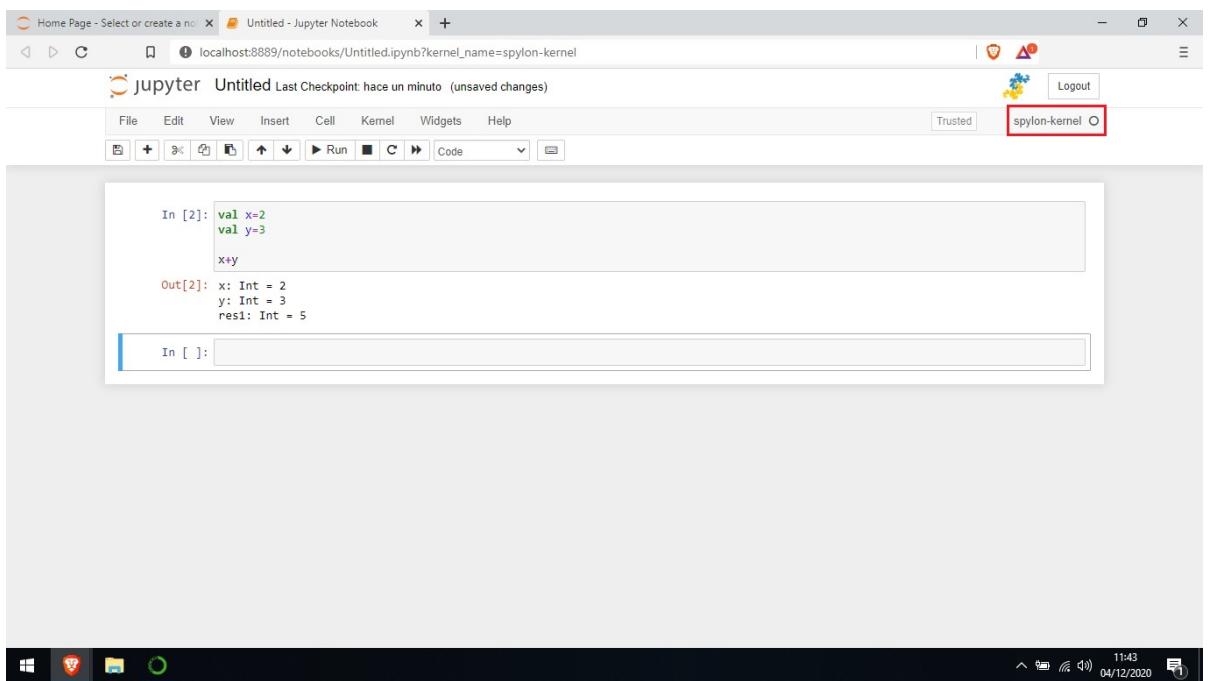
En CentOS:

```
python -m spylon_kernel install --user
```

Ahora si abrimos un notebook desde anaconda veremos como se indica en la foto que nos da la opción de crear un notebook de Scala:



Y podemos verlo en funcionamiento:



NOTA: Por comodidad, podemos definir cuál el directorio por defecto que utilizará Jupyter Notebook cuando lo abramos. En esta carpeta es donde se guardarán y leerán los archivos que utilicemos salvo que indiquemos otra ruta. Para ello, nos dirigimos al directorio *C:/Users/usuario/.jupyter* y abrimos el fichero *jupyter_notebook_config.py* con el bloc de notas.

Una vez en el bloc de notas buscamos la siguiente línea:

```
c.NotebookApp.notebook_dir = ''
```

Por defecto viene comentada, por lo que la descomentamos quitando la almohadilla del principio de la línea. Por último, escribimos el directorio que queremos tener por defecto y guardamos el archivo. Por ejemplo:

```
c.NotebookApp.notebook_dir = 'C:/Users/usuario/Documents/SparkProjectsFolder'
```

La próxima vez que abramos Jupyter ya tendremos por defecto la carpeta indicada.

2.4.9. R desde Jupyter Notebook

Podemos añadir el lenguaje R a nuestro Jupyter Notebook ejecutando el siguiente comando en Anaconda Prompt:

```
conda install -c r r-irkernel
```

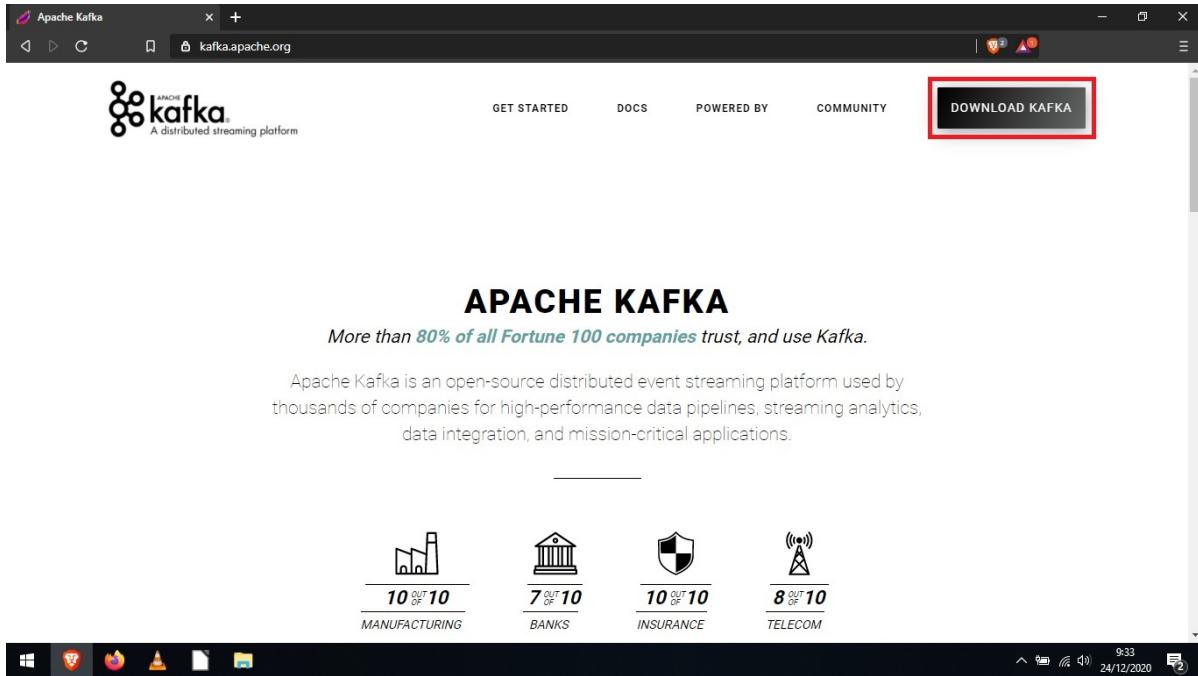
2.5. Apache Kafka

2.5.1. Descarga

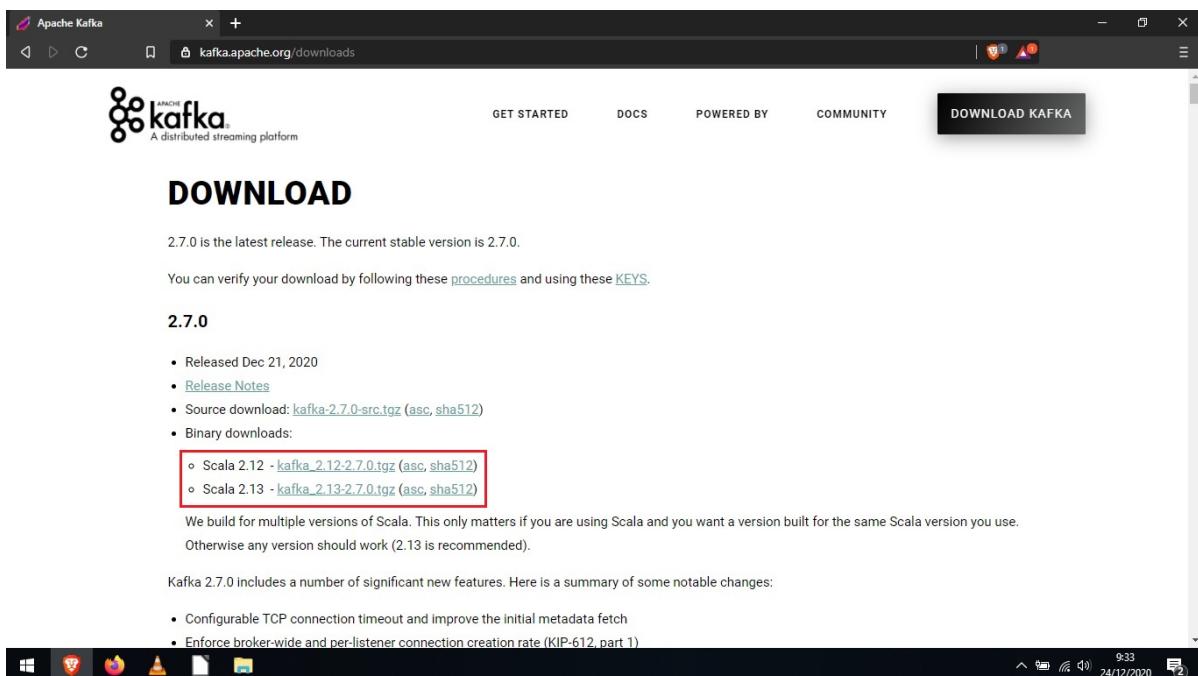
Para instalar Apache Kafka vamos a ir a su web oficial:

<https://kafka.apache.org/>

Le damos a “DOWNLOAD KAFA” en la parte superior derecha como se puede ver en la siguiente imagen:



Y descargamos el archivo tgz haciendo click en el nombre del archivo:

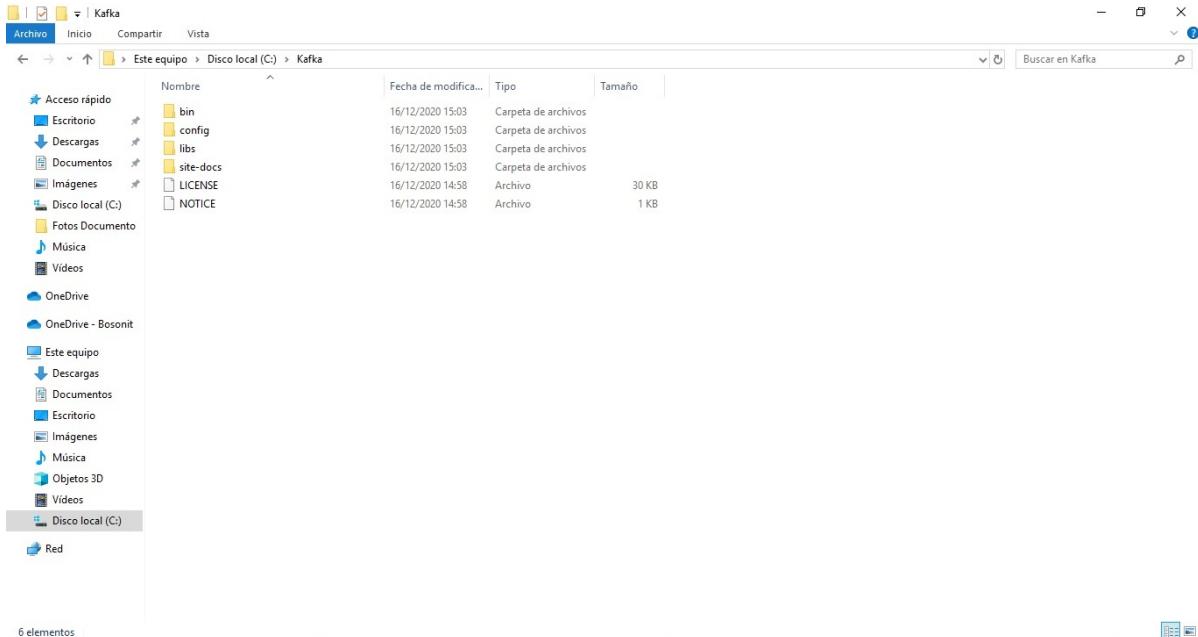


2.5.2. Instalación

De la misma forma que con Java o Spark, tendremos que crear en la raíz del sistema una carpeta Kafka de forma que quede: C:/Kafka

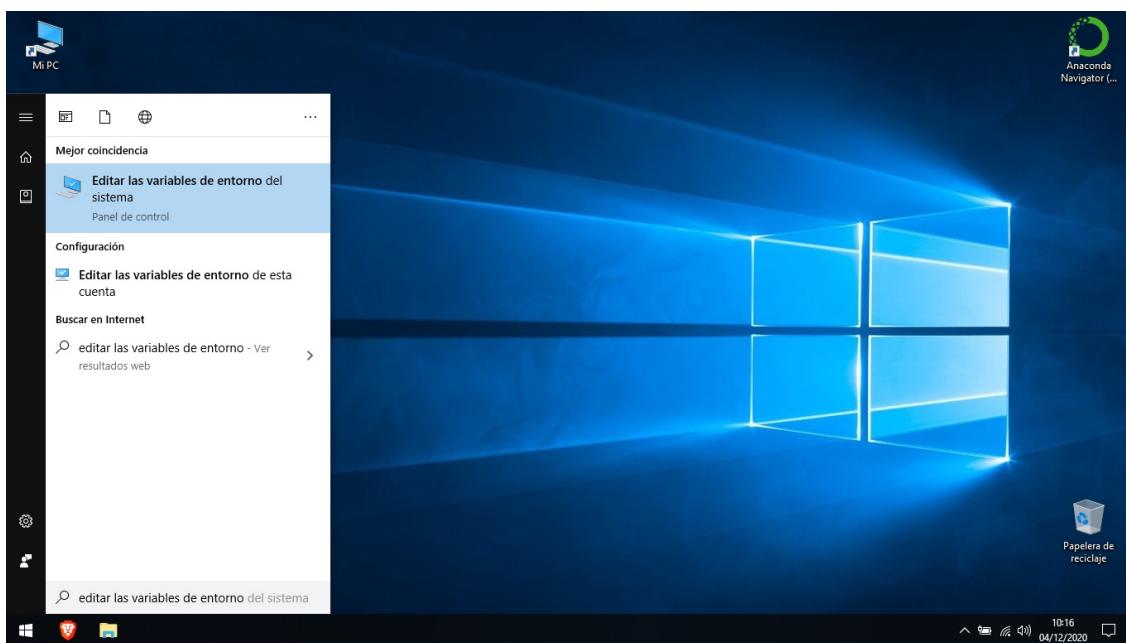
Realmente he elegido el nombre Kafka para la carpeta por facilitar el enrutamiento, pero realmente el nombre daría igual siempre y cuando la configuremos bien en las variables de entorno.

En esta carpeta será donde descomprimiremos el archivo de Apache Kafka recién descargado. Quedará de la siguiente forma:

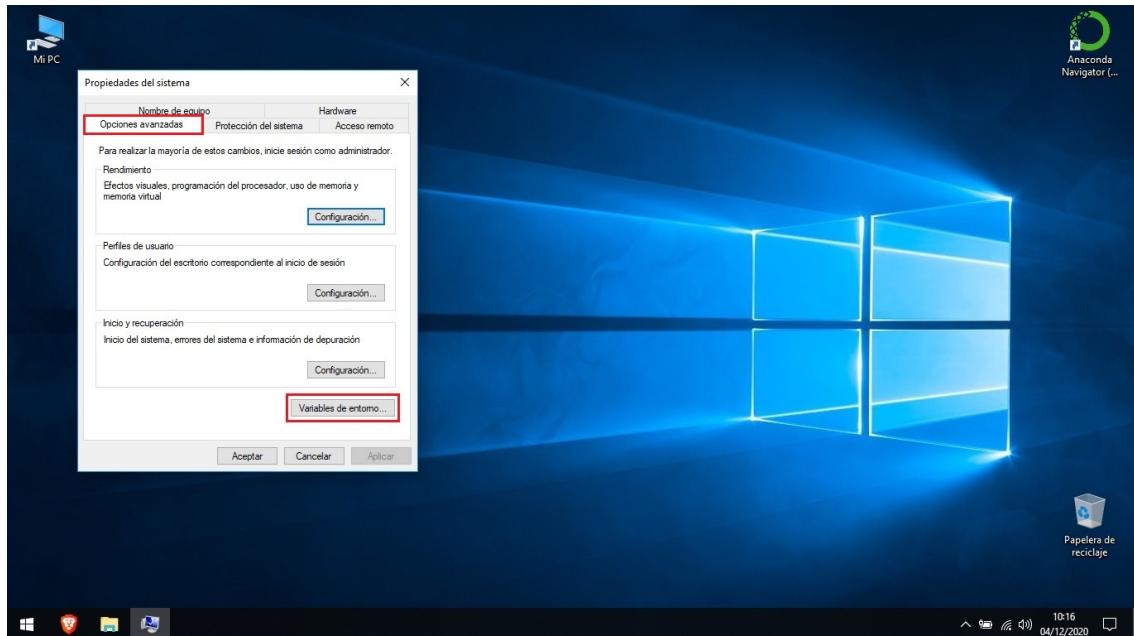


2.5.3. Variables de entorno

Repetiremos los mismos pasos que en los casos anteriores. De todas formas repetiremos los pasos que hay que seguir. Hacemos click en inicio y escribiremos 'Editar las variables de entorno del sistema' como en la siguiente imagen:

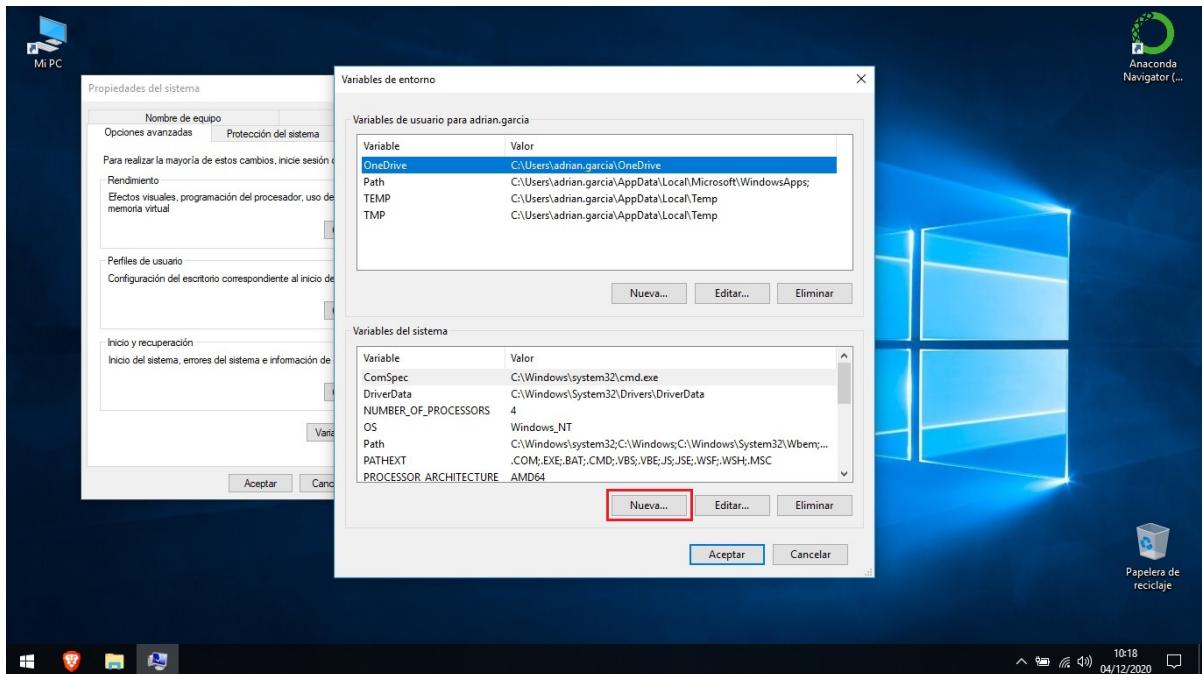


Al abrir el editor veremos que se abre la ventana de Propiedades del sistema. Hacemos click en la pestaña de 'Opciones avanzadas' y abajo hacemos click en 'Variables de entorno':

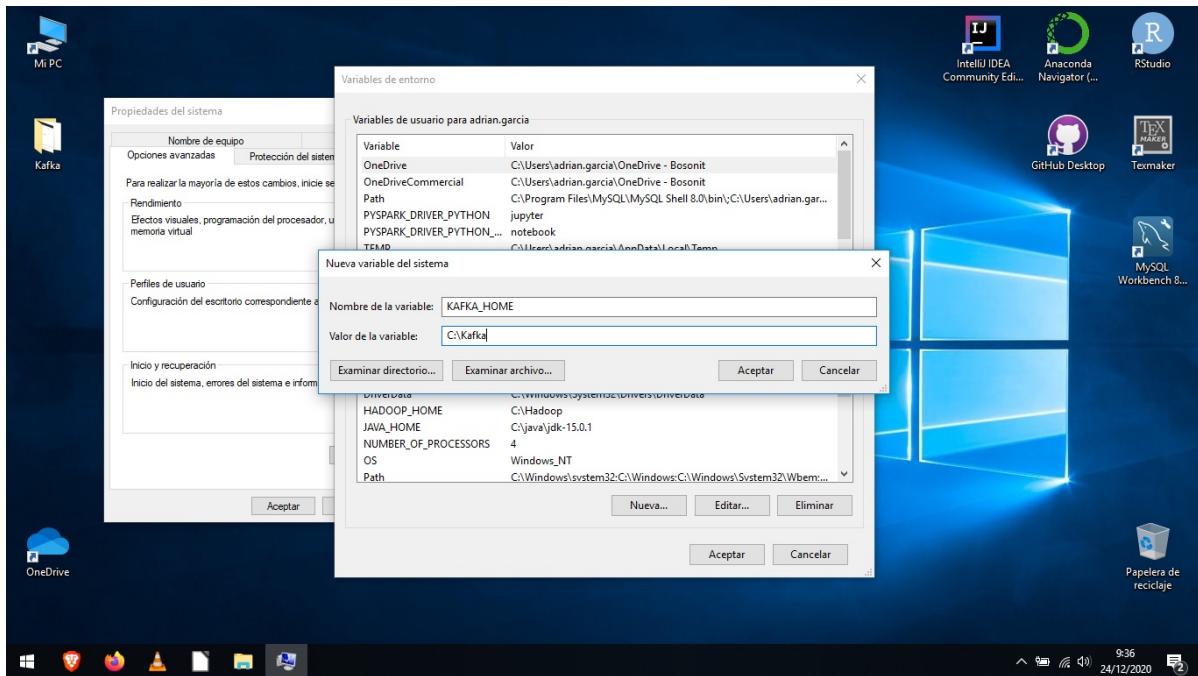


En la ventana que se nos ha abierto veremos dos recuadros. En uno pondrá *Variables de usuario* y en el de debajo *Variables del sistema*. Lo mas recomendable es crear las variables de entorno para el sistema, con el fin de que cualquier usuario tenga acceso.

Entonces en el grupo de Variables del sistema hacemos click en el botón 'Nueva'.



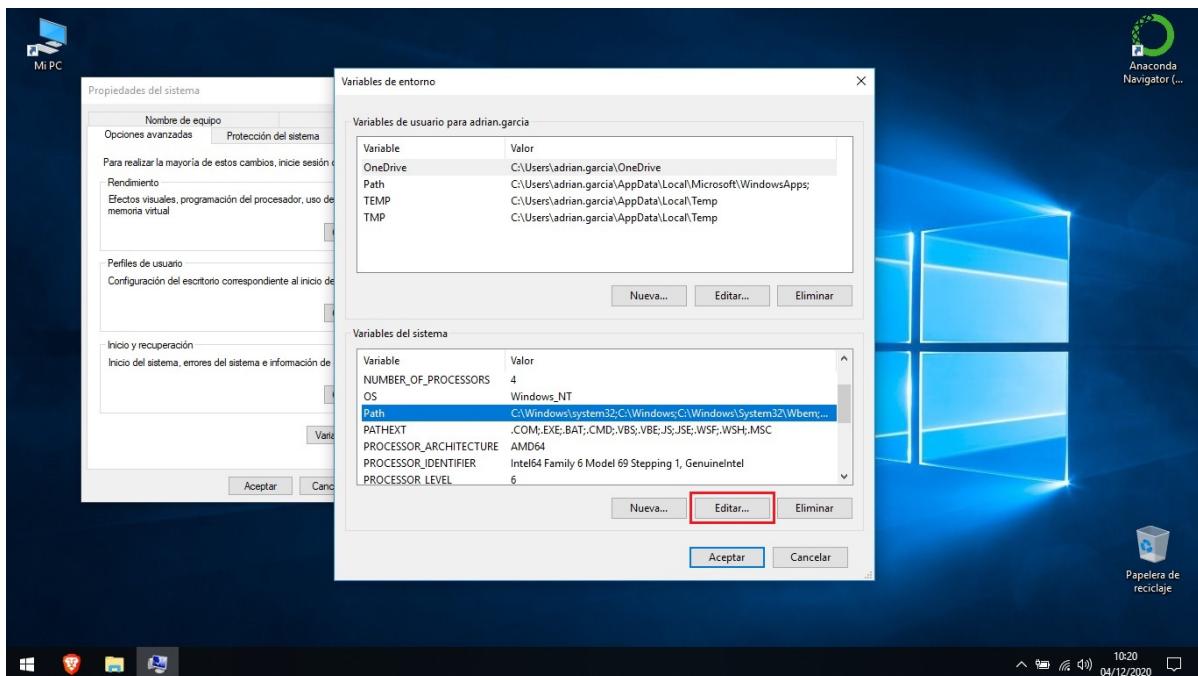
En el nombre escribiremos 'KAFKA_HOME', mientras que en valor la variable escribiremos la ruta donde instalamos Apache Kafka, en nuestro caso será *C:/Kafka*



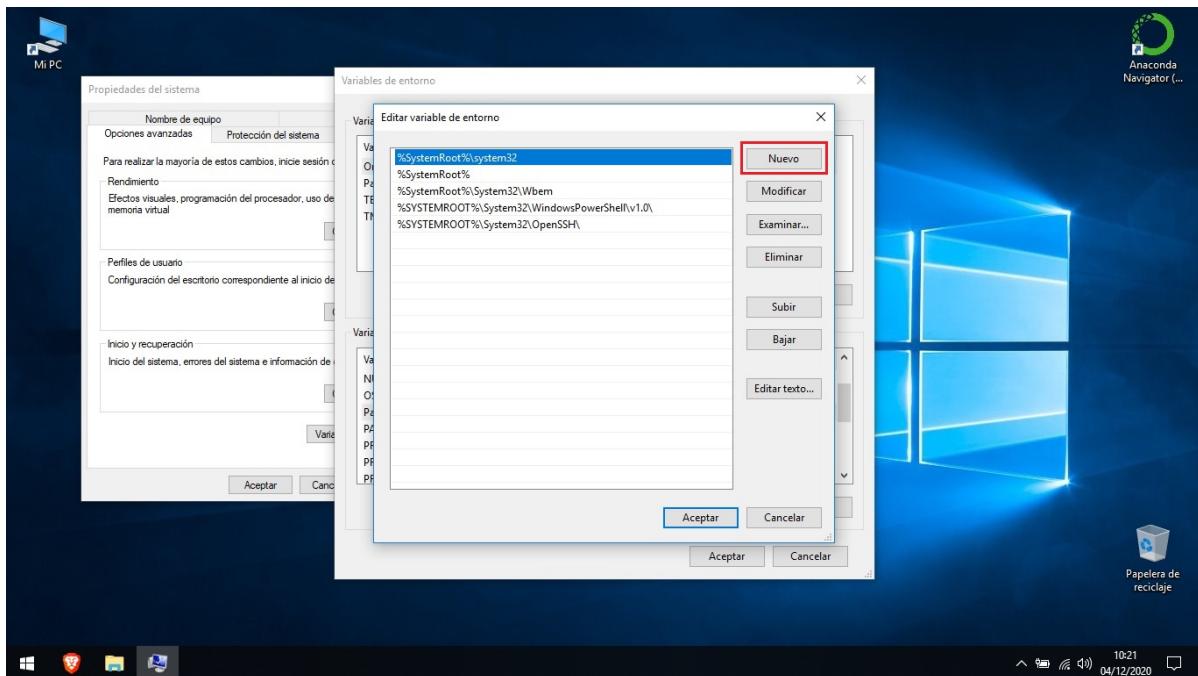
Hacemos clic en aceptar.

