



# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Instalación y Configuración</b>	<b>4</b>
2.1. Requisitos . . . . .	4
2.2. Java JDK . . . . .	5
2.2.1. Descarga . . . . .	5
2.2.2. Instalacion . . . . .	6
2.2.3. Variables de entorno . . . . .	7
2.3. Anaconda . . . . .	12
2.3.1. Descarga . . . . .	12
2.3.2. Instalacion . . . . .	14
2.4. Apache Spark . . . . .	16
2.4.1. Descarga . . . . .	16
2.4.2. Instalación . . . . .	17
2.4.3. Variables de entorno . . . . .	18
2.4.4. Scala y Python desde consola . . . . .	19
2.4.5. Ejecutando Spark-shell y pyspark desde consola . . . . .	19
2.4.6. Jupyter Notebook desde consola con Pyspark . . . . .	20
2.4.7. Pyspark en anaconda . . . . .	21
2.4.8. Scala desde Jupyter Notebook . . . . .	21
<b>3. Servicios en la nube</b>	<b>22</b>
3.1. Databricks . . . . .	22
3.1.1. Registro . . . . .	23
3.1.2. Creacion de Cluster . . . . .	24
3.1.3. Creacion y carga de Notebooks . . . . .	28
3.2. Google Cloud . . . . .	31
3.2.1. Registro . . . . .	32
3.2.2. Storage . . . . .	35
3.2.3. Dataproc . . . . .	40

## 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

### 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 página principal, donde empieza con Download, haremos click en [jdk.java.net/15](#). Esto nos llevará a la página de descargas, donde veremos versiones para los distintos sistemas operativos. En nuestro caso seleccionaremos la versión de Windows/x64.

Workshop  
OpenJDK FAQ  
Installing  
Contributing  
Sponsoring  
Developers' Guide  
Vulnerabilities  
Mailing lists  
IRC - Wiki  
Bylaws - Census  
Legal  
JEP Process

Source code  
Mercurial  
GitHub

Groups  
(overview)  
2D Graphics  
Adoption  
AWT  
Build  
Compatibility & Specification  
Runtime  
Compiler  
Conformance  
Core Libraries  
Governing Board  
Hotspot  
IDE Tooling & Support  
Internationalization  
JMX  
Members  
Networking  
Porters  
Quality  
Security  
Serviceability  
Sound  
Swing  
Vulnerability  
Web

Projects  
(overview)  
Amber  
Annotations Pipeline  
2.0



**What is this?** The place to collaborate on an open-source implementation of the Java Platform, Standard Edition, and related projects. (Learn more.)

**Download** and install the open-source JDK for most popular Linux distributions. Oracle's free, GPL-licensed production-ready OpenJDK JDK 15 binaries are at [jdk.java.net/15](#). Oracle's commercially-licensed JDK 15 binaries for Linux, macOS, and Windows, based on the same code, are here.

**Learn how to use the JDK** to write applications for a wide range of environments.

**Hack on the JDK itself**, right here in the OpenJDK Community: Browse the code on the web, clone a Mercurial repository to make a local copy, and contribute a patch to fix a bug, enhance an existing component, or define a new feature.

(a) Página principal

[jdk.java.net](#)  
GA Releases  
JDK 15  
JMC 7  
Early-Access  
Releases  
JDK 17  
JDK 16  
Lamai  
Loon  
Metropolis  
Panama  
Valhalla  
Reference  
Implementations  
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  
Feedback  
Report a bug  
Archive

#### JDK 15.0.1 General-Availability Release

This page provides production-ready open-source builds of the Java Development Kit, version 15, an implementation of the Java SE 15 Platform under the GNU General Public License, version 2, with the Classpath Exception. Commercial builds of JDK 15.0.1 from Oracle, under a non-open-source license, can be found at the Oracle Technology Network.

#### Documentation

- Features
- Release notes
- API Javadoc

#### Builds

<b>Linux/AArch64</b>	<a href="#">tar.gz (sna256)</a>	170492774 bytes
<b>Linux/x64</b>	<a href="#">tar.gz (sna256)</a>	195347356
<b>macOS/x64</b>	<a href="#">tar.gz (sna256)</a>	192652449
<b>Windows/x64</b>	<a href="#">zip (sna256)</a>	195936491

#### Notes

- The Alpine Linux build previously available on this page was removed as of the first JDK 15 release candidate. It's not production-ready because it hasn't been tested thoroughly enough to be considered a GA build. Please use the early-access JDK 16 Alpine Linux build in its place.
- If you have difficulty downloading any of these files please contact [jdk-download-help\\_ww@oracle.com](#).

#### Feedback

If you have suggestions or encounter bugs, please submit them using the usual Java SE bug-reporting channel. Be sure to include complete version information from the output of the `java --version` command.

#### International use restrictions

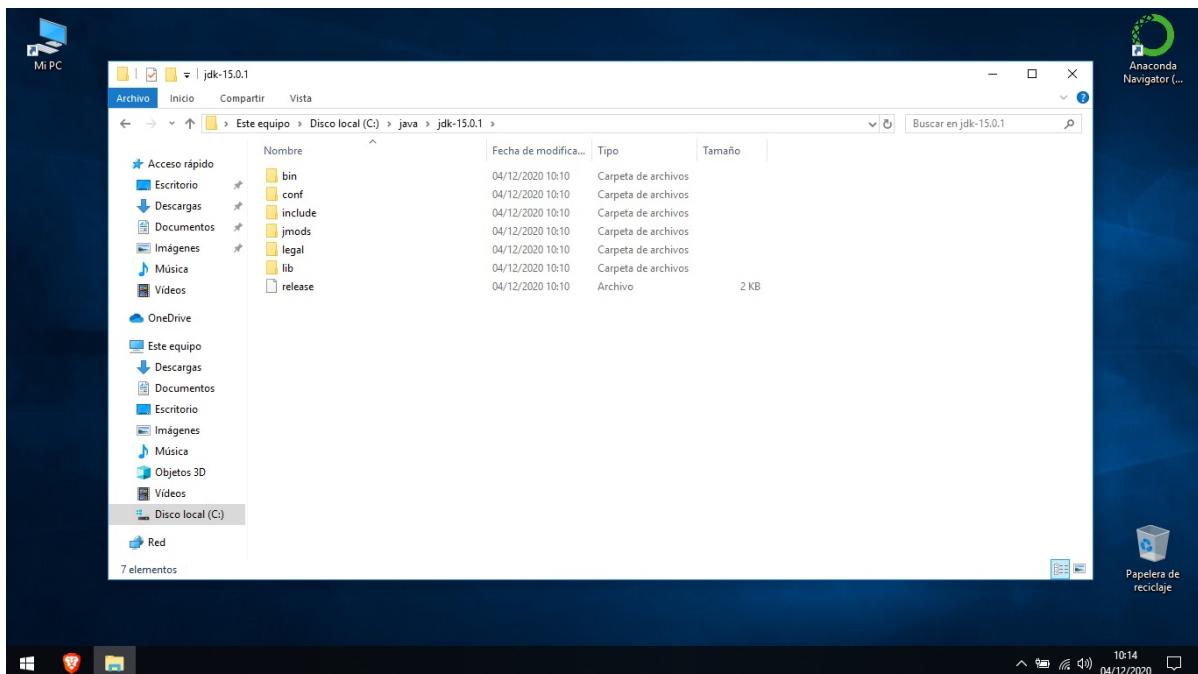
Due to limited intellectual property protection and enforcement in certain

(b) Página de descargas

### 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:



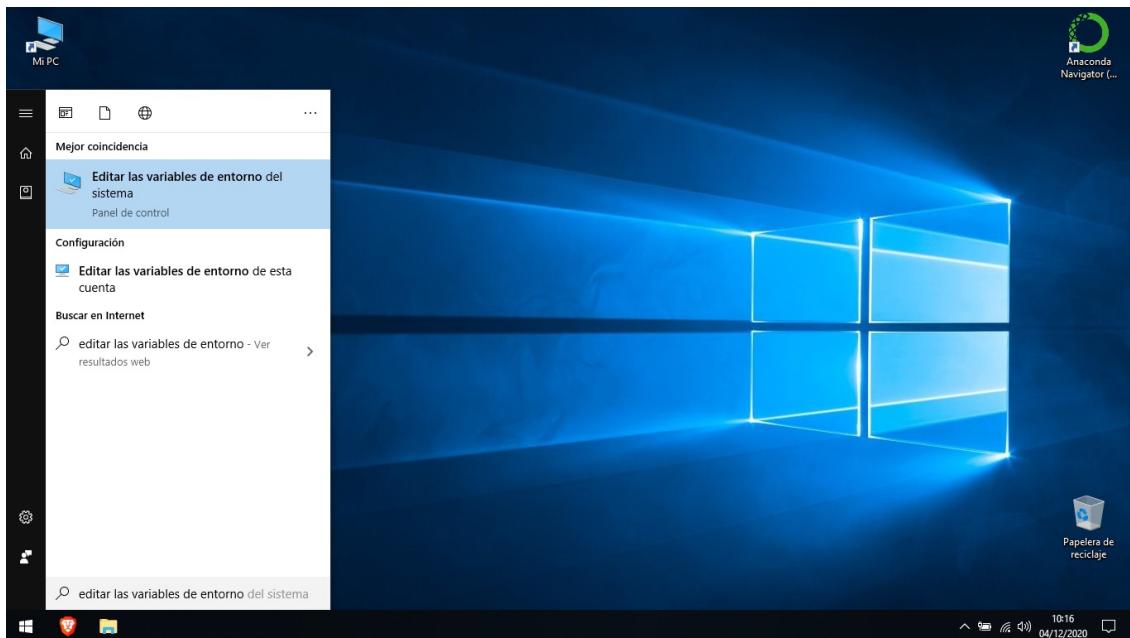
### 2.2.3. Variables de entorno

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.

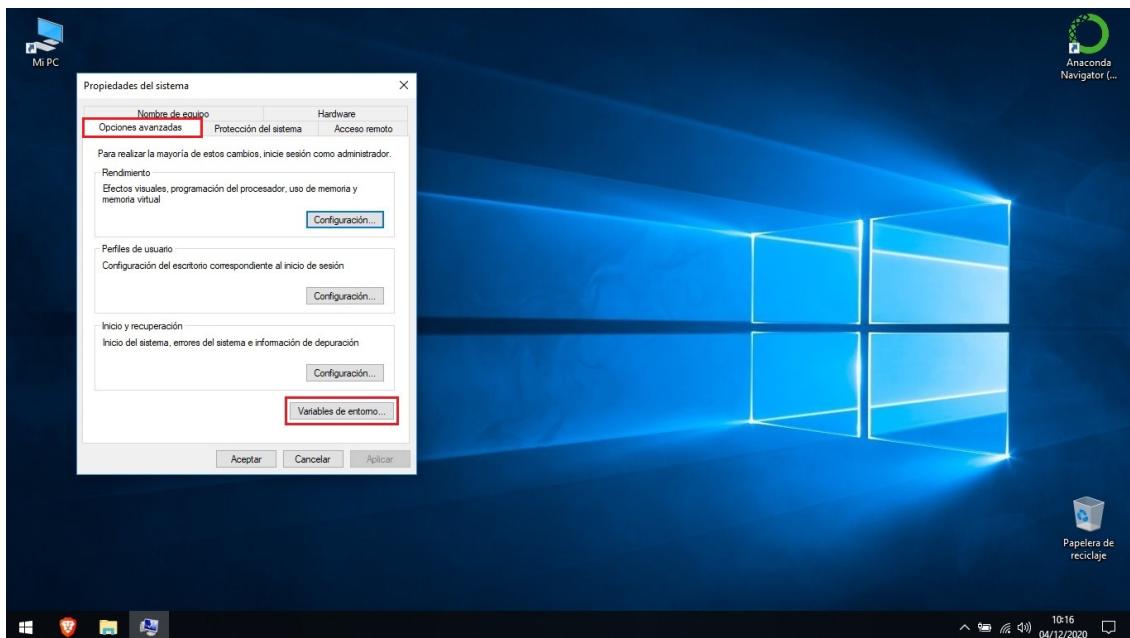
En este caso, deberemos crear la variable de entorno **JAVA\_HOME** y actualizar la variable de entorno **PATH**. La primera se utilizará para que Java sepa dónde se encuentra la instalación de Java JDK y la segunda es para poder ejecutar los comandos de Java (como javac, java, etc) desde cualquier lugar,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

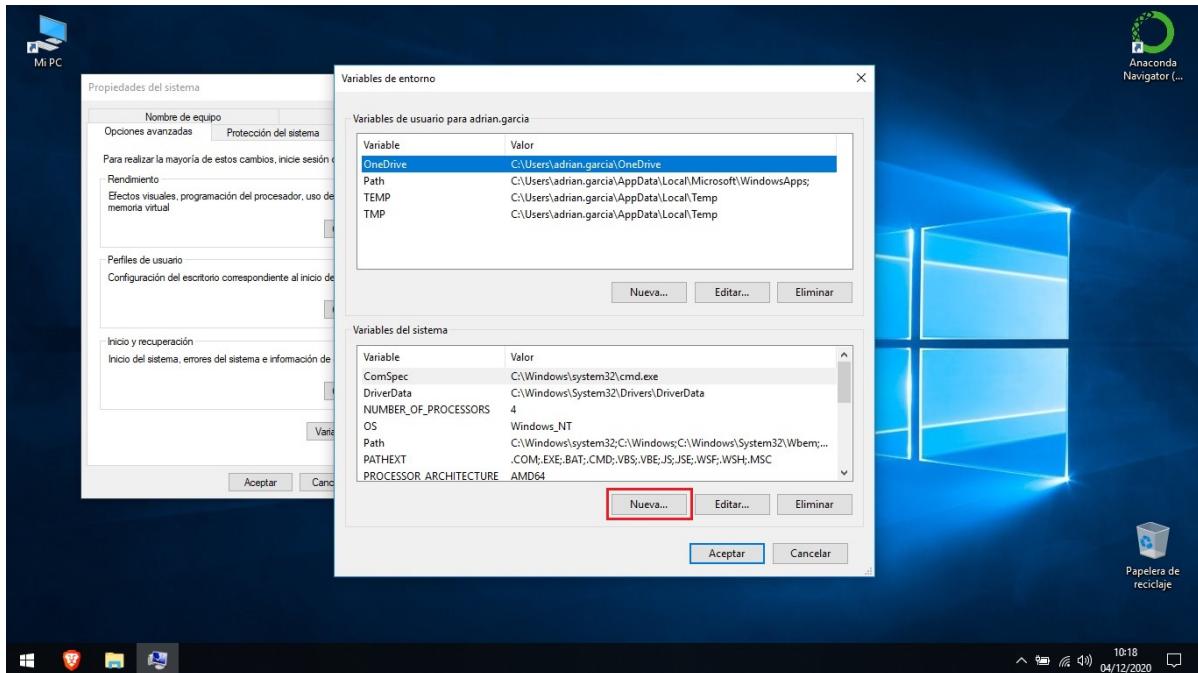


Al abrir el editor veremos que se abre la ventana de Propiedades del sistema. Hacemos click en la pestaña de 'Opciones avanzadas' y abajo hacemos click en 'Variables de entorno' como en la siguiente imagen:

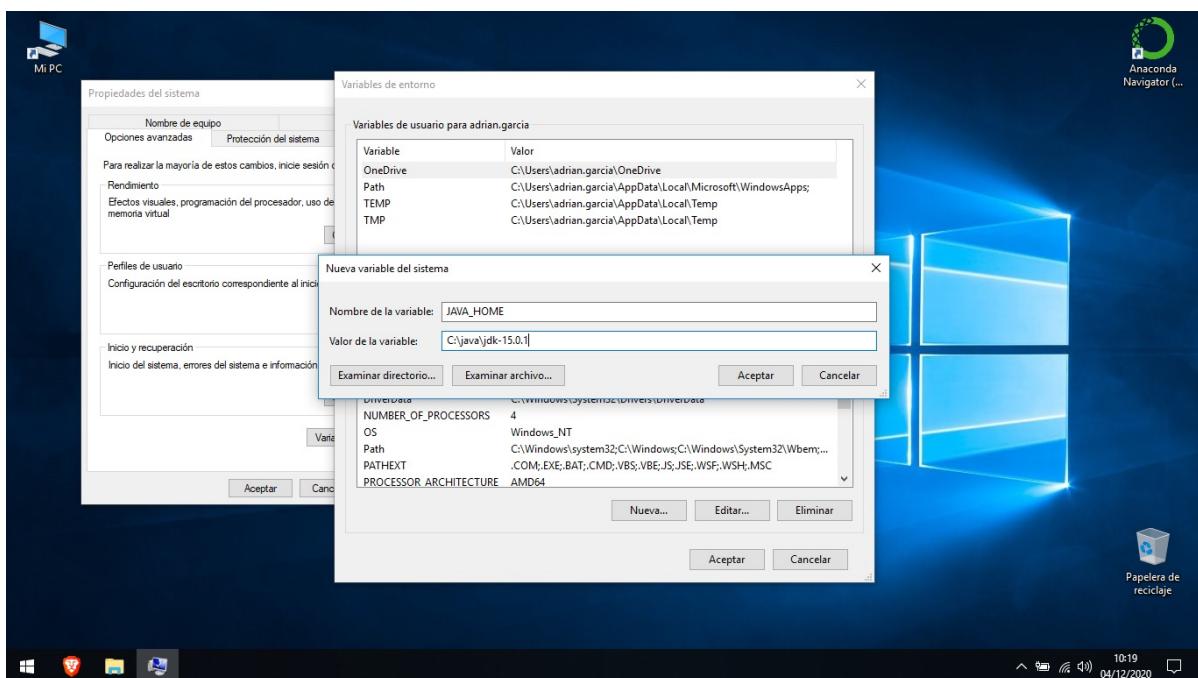


En la ventana que se nos ha abierto veremos dos recuadros. En uno pondrá *Variables de usuario* y en el de debajo *Variables del sistema*. Lo mas recomendable es crear las variables de entorno para el sistema, con el fin de que cualquier usuario tenga acceso a la ejecución de java.

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

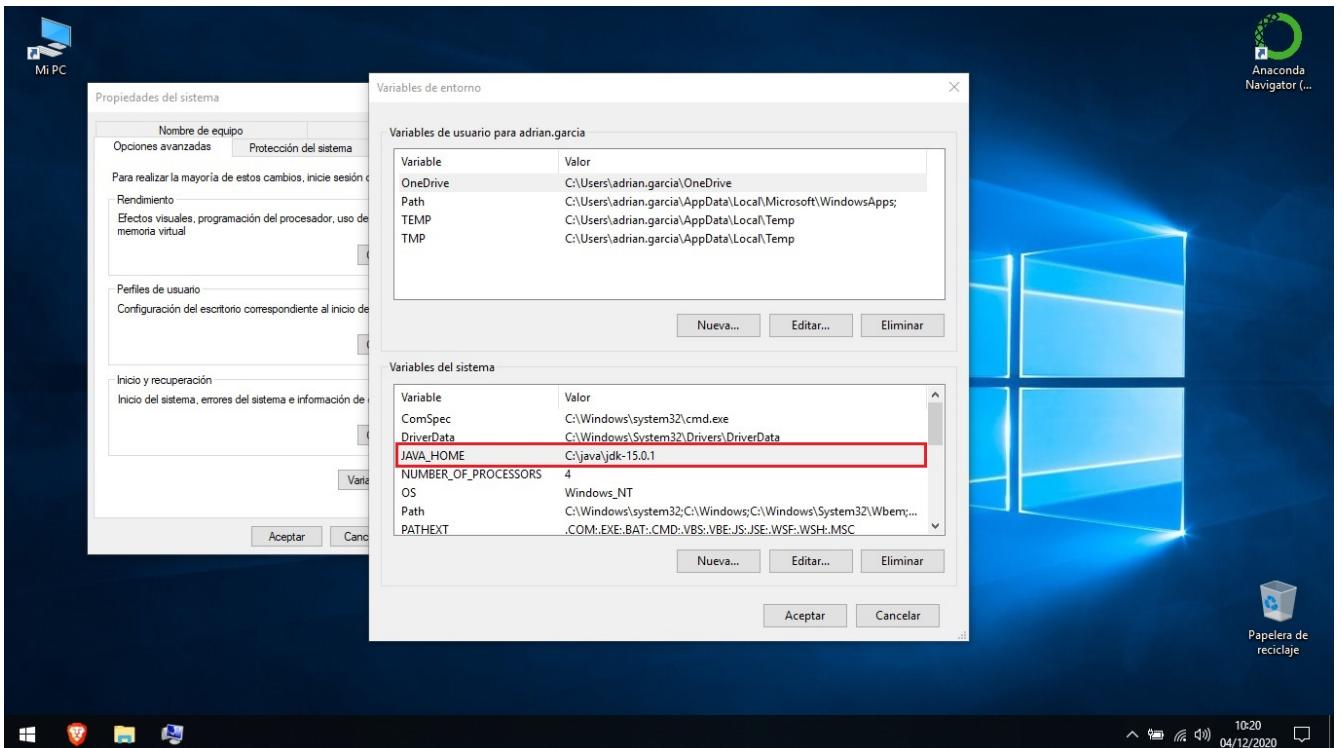


En el nombre escribiremos 'JAVA\_HOME', mientras que en valor la variable escribiremos la ruta donde se instaló el Java JDK, en nuestro caso será *C:/java/jdk-11.0.2*



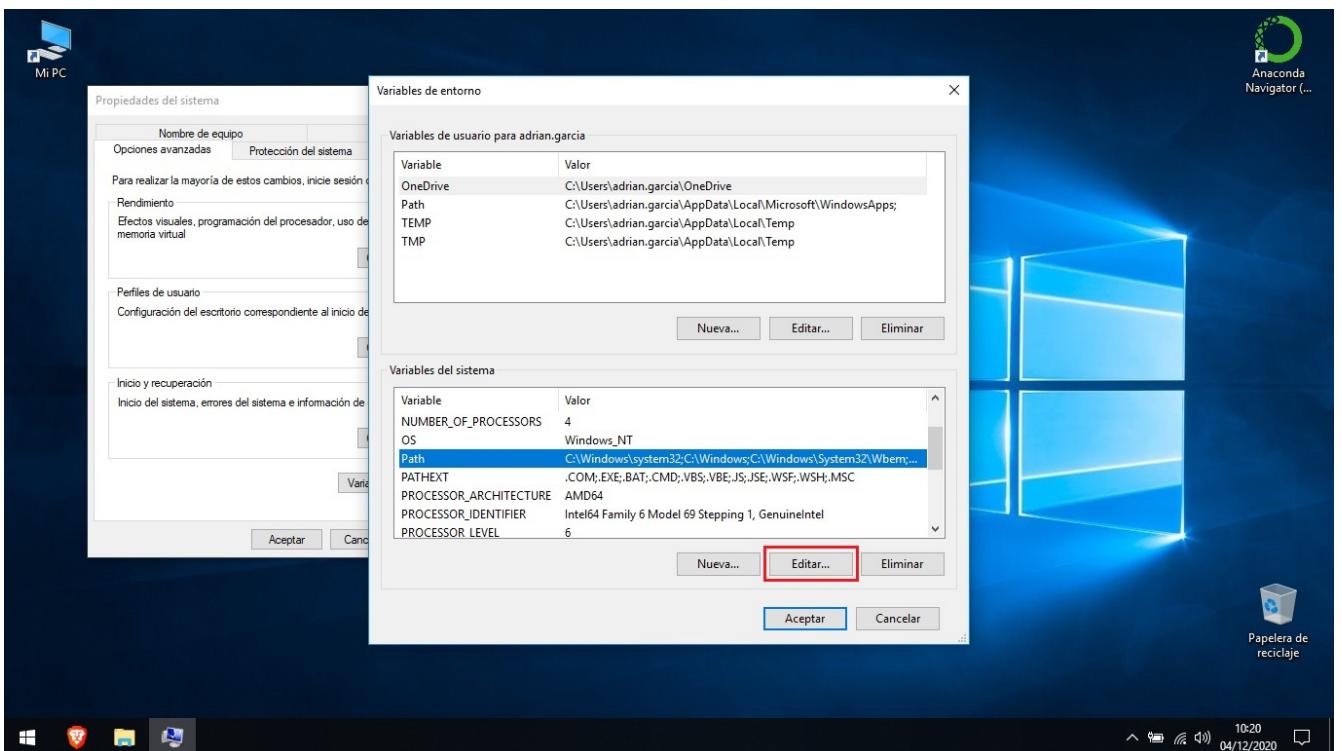
Hacemos clic en aceptar.

Debería quedarnos de la siguiente manera:

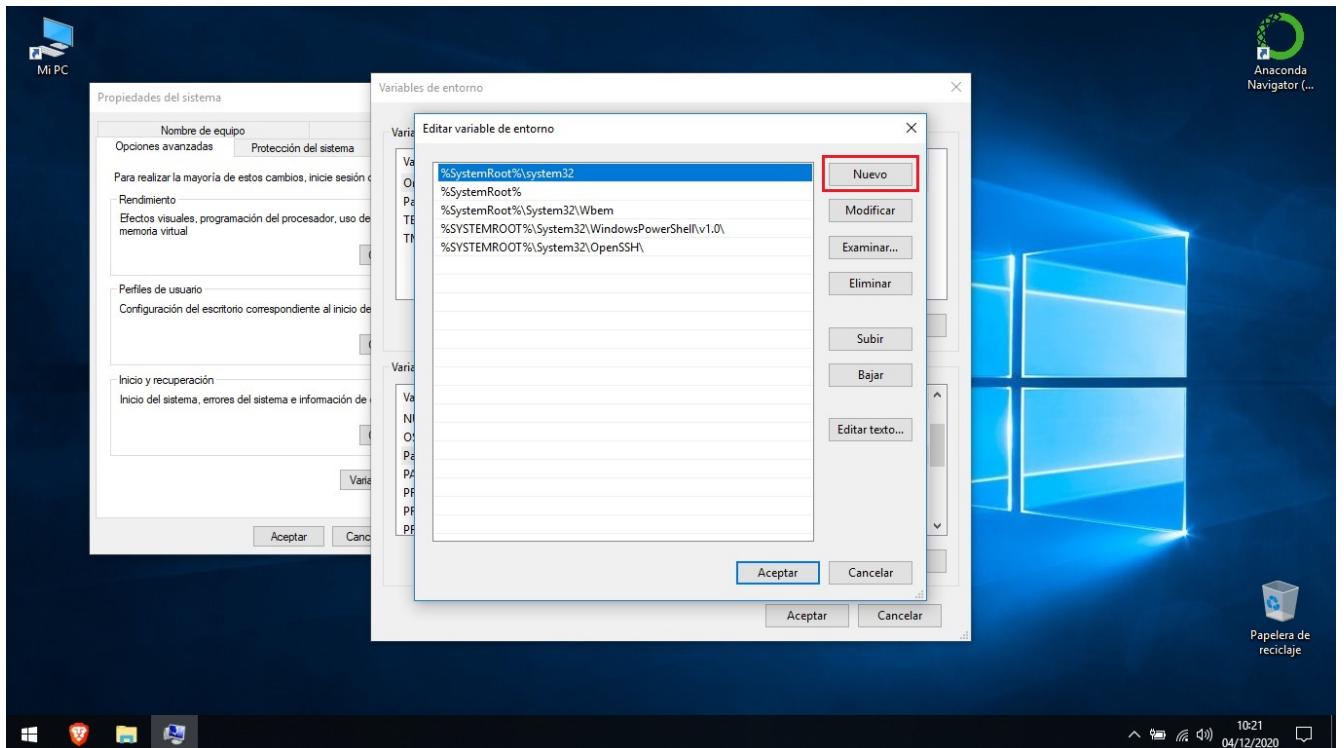


## • PATH

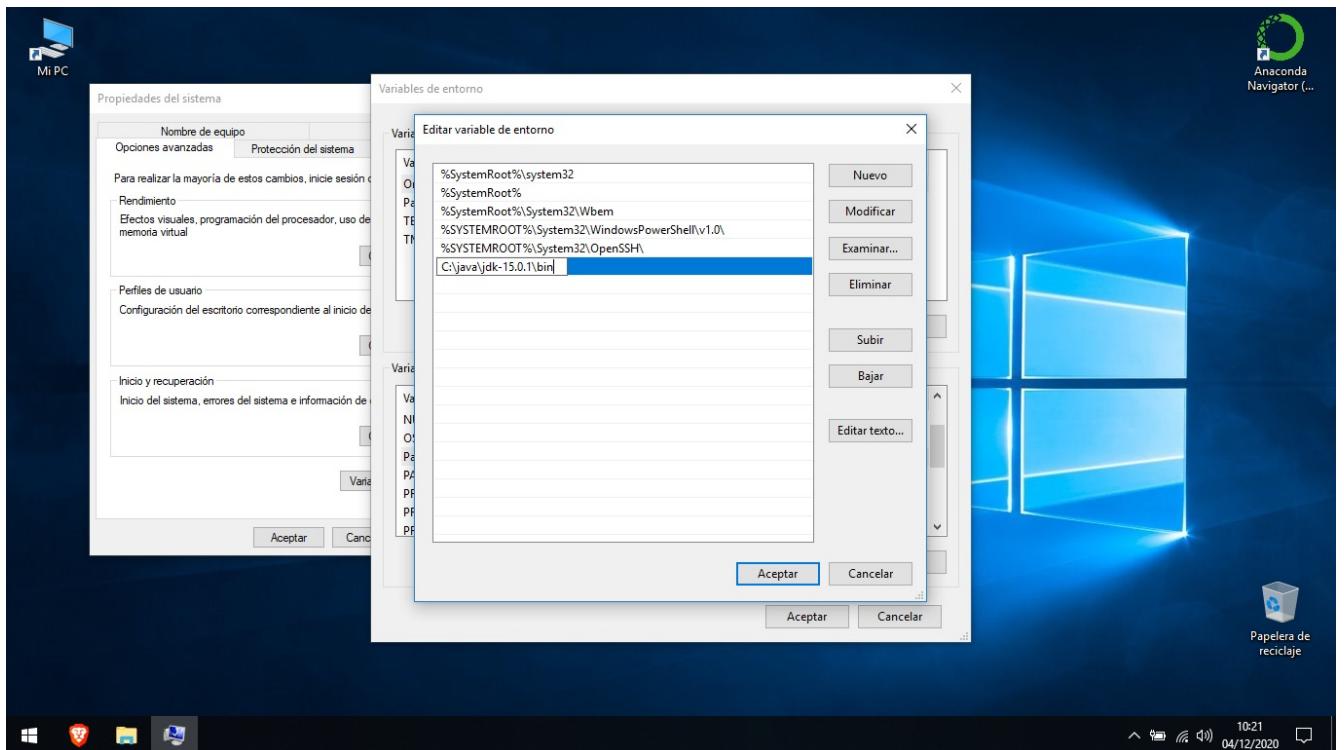
Si por alguna razón no tienes abierta la ventana de variables de entorno, repite los pasos anteriores. En este caso vamos a editar la variable Path que ya existe dentro de variables del sistema. La seleccionamos y le damos a editar:



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

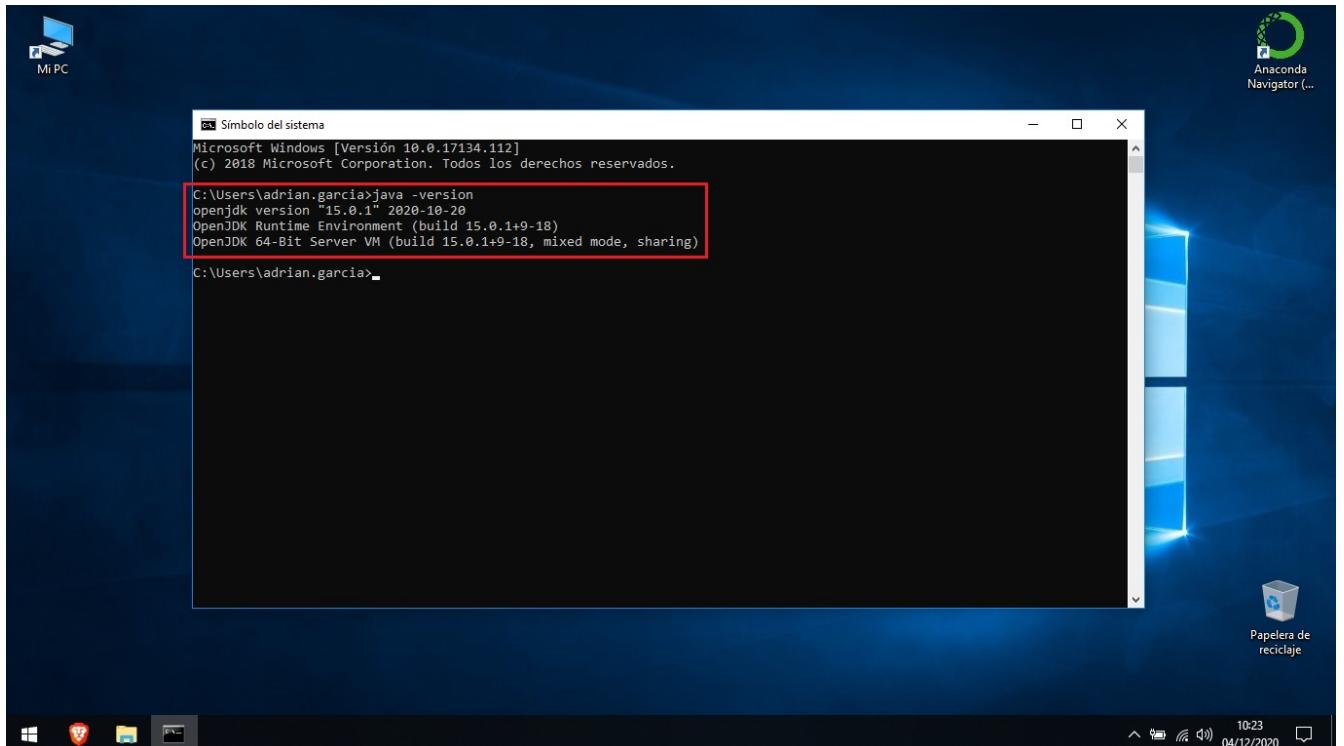


Escribiremos la ruta a la instalación de java, pero en este caso direccionandolo a la carpeta bin. Si has seguido todos los pasos hasta aquí, la ruta sera *C:/java/jdk-11.0.2/bin*



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.

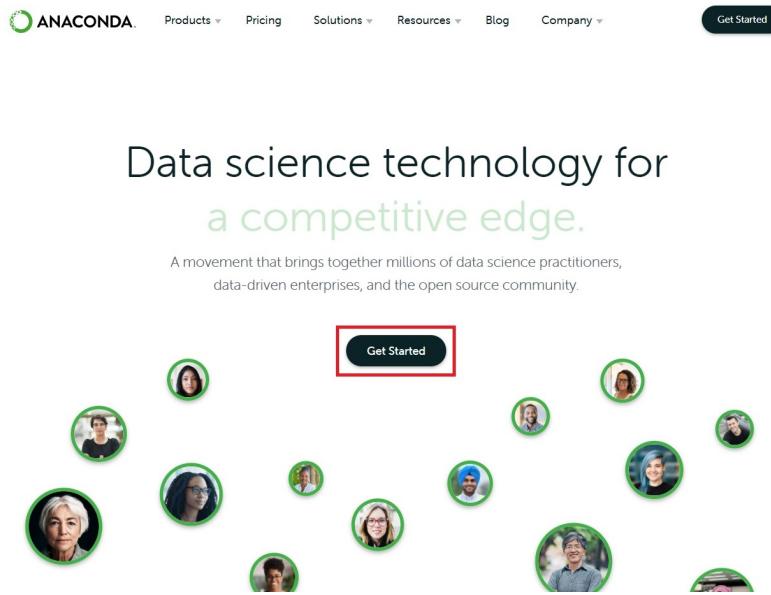
Nos va a servir para el procesamiento de datos a gran escala, el análisis predictivo y la informática científica, que tiene como objetivo simplificar la gestión de empaquetado y distribución. Esta es quizás la Suite más completa para la Ciencia de datos con Python y que nos brinda una gran cantidad de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

### 2.3.1. Descarga

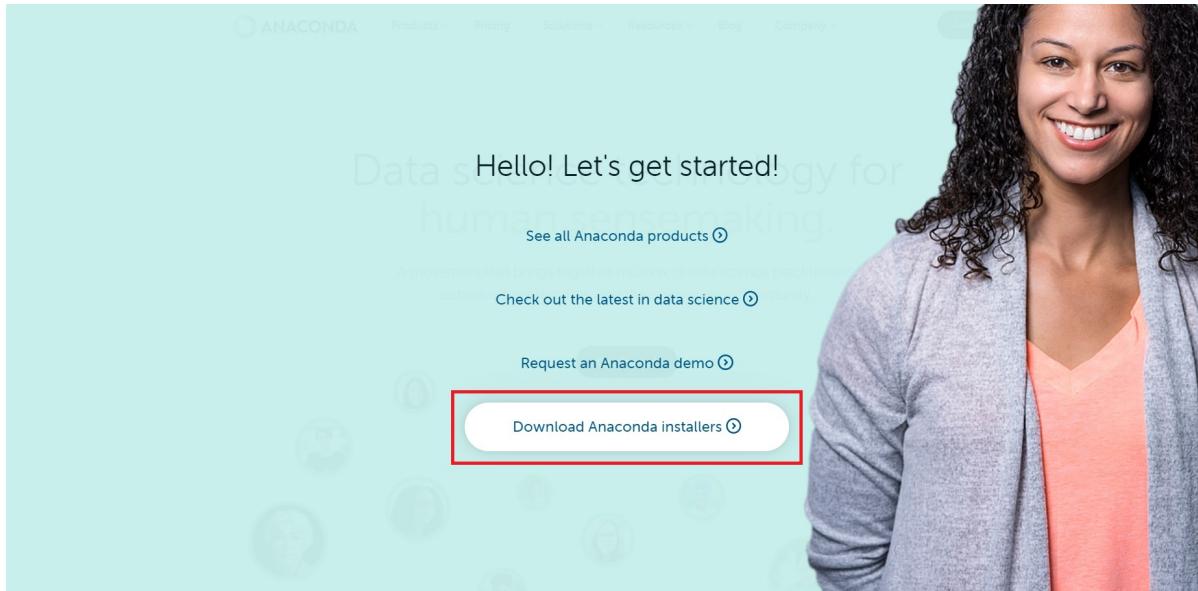
El primero paso será ir a la web oficial de Anaconda:

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

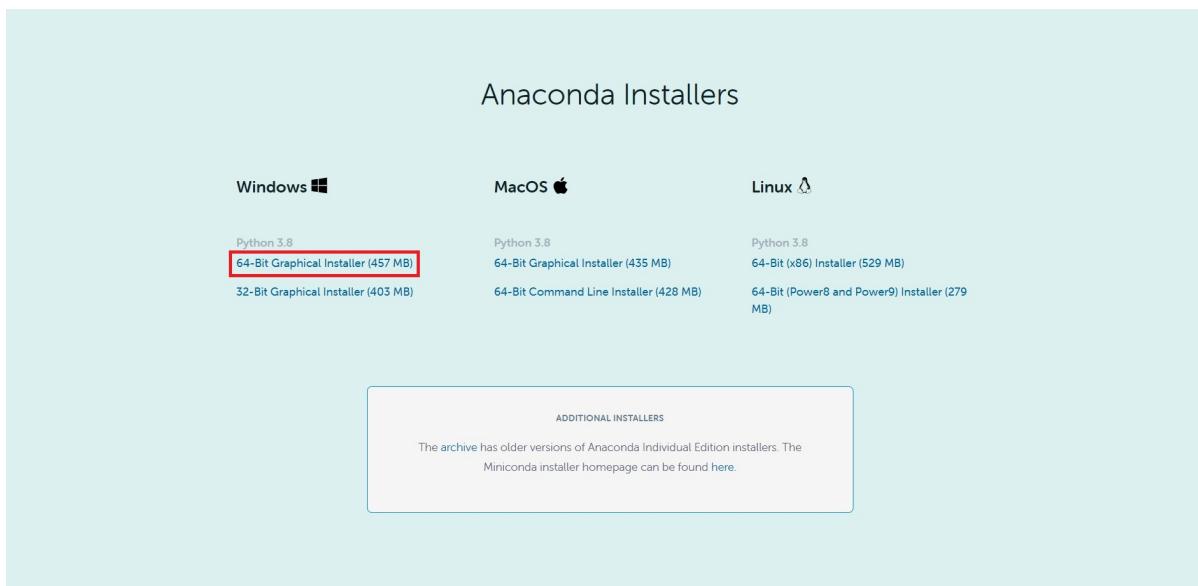
En la web principal haremos click en *Get Started*:



En la ventana emergente seleccionamos *Download Anaconda Installers*

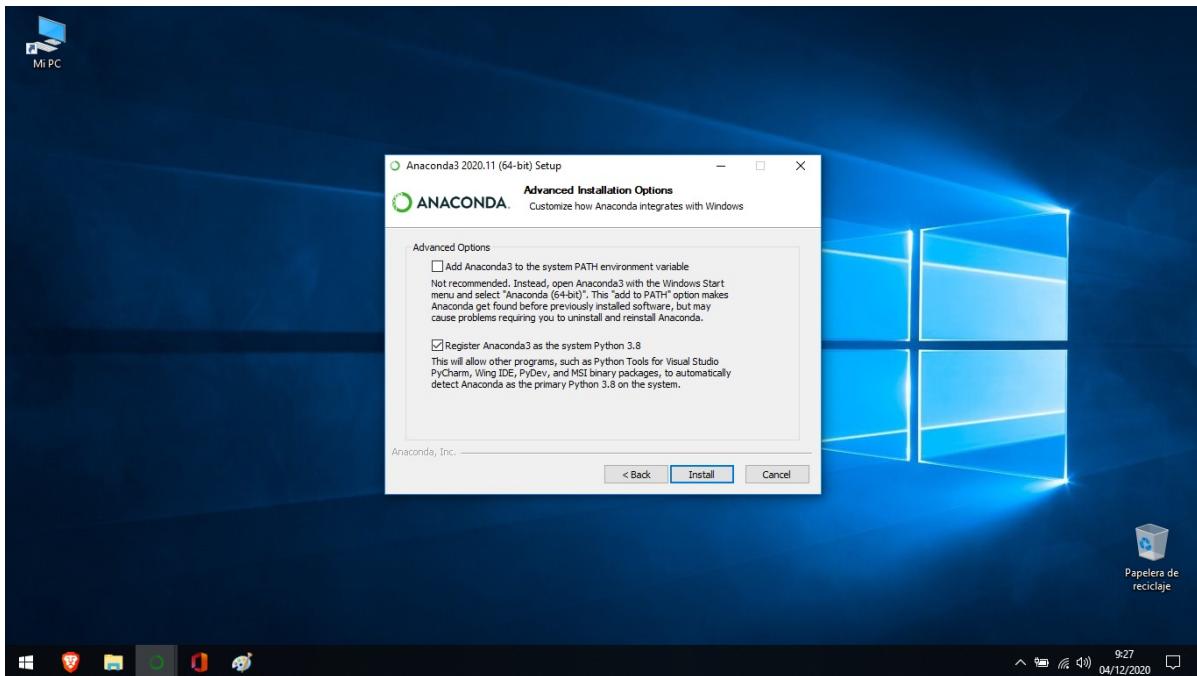


Y seleccionamos la versión compatible con nuestro sistema, en este caso Windows 10 de 64 bits:

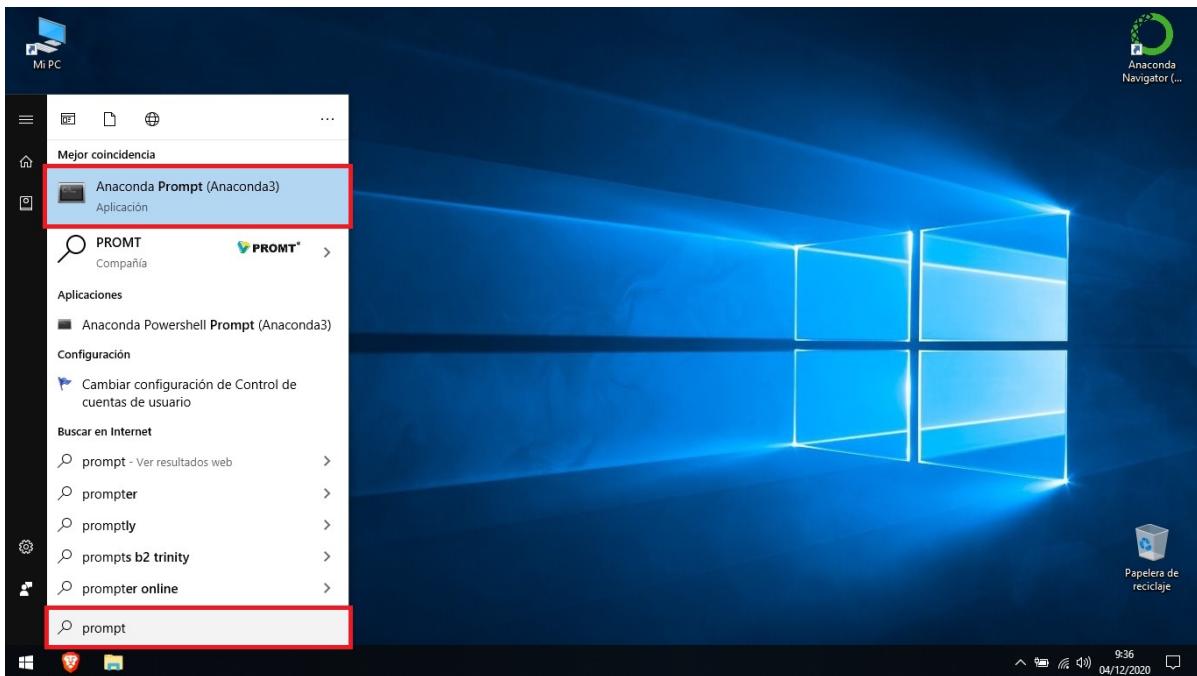


### 2.3.2. Instalacion

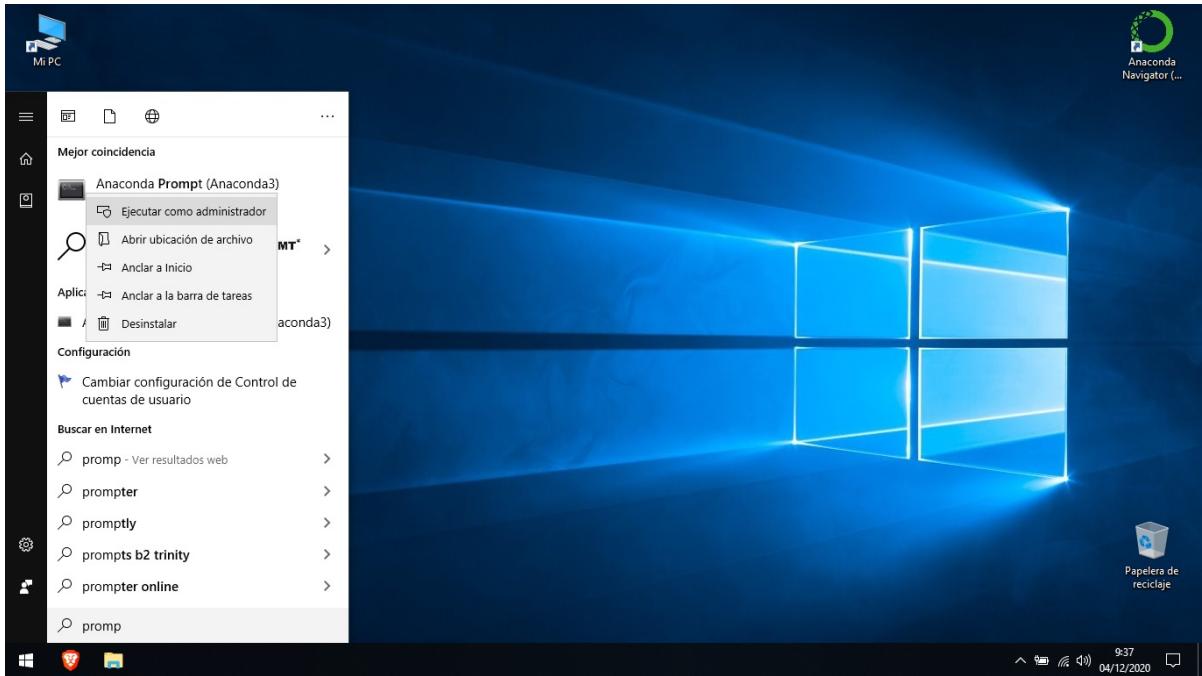
Vamos a nuestra carpeta de descargas e iniciamos el instalador. Durante la instalación dejaremos todos los valores por defecto.



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:

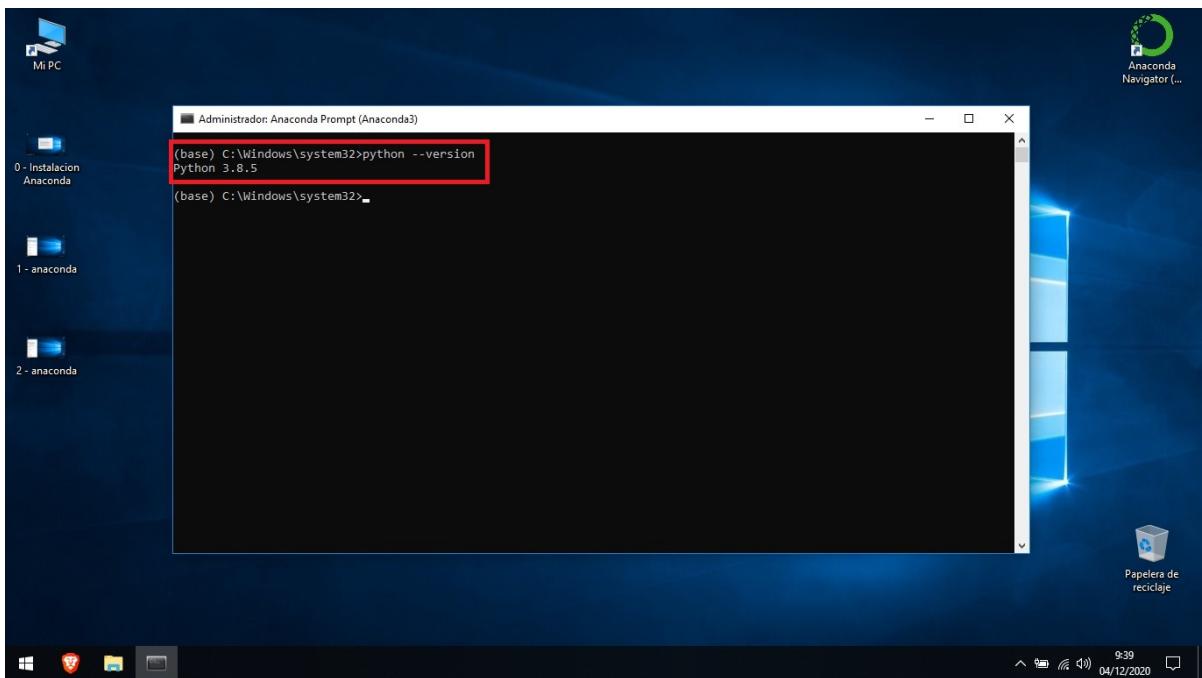


Haremos click derecho y ejecutaremos como administrador. Esto hará que salga una pantalla emergente de windows que nos preguntara si estamos seguros de que queremos permitir que este programa haga modificaciones en el equipo, aceptamos.



**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.

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:



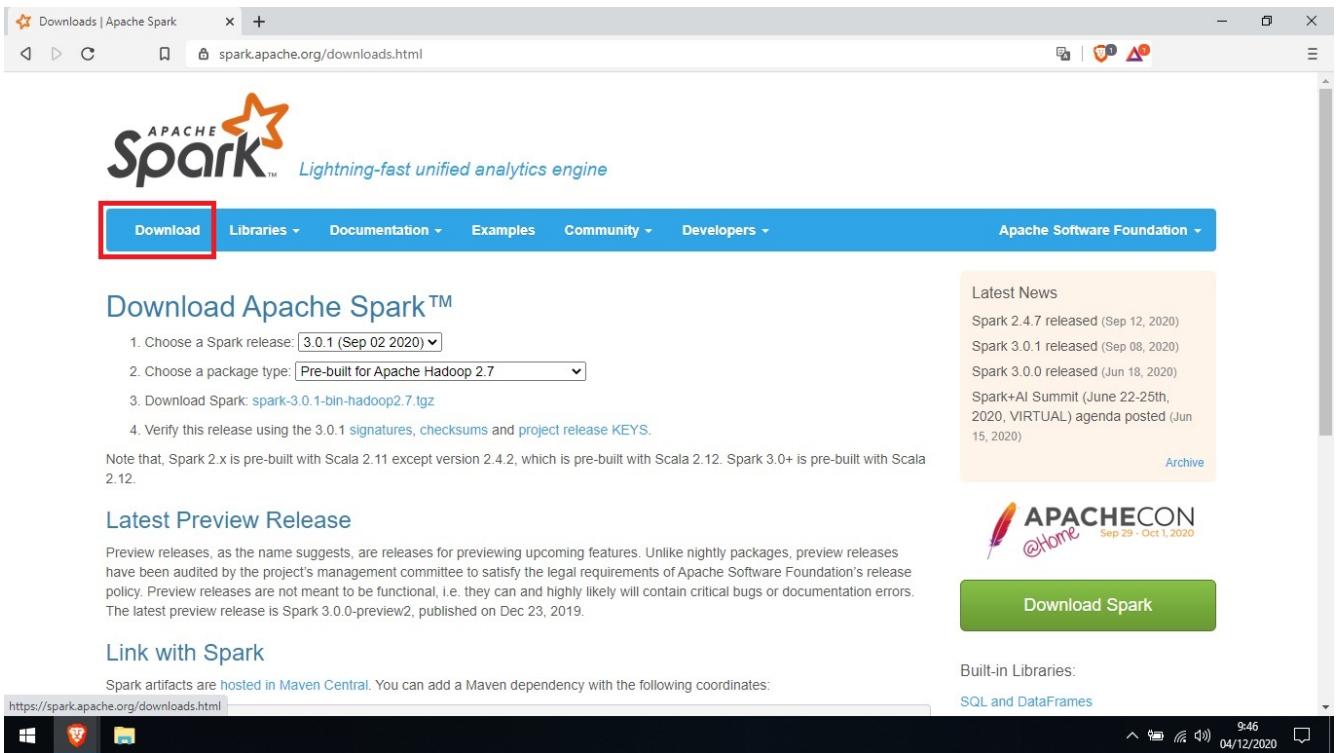
## 2.4. Apache Spark

### 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”

**Download Apache Spark™**

1. Choose a Spark release: [3.0.1 (Sep 02 2020) ▾]
2. Choose a package type: [Pre-built for Apache Hadoop 2.7 ▾]
3. Download Spark: [Spark-3.0.1-bin-hadoop2.7.tgz](#) [highlighted]
4. Verify this release using the 3.0.1 signatures, checksums and project release KEYS.

Note that, Spark 2.x is pre-built with Scala 2.11 except version 2.4.2, which is pre-built with Scala 2.12. Spark 3.0+ is pre-built with Scala 2.12.

**Latest Preview Release**

Preview releases, as the name suggests, are releases for previewing upcoming features. Unlike nightly packages, preview releases have been audited by the project's management committee to satisfy the legal requirements of Apache Software Foundation's release policy. Preview releases are not meant to be functional, i.e. they can and highly likely will contain critical bugs or documentation errors. The latest preview release is Spark 3.0.0-preview2, published on Dec 23, 2019.

**Link with Spark**

Spark artifacts are hosted in [Maven Central](#). You can add a Maven dependency with the following coordinates:

```
https://spark.apache.org/downloads.html
```

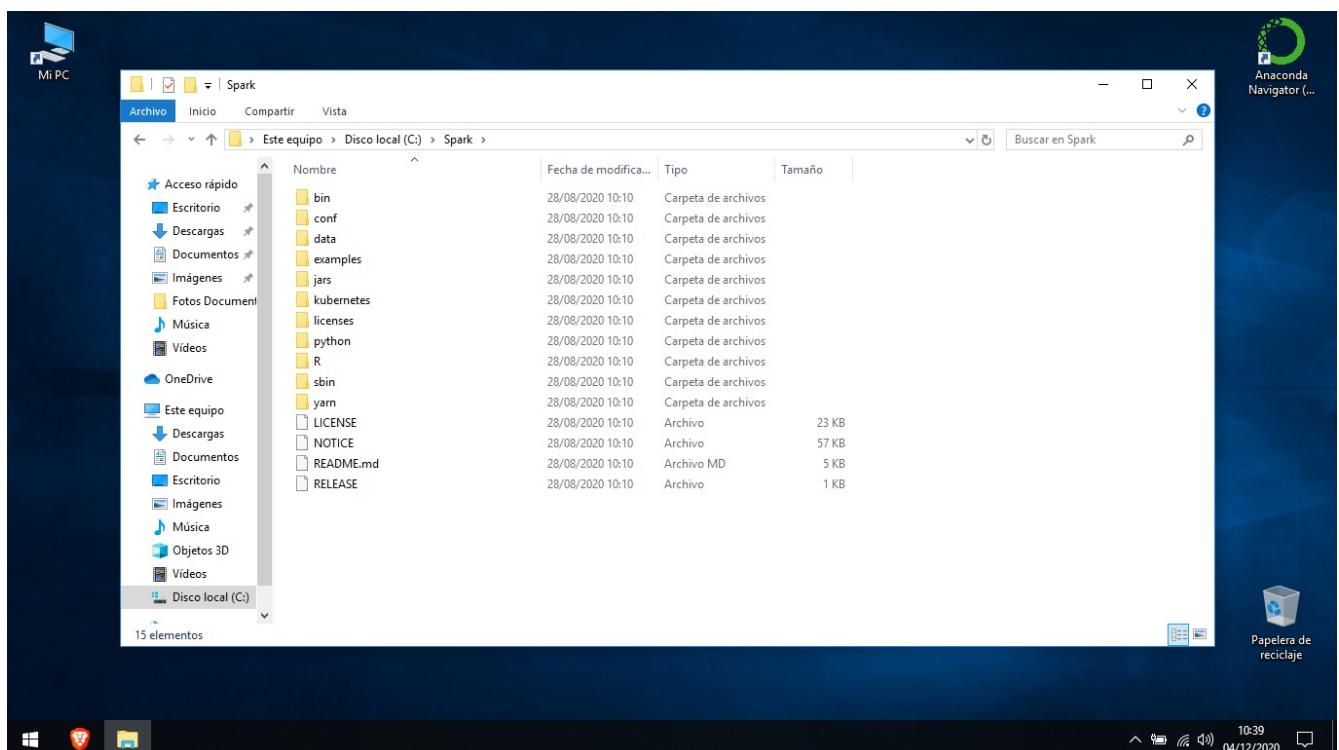
Windows taskbar icons are visible at the bottom left, and system status at the bottom right.

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

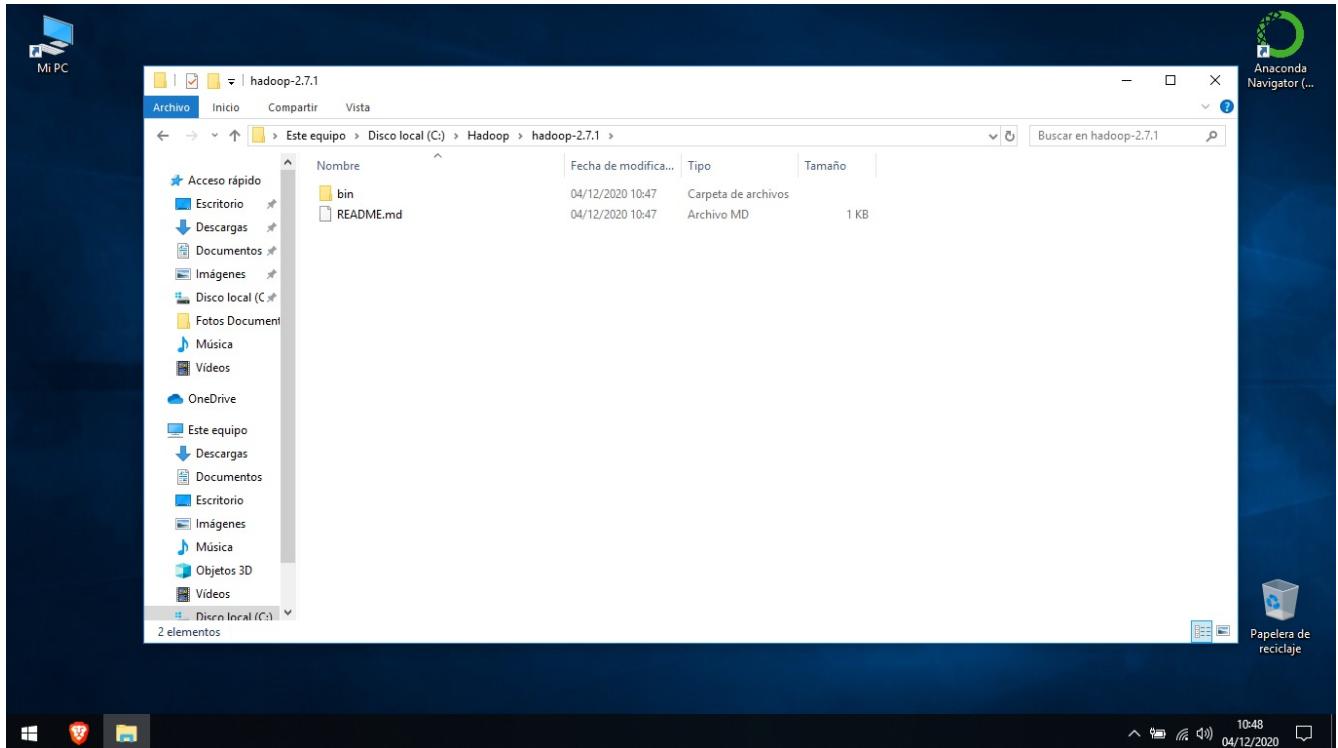
En esta carpeta sera donde descomprimiremos el archivo de Apache Spark recién descargado. Quedara de la siguiente forma:



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 tres 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
- 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

Anconda Navigator (...)

```
(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

Mi PC

Anaconda Navigator ...

```
[base] C:\Users\adrian.garcia>spark-shell
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 11:04:31 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).
Spark context Web UI available at http://EM2020002537.bosonit.local:4040
Spark context available as 'sc' (master = local[*], app id = local-1607076281867).
Spark session available as 'spark'.
Welcome to

    \_____
   /       \
  /  _   _ \
 /  /\ \  / \
/  /\_\ \/\_\
 \_\_/\_\_/\_\
               version 3.0.1

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 15.0.1)
Type in expressions to have them evaluated.
Type :help for more information.

scala> 20/12/04 11:04:52 WARN ProcfsMetricsGetter: Exception when trying to compute pagesize, as a result reporting of ProcessTree metrics is stopped

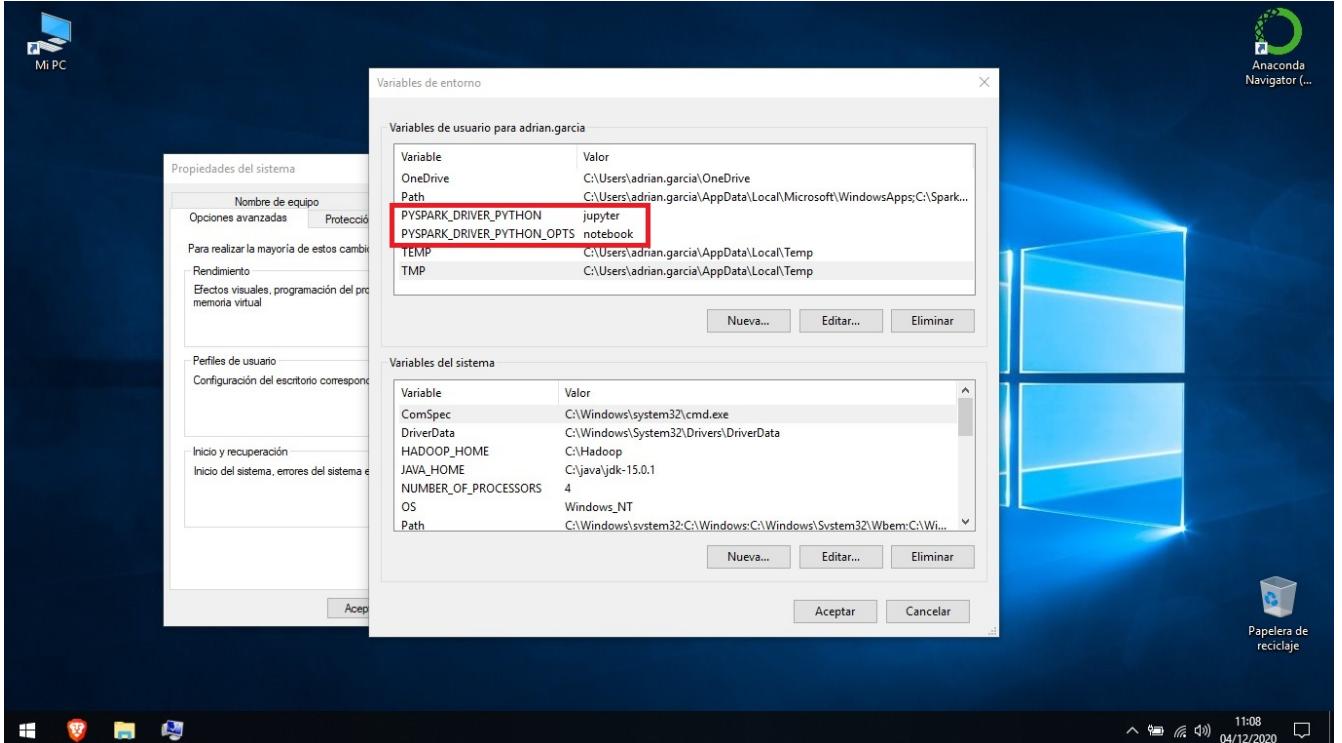
scala>
```

Papelera de reciclaje

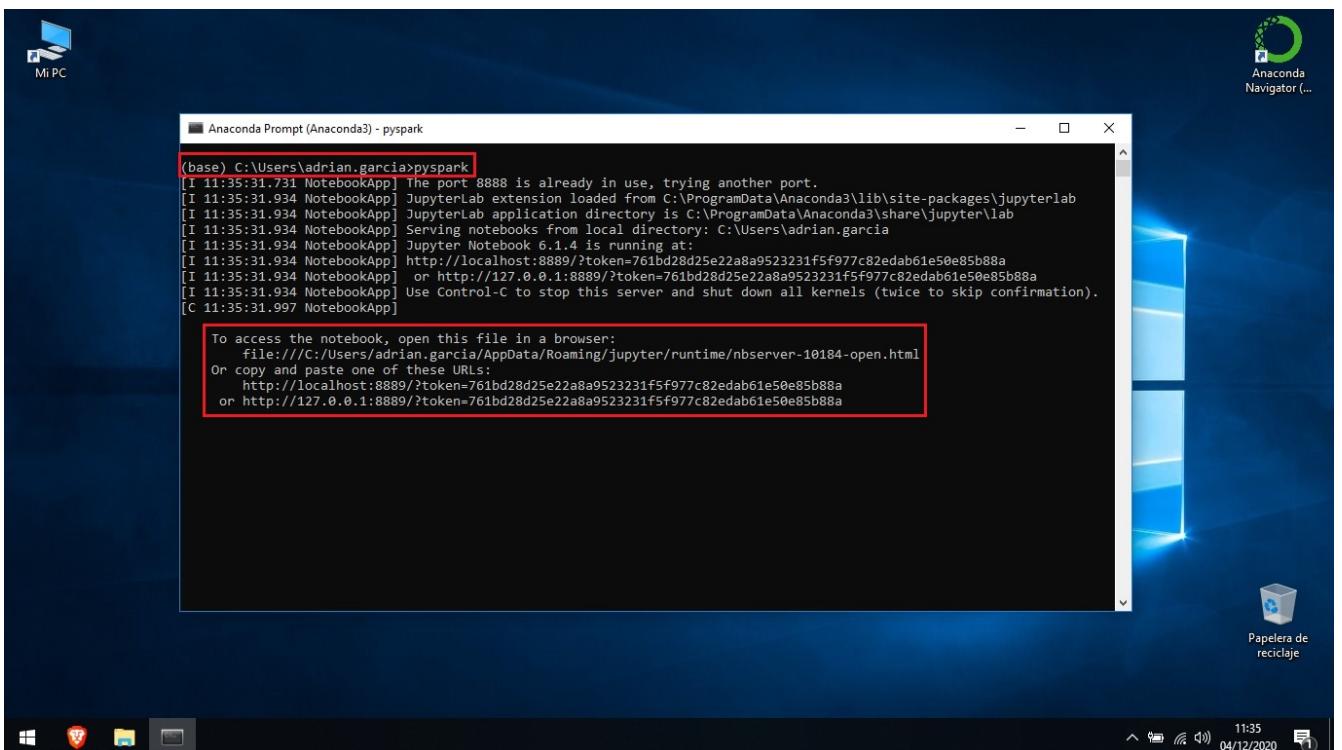
Ejecutar en la consola de anaconda pip install pyspark

#### 2.4.6. Jupyter Notebook desde consola con Pyspark

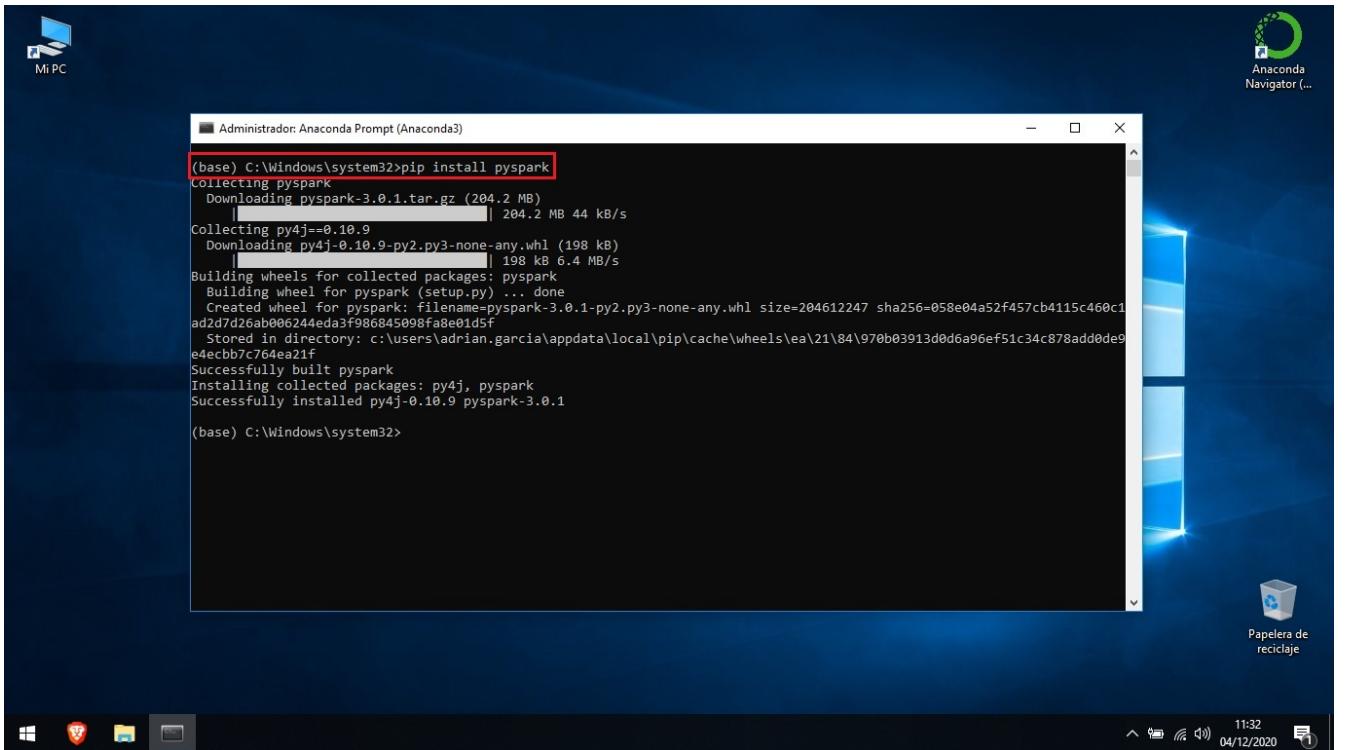
Añadimos las variables de entorno de jupyter y notebook



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



#### 2.4.7. Pyspark en anaconda



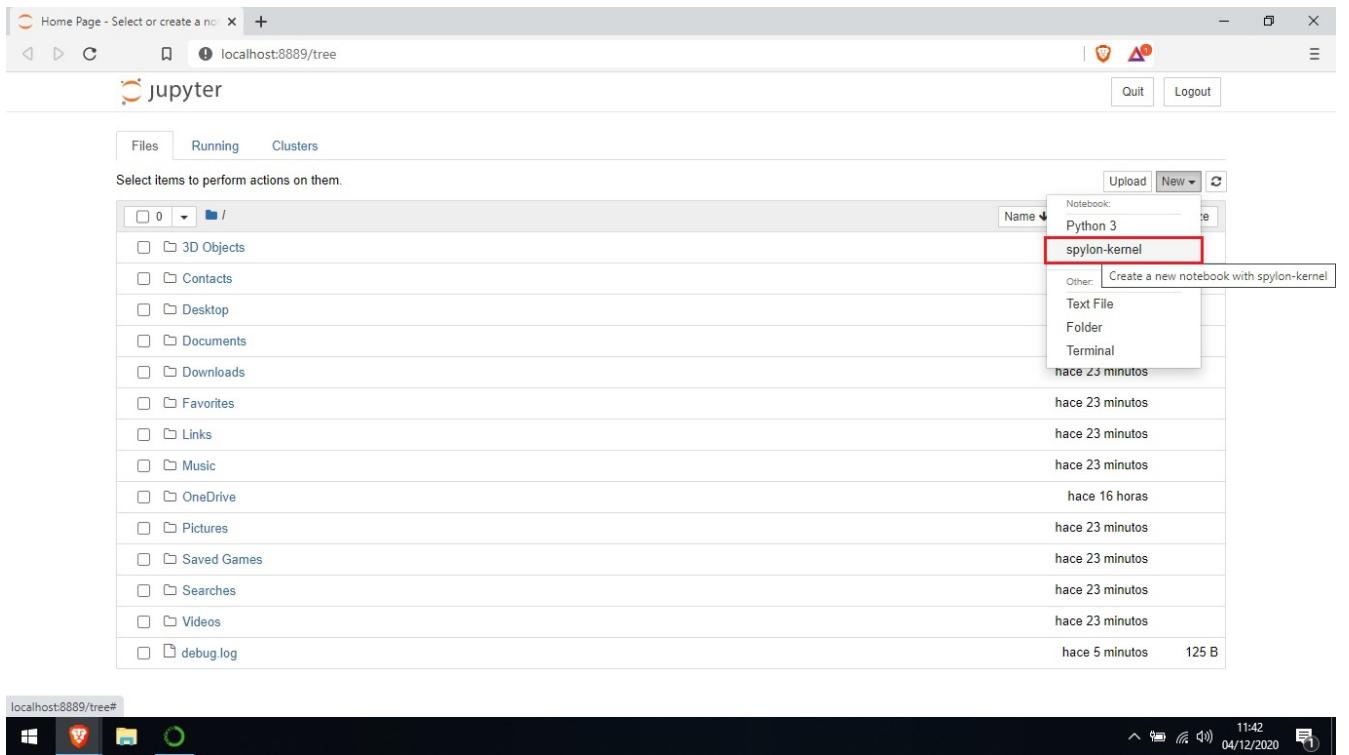
#### 2.4.8. Scala desde Jupyter Notebook

Abrimos la consola de anaconda

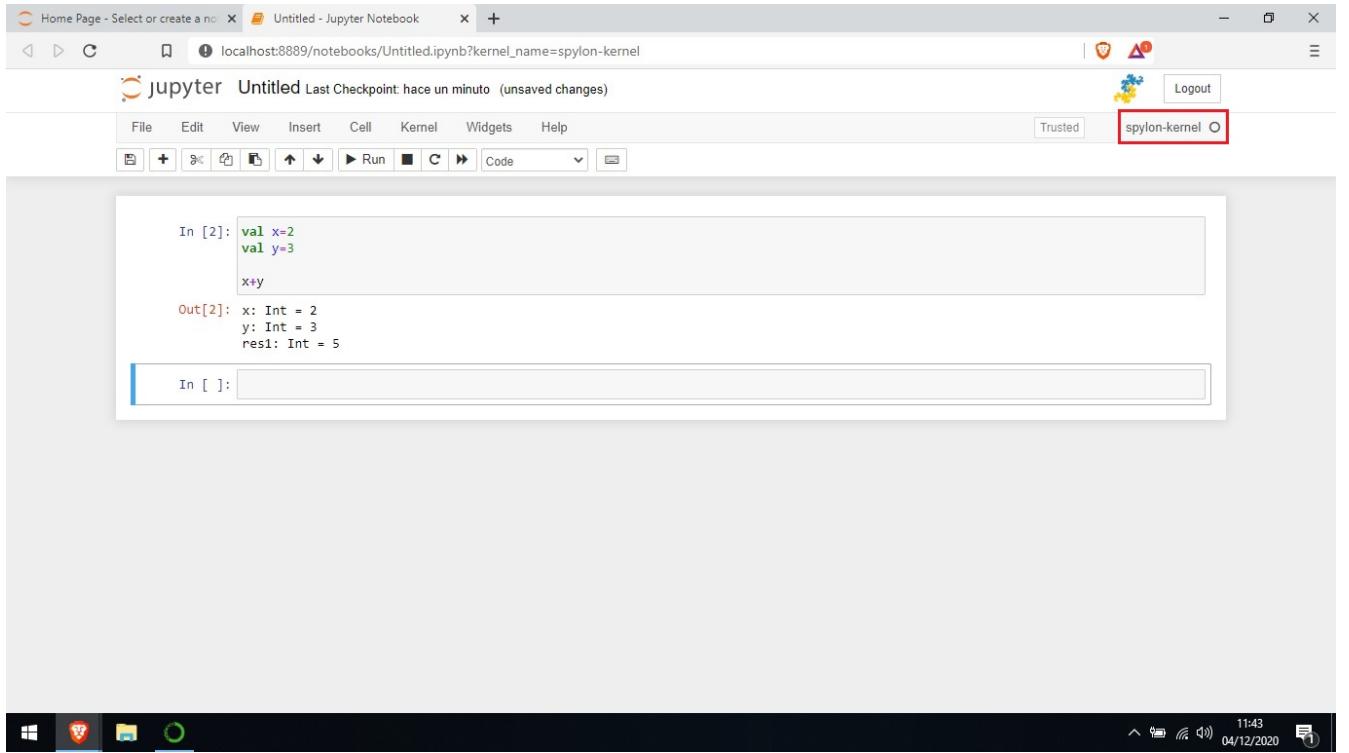
Ejecutamos “pip install spylon-kernel”

Añadimos que nos permita seleccionar el kernel de scala desde el notebook. Para ello ejecutamos ‘python -m spylon\_kernel install’

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:



### 3. Servicios en la nube

#### 3.1. 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

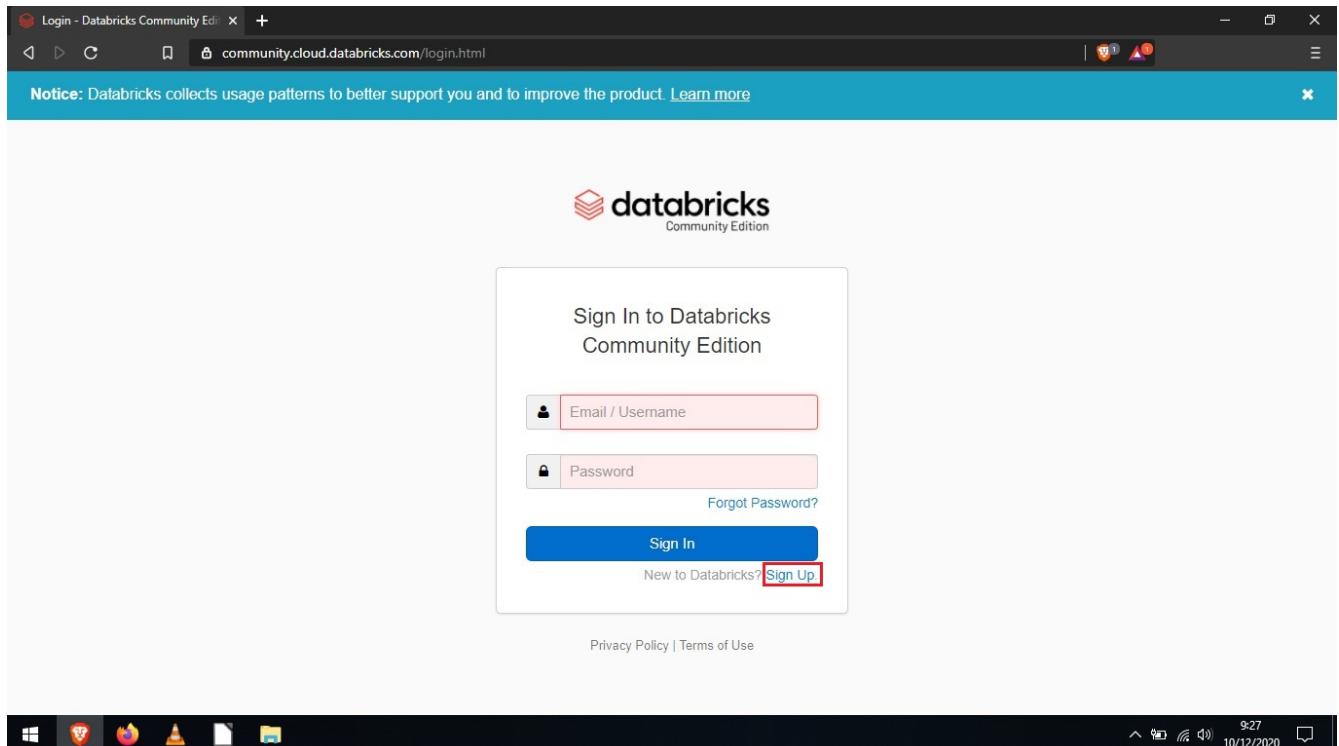
analítica Big Data e inteligencia artificial con Spark de una forma sencilla y colaborativa. Esta plataforma tambien está disponible como servicio cloud en Microsoft Azure y Amazon Web Services (AWS).

### 3.1.1. Registro

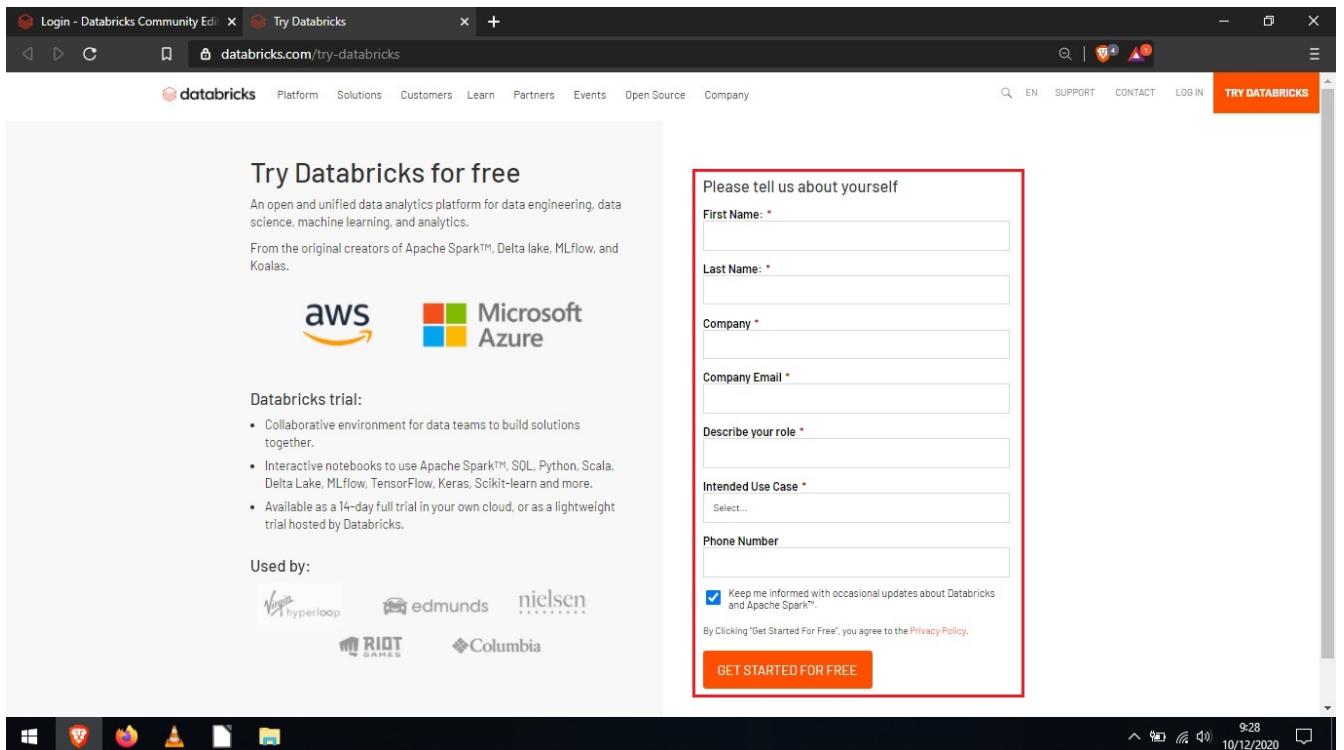
Databricks nos permite probar su funcionamiento con una serie de limitaciones por medio de la llamada 'Databricks Community Edition'. Lo primero que deberemos hacer para poder acceder a esta version de prueba sera entrar en la siguiente web:

<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 correo ya podemos iniciar sesión. Lo que encontraremos al iniciar sesión será un Dashboard de esta manera:

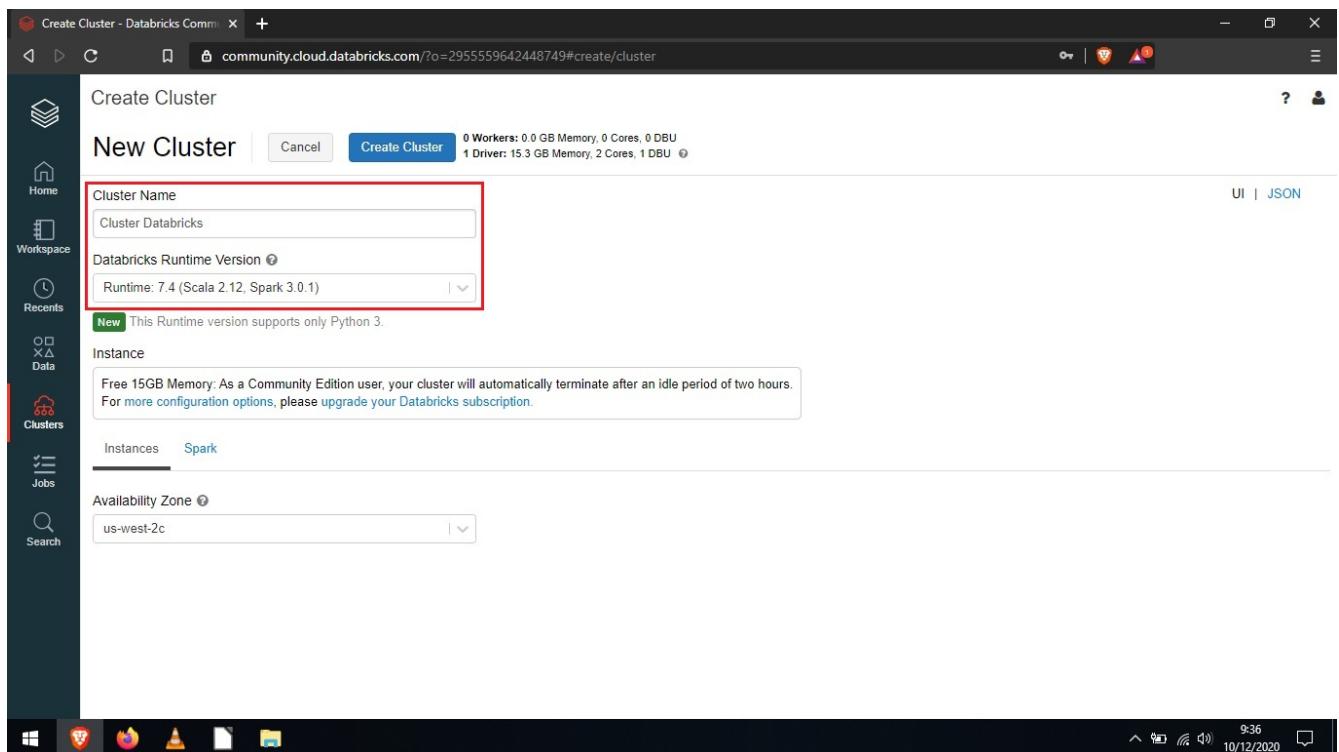
### 3.1.2. Creación de Cluster

Como podemos ver, se nos incluye una guía rápida y las principales funciones básicas. Lo que haremos ahora es ir a 'Clusters' en el menú lateral para crear nuestro Cluster personalizado.

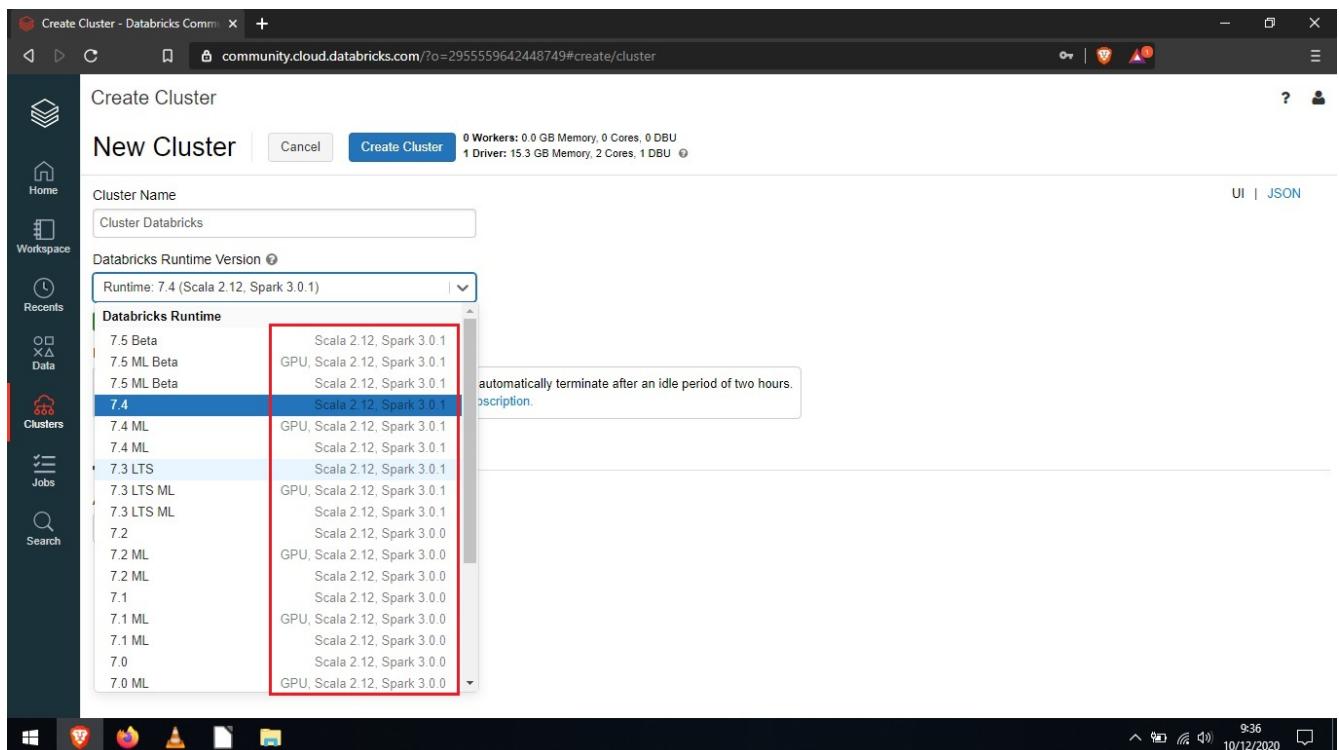
The screenshot shows the Databricks Community Edition homepage. On the left, there's a sidebar with icons for Home, Workspace, Recents, Data, Clusters (which is highlighted with a red box), Jobs, and Search. The main content area has a title 'Welcome to databricks'. It features three cards: 'Explore the Quickstart Tutorial' (Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.), 'Import & Explore Data' (Quickly import data, preview its schema, create a table, and query it in a notebook.), and 'Create a Blank Notebook' (Create a notebook to start querying, visualizing, and modeling your data.). Below these cards are sections for 'Common Tasks' (New Notebook, Create Table, New Cluster, New Job, New MLflow Experiment), 'Recents' (Recent files appear here as you work), and 'What's new in v3.34' (View latest release notes). At the bottom, there's a navigation bar with browser icons and a status bar showing the URL <https://community.cloud.databricks.com/?o=2955559642448749#setting/clusters>, the date 10/12/2020, and the time 9:35.

The screenshot shows the 'Clusters' page in Databricks. The sidebar is identical to the previous screenshot. The main area has a title 'Clusters' and tabs for 'All-Purpose Clusters' (selected) and 'Job Clusters'. A red box highlights the '+ Create Cluster' button. Below the tabs is a search bar with filters for 'All', 'Created by me', 'Accessible by me', and a 'Filter...' button. A table below shows columns for Name, State, Nodes, Runtime, Driver, Worker, Creator, and Actions. The table is currently empty, displaying the message 'No Clusters'. At the bottom, there's a pagination bar showing '0 - 0 of 0' and a status bar at the bottom right.

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 subtitulos 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.

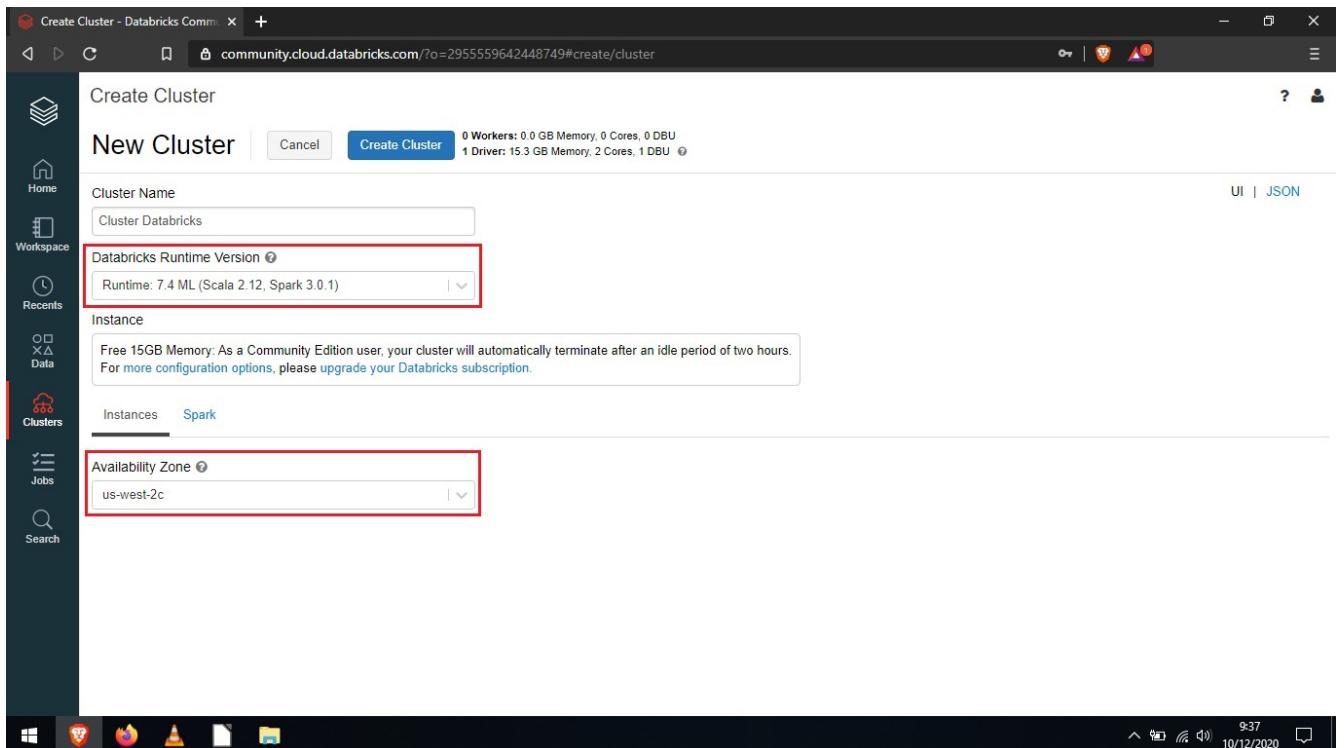
- Beta: Version de prueba que esta en fase de pulido. Solo se recomienda si vamos a probar nuevas funcionalidades no incluidas en versiones anteriores mas estables.

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:

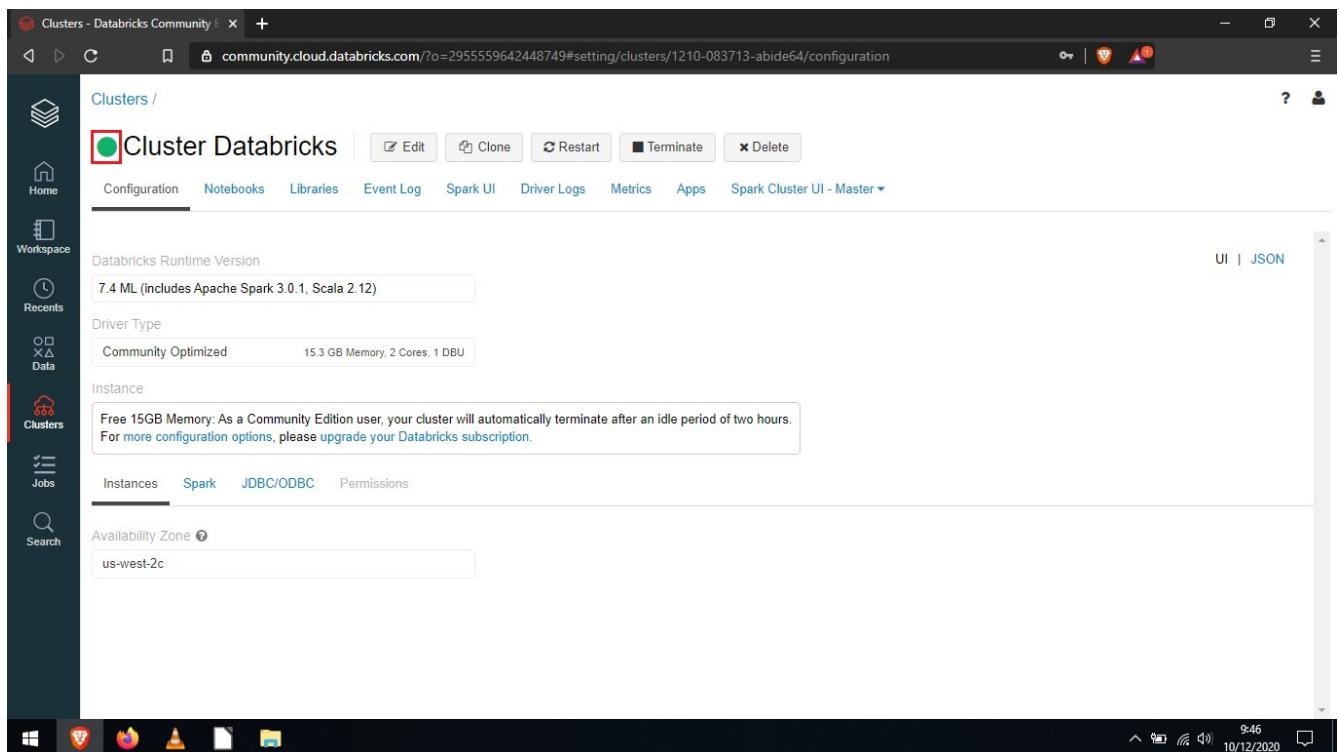
<https://docs.databricks.com/release-notes/runtime/releases.html>

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.



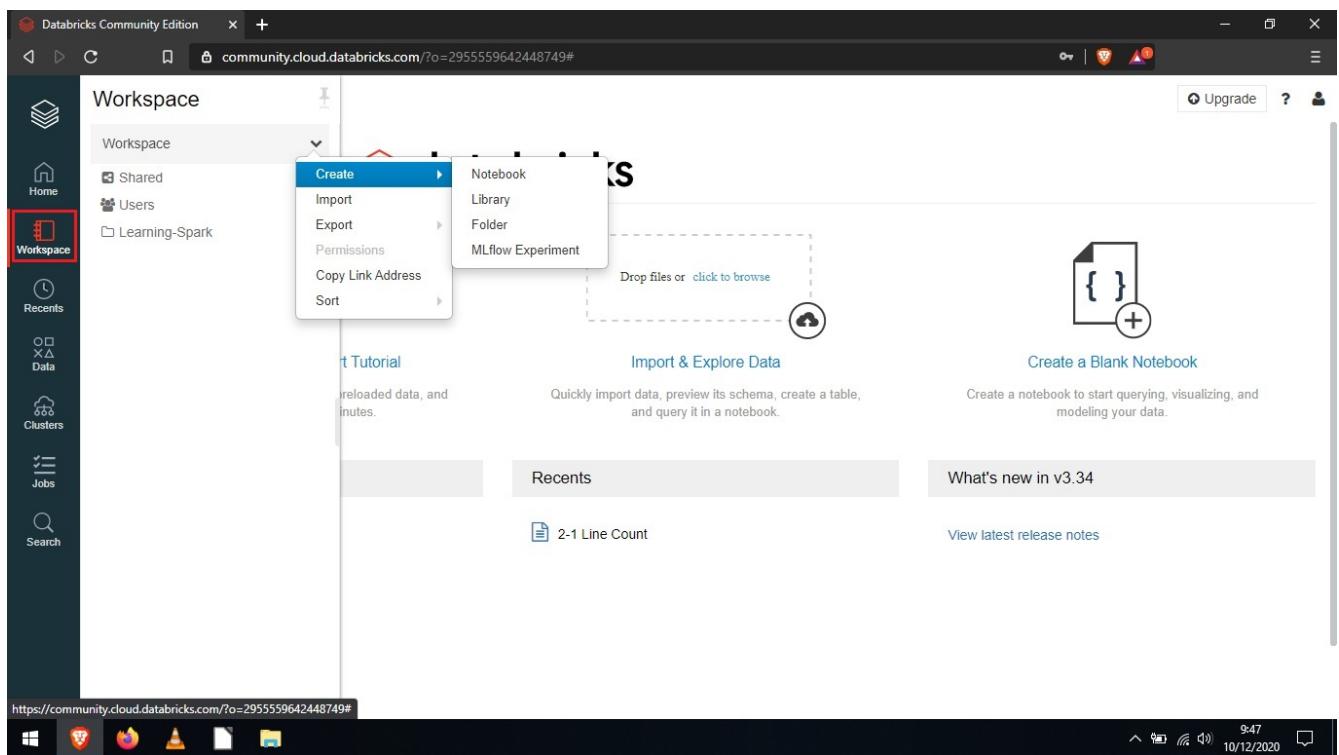
La ultima opcion que vemos es la de 'Availability Zone' que nos indica donde estara hospedado nuestro cluster. Esto nos afectara en la velocidad entre envio de peticion y respuesta. En este caso solo tenemos opciones de Estados Unidos por lo que lo dejamos tal cual esta.

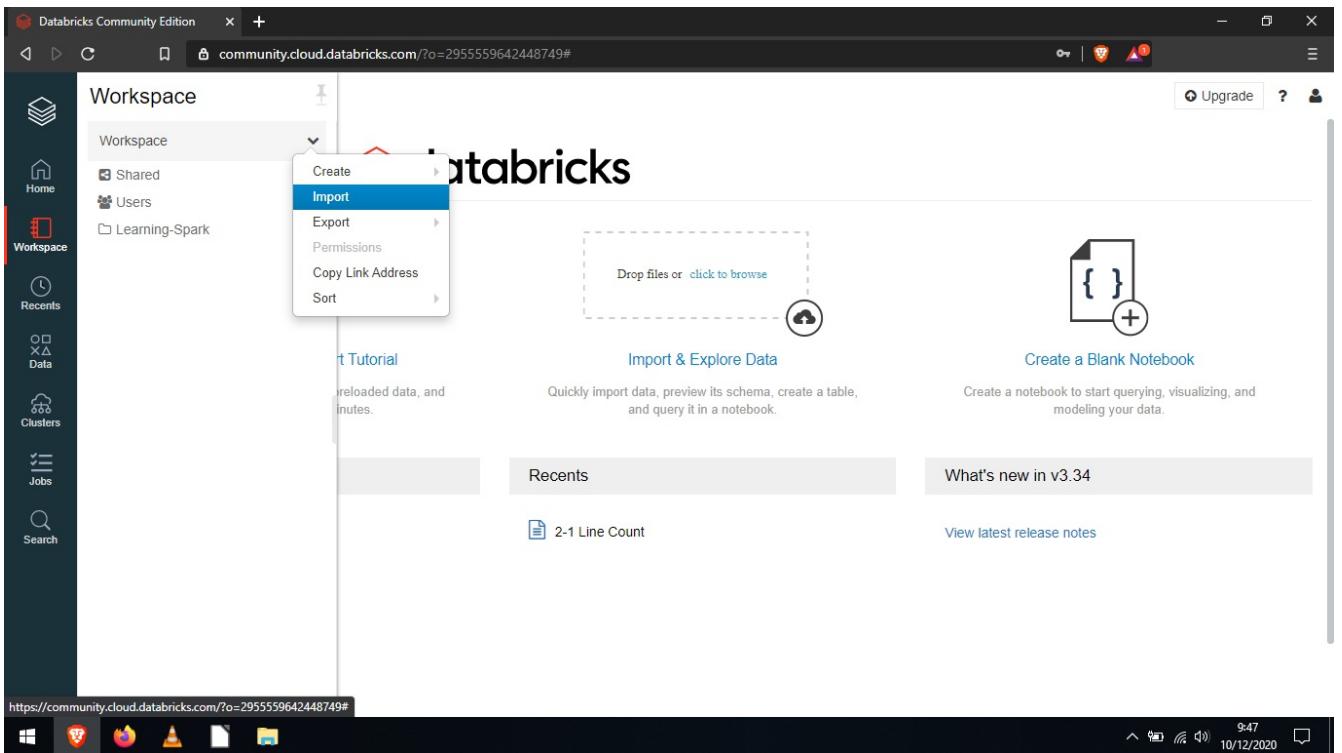
Una vez tenemos todas las caracteristicas deseadas solo tenemos que hacer click en 'Create Cluster'. El proceso de creacion puede tardar varios minutos, en cuanto este desplegado nuestro cluster veremos un indicador verde en la parte superior como en la siguiente imagen:



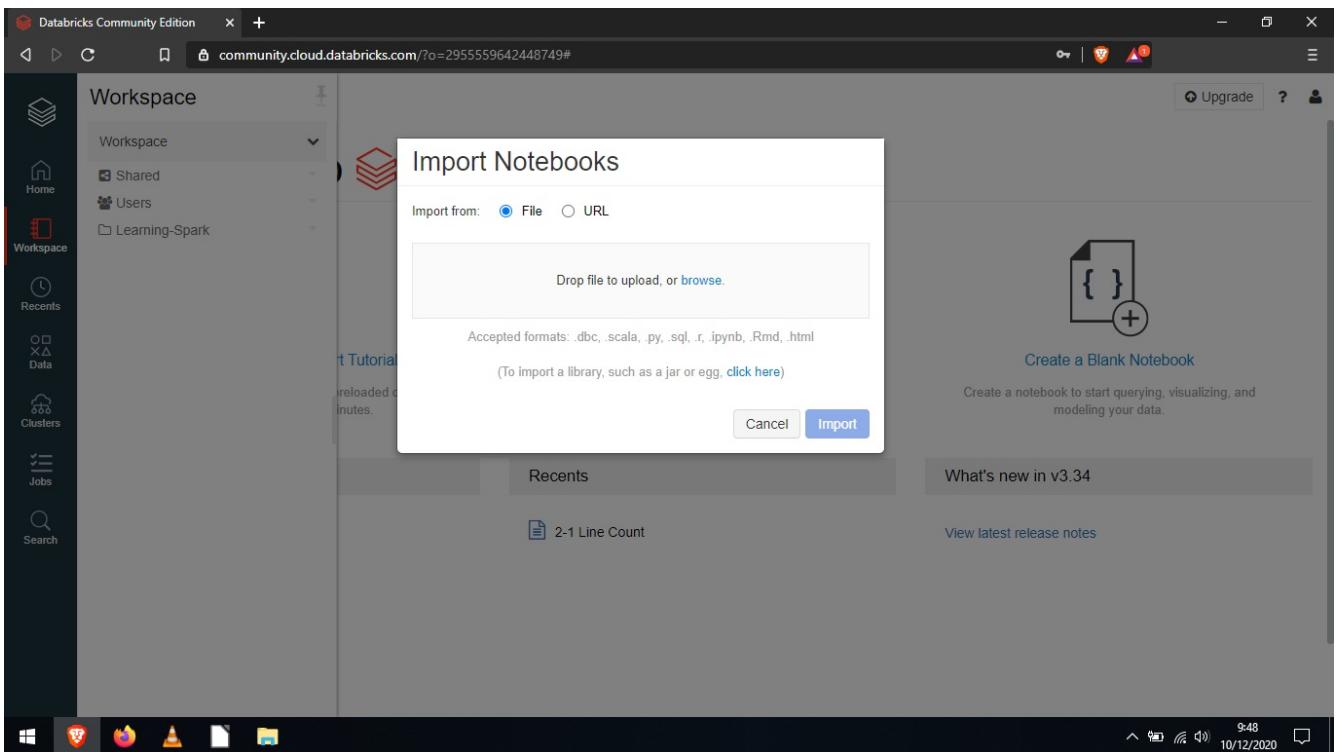
### 3.1.3. Creacion y carga de Notebooks

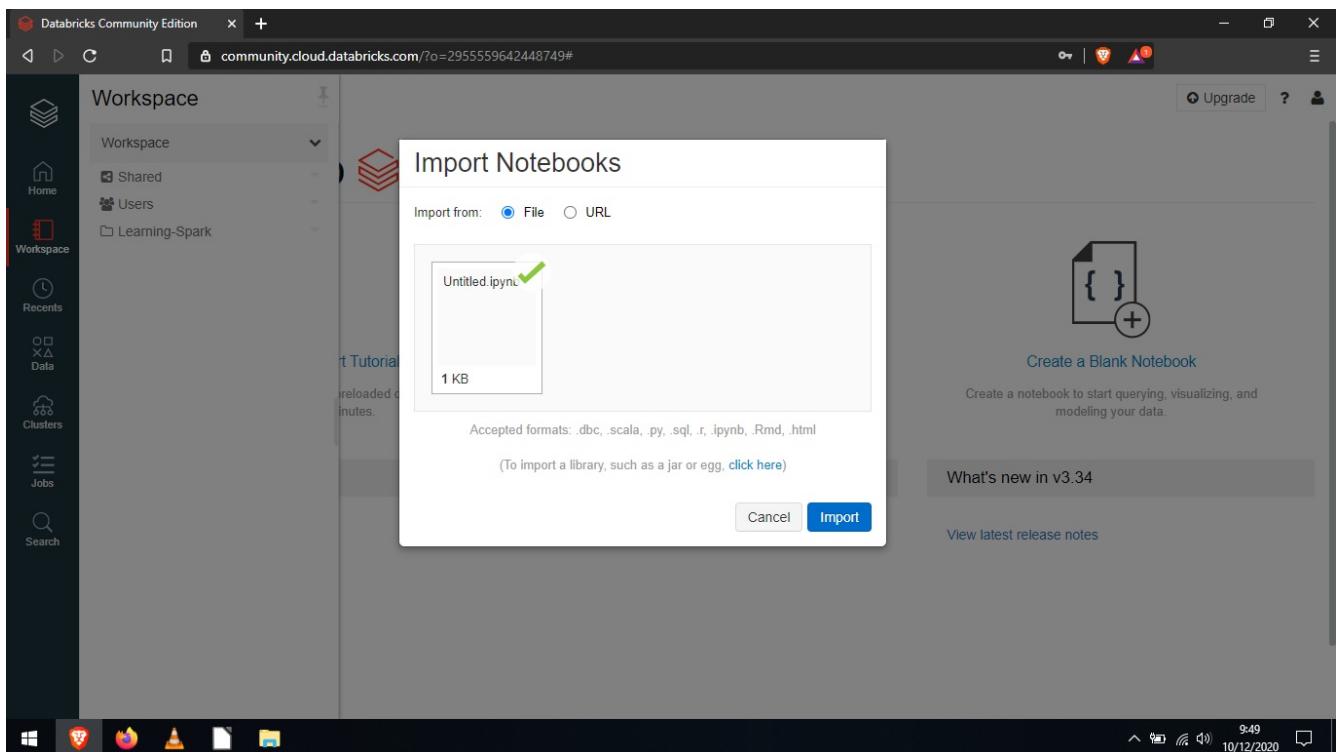
Ahora para poder trabajar con nuestro cluster tendremos que ir a la sección 'workspace' en el menu lateral. En el como podemos ver en las siguientes imagenes, podremos crear un nuevo notebook o importar uno.



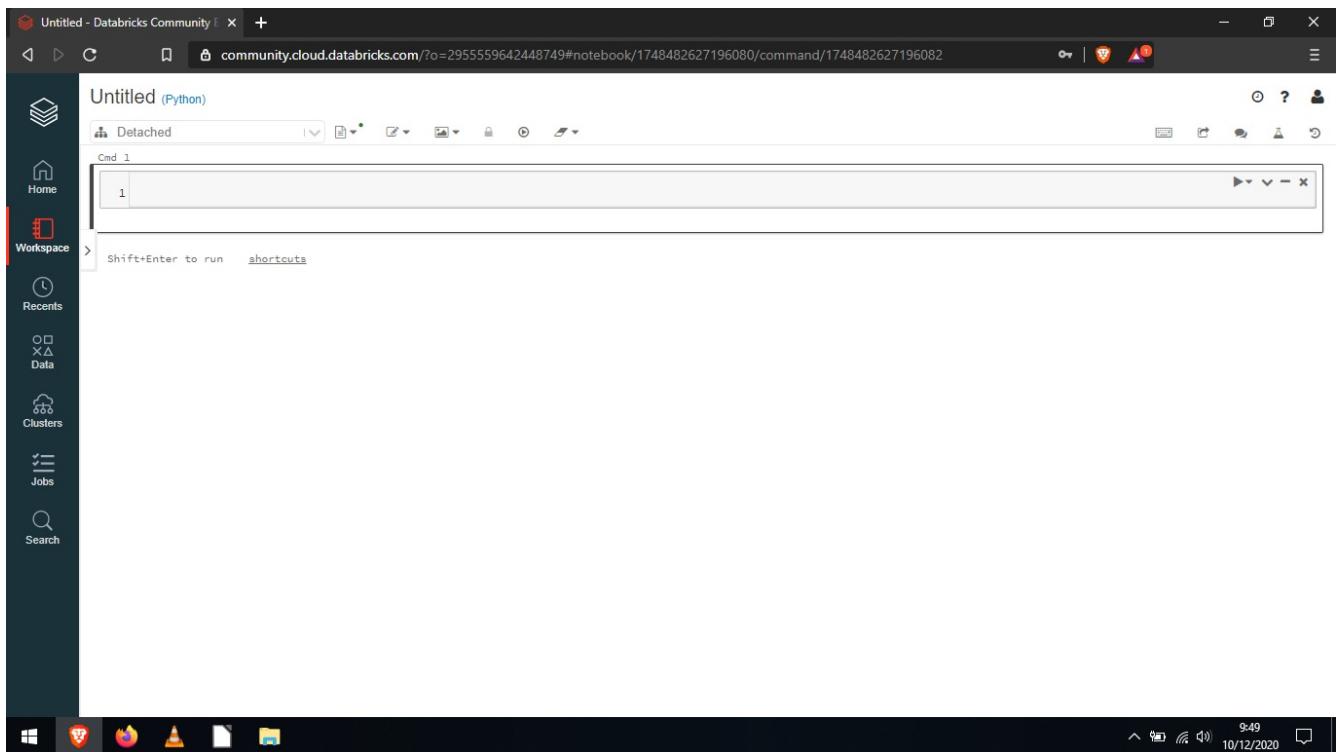


Si le damos a importar se nos abrirá una ventana donde podremos arrastrar el documento o seleccionarlo dentro de nuestro ordenador:

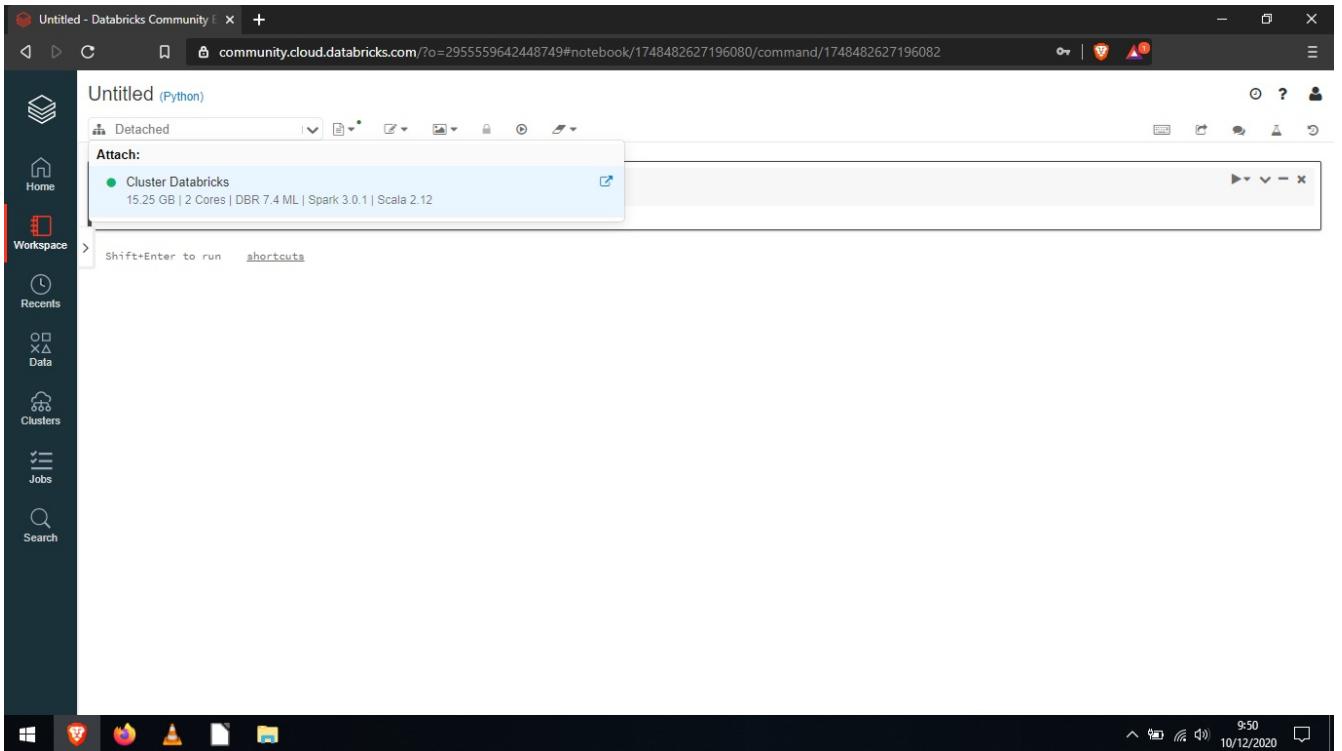