• PATH

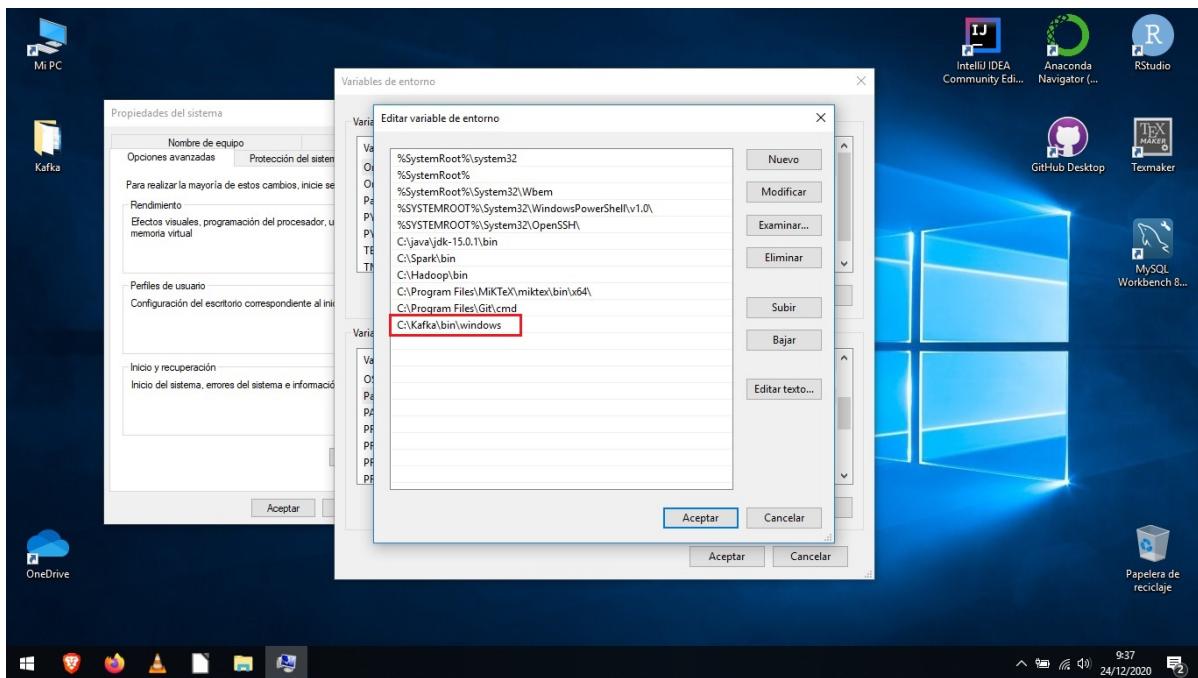
Si por alguna razón no tienes abierta la ventana de variables de entorno, repite los pasos anteriores. Ahora vamos a editar la variable Path que ya existe dentro de variables del sistema. La seleccionamos y le damos a editar:



Podrás ver todos los valores que tiene por defecto la variable Path. No los modifiques o elimines, solo haz click en 'Nuevo'



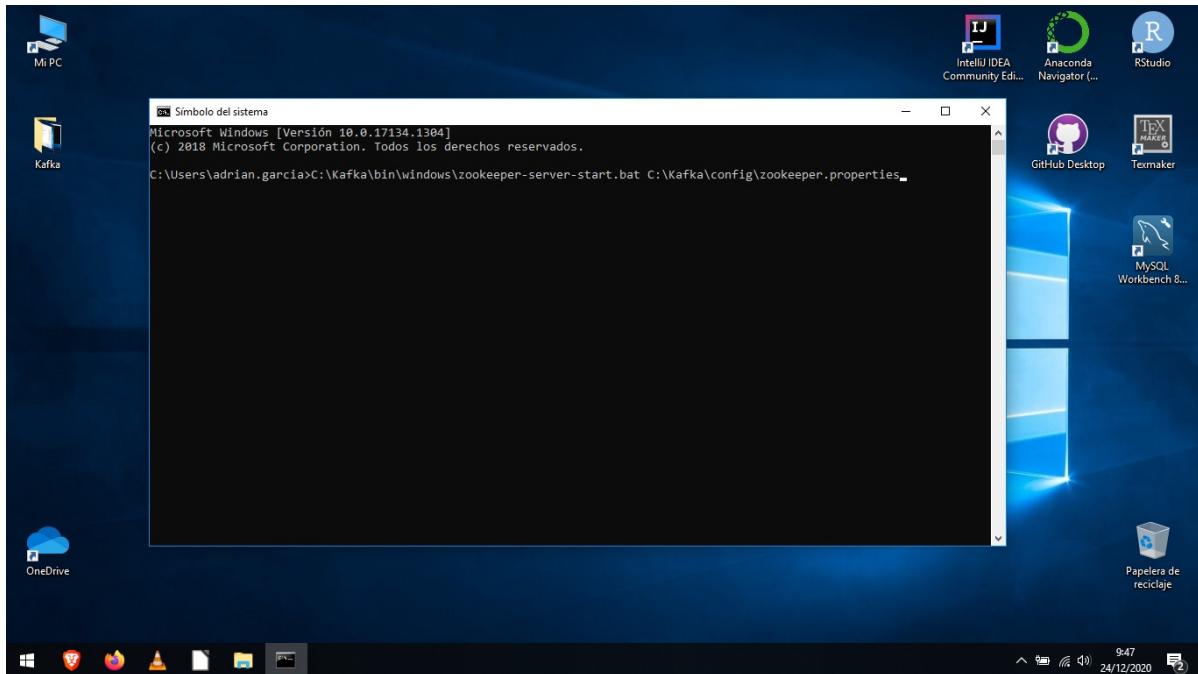
Escribiremos la ruta a la instalación de Kafka, pero en este caso direccionandolo a la carpeta windows dentro de la carpeta bin. Si has seguido todos los pasos de esta guia, la ruta sera *C:/Kafka/bin/windows*



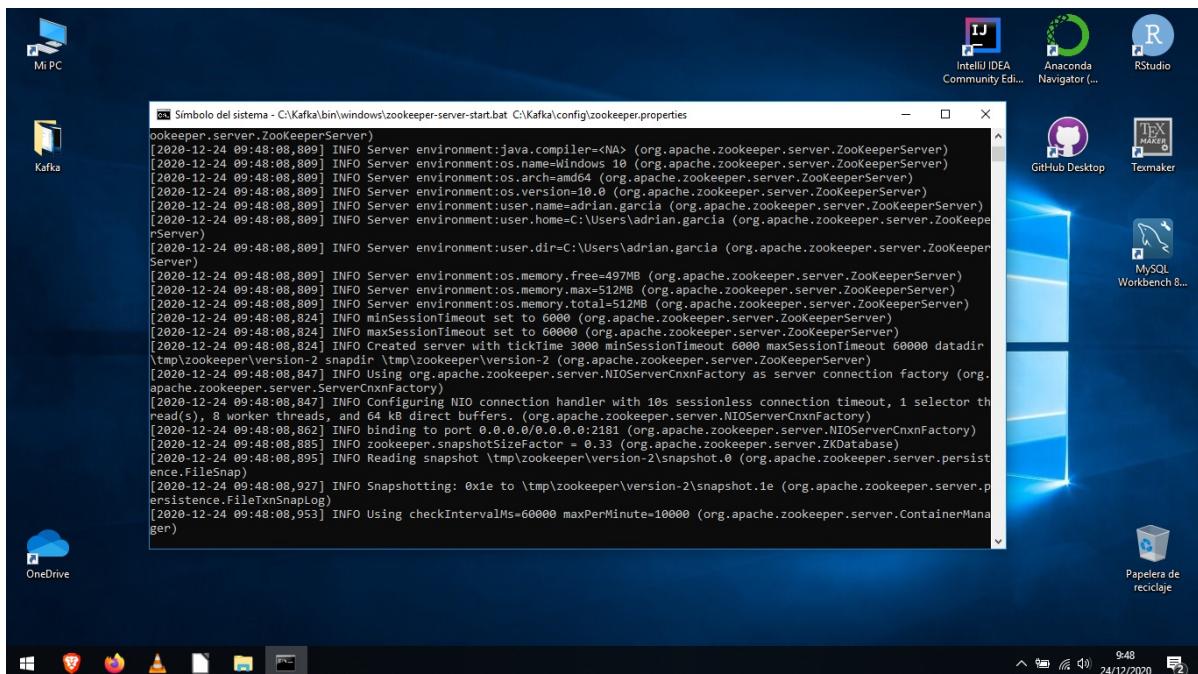
Aceptamos en la ventana de Path y volvemos a aceptar en la ventana de variables de entorno y en propiedades del sistema.

2.5.4. Verificación de Funcionamiento

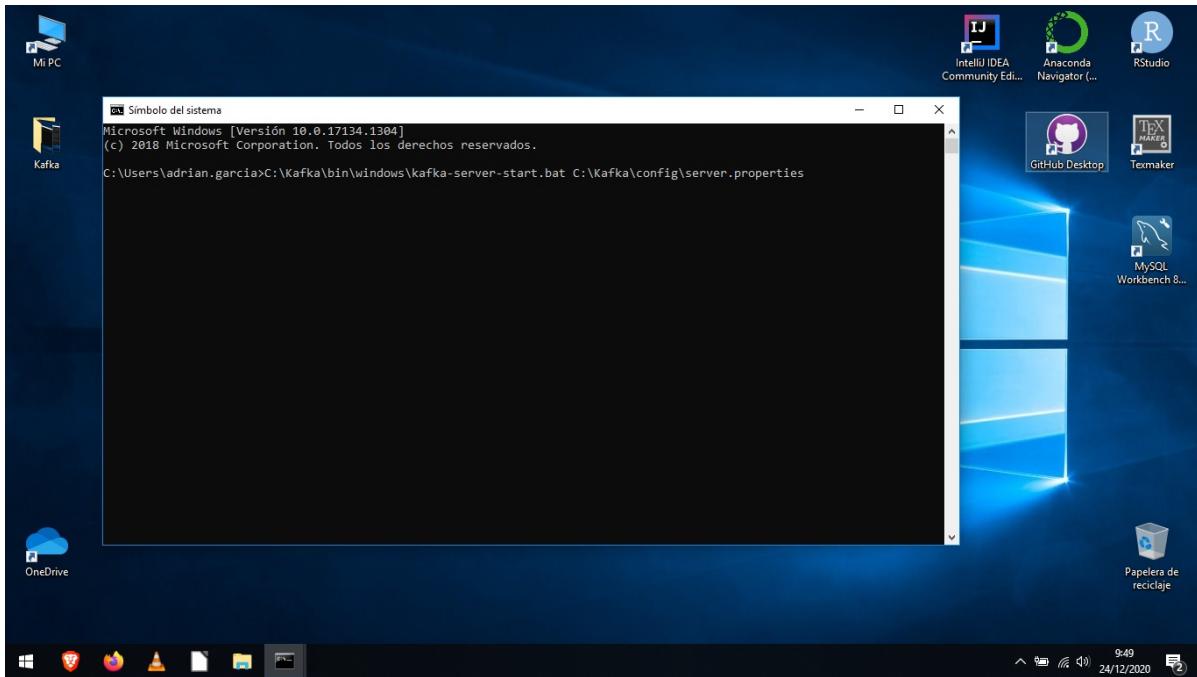
Ahora comprobaremos que la instalación se ha efectuado de forma correcta. Para ello abriremos una consola del sistema. Para ello le damos a inicio, escribiremos *cmd* y ejecutaremos *Símbolo del sistema*. Lo primero que deberemos hacer sera inicializar Apache Zookeeper. Para ello escribiremos **C:\Kafka\bin\windows\zookeeper-server-start.bat** **C:\Kafka\config\zookeeper.properties**:



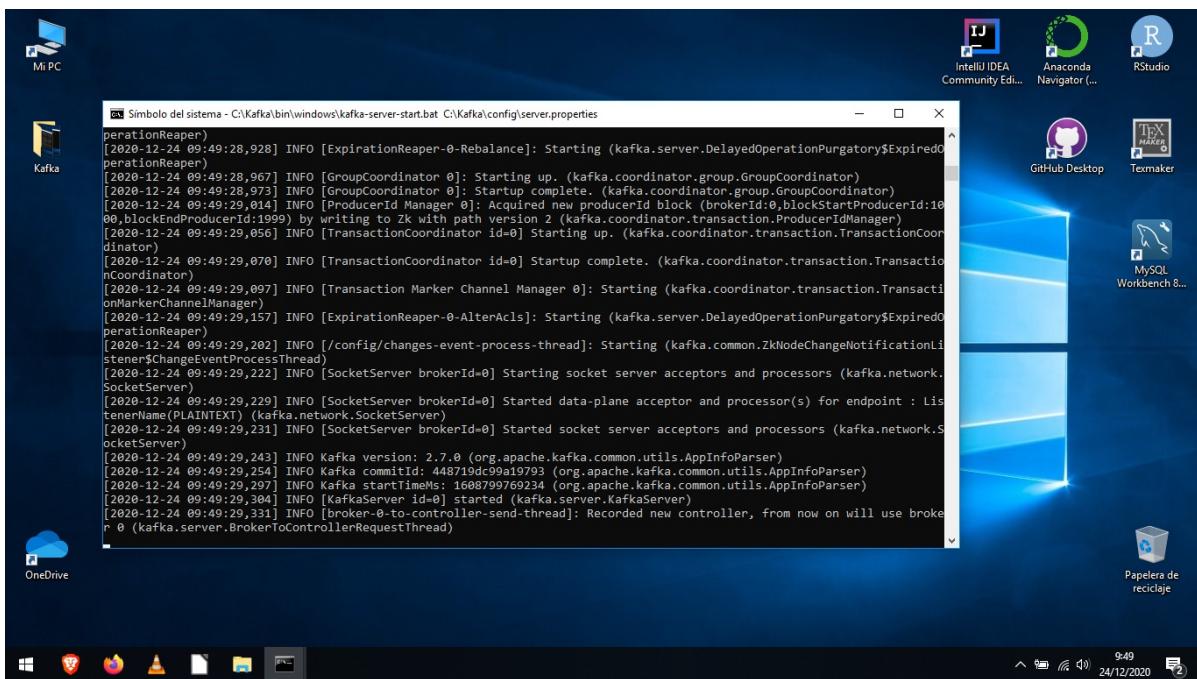
Al ejecutar este comando debería salirnos algo similar a esto:



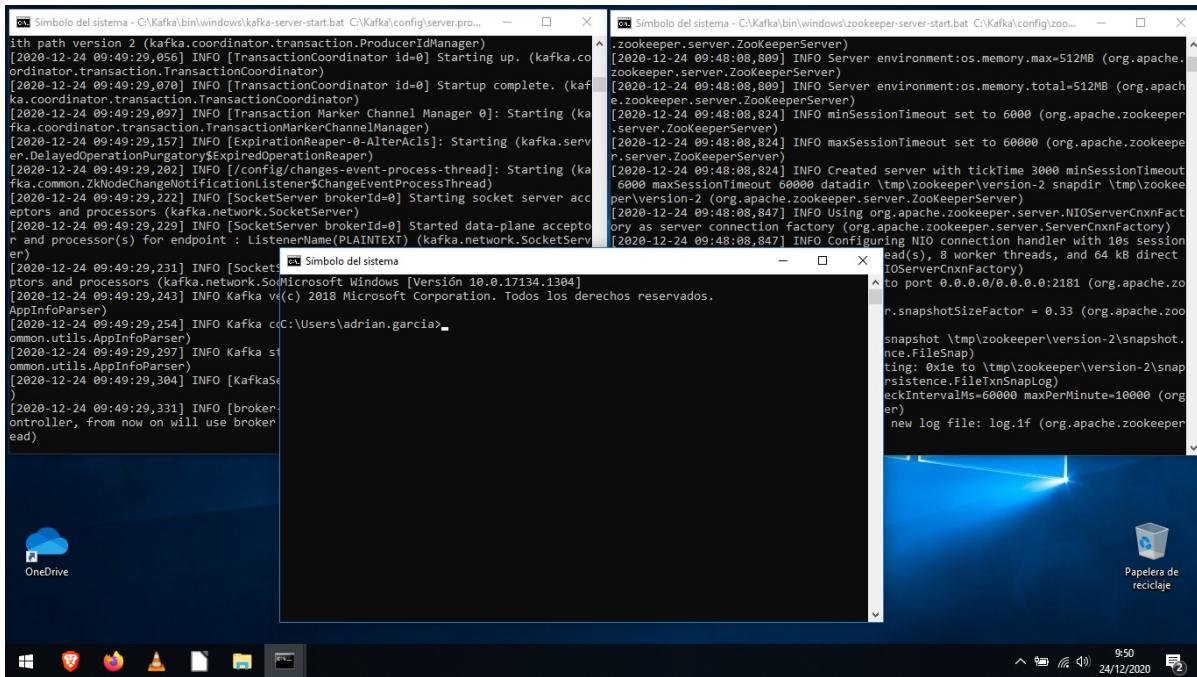
Ahora deberemos abrir otra consola del sistema **nueva** de la misma forma que antes y sin cerrar la consola donde hemos iniciado Apache Zookeeper. Ahora ejecutaremos Kafka Broker/Server. En este caso escribiremos **C:\Kafka\bin\windows\kafka-server-start.bat** **C:\Kafka\config\server.properties**:



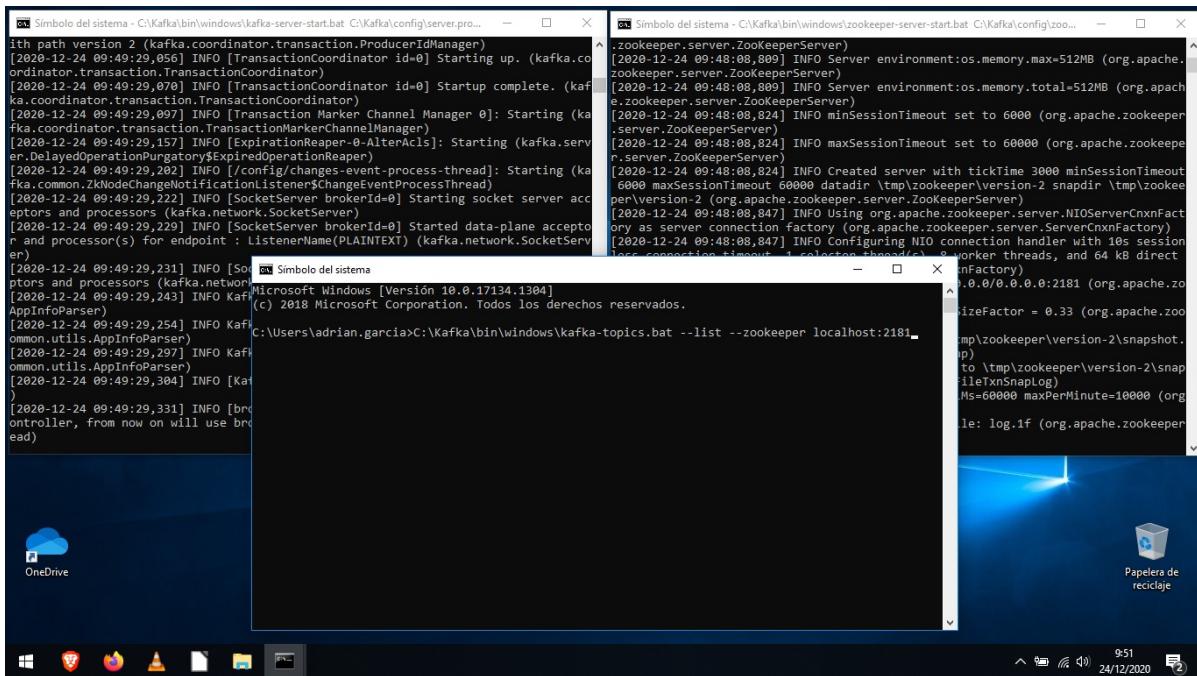
Y tambien deberiamos obtener algo similar a:



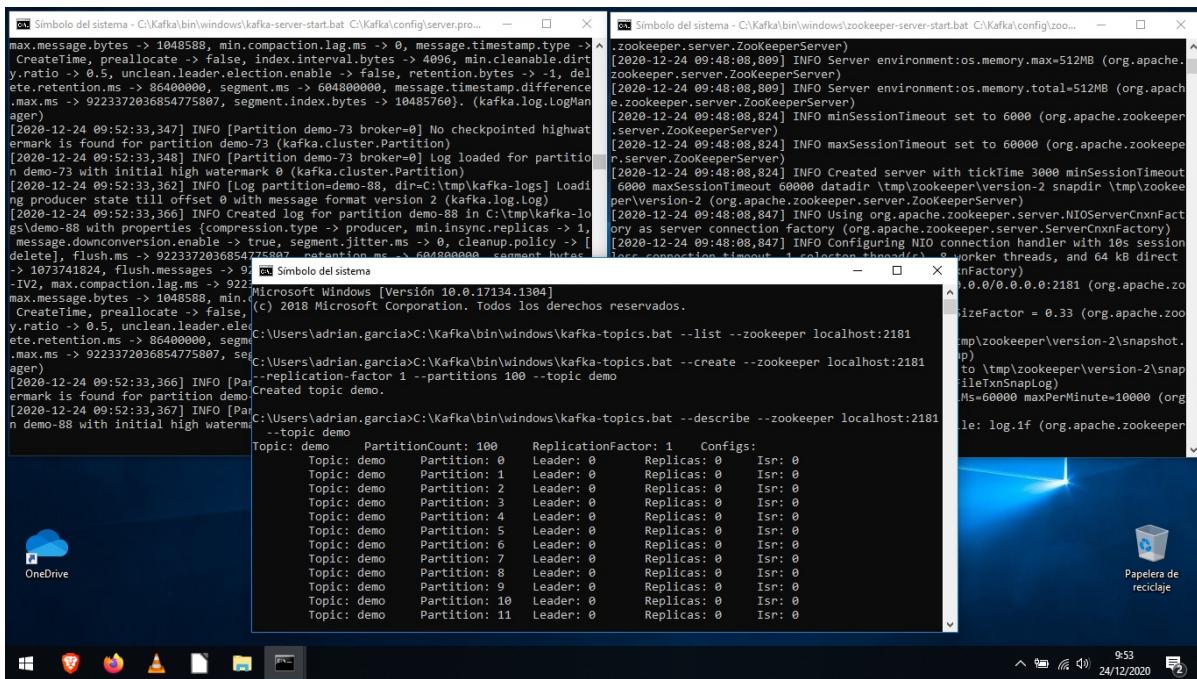
Ahora que tenemos Apache Zookeeper y Kafka Broker funcionando, sin cerrar ningun simbolo del sistema abriremos un tercero para probar que todo se ha ejecutado y funciona perfectamente.



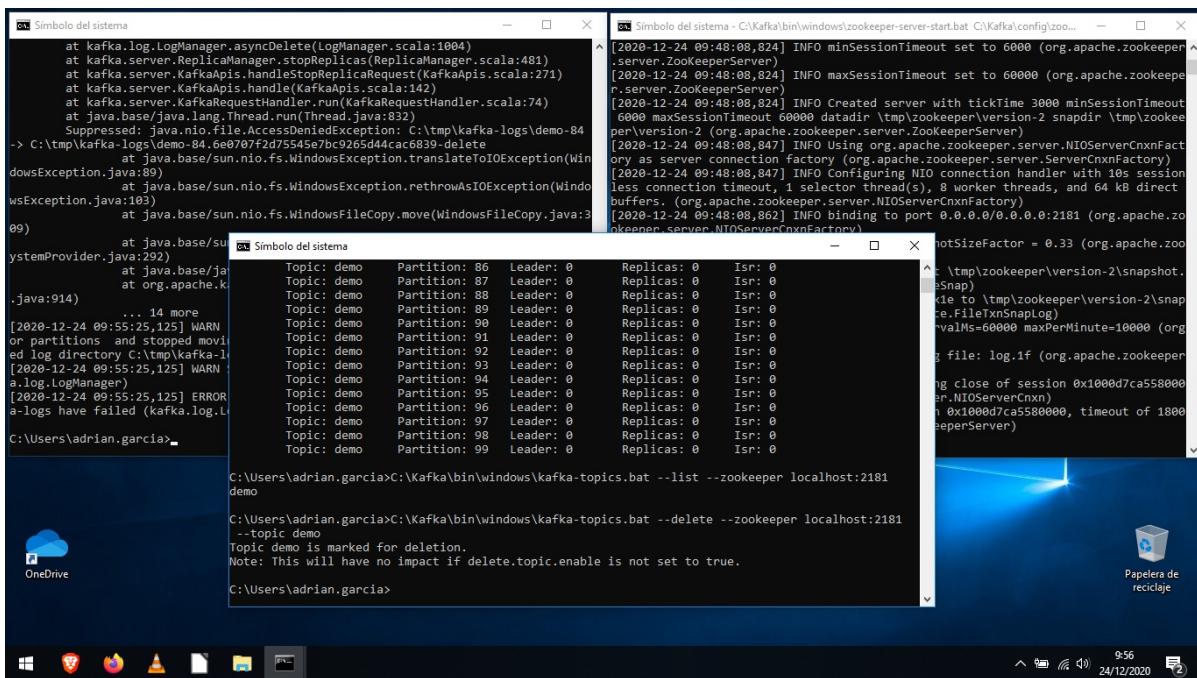
Primero solicitaremos que liste todos los topics, que en principio no deberíamos tener ninguno. Para ello escribiremos **C:\Kafka\bin\windows\kafka-topics.bat -list --zookeeper localhost:2181**



De la misma forma podemos ejecutar un comando de creacion de topics haciendo **C:\Kafka\bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 100 --topic demo**. Y para mostrar las propiedades de este topic recien creado podemos hacer **C:\Kafka\bin\windows\kafka-topics.bat --describe --zookeeper localhost:2181 --topic demo**



Para ver la lista de topics C:\Kafka\bin\windows\kafka-topics.bat –list –zookeeper localhost:2181 y para eliminarlo C:\Kafka\bin\windows\kafka-topics.bat –delete –zookeeper localhost:2181 –topic demo



3. Instalación y Configuración en Ubuntu

3.1. Requisitos

Sistema operativo Ubuntu de 64bits

3.2. Java JDK

El Java JDK es el Java Development Kit, que traducido al español significa, Herramientas de desarrollo para Java. Aquí nos encontraremos con el compilador javac que es el encargado de convertir nuestro código fuente (.java) en bytecode (.class), el cual posteriormente sera interpretado y ejecutado en la JVM, Java Virtual Machine por sus siglas en inglés, que nuevamente en español significa, La Maquina Virtual de Java.

Puede que nos suene mas Java JRE, este es el Java Runtime Environment, que en español significa, Entorno de Ejecución de Java. En palabras del propio portal de Java es la implementación de la Máquina virtual de Java que realmente ejecuta los programas de Java, esto quiere decir que aquí encontraremos todo lo necesario para ejecutar nuestras aplicaciones escritas en Java.

Normalmente el JRE esta destinado a usuarios finales que no requieren el JDK, pues a diferencia de este, no contiene los programas necesarios para crear aplicaciones en el lenguaje Java, es así, que el JRE se puede instalar sin necesidad de instalar el JDK, pero al instalar el JDK, este siempre cuenta en su interior con el JRE.

3.3. Anaconda

Anaconda es una solución flexible de código abierto que proporciona las utilidades para crear, distribuir, instalar, actualizar y administrar software de manera multiplataforma. Ademas nos facilita la gestión de múltiples entornos de datos que se pueden mantener y ejecutar por separado sin interferencias entre sí.

Nos va a servir para el procesamiento de datos a gran escala, el análisis predictivo y la informática científica, que tiene como objetivo simplificar la gestión de empaquetado y distribución. Esta es quizás la Suite más completa para la Ciencia de datos con Python y que nos brinda una gran cantidad de funcionalidades que nos van a permitir desarrollar aplicaciones de una manera más eficiente, rápida y sencilla.

3.3.1. Instalación

Para poder instalar la suite Anaconda necesitaremos el paquete CURL instalado en Ubuntu. Si es una instalacion limpia, abriremos una terminal y escribiremos:

```
sudo apt update
```

4. Apache Spark

4.1. Tipos de administradores de clústers

Actualmente (Version 3.0.2 de Spark), el sistema admite varios administradores de clusters:

- Standalone – un administrador de clúster simple incluido con Spark que facilita la configuracion de un cluster.
- Apache Mesos – un administrador de clúster general que también puede ejecutar Hadoop MapReduce y aplicaciones de servicio.
- Hadoop YARN – el administrador de recursos a partir de Hadoop 2.
- Kubernetes – un sistema de codigo abierto para automatizar el despliegue, el escalado y la gestion de aplicaciones en contenedores.

Existe un proyecto de terceros (no compatible con el proyecto Spark) para agregar compatibilidad con Nomad como administrador de clúster.

5. Servicios en la nube

5.1. Databricks

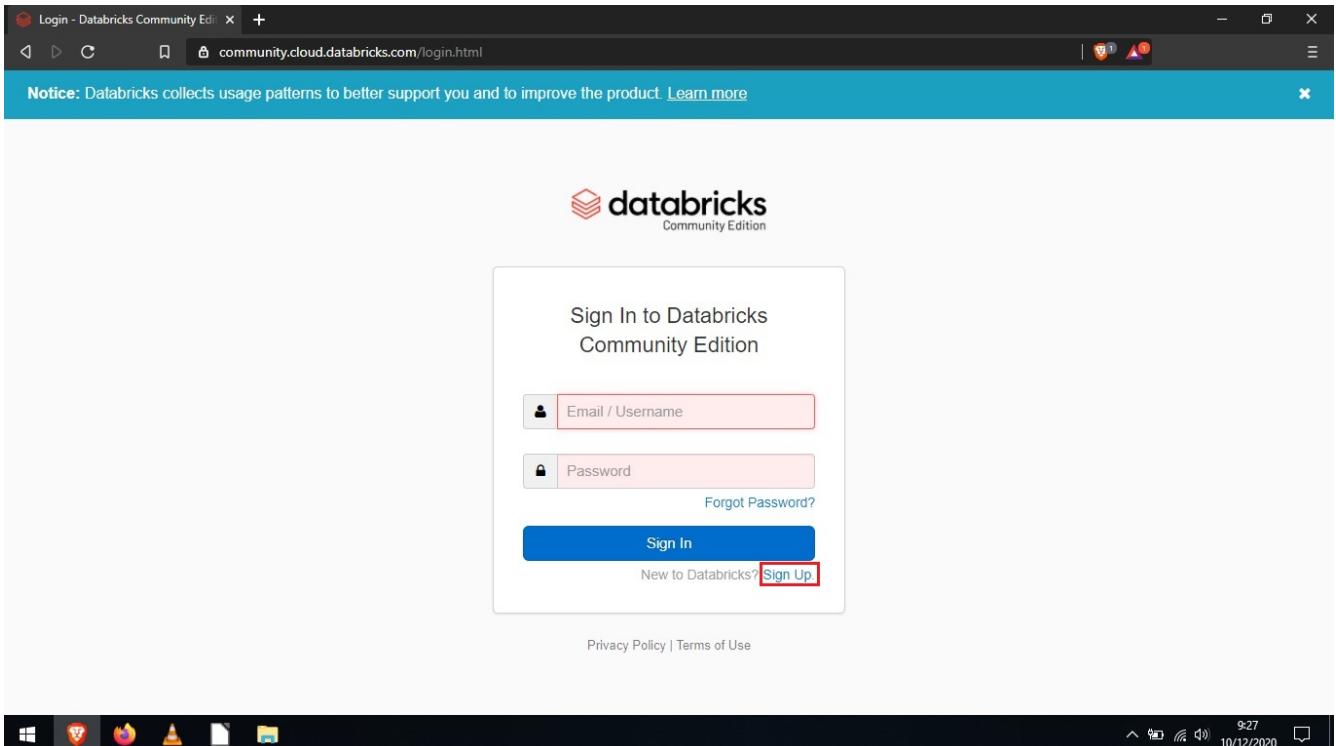
Databricks es el nombre de la plataforma analítica de datos basada en Apache Spark desarrollada por la compañía con el mismo nombre. La empresa se fundó en 2013 con los creadores y los desarrolladores principales de Spark. Permite hacer analítica Big Data e inteligencia artificial con Spark de una forma sencilla y colaborativa. Esta plataforma tambien está disponible como servicio cloud en Microsoft Azure y Amazon Web Services (AWS).

5.1.1. Registro

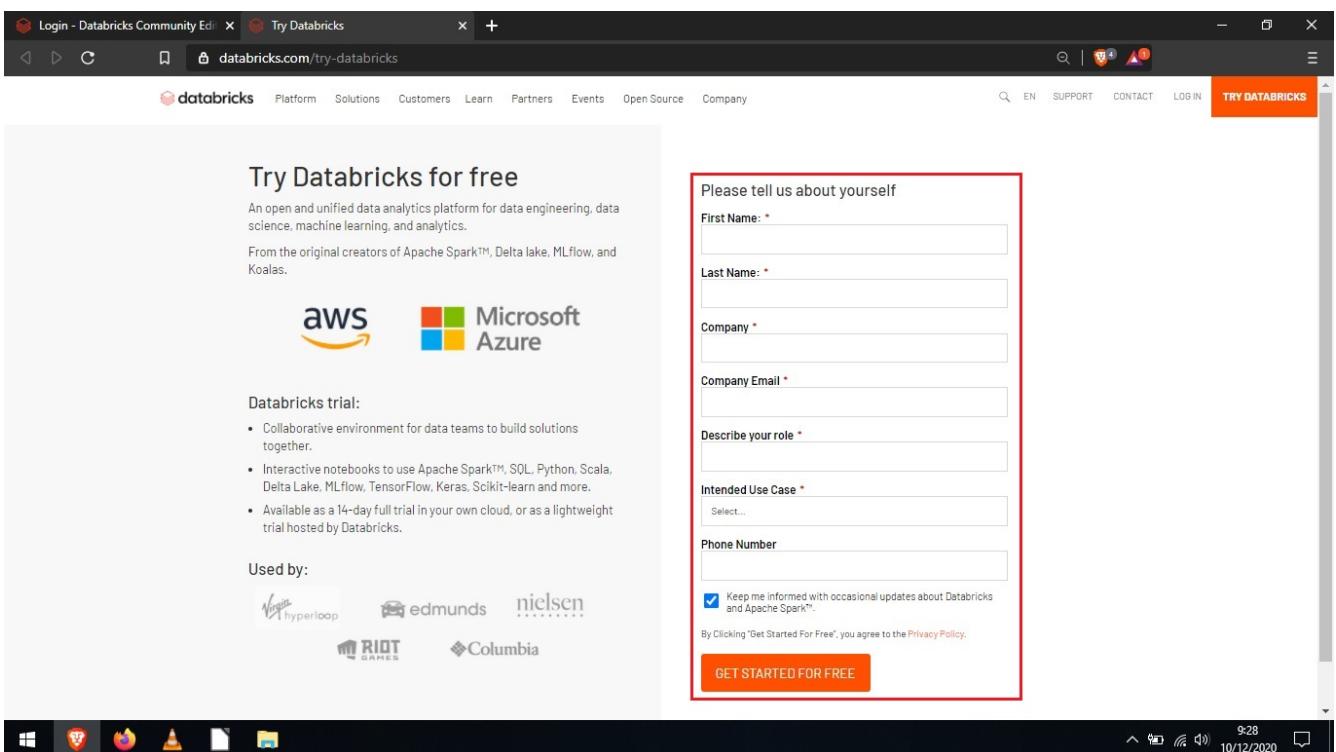
Databricks nos permite probar su funcionamiento con una serie de limitaciones por medio de la llamada 'Databricks Community Edition'. Lo primero que deberemos hacer para poder acceder a esta version de prueba sera entrar en la siguiente web:

<https://community.cloud.databricks.com/>

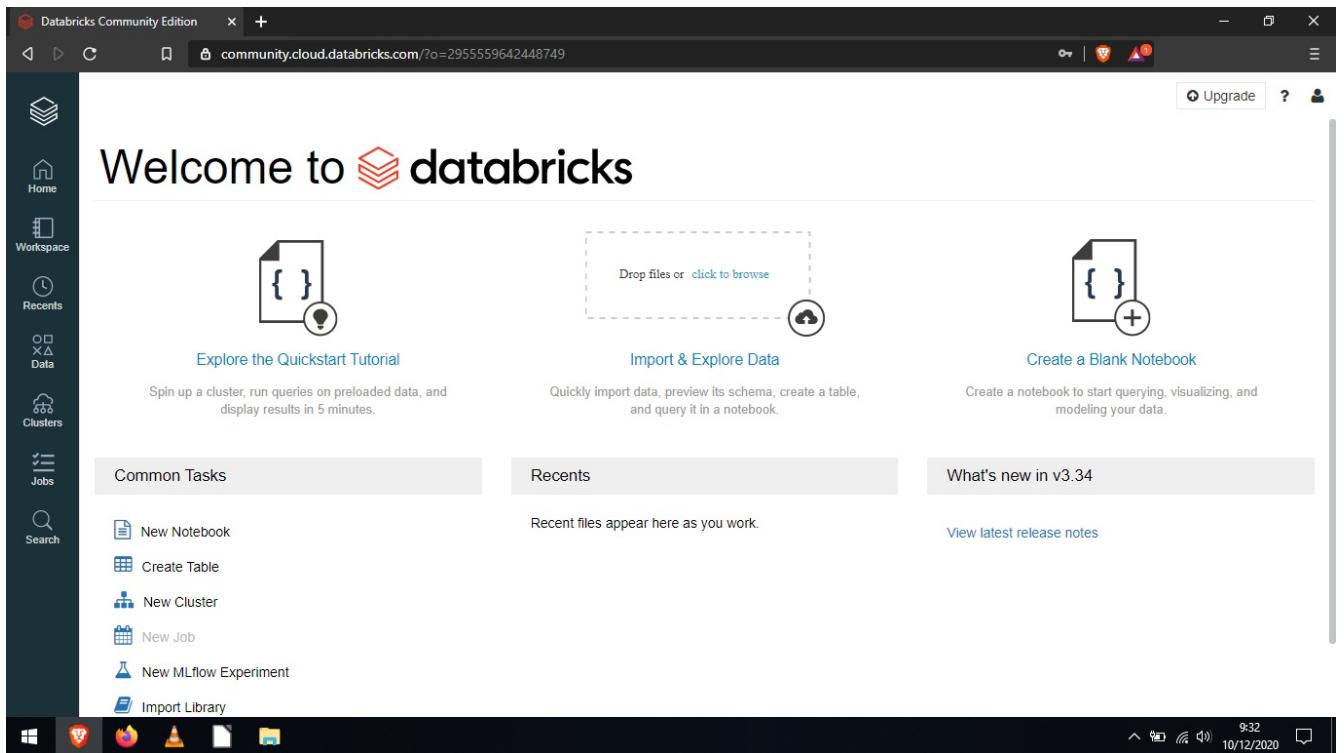
Al entrar veremos la siguiente pagina web:



Lo primero que deberemos hacer es registrarnos en la web para tener un nombre de usuario y contraseña. Para ello hacemos click en 'Sign Up' y completamos los datos que se nos solicitan en la web.

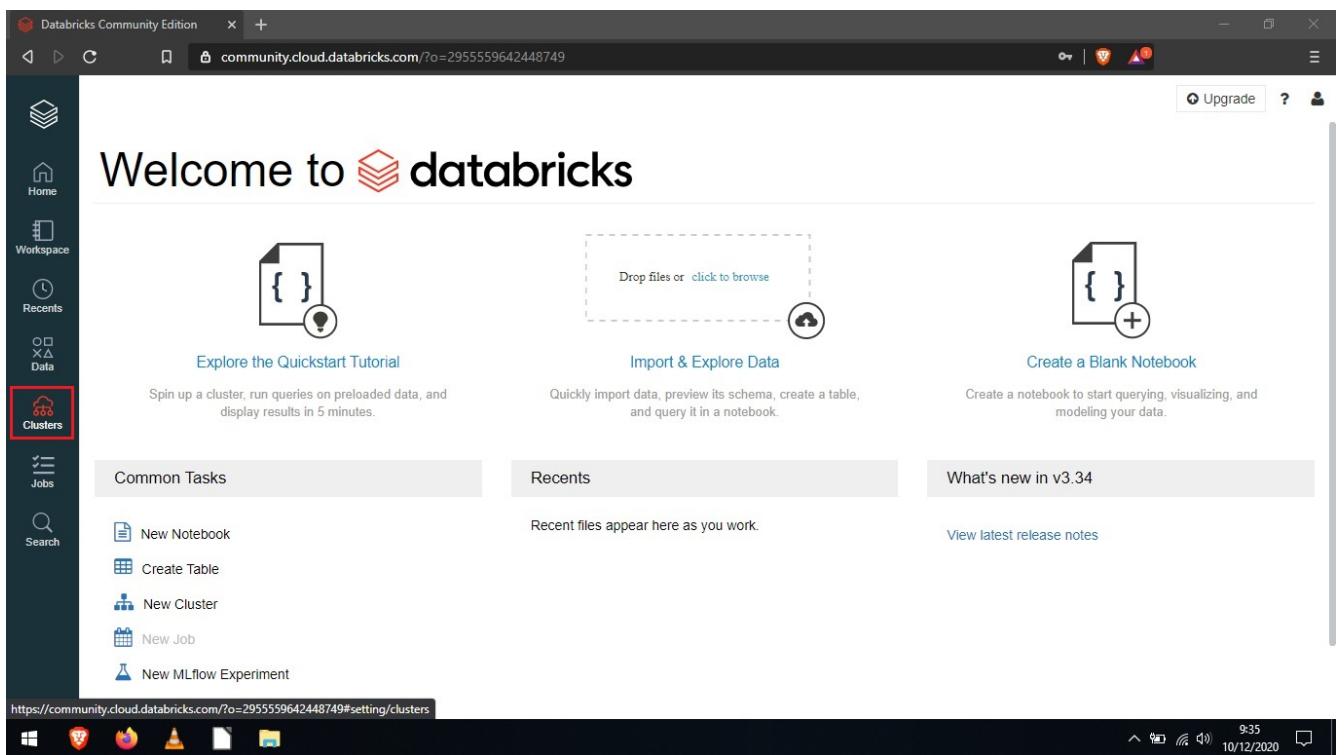


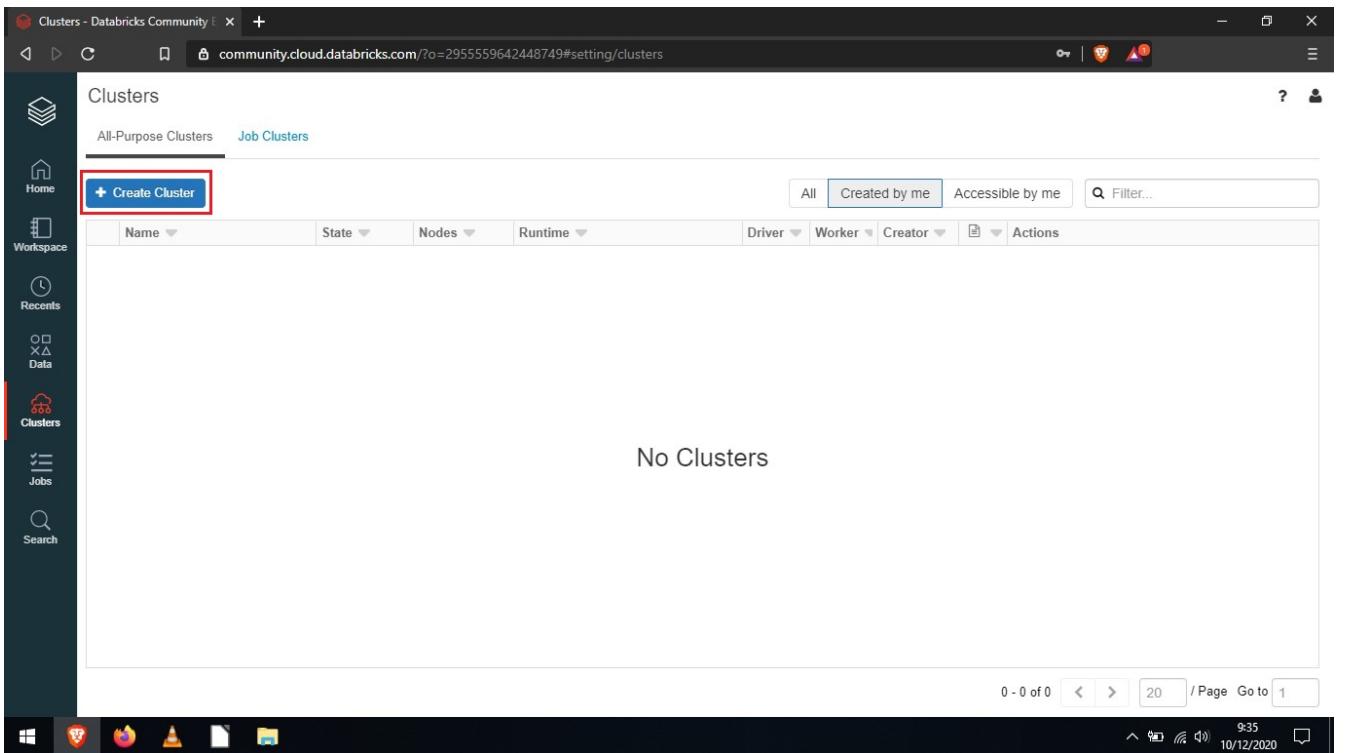
Una vez completado el registro y verificado nuestra dirección de email ya podemos iniciar sesión. Lo que encontraremos al iniciar sesión será un Dashboard de esta manera:



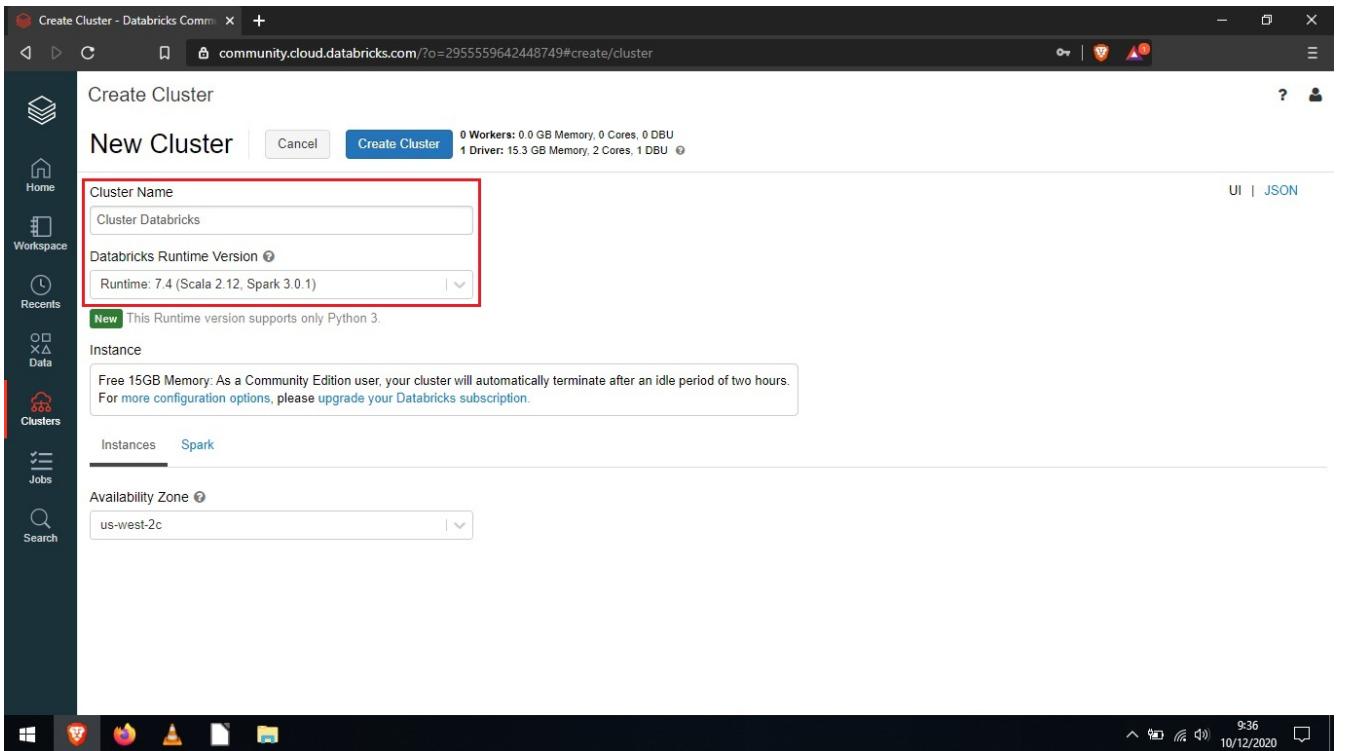
5.1.2. Creacion de Cluster

Como podemos ver, se nos incluye una guia rapida y las principales funciones basicas. Lo que haremos ahora es ir a 'Clusters' en el menu lateral para crear nuestro Cluster personalizado.

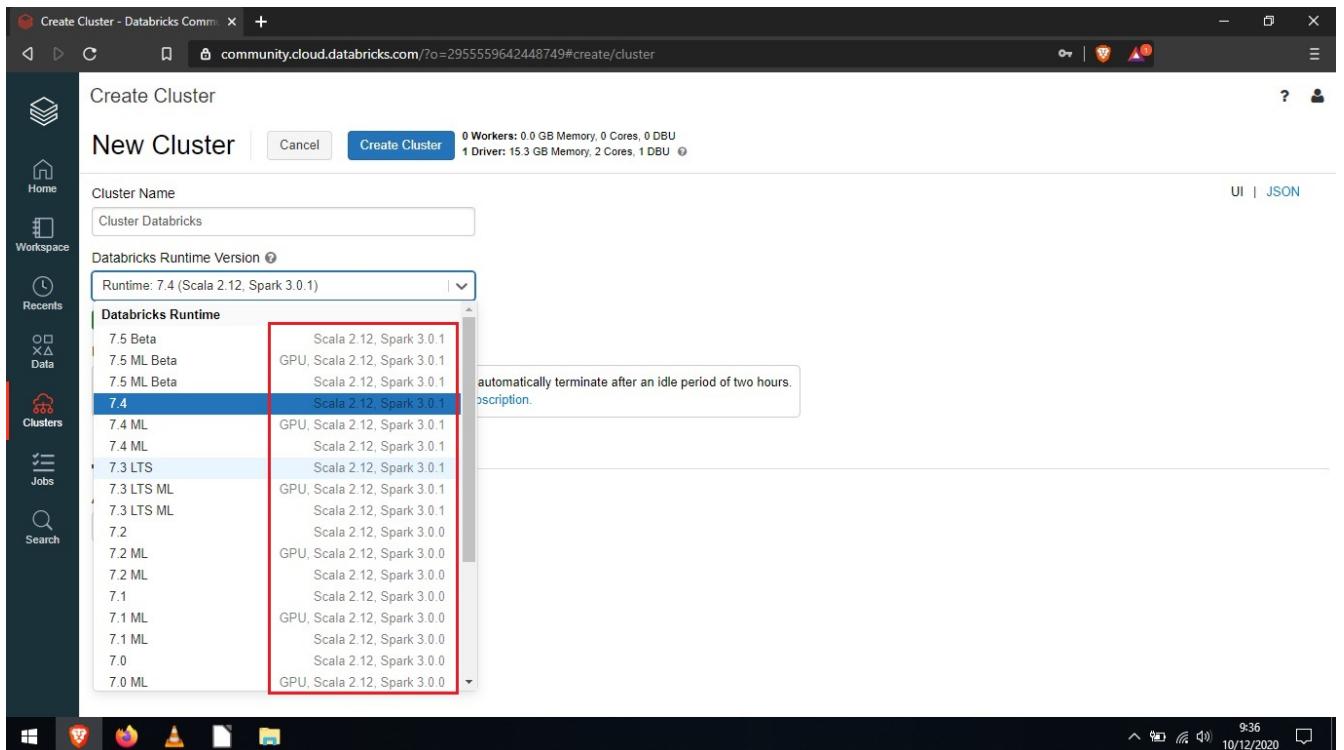




Ahora haremos click en 'Create Cluster':



Como podemos ver en la imagen, en primer lugar se nos pide un nombre para nuestro cluster. Lo siguiente sera elegir que version del sistema queremos para nuestro cluster en el que se indicara la version de Scala y Spark que tendra. Si hacemos click en él, veremos un desplegable:



Veremos que ademas de las diferentes versiones del sistema, aparecen algunos subtítulos que tienen el siguiente significado:

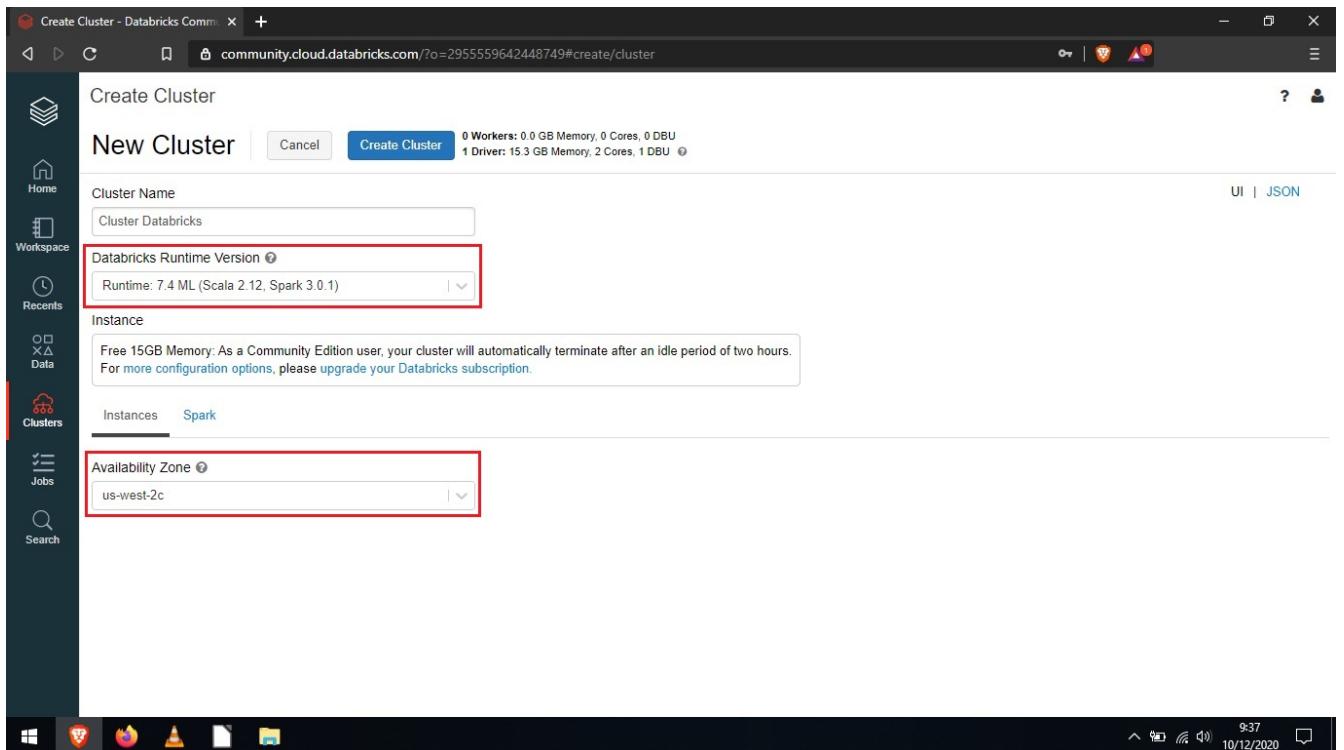
- LTS: en ingles 'Long Term Support'. Es un termino informatico usado para nombrar versiones o ediciones de software diseñadas para tener soporte durante un periodo de tiempo mayor de lo normal. Se aplica normalmente a proyectos de software de codigo abierto. Es la mas recomendable si vamos a hacer un proyecto a largo plazo en el que no queramos quedarnos en algun momento sin soporte.
- ML: Como se dice en la documentacion de Databricks, las versiones con ML contiene las librerias de machine learning mas populares, incluyendo TensorFlow, Pytorch y XGBoost. Ademas tambien es compatible con deep learning distribuido usando Horovod. Se recomienda si vamos a usar tecnicas de machine learning.
- Beta: Version de prueba que esta en fase de pulido. Solo se recomienda si vamos a probar nuevas funcionalidades no incluidas en versiones anteriores mas estables.

En la parte derecha tambien veremos que existen versiones estandar y versiones que incluyen GPU. Estas versiones con GPU solo deben seleccionarse si vamos a utilizar el procesamiento en paralelo de estas.

Si necesitamos ampliar la informacion sobre las diferentes versiones que nos proporciona la plataforma podemos acceder a la web:

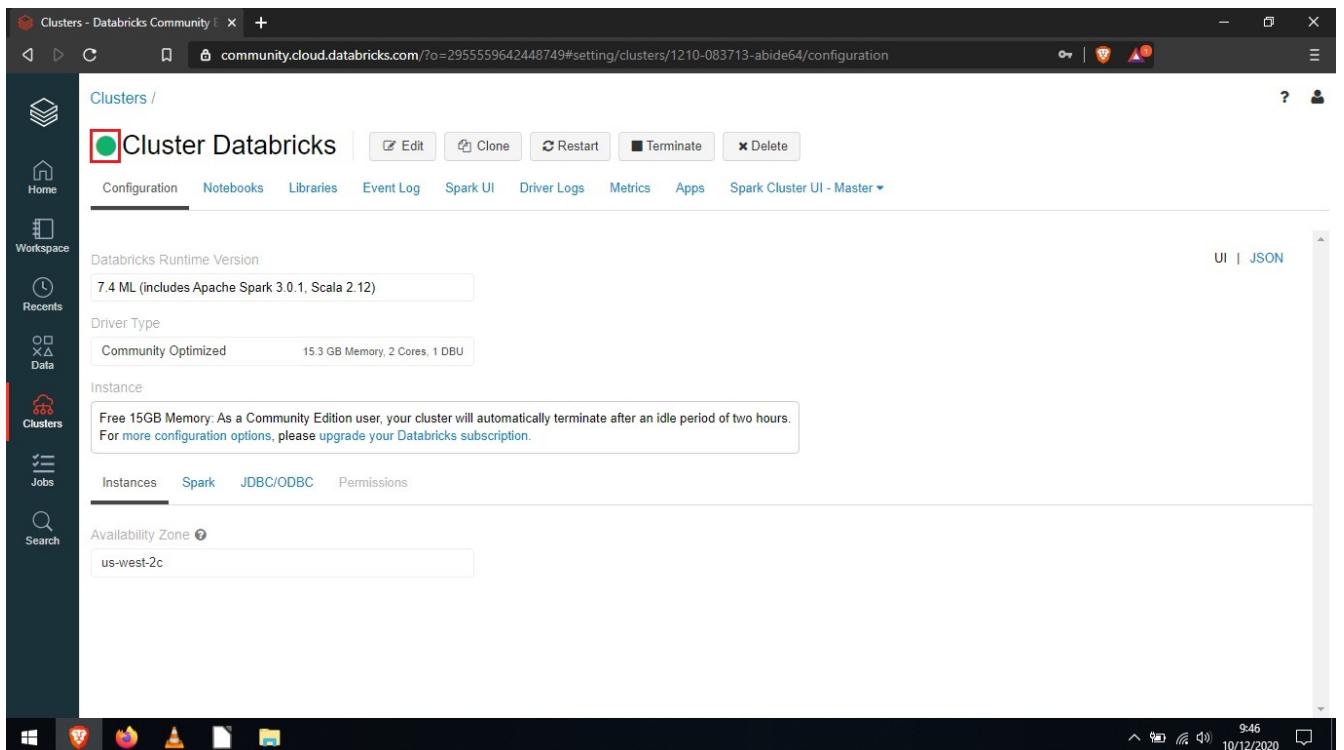
<https://docs.databricks.com/release-notes/runtime/releases.html>

En nuestro caso seleccionaremos la version 7.4 ML sin GPU, que incluye la version 2.12 de Scala y la 3.01 de Spark.



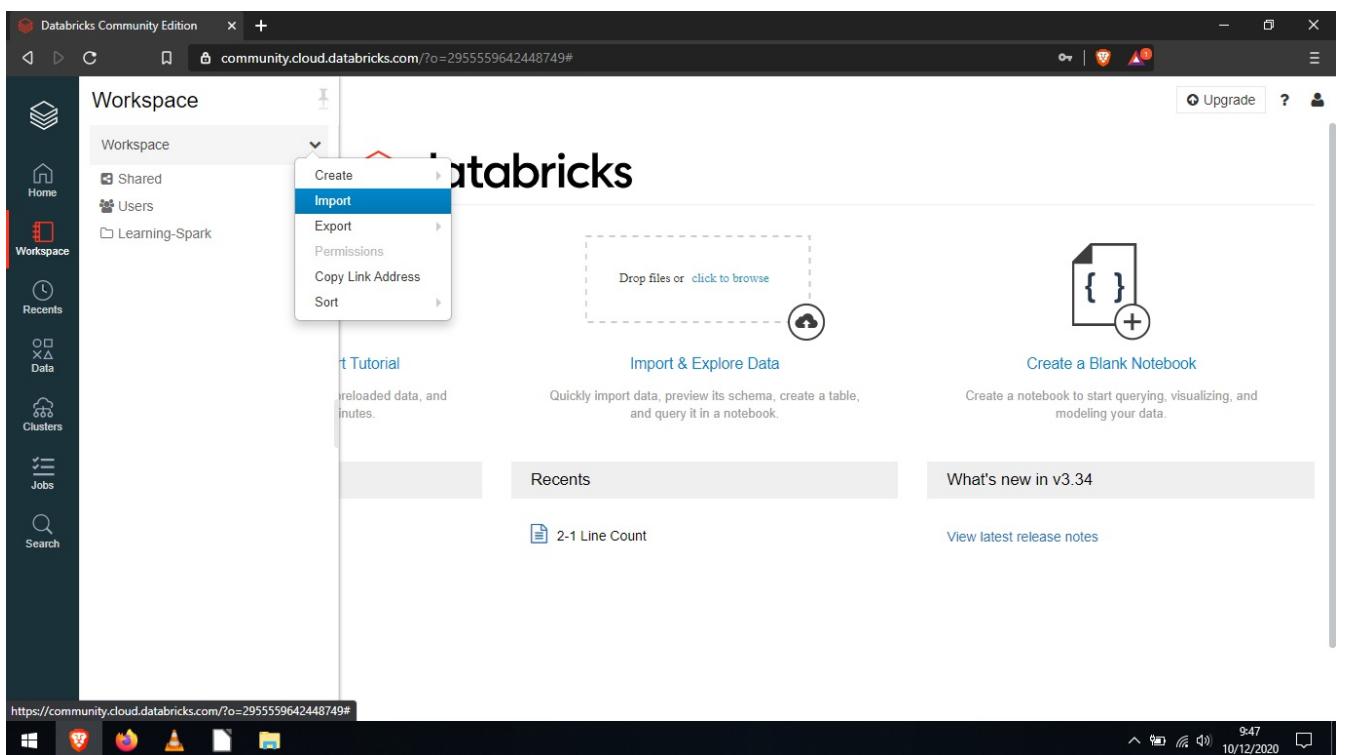
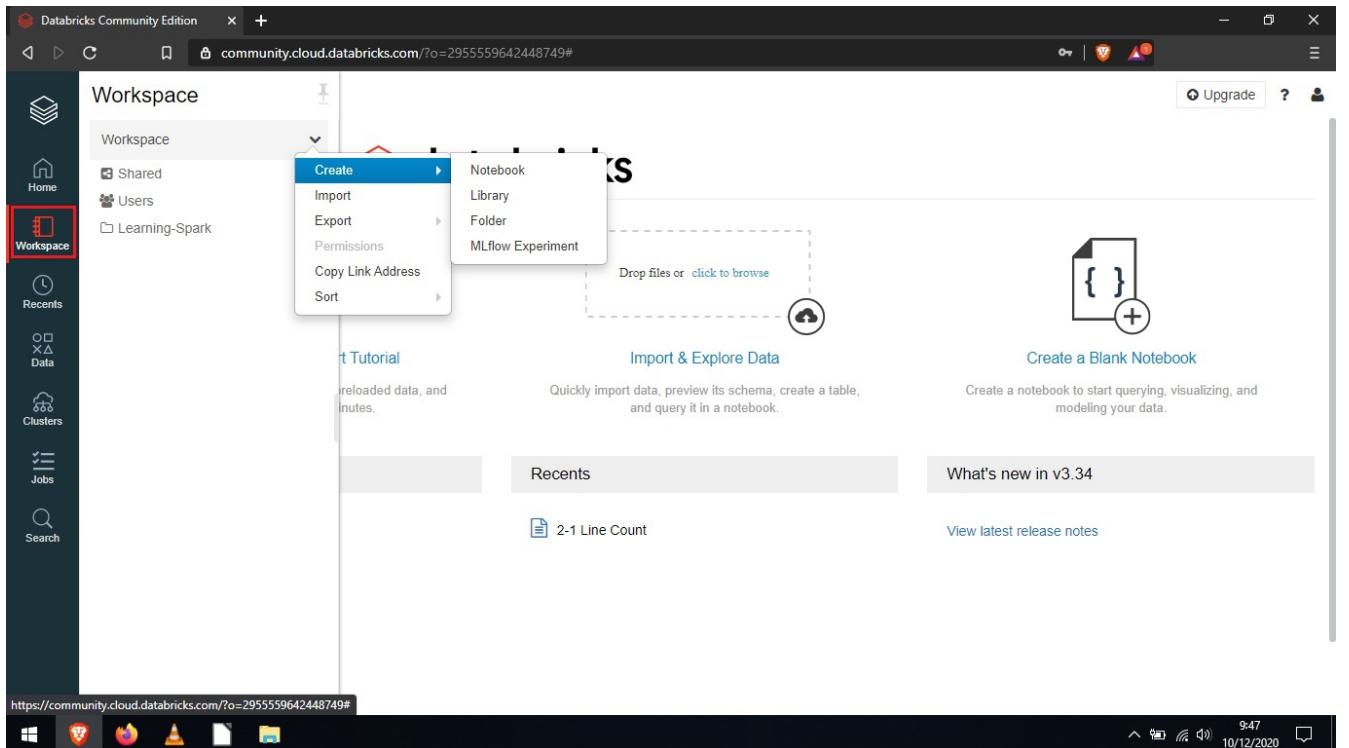
La ultima opcion que vemos es la de 'Availability Zone' que nos indica donde estara hospedado nuestro cluster. Esto nos afectara en la velocidad entre envio de peticion y respuesta. En este caso solo tenemos opciones de Estados Unidos por lo que lo dejamos tal cual esta.

Una vez tenemos todas las caracteristicas deseadas solo tenemos que hacer click en 'Create Cluster'. El proceso de creacion puede tardar varios minutos, en cuanto este desplegado nuestro cluster veremos un indicador verde en la parte superior como en la siguiente imagen:

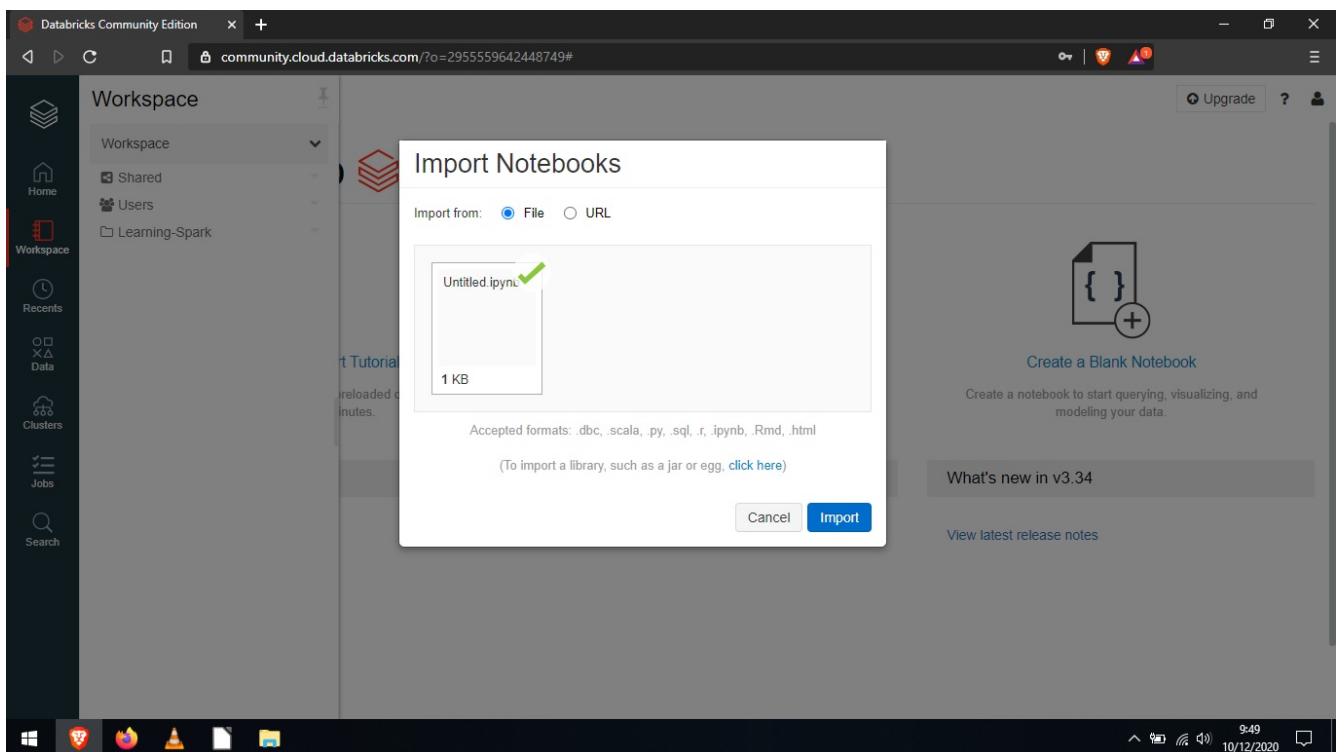
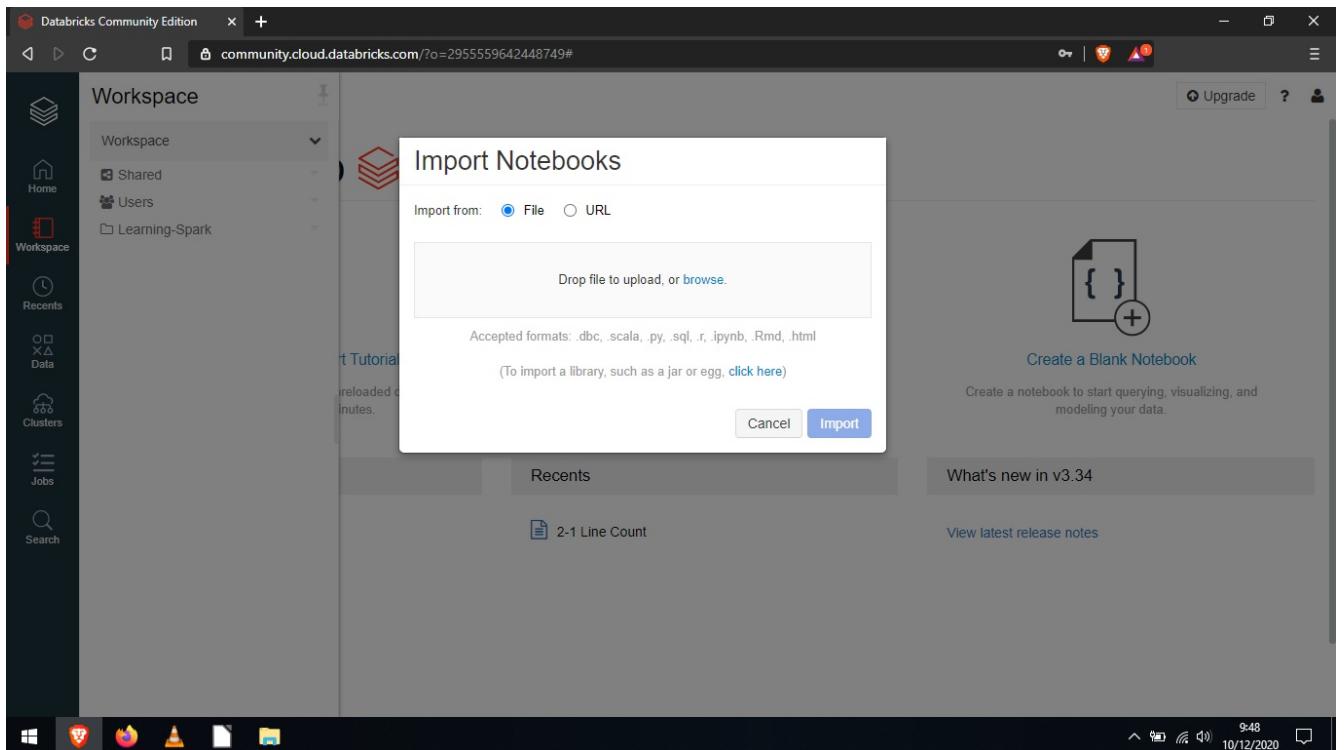


5.1.3. Creacion y carga de Notebooks

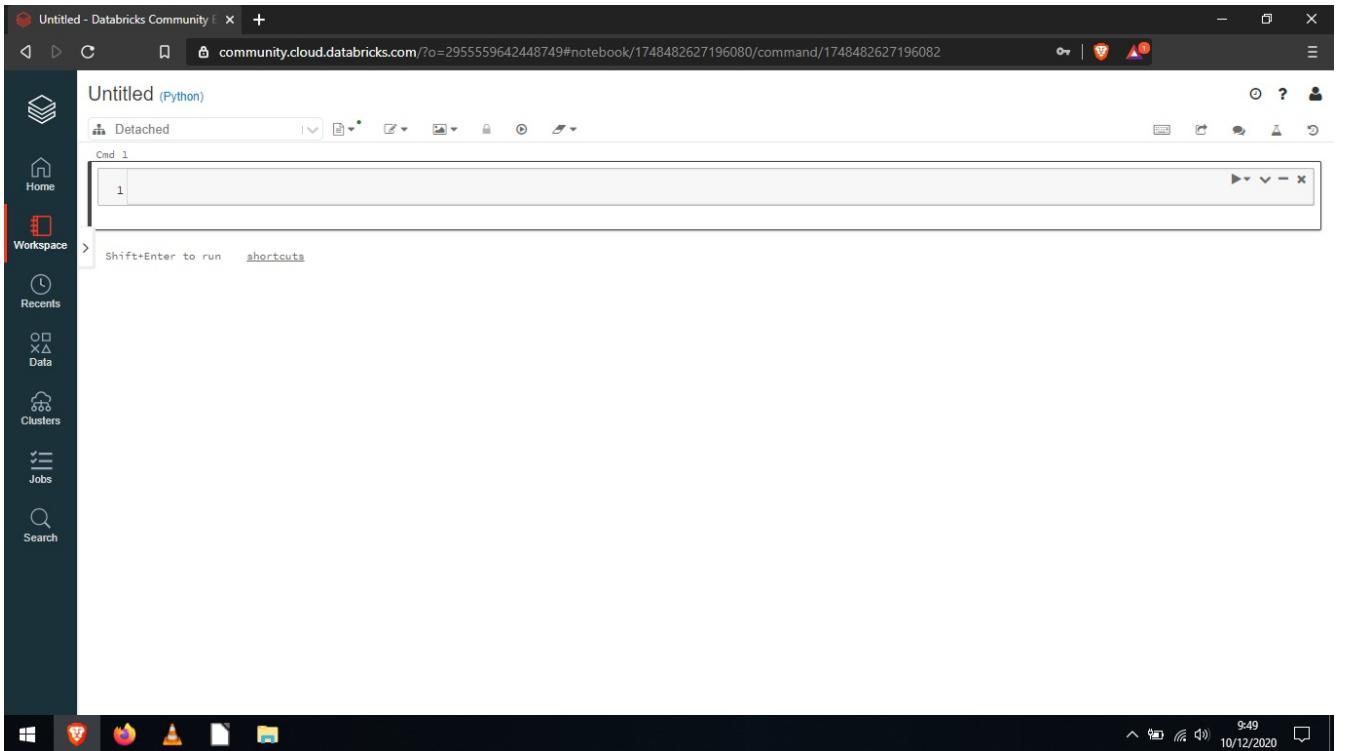
Ahora para poder trabajar con nuestro cluster tendremos que ir a la sección 'workspace' en el menu lateral. En el como podemos ver en las siguientes imagenes, podremos crear un nuevo notebook o importar uno.



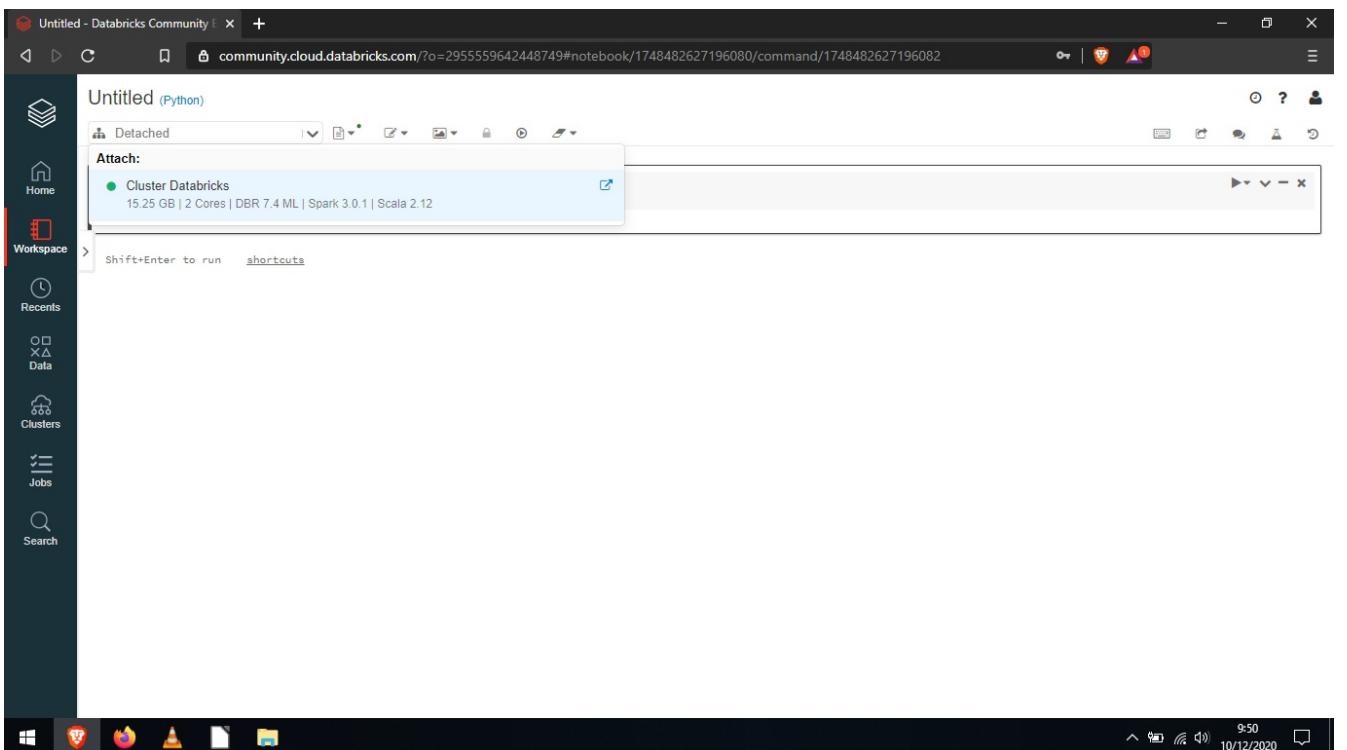
Si le damos a importar se nos abrirá una ventana donde podremos arrastrar el documento o seleccionarlo dentro de nuestro ordenador:



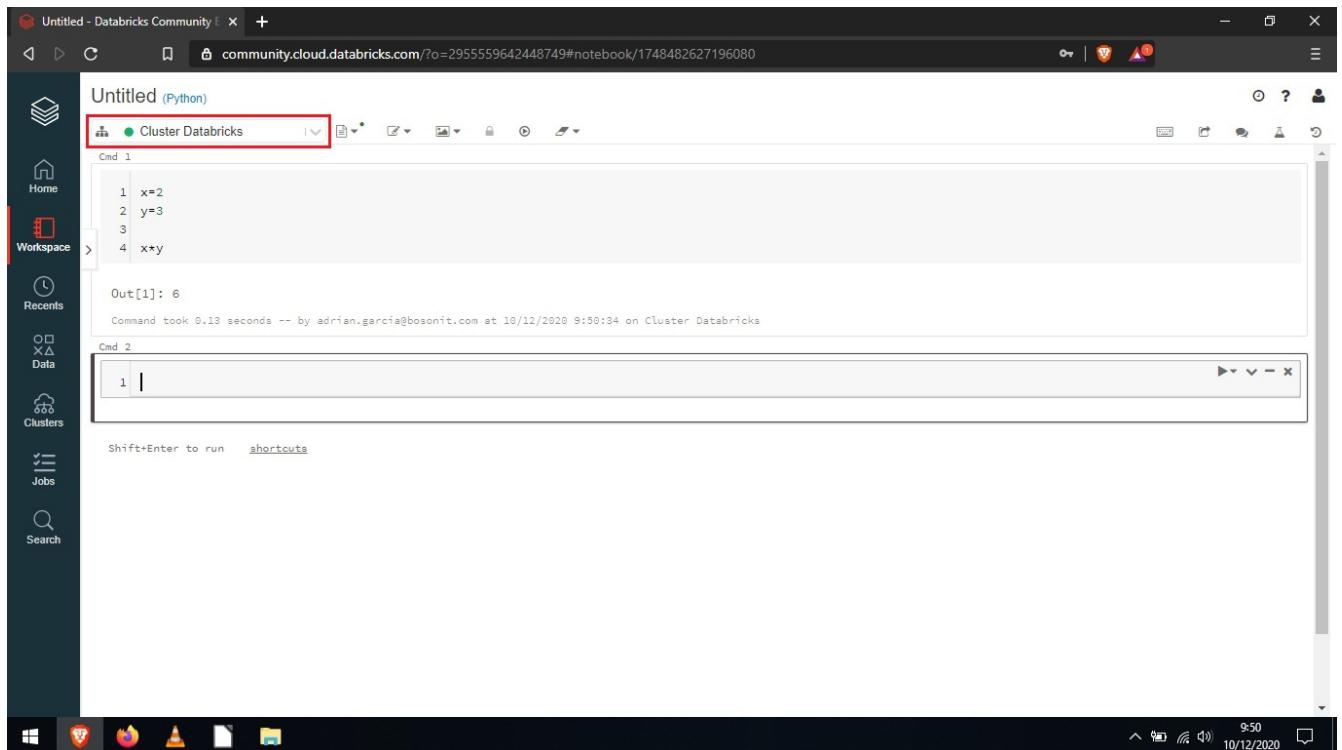
Entonces se nos abrirá el notebook de la siguiente forma:



En la parte superior podremos seleccionar el cluster donde queremos que se ejecute el notebook, en nuestro caso solo tendremos una opcion:



Ahora que hemos seleccionado el cluster podemos comprobar que todo funciona como si estuvieramos programando en nuestro ordenador personal, solo que ahora el procesamiento se esta ejecutando remotamente en un cluster externo.



5.2. Google Cloud

De la misma forma que databricks nos ofrece el procesamiento en la nube. Google tambien dispone de su propia plataforma. Aunque tiene multitud de herramientas para distintos desarrollos y funciones diferentes, nos centraremos en como crear un almacenamiento permanente, inicializacion de cluster y creacion de notebooks.

5.2.1. Registro

Google Cloud nos ofrece con nuestra cuenta de gmail la opcion de probar todas las funciones de Google Cloud durante 1 año o hasta 300\$ de gasto. En este tipo de servicios, los gastos son en funcion de la utilizacion de recursos tanto de procesamiento como de almacenamiento. Para poder aprender a utilizar la plataforma y familiarizarnos con todas sus funciones la version de prueba nos sobra, ya que consumiremos antes el año que los 300\$.

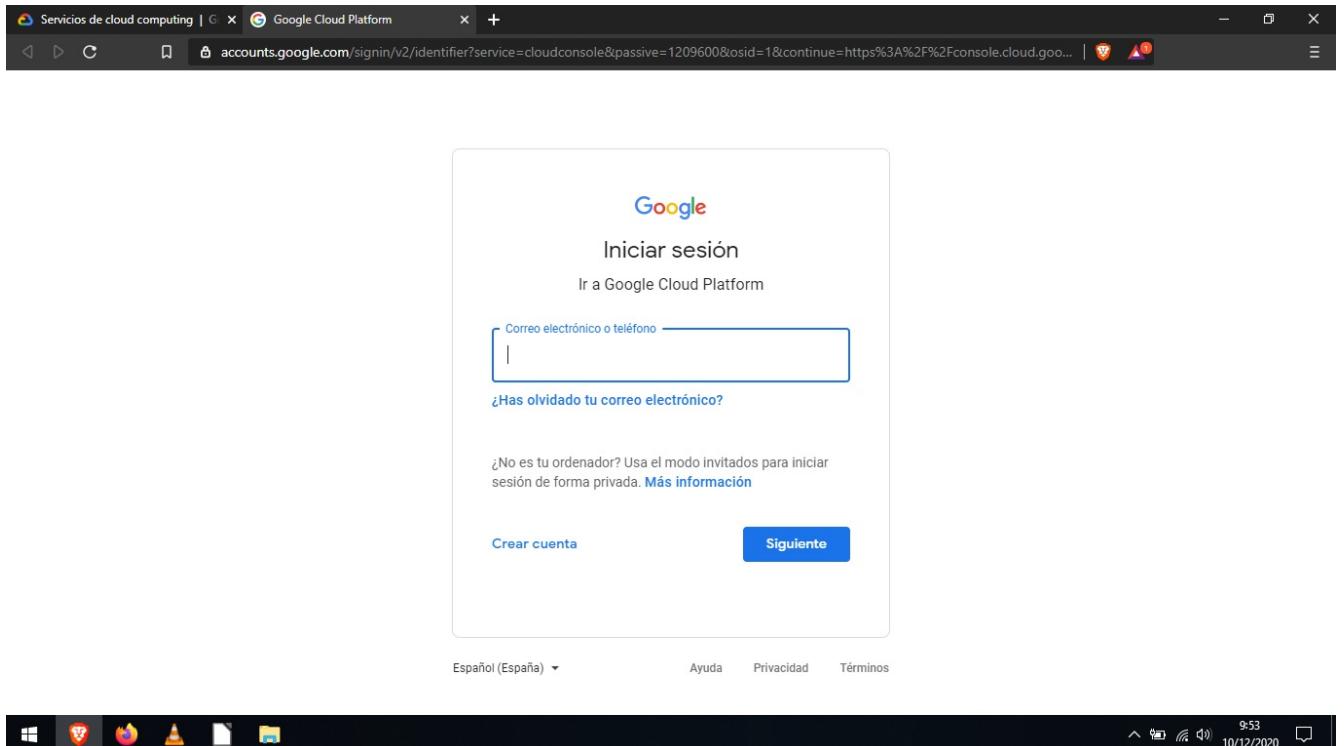
Lo primero que tendremos que hacer sera registrarnos si no tenemos cuenta de Google o crear una. Una vez la tengamos iremos a la siguiente pagina web:

<https://cloud.google.com/>

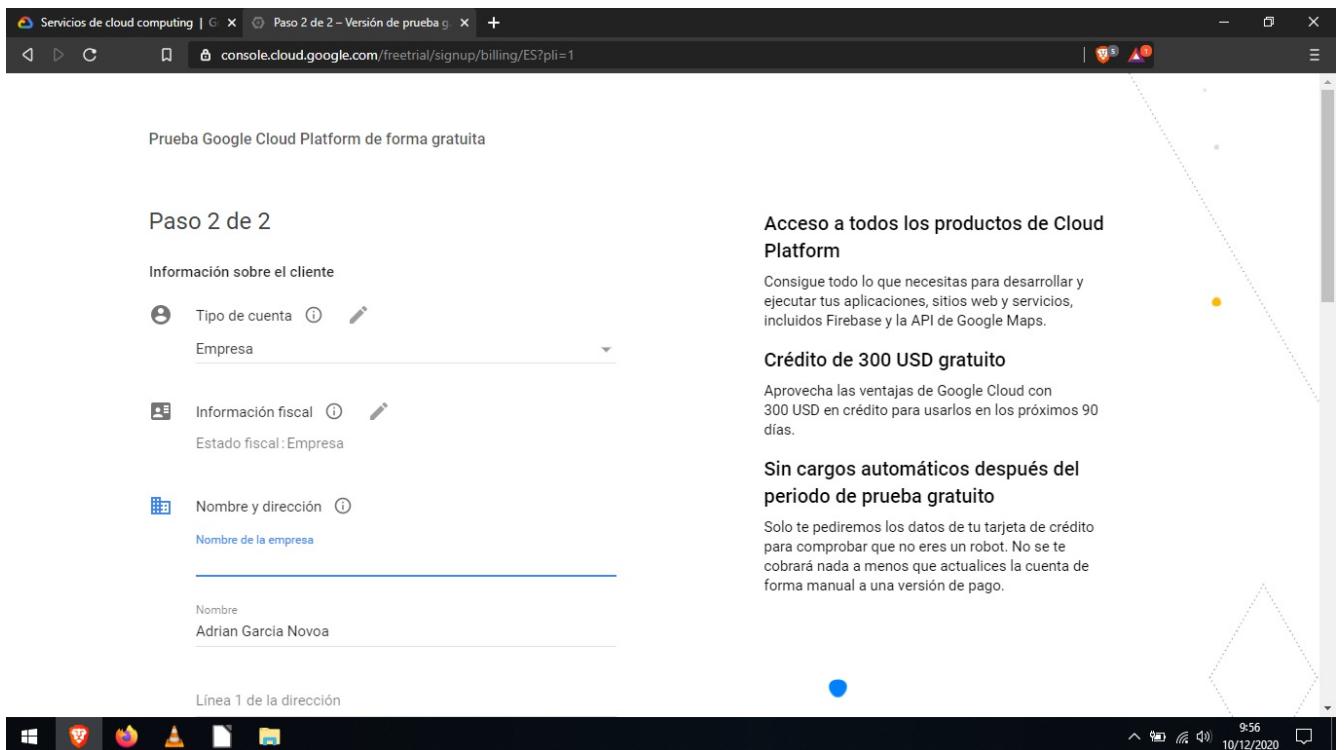
La pagina que nos aparecera sera la siguiente:

The screenshot shows the Google Cloud homepage. At the top, there's a navigation bar with links for 'Por Qué Elegir Google', 'Soluciones', 'Productos', 'Precios', and 'Primeros Pasos'. Below the navigation is a search bar and a language dropdown set to 'Español'. On the right side of the header are 'Documentos', 'Asistencia', 'Language', and 'Acceder' buttons. A prominent blue button labeled 'Empezar gratis' is located in the center of the page. Above this button, a yellow banner states: 'Los nuevos clientes reciben 300 USD en crédito gratuito para invertirlos en Google Cloud. Todos los clientes pueden disfrutar del uso gratuito de más de 20 productos. [Ver detalles de la oferta](#)'. The main content area features a large heading 'Google Cloud te ayuda a encontrar soluciones' with a subtext: 'Si tu empresa tiene dificultades, plántales cara con los servicios de cloud computing de Google'. Below this is another 'Empezar gratis' button. To the right, there's a promotional section for the 'Public Sector Summit' scheduled for December 8-9, with a 'Register now' button and icons for various sectors like healthcare and education. At the bottom, there are four service offerings: 'Moderniza tus cargas de trabajo con', 'Protege tus datos con seguridad', 'Toma mejores decisiones con', and 'Usa entornos de nube híbrida y'. The footer contains standard links and a date/time stamp: '10/12/2020 9:52'.

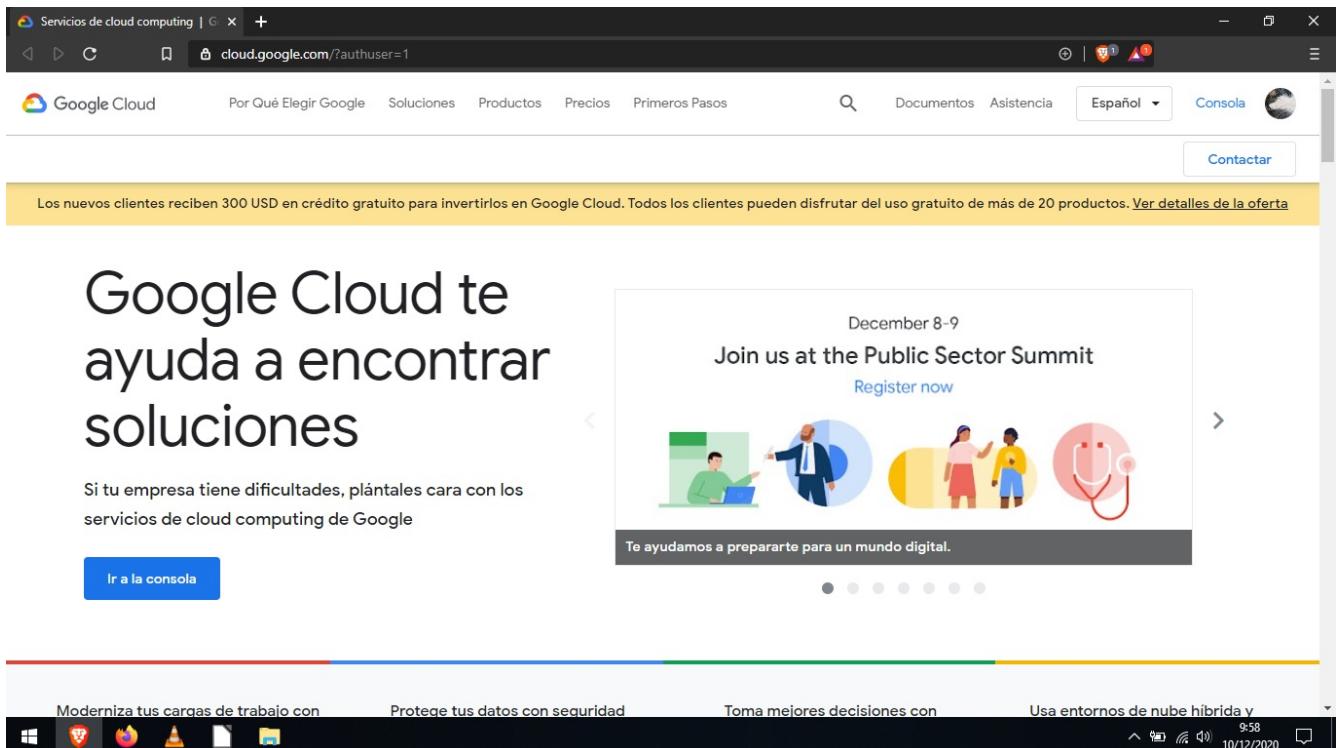
Como podemos observar en la parte superior, ya se nos ofrece el servicio gratuito de prueba. Le damos a 'Empezar gratis' y nos pedira que iniciemos sesion.



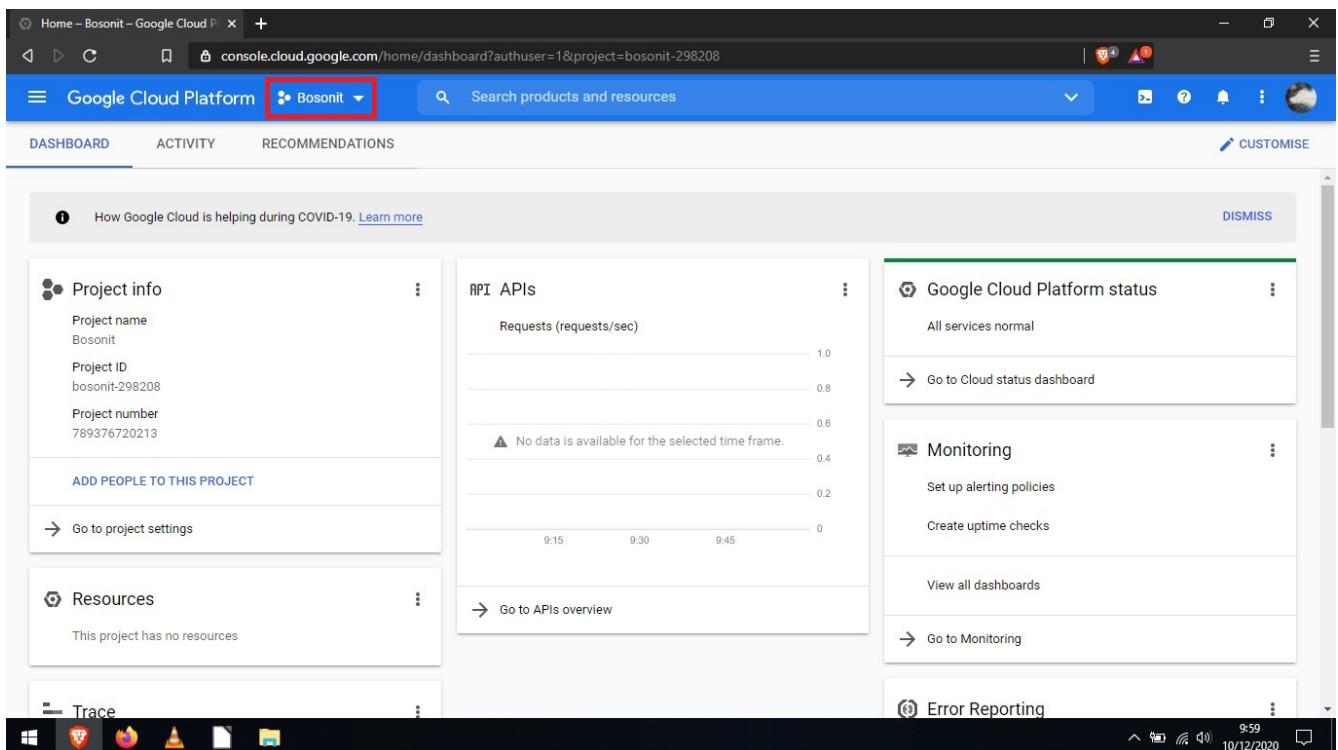
Ademas de aceptar las condiciones del servicio, se nos pedira un numero de tarjeta de credito. En ningun momento se nos cobrara nada, es simplemente como comprobacion de que es una persona fisica la que solicita el permiso y evitar duplicaciones de cuentas. Ademas al terminar el año no se cobrara nada, solo si se hace la actualizacion a version de pago de forma manual y explicita nos mantendran el servicio activo.



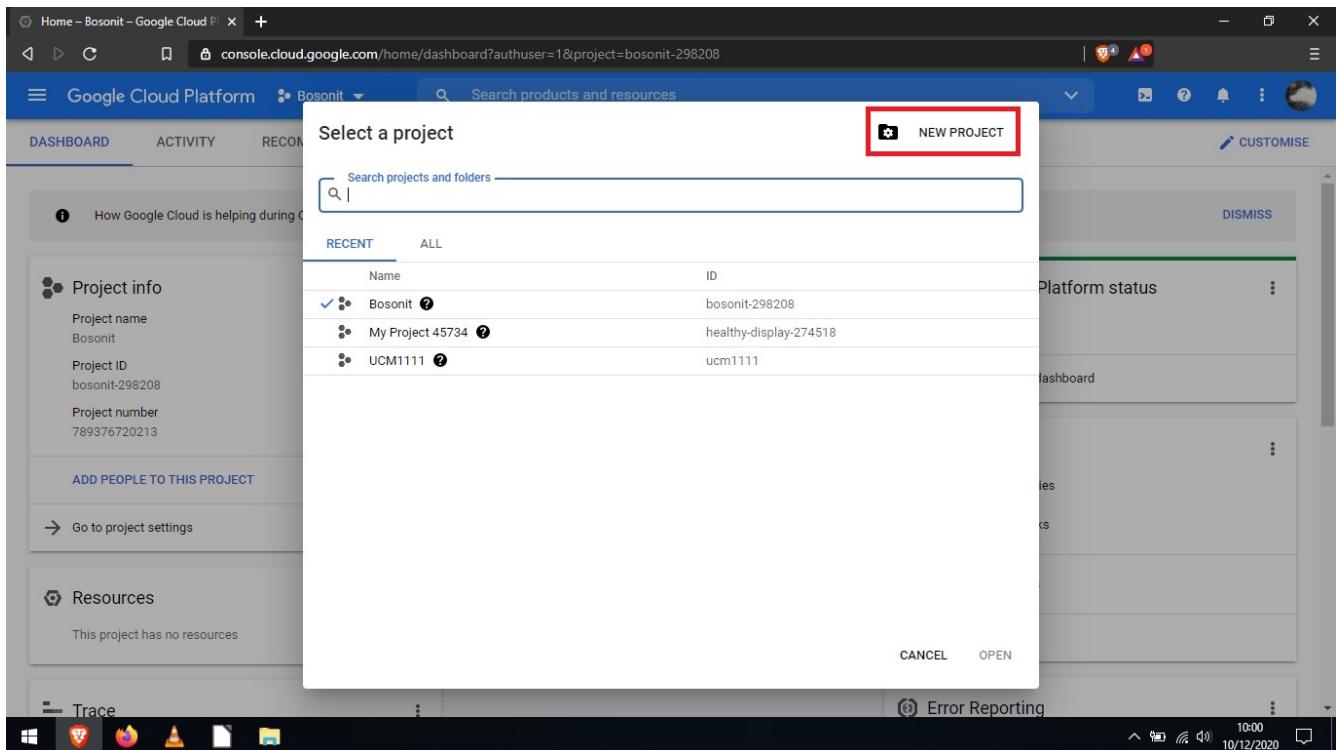
Una vez hemos completado los pasos y hemos iniciado sesion la web principal sera asi:



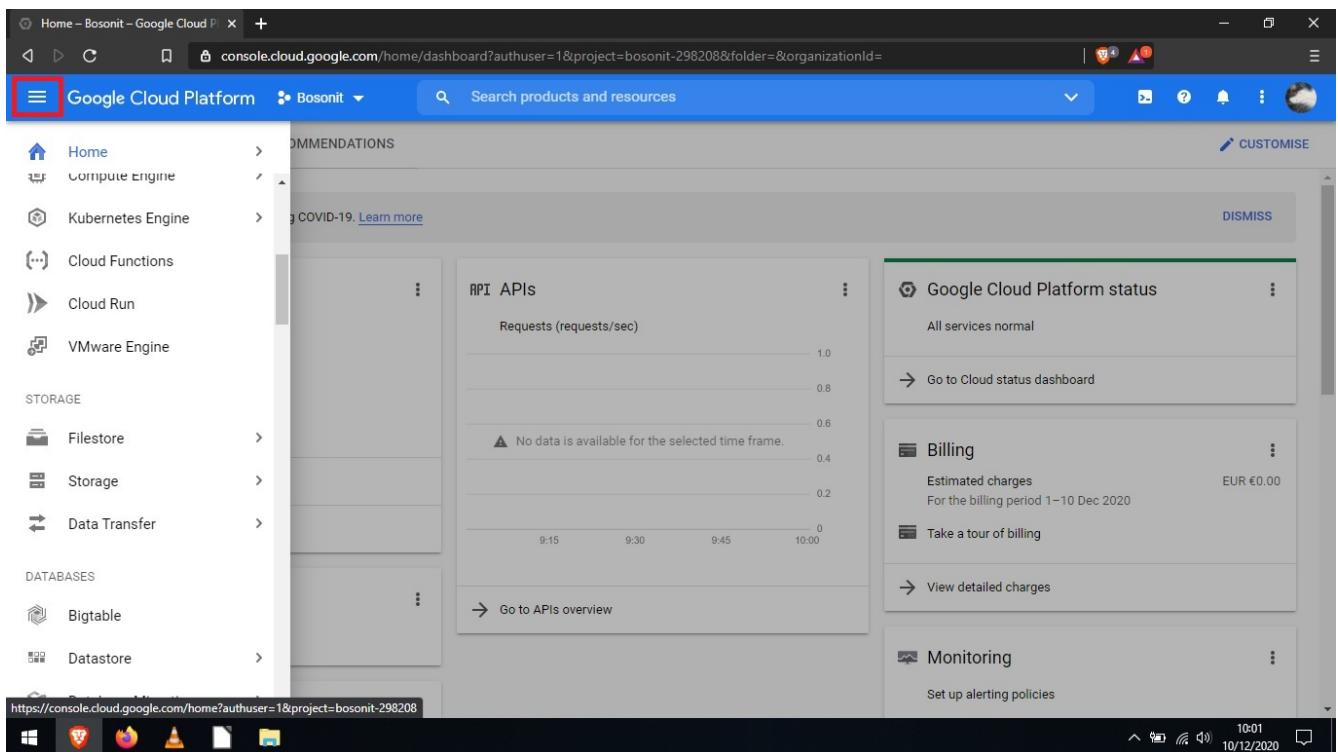
Donde ahora 'Empezar gratis' se ha convertido en 'Ir a la consola'. Hacemos click y nos lleva al Dashboard de la plataforma, donde tendremos que crear un proyecto haciendo click en 'Select project' en la parte superior:



Solo tendremos que elegir un nombre para nuestro proyecto y abrirlo.



Veremos que en la parte izquierda tendremos un menu con muchisimas opciones. Todo eso son diferentes funcionalidades y herramientas de las que dispone la plataforma de Google Cloud.



5.2.2. Storage

Cuando vamos a trabajar con clusters, tenemos que crear un lugar de almacenamiento permanente. Esto se debe a que cuando desplegamos el cluster, este se crea y se mantiene activo el tiempo que nosotros deseemos, pero una vez hemos terminado, el cluster y sus datos internos son eliminados. Para ello crearemos primero un almacenamiento permanente que funciona de forma similar a Google Drive, de esta forma cuando dejemos de trabajar y eliminemos nuestro cluster,

los archivos permaneceran y podremos volver a crear otro cluster y continuar trabajando donde lo habiamos dejado.

Para ello iremos en el menu lateral a donde pone 'Storage' como podemos ver en la siguiente imagen:

The screenshot shows the Google Cloud Platform dashboard. On the left sidebar, under the 'STORAGE' category, the 'Storage' option is selected and has a red circle with a pin icon next to it, indicating it is pinned. A dropdown menu is open over the 'Storage' option, with the 'Browser' item highlighted and also having a red box around it. The main content area displays the 'API APIs' section with a chart showing requests per second over time, and other status cards for Google Cloud Platform status, Billing, and Monitoring.

Recomiendo que hagais click en la chincheta (Pin) para que nos guarde la sección Storage en la parte superior y no tener que buscarlo cada vez que queramos trabajar con el.

Una vez entramos veremos algo asi:

The screenshot shows the 'Storage browser' interface within the Google Cloud Platform. The top navigation bar shows 'Storage browser - Storage'. The main header has a 'CREATE BUCKET' button highlighted with a red box. The left sidebar lists 'Storage', 'Browser', 'Monitoring', and 'Settings'. The central area displays a table for filtering buckets, with a note about sorting and filtering. Below the table is a large graphic of Earth with colored dots. At the bottom, there is a call-to-action for creating a bucket.

Ahora crearemos nuestro Bucket donde almacenaremos los datos que queremos de forma permanente. Se nos ofreceran

diferentes opciones como nombre, region donde queremos que se almacenen los datos, si queremos los datos centralizados o replicados en varias regiones, el tipo de acceso que haremos a el...

En nuestro caso como podemos ver en las siguientes imágenes dejaremos todo por defecto, con las opciones mas simples. Solo modificaremos el apartado 'Location' donde seleccionaremos una region de Europa para minimizar el tiempo de acceso.

Create a Bucket

Name your bucket
Pick a globally unique, permanent name. [Naming guidelines](#)

bosonit

Tip: Don't include any sensitive information

CONTINUE

Choose where to store your data

Choose a default storage class for your data

Choose how to control access to objects

Advanced settings (optional)

CREATE **CANCEL**

Monthly cost estimate

Enter values below to check this bucket's monthly cost. For guidance only. [Pricing details](#)

Storage and retrieval

Storage size: 0.026 GB-month (\$0.026 per GB-month)

Data retrieval size: Free

Operations

Class A operations: \$0.005 per 1,000 ops

Class B operations: \$0.0004 per 1,000 ops

Availability SLA: 99.95%

Create a Bucket

Choose where to store your data

This permanent choice defines the geographic placement of your data and affects cost, performance and availability. [Learn more](#)

Location type

Region Lowest latency within a single region

Dual-region High availability and low latency across 2 regions

Multi-region Highest availability across largest area

Location

us-west4 (Las Vegas)

europe-north1 (Finland)

europe-west1 (Belgium)

europe-west2 (London)

europe-west3 (Frankfurt)

europe-west4 (Netherlands)

europe-west6 (Zurich)

Monthly cost estimate

Enter values below to check this bucket's monthly cost. For guidance only. [Pricing details](#)

Storage and retrieval

Storage size: 0.020 GB-month (\$0.020 per GB-month)

Data retrieval size: Free

Operations

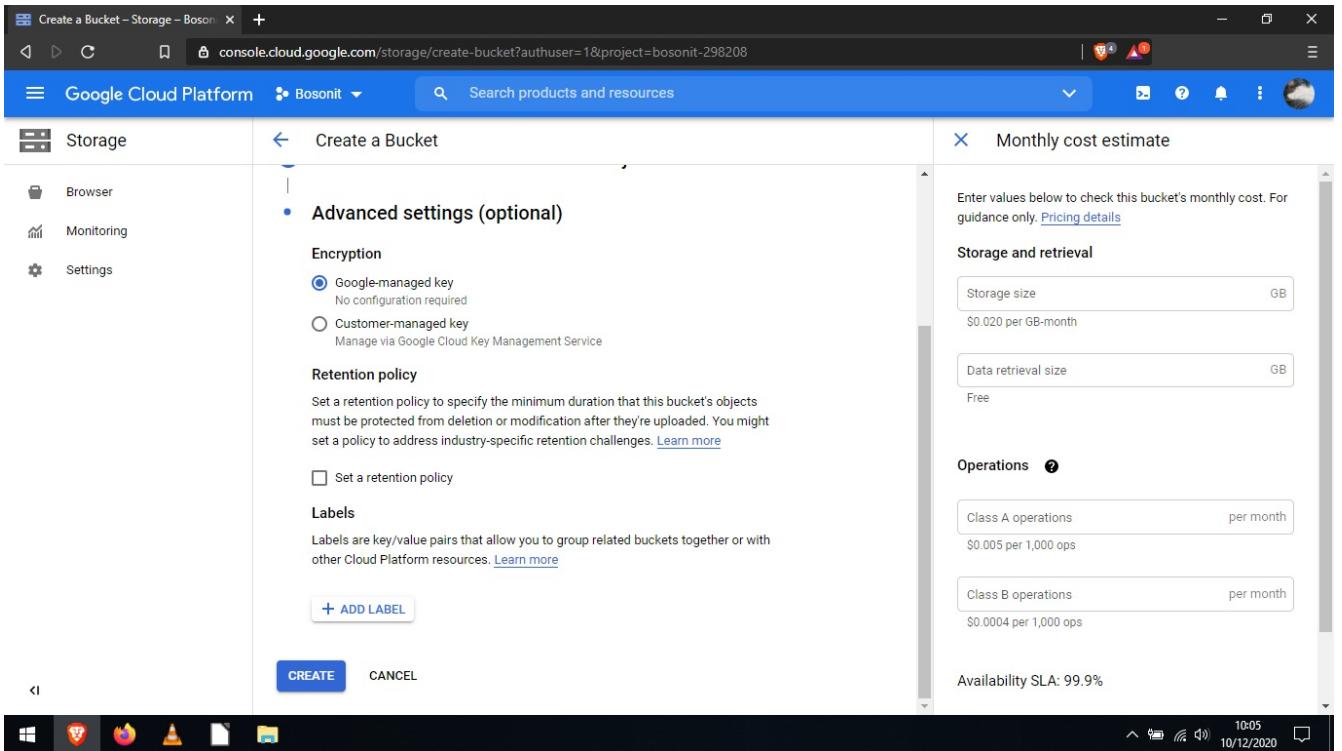
Class A operations: \$0.005 per 1,000 ops

Class B operations: \$0.0004 per 1,000 ops

Availability SLA: 99.9%

The screenshot shows the 'Create a Bucket' wizard on the Google Cloud Platform. The left sidebar shows 'Storage' selected. The main panel is titled 'Create a Bucket' and has a sub-section 'Choose where to store your data'. It includes a list of storage classes: Standard (selected), Nearline, Coldline, and Archive. A note says: 'A storage class sets costs for storage, retrieval and operations. Pick a default storage class based on how long you plan to store your data and how often it will be accessed.' Below this is a 'CONTINUE' button. To the right, there's a 'Monthly cost estimate' section with fields for 'Storage size' (0.020 per GB-month) and 'Data retrieval size' (Free). The bottom right shows system status: 10:04, 10/12/2020.

This screenshot shows the continuation of the 'Create a Bucket' wizard. The left sidebar still shows 'Storage'. The main panel now has a checked box for 'Name your bucket'. Below it, the 'Choose where to store your data' and 'Choose a default storage class for your data' steps also have checked boxes. Under 'Access control', the 'Fine-grained' option is selected, with a note about specifying access to individual objects using object-level permissions (ACLs). A 'CONTINUE' button is present. The right side of the screen remains the same as the previous step, showing monthly cost estimates and system status.



Una vez creado veremos que ahora en la sección principal aparece nuestro Bucket con el nombre que hemos seleccionado.

Name	Created	Location type	Location	Default storage class	Updated	Public access
bosonit	10 Dec 2020, 10:05:11	Region	europe-west1 ...	Standard	10 Dec 2020, 10:05:11	Subject to object ACLs

Y si hacemos click en el nombre nos lleva a la sección donde podremos configurarlo y subir archivos:

The screenshot shows the Google Cloud Platform Storage browser interface. On the left, there's a sidebar with options like Storage, Browser, Monitoring, and Settings. The main area is titled 'Storage browser' and shows a table of buckets. A single row is visible for the bucket 'bosonit', which was created on 10 Dec 2020 at 10:05:11. It is located in the 'europe-west1' region and has a 'Standard' default storage class. The table includes columns for Name, Created, Location type, Location, Default storage class, Updated, and Public access.

5.2.3. Dataproc

Dataproc es un servicio en la nube rápido, fácil de usar y totalmente gestionado para ejecutar clústeres de Apache Spark y Apache Hadoop de una manera rápida y sencilla.

De la misma forma que en la sección anterior, buscaremos en el menú izquierdo la sección 'Dataproc', igual que antes, también recomiendo fijarla por medio de la chincheta.

La primera vez que hagamos click seguramente se nos solicite activar la API, aceptamos y nos llevará a la siguiente pantalla:

The screenshot shows the Google Cloud Platform Marketplace page for the 'Cloud Dataproc API'. At the top, there's a circular icon with a white arrow pointing up and the text 'Cloud Dataproc API' and 'Google'. Below it, a brief description states: 'Manages Hadoop-based clusters and jobs on Google Cloud Platform.' There are two buttons: 'ENABLE' (in blue) and 'TRY THIS API' (in grey). At the bottom of this section, there are tabs for 'OVERVIEW' (which is selected) and 'DOCUMENTATION'. The 'OVERVIEW' section contains an 'Overview' heading and a paragraph about managing Hadoop-based clusters and jobs. It also includes an 'About Google' section with a paragraph about Google's mission. The 'Additional details' section provides information such as Type: APIs & services, Last updated: 10/12/2019, and Service name: dataproc.googleapis.com.

Hacemos click en 'Enable', esperamos y ya podemos continuar en la siguiente pantalla:

Aqui veremos de forma similar a la web de Databricks que sera donde crearemos y gestionaremos el cluster. Le damos a 'Create Cluster' y nos llevara a la pagina de configuracion:

De forma similar al resto de plataformas, se nos solicitara el nombre del cluster, la region en la que queremos que sea desplegado y el tipo de cluster.

Como ya sabemos, existen diferentes tipos de cluster. En nuestro caso seleccionaremos la version de un 1 Master y N Workers.

Tambien se nos da la opcion de activar que autoescale, es decir, que aumente el numero de Workers segun aumente la solicitud de procesamiento. En nuestro caso no va a ser necesario.

Si seguimos bajando veremos el tipo de sistema operativo que queremos que tenga el cluster y podremos desplegar todos los disponibles:

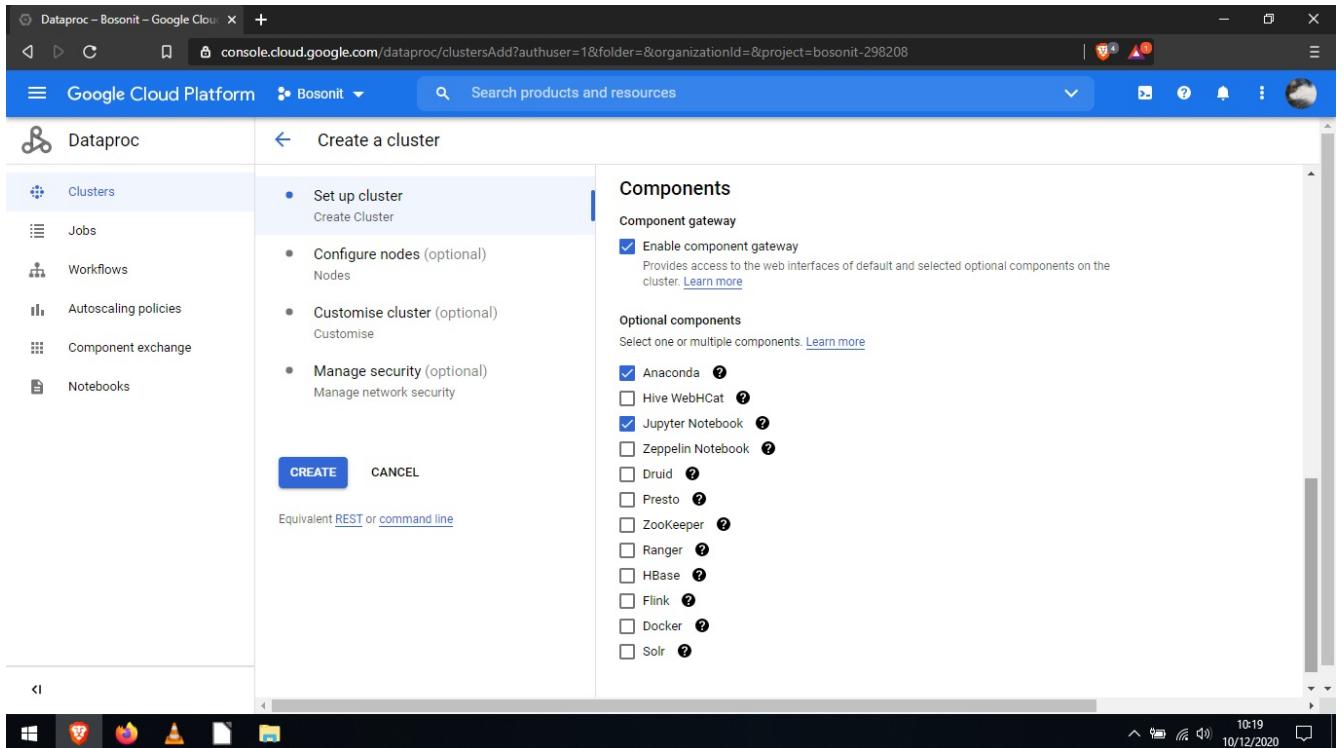
Versioning
Use a custom image to load pre-installed packages. [Learn more](#)
Image Type and Version
1.3-debian10
Release date
First released on 8/16/2018.
Components
Component gateway
 Enable component gateway
Provides access to the web interfaces of default and selected optional components on the cluster. [Learn more](#)
Optional components
Select one or multiple components. [Learn more](#)
 Anaconda [?](#)
 Hive WebHCat [?](#)
 Jupyter Notebook [?](#)
 Zeppelin Notebook [?](#)

Image Version	Description
1.5 (Ubuntu 18.04 LTS, Hadoop 2.10, Spark 2.4)	First released on 25 March 2020.
1.5 (Debian 10, Hadoop 2.9, Spark 2.4)	First released on 25 March 2020.
1.4 (Debian 10, Hadoop 2.9, Spark 2.4)	First released on 22/03/2019.
1.4 (Ubuntu 18.04 LTS, Hadoop 2.9, Spark 2.4)	First released on 22/3/2019.
1.3 (Debian 10, Hadoop 2.9, Spark 2.3)	First released on 8/16/2018.
PREVIEW 2.0 (Ubuntu 18.04 LTS, Hadoop 3.2, Spark 3.0)	Preview released on 6/10/2020.
PREVIEW 2.0 (Debian 10, Hadoop 3.2, Spark 3.0)	Preview released on 6/10/2020.

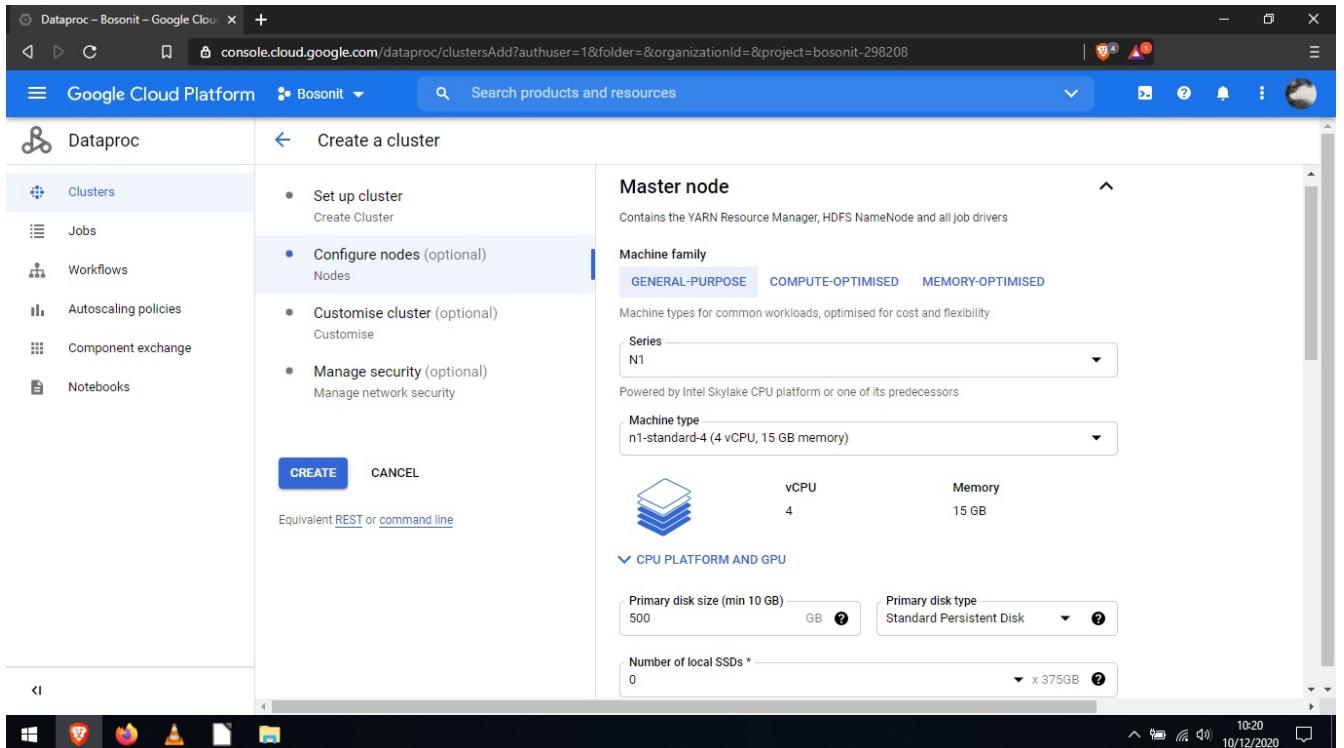
En nuestro caso recomendamos seleccionar la version 1.4 que incluye el sistema operativo Debian 10 con Hadoop 2.9 y Spark 2.4.

En la ultima seccion veremos los componentes que queremos incluir en nuestro cluster, que en nuestro caso hemos

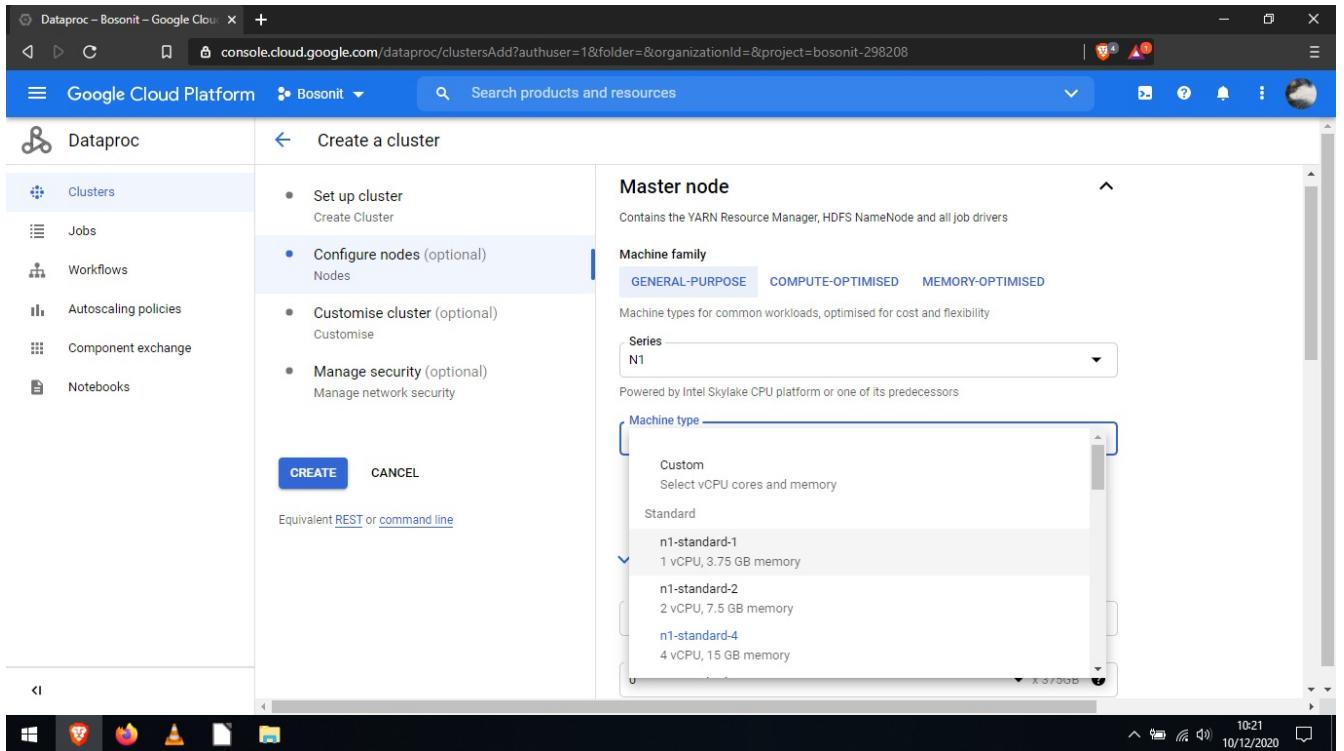
seleccionado Anaconda y Jupiter Notebook. Y no nos olvidemos de marcar 'Enable component gateway' para poder trabajar sin errores en el navegador.



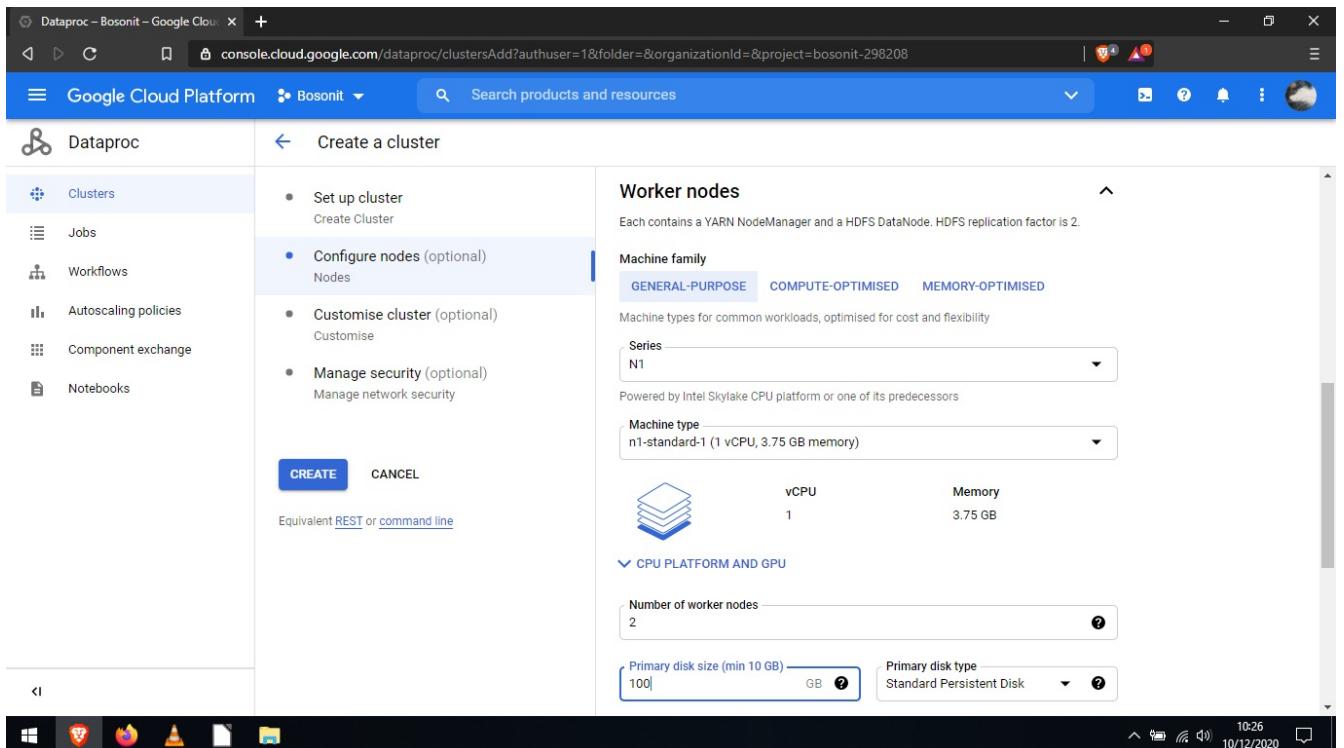
En el segundo apartado de la configuracion veremos y podremos modificar las caracteristicas del Master y los Workers:



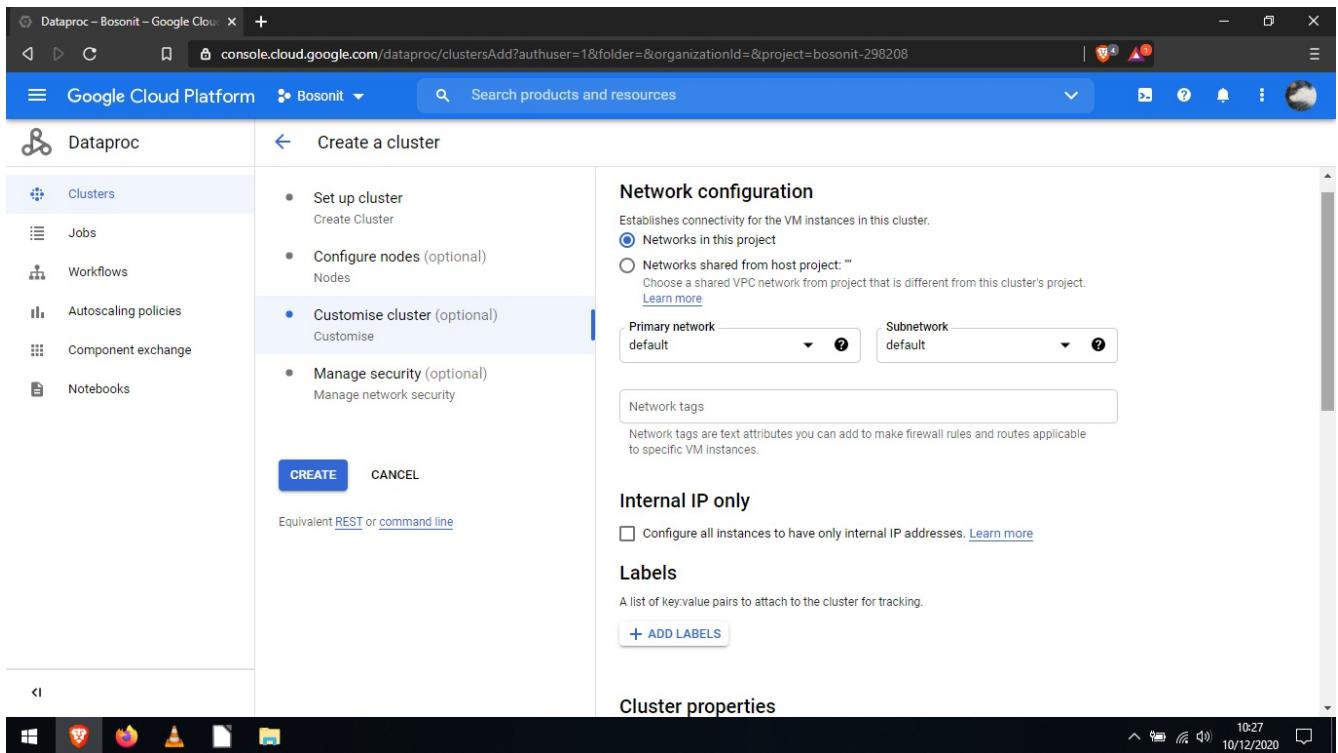
Para el Master vamos a seleccionar el mas pequeño ya que para nuestro aprendizaje no vamos a necesitar una gran cantidad de potencia de calculo y las caracteristicas por defecto en cuanto a tamaño de disco y tipo de disco:



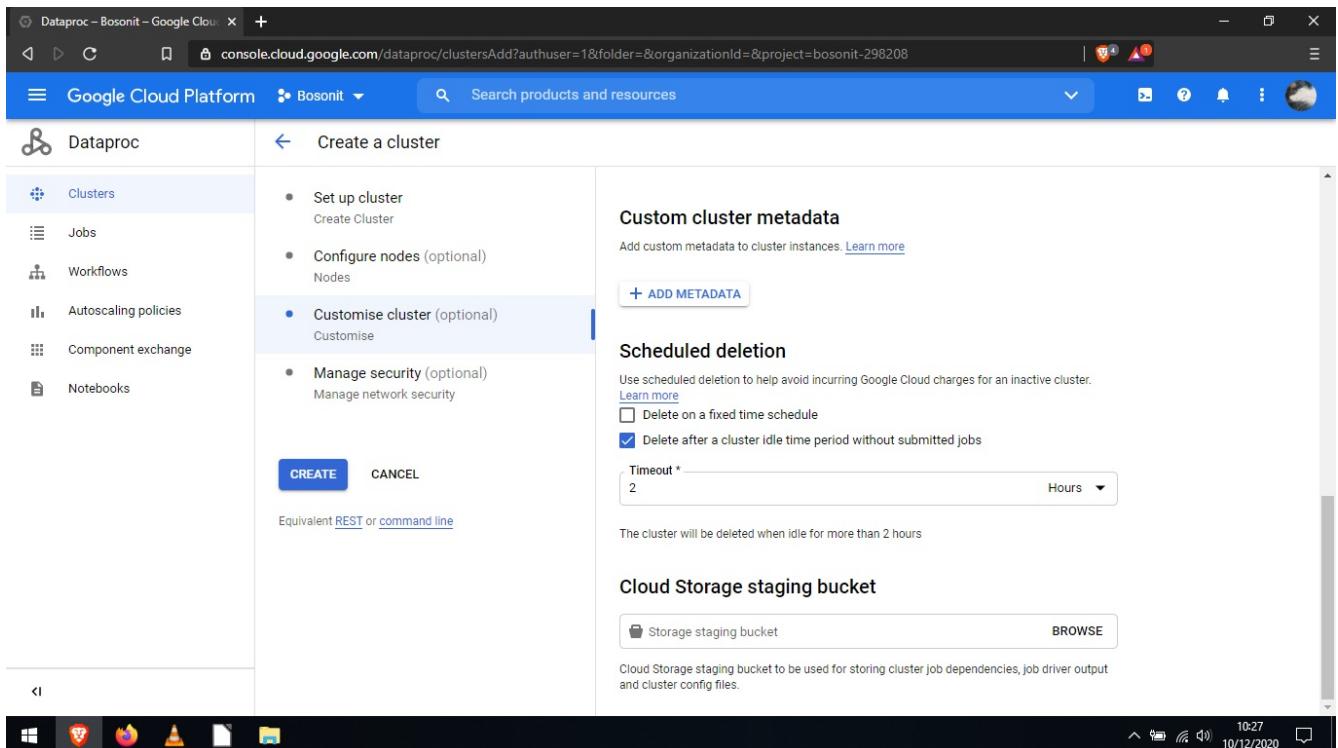
Si bajamos vemos la sección de Workers, donde podremos seleccionar cuantos queremos, la capacidad de cada uno de ellos y el tamaño de disco y tipo de disco. En mi caso como en el Master selecciono el de menos potencia y pongo el tamaño de los discos a 100. De todas formas se puede dejar por defecto sin ningún problema.



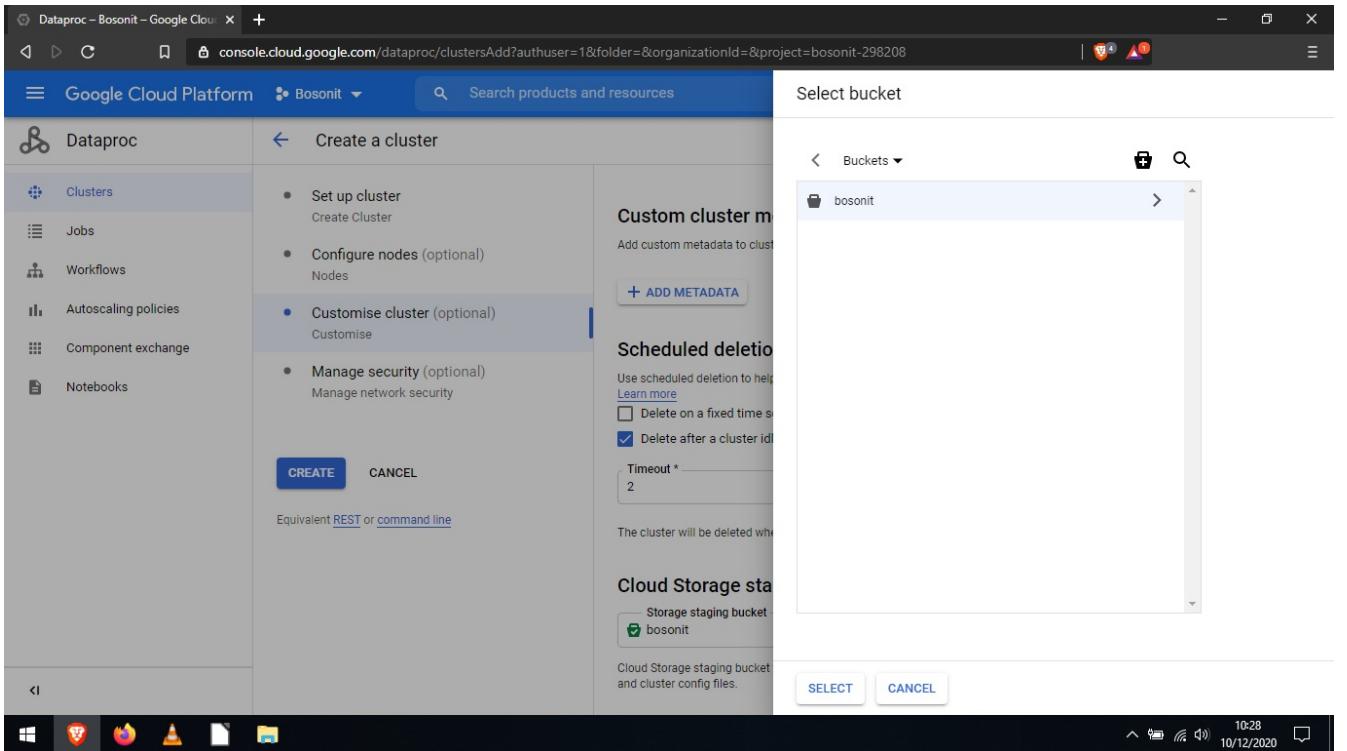
En la tercera sección vamos a dejar la primera parte de la siguiente manera por defecto:



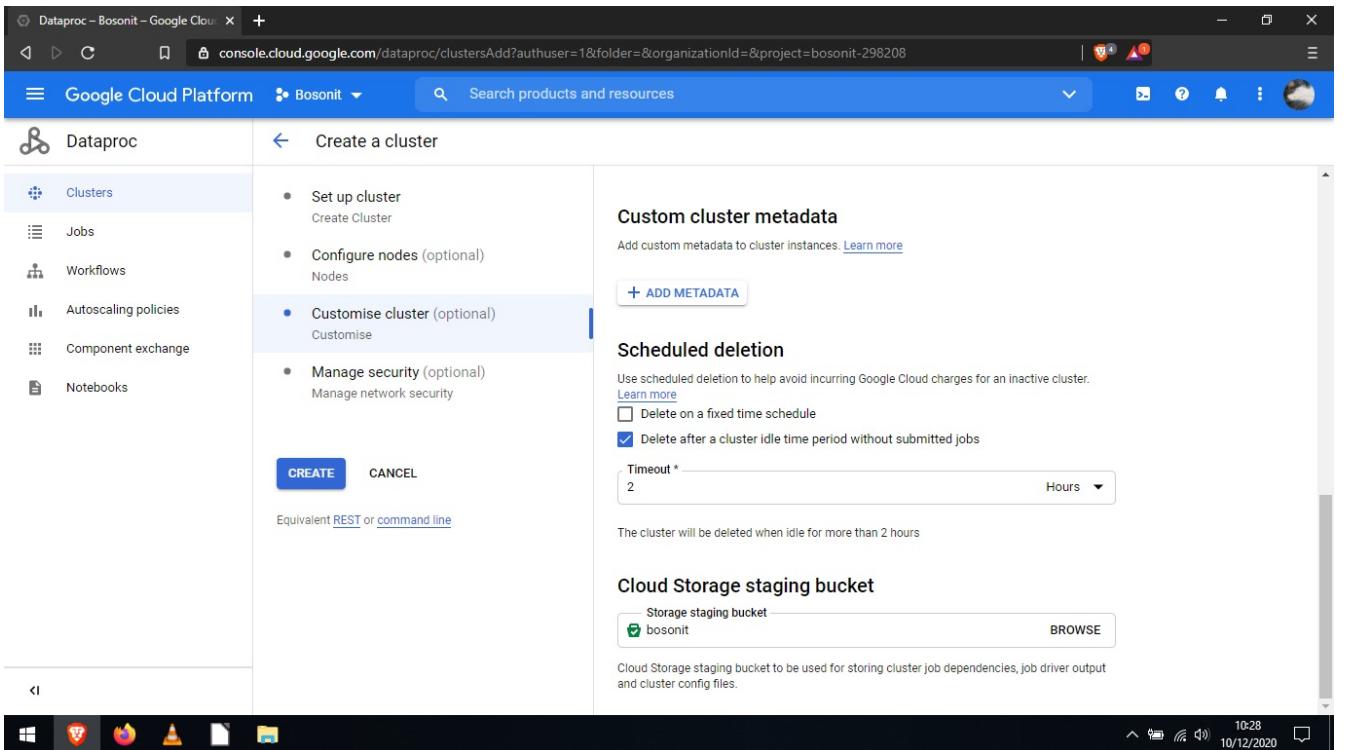
Y en la parte de abajo debemos fijarnos en activar la función de apagado tras un periodo de inactividad (2h en mi caso) para que el cluster se elimine cuando dejemos de usarlo, por si se nos olvida apagarlo.



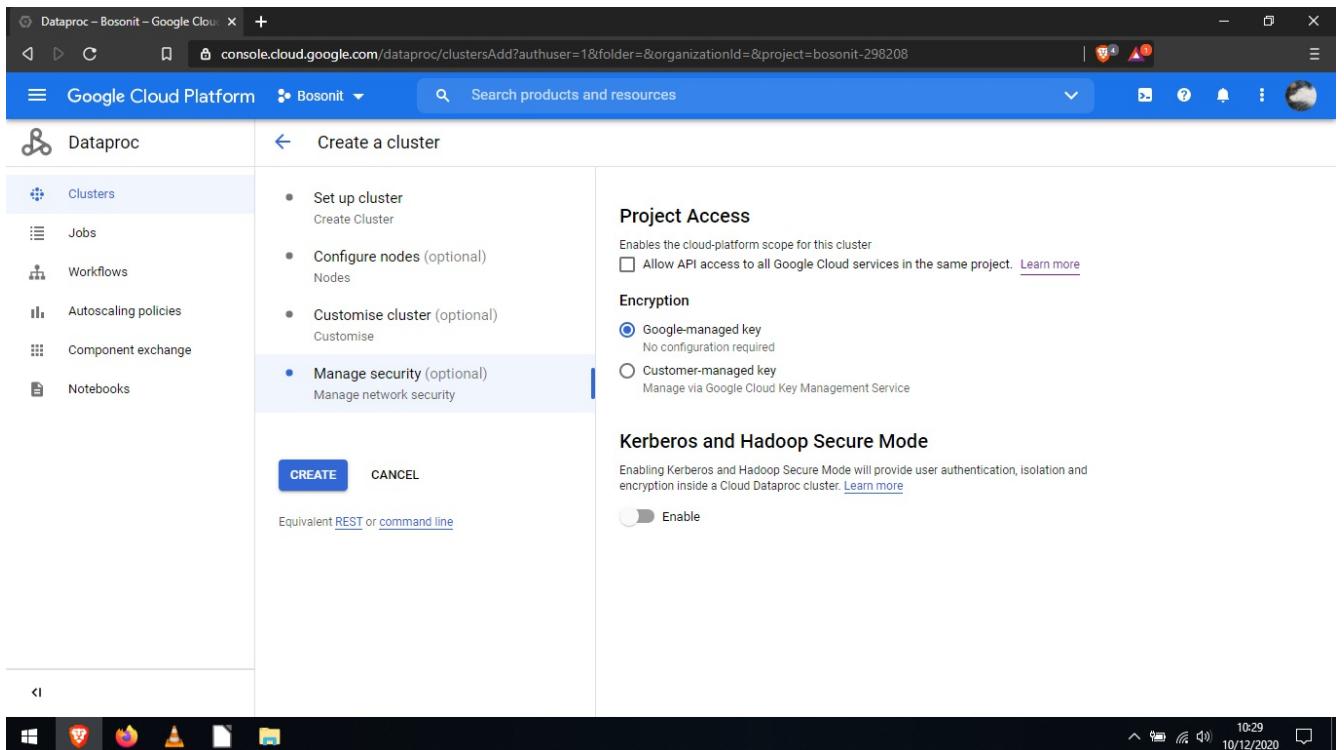
Pero aun mas importante es seleccionar 'Cloud Storage staging Bucket'. Aqui es donde seleccionaremos el Bucket que hemos creado en la sección anterior y donde guardaremos nuestros Dataset y Notebooks para que el cluster pueda trabajar con ellos y queden guardados aunque el cluster sea eliminado. Lo seleccionamos asi:



Y deberia quedarnos algo asi:



En la ultima seccion no tenemos que tocar nada:



Solo darle a 'CREATE' y esperar a que se despliegue ya que suele tardar unos pocos minutos. Una vez ha sido creado nos saldra algo asi:

Name	Region	Zone	Total worker nodes	Scheduled deletion	Cloud Storage staging bucket	Created	Status
europewest2-cluster-13cf	europe-west2	europe-west2-b	2	On	bosonit	10 Dec 2020, 10:29:37	Running

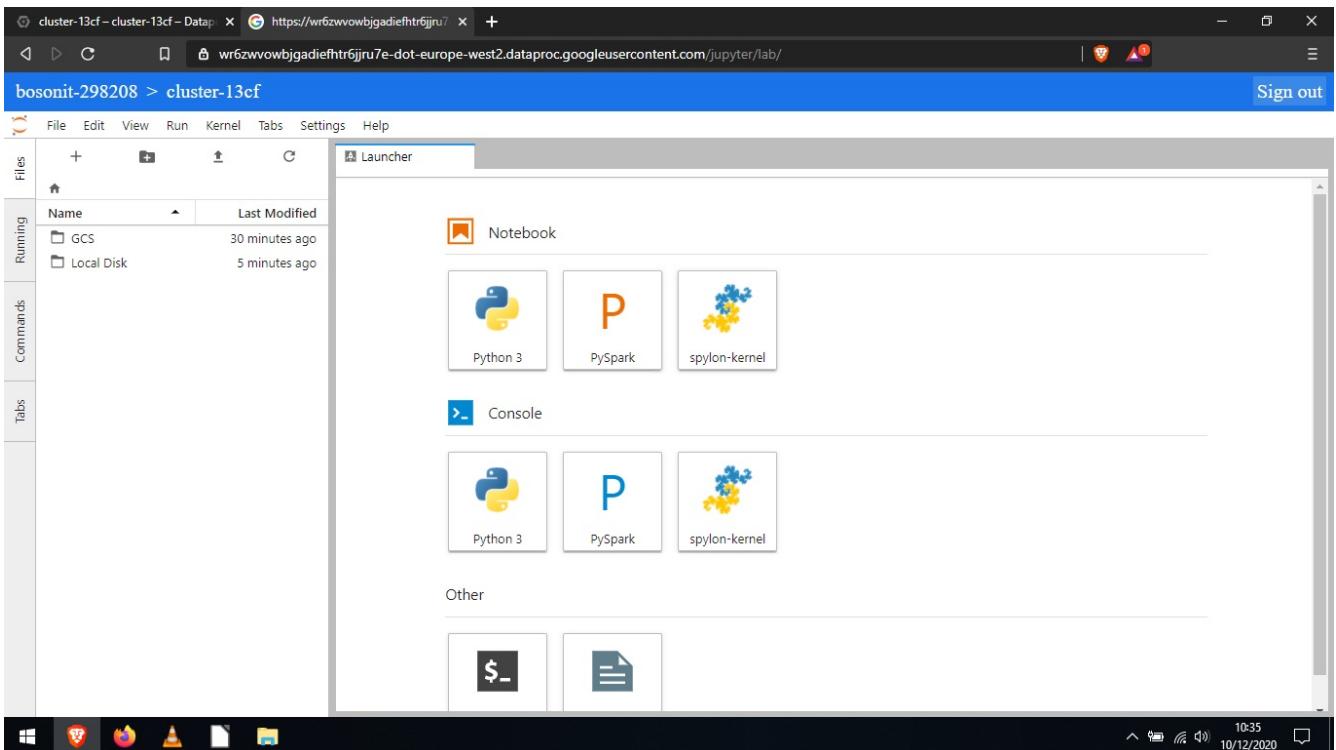
Si hacemos click en el nombre del Cluster nos llevara a la informacion general donde podremos ver la monitorizacion de los procesos y usos del cluster.

The screenshot shows the Google Cloud Platform DataProc cluster monitoring interface for 'cluster-13cf'. The left sidebar has 'Clusters' selected. The main area displays cluster details: Name (cluster-13cf), Cluster UUID (b47d9b55-d60a-4c01-a085-3ce3e4a634f9), Type (DataProc cluster), and Status (Running). Below this are tabs for MONITORING, JOBS, VM INSTANCES, CONFIGURATION, and WEB INTERFACES. The MONITORING tab is active, showing 'YARN memory' usage from 9:45 to 10:30. The YARN pending memory chart shows values: 953.67 MB, 762.94 MB, 572.2 MB, 381.47 MB, and 190.73 MB. The YARN memory chart shows 0 B. A message at the top says: 'Creating clusters using the n1-standard-1 machine type is not recommended. Consider using a machine type with higher memory.' A 'MORE' link is also present.

Pero lo importante es la sección 'WEB INTERFACES' donde accederemos a los componentes instalados como Jupiter Notebook.

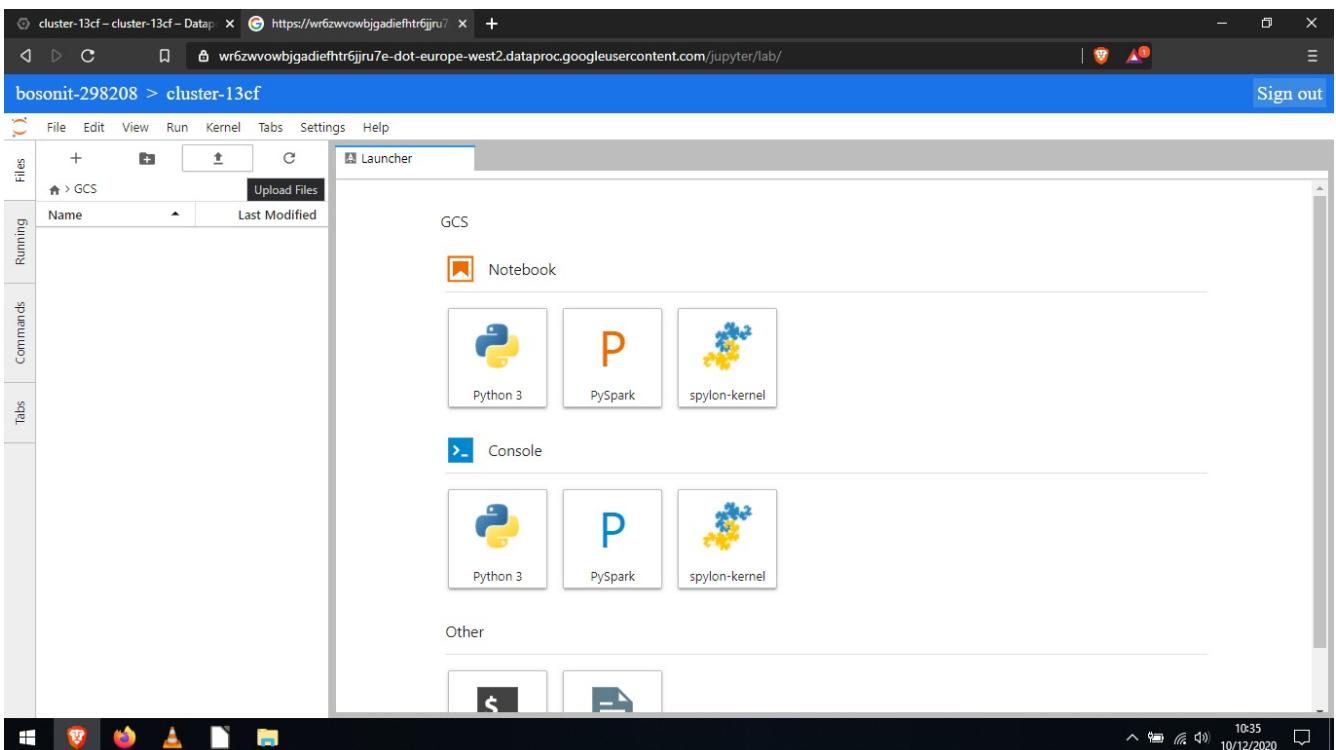
The screenshot shows the Google Cloud Platform DataProc cluster interfaces interface for 'cluster-13cf'. The left sidebar has 'Clusters' selected. The main area displays cluster details: Status (Running). Below this are tabs for MONITORING, JOBS, VM INSTANCES, CONFIGURATION, and WEB INTERFACES. The WEB INTERFACES tab is active, showing a list of available web interfaces: SSH tunnel, Component gateway, YARN ResourceManager, MapReduce Job History, YARN Application Timeline, Spark History Server, HDFS NameNode, Tez, Jupyter, JupyterLab, and Equivalent REST. Each item has a small icon and a 'Learn more' link.

Hacemos click en 'JupyterLab' y veremos lo siguiente:

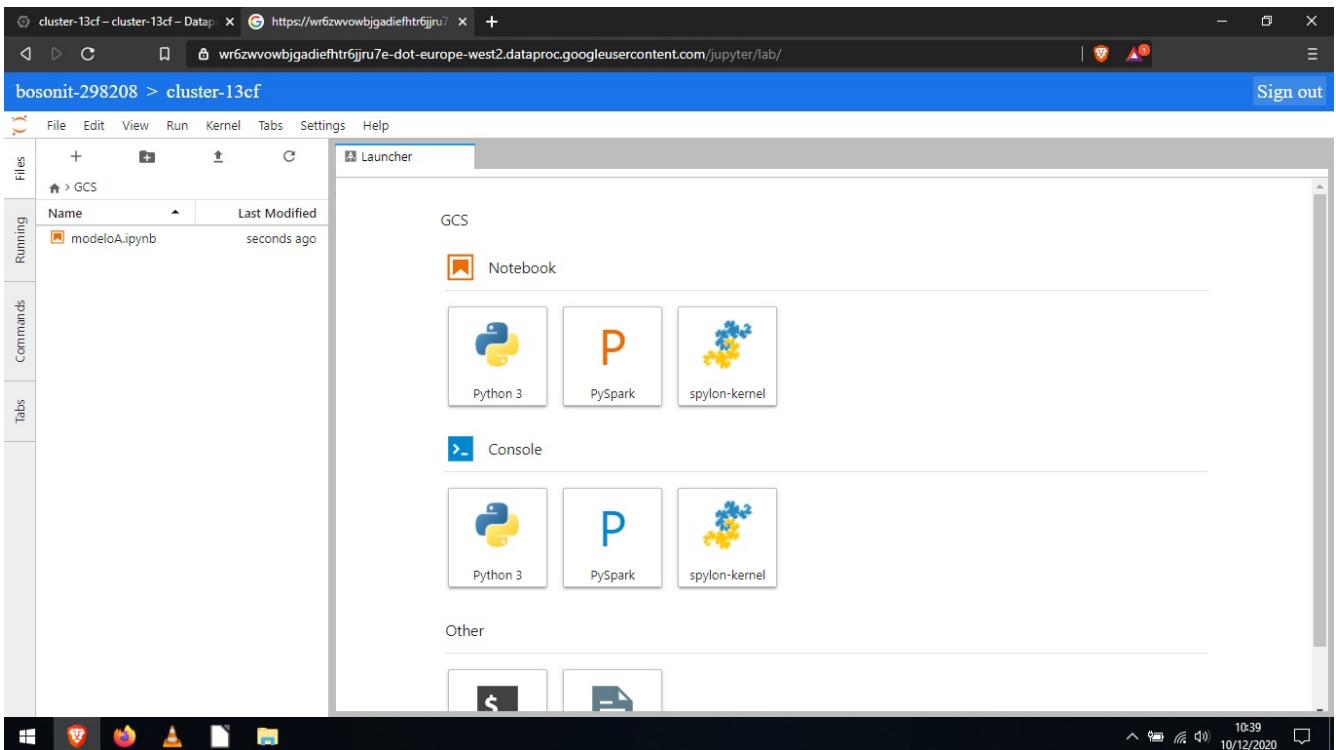


En la parte izquierda veremos dos carpetas, una GCS que es 'Google Cloud Storage' que se trata del almacenamiento permanente del que hablamos y Local que es el almacenamiento del propio cluster donde si entramos podremos ver los diferentes archivos del sistema operativo y demás del propio cluster. Esta carpeta Local es la que se eliminará al borrar el Cluster, por eso nos interesa almacenar todo en GCS.

Si entramos en la carpeta GCS podremos subir un notebook de prueba:

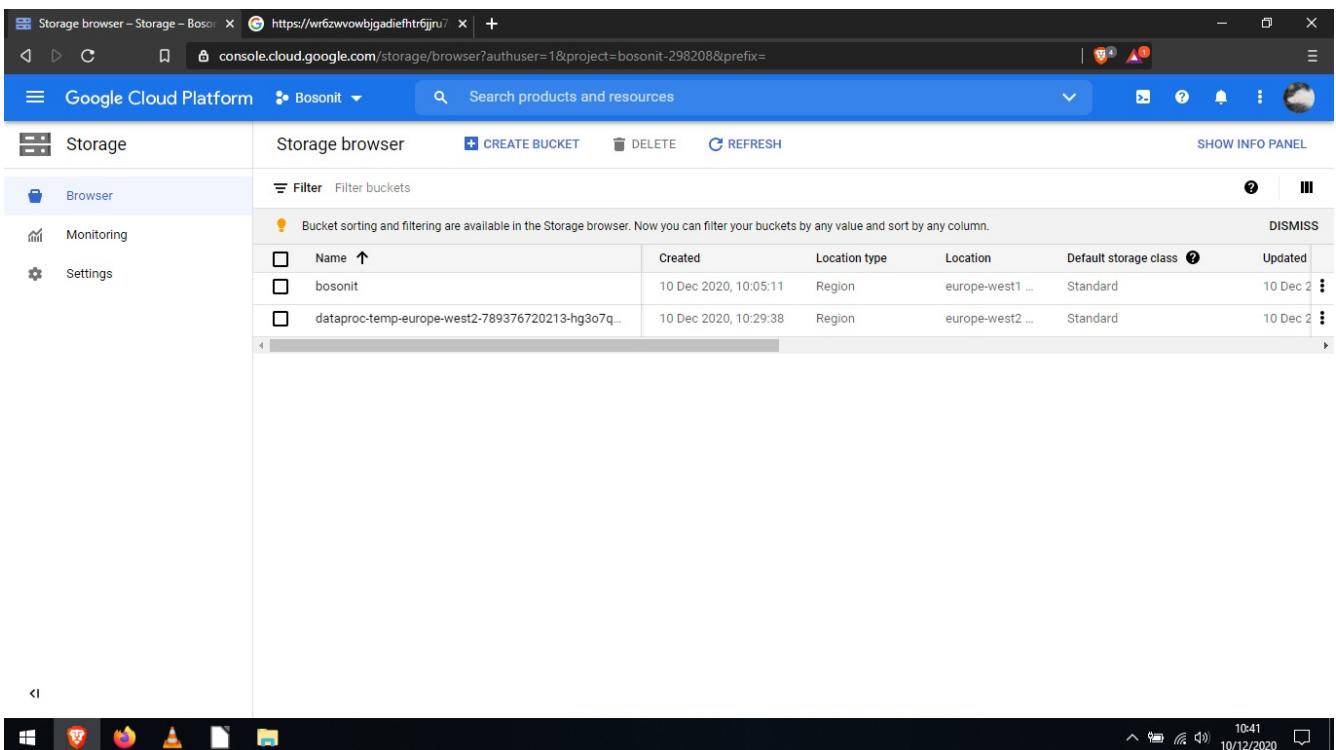


Y quedará de la siguiente manera:

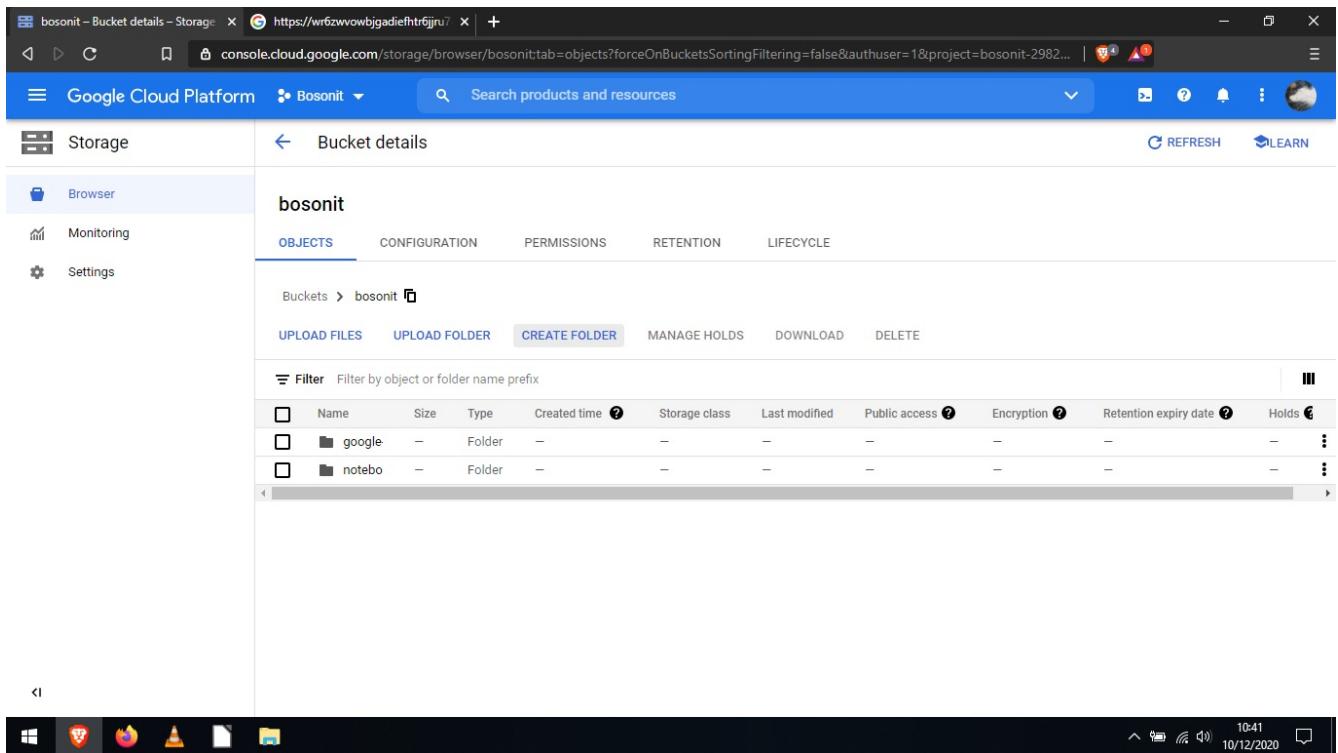


Este archivo quedara guardado de forma permanente en Storage hasta que lo eliminemos manualmente, independientemente de que el cluster este activo o no.

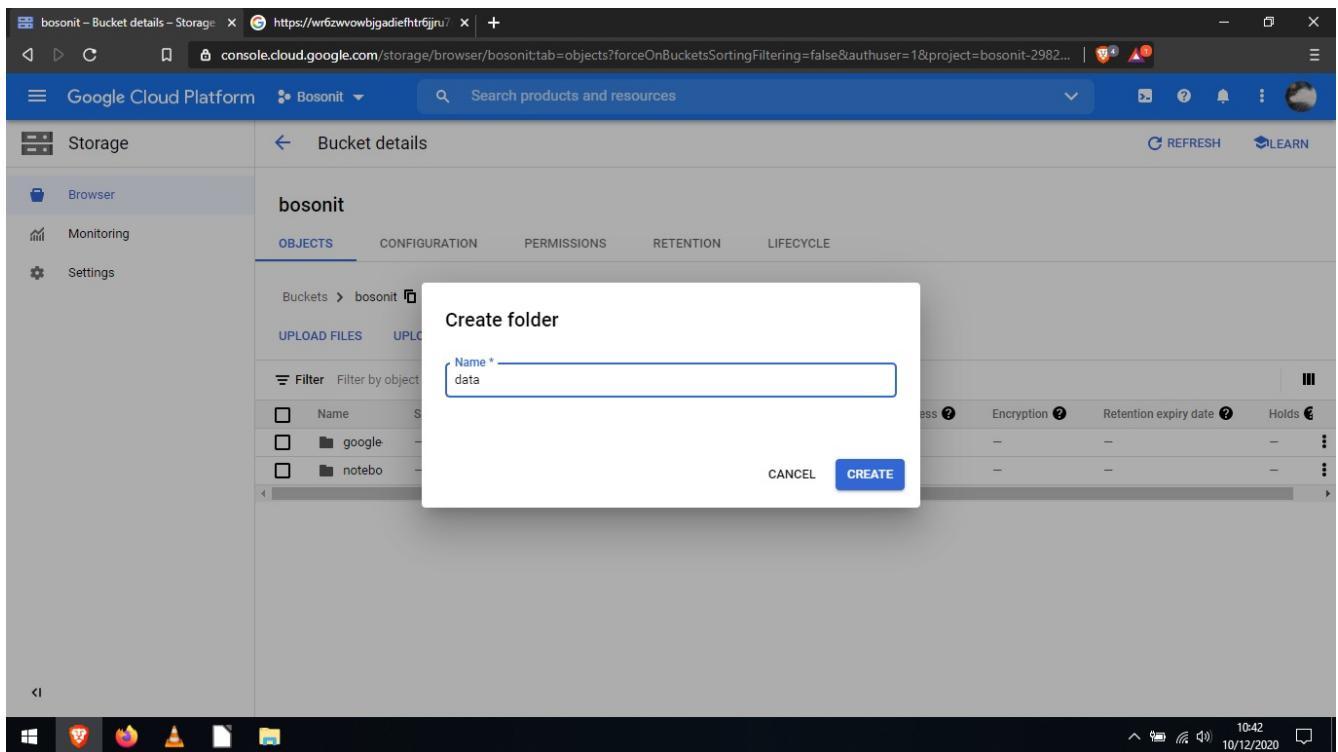
Ahora vamos a subir un Dataset para poder comprobar que todo funciona correctamente. Si intentamos subirlo desde aqui nos dira que solo podemos subir archivos con un tamaño maximo de 15Mb por lo tanto vamos a volver a la seccion 'Storage':

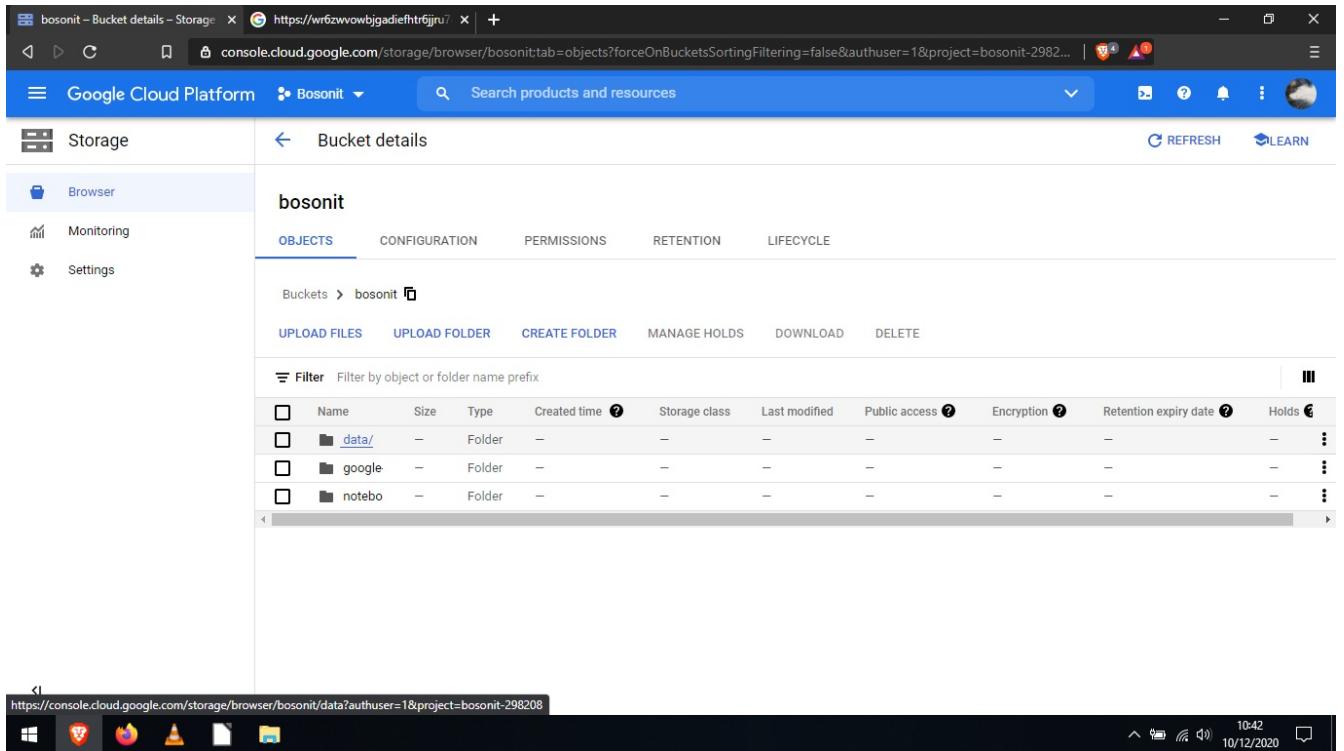


Entramos en la carpeta 'bosonit' que habiamos creado y veremos:



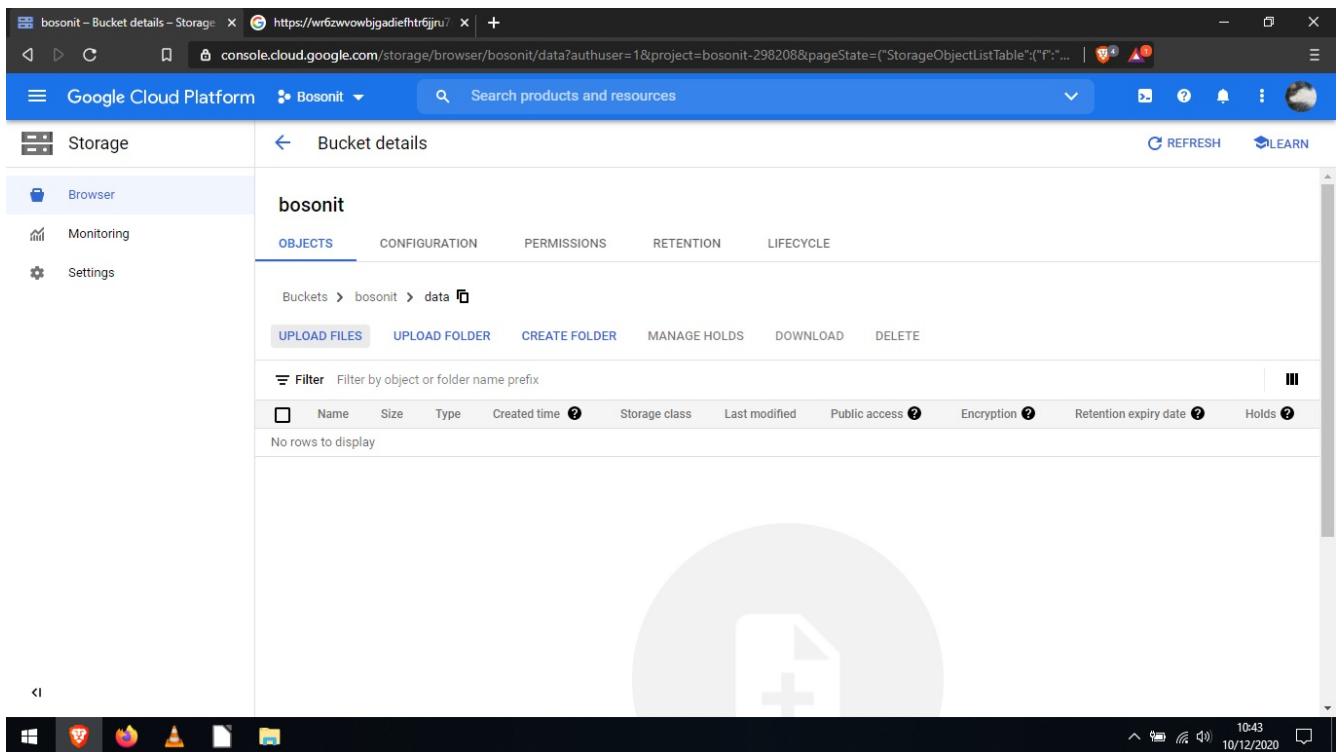
Crearemos una carpeta dandole a 'Create Folder' y le pondremos de nombre data:



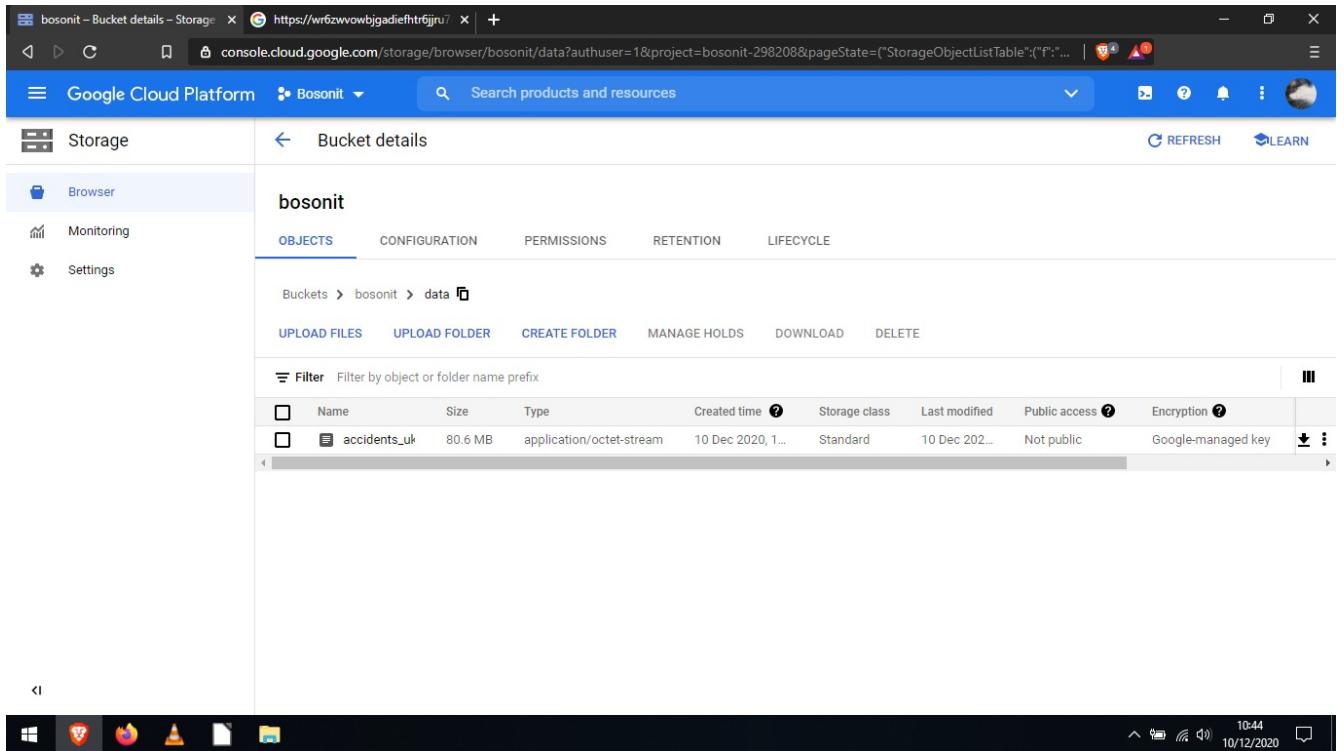


The screenshot shows the Google Cloud Platform Storage interface. A sidebar on the left lists 'Storage', 'Browser', 'Monitoring', and 'Settings'. The main area is titled 'Bucket details' for 'bosonit'. Below this, there are tabs for 'OBJECTS', 'CONFIGURATION', 'PERMISSIONS', 'RETENTION', and 'LIFECYCLE'. Under 'OBJECTS', it says 'Buckets > bosonit'. There are buttons for 'UPLOAD FILES', 'UPLOAD FOLDER', 'CREATE FOLDER', 'MANAGE HOLDS', 'DOWNLOAD', and 'DELETE'. A filter bar allows filtering by object or folder name prefix. A table lists objects with columns: Name, Size, Type, Created time, Storage class, Last modified, Public access, Encryption, Retention expiry date, and Holds. Three entries are shown: 'data/' (Folder), 'google' (Folder), and 'notebo' (Folder). The 'data/' entry is highlighted with a blue background.

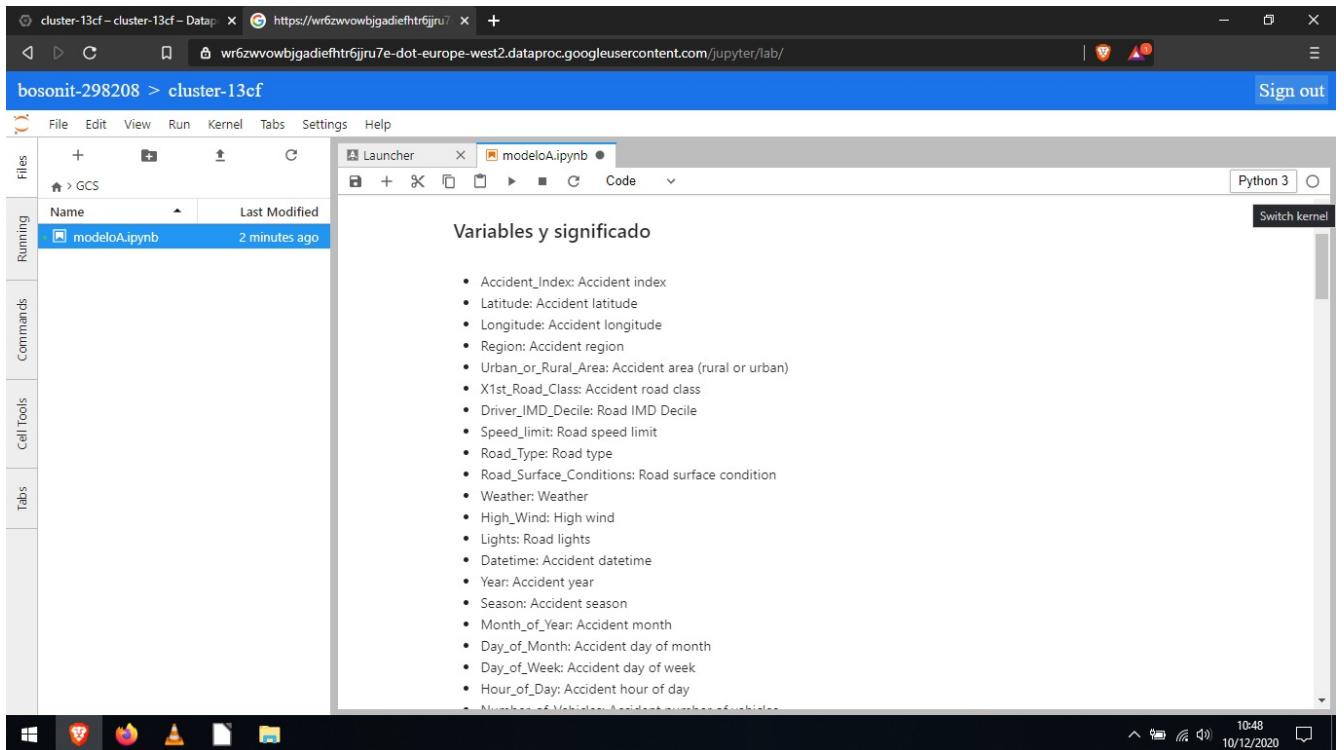
Entramos en ella y haciendo click en Upload subiremos nuestro dataset:



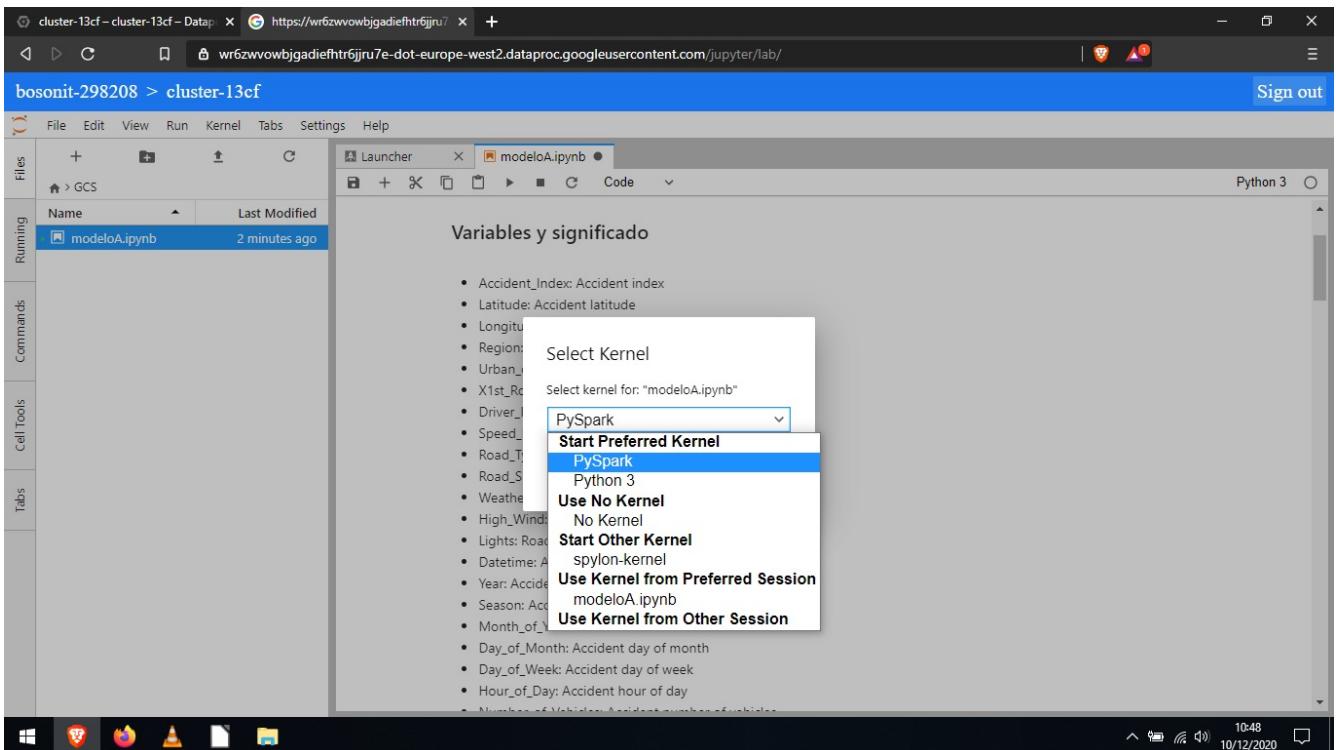
The screenshot shows the same Google Cloud Platform Storage interface as before, but now the 'data' folder is selected. The 'Upload Files' button is highlighted. A large circular icon with a plus sign in the center is displayed, indicating where to click to upload files. The rest of the interface remains the same, showing the table of objects and the sidebar with monitoring and settings options.



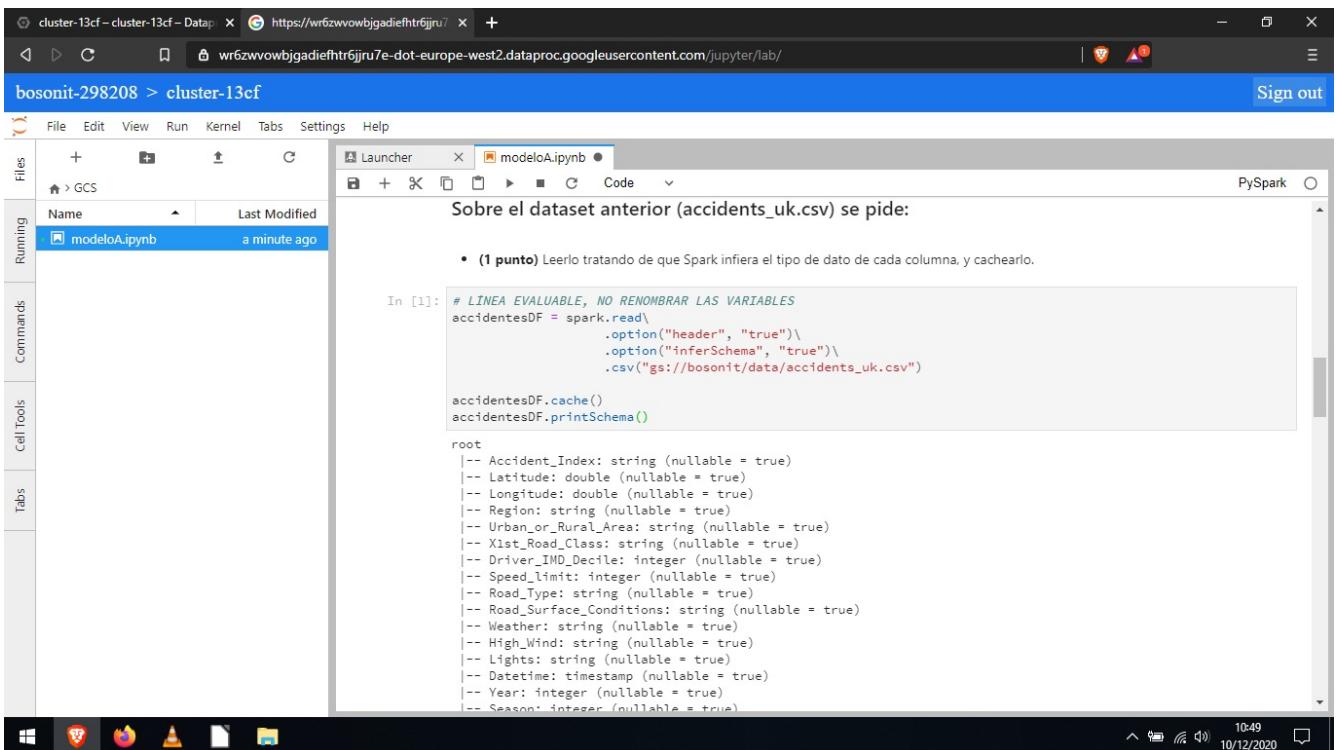
Ahora si volvemos a nuestro Notebook anterior y entramos en el:



Arriba a la derecha seleccionaremos el kernel que queremos utilizar, en nuestro caso Pyspark:



Y probaremos que todo funciona de la siguiente manera:



Como puedes apreciar, la ruta para acceder a los datos almacenados en Storage tienen la ruta: gs://bosonit/data/dataset.csv , donde como vemos la ruta raiz es gs:// y luego el nombre del bucket, seguido de la carpeta data creada anteriormente y el nombre del dataset.

6. Optimización

6.1. Elegir un tipo de compresión

La elección de un cierto tipo de compresión y formato puede tener un impacto importante en el rendimiento. Hay tres importantes lugares donde considerar la compresión de los datos, son los trabajos de MapReduce y Spark Jobs, los datos almacenados en HBase y las queries de Impala. En la mayoría de los casos, los principios son similares para cada uno.

Cuando debemos decidir el formato y compresión a utilizar, debemos equilibrar la capacidad de procesamiento (CPU) necesario en la compresión y descompresión, la E/S del disco al leer o escribir datos y el ancho de banda necesario para enviar los datos a través de la red. El equilibrio de estos factores depende de las características específicas del clúster y de los datos, así como del patrón de uso de estos.

Por supuesto, la compresión no es recomendable cuando los datos ya tienen una cierta compresión (por ejemplo imágenes en formato JPEG). Es más, el archivo comprimido puede ser mayor que el original.

Los formatos más comunes de compresión son los siguientes:

Compression format	Tool	Algorithm	Filename extension	Splittable?
DEFLATE ^a	N/A	DEFLATE	.deflate	No
gzip	gzip	DEFLATE	.gz	No
bzip2	bzip2	bzip2	.bz2	Yes
LZO	lzo	LZO	.lzo	No ^b
LZ4	N/A	LZ4	.lz4	No
Snappy	N/A	Snappy	.snappy	No

Figura 1: Tabla 5.1 de Hadoop The Definitive Guide 4th Edition

- **GZIP** este formato de compresión usa más recursos de la CPU que Snappy o LZO, pero proporciona una relación de compresión más alta. GZip es una buena opción para datos en frío, a los que se accede con poca frecuencia. Snappy o LZO son una mejor opción para datos en caliente, a los que se accede de forma frecuente. Como detalle, el formato GZIP comprime los datos hasta un 30% más que Snappy y usa 2 veces más CPU cuando se leen en comparación con la lectura de un archivo en formato Snappy.
- **BZip2** puede conseguir más compresión que GZip para algunos tipos de datos, a costa de una menor velocidad de compresión y descompresión. HBase no soporta Bzip2.
- **LZO** se enfoca en una mayor velocidad de descompresión y bajo uso de CPU y una mayor compresión a costa de más consumo CPU. Los archivos LZO no se pueden dividir de forma nativa, pero existe el proyecto [Hadoop-LZO](#) que hace posible la divisibilidad de este formato.
- **Snappy** a menudo funciona mejor que LZO. Merece la pena realizar pruebas para ver si se detecta una diferencia significativa.
- **Divisibilidad** *Para MapReduce y Spark, es necesario que los datos comprimidos se puedan dividir, y los formatos BZip2 y LZO no son divisibles. Los bloques Snappy y GZip tampoco son divisibles, pero los archivos con bloques Snappy dentro de un formato de archivo contenedor como SequenceFile, Avro o Parquet si se pueden dividir. Snappy además está diseñado para usarse con un formato contenedor, como SequenceFiles, archivos de datos Avro o Parquet, en lugar de usarse directamente en texto plano sin formato, por ejemplo, este último no se puede dividir y por lo tanto no se puede procesar en paralelo. En el caso de datos para HBase, la capacidad de división no es relevante.*

Para MapReduce se pueden comprimir los datos intermedios, los de salida o ambos. Debemos ajustar los parametros que añadimos al job de MapReduce en consecuencia. Los siguientes ejemplos comprimen tanto los datos intermedios como los de salida. MR2 se muestra primero, seguido de MR1.

- MRv2

```
hadoop jar hadoop-examples-.jar sort "-Dmapreduce.compress.map.output=true"
"-Dmapreduce.map.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec"
"-Dmapreduce.output.compress=true"
"-Dmapreduce.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec" -outKey
org.apache.hadoop.io.Text -outValue org.apache.hadoop.io.Text input output
```

- MRv1

```
hadoop jar hadoop-examples-.jar sort "-Dmapred.compress.map.output=true"
"-Dmapred.map.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec"
"-Dmapred.output.compress=true"
"-Dmapred.output.compression.codec=org.apache.hadoop.io.compress.GzipCodec" -outKey
org.apache.hadoop.io.Text -outValue org.apache.hadoop.io.Text input output
```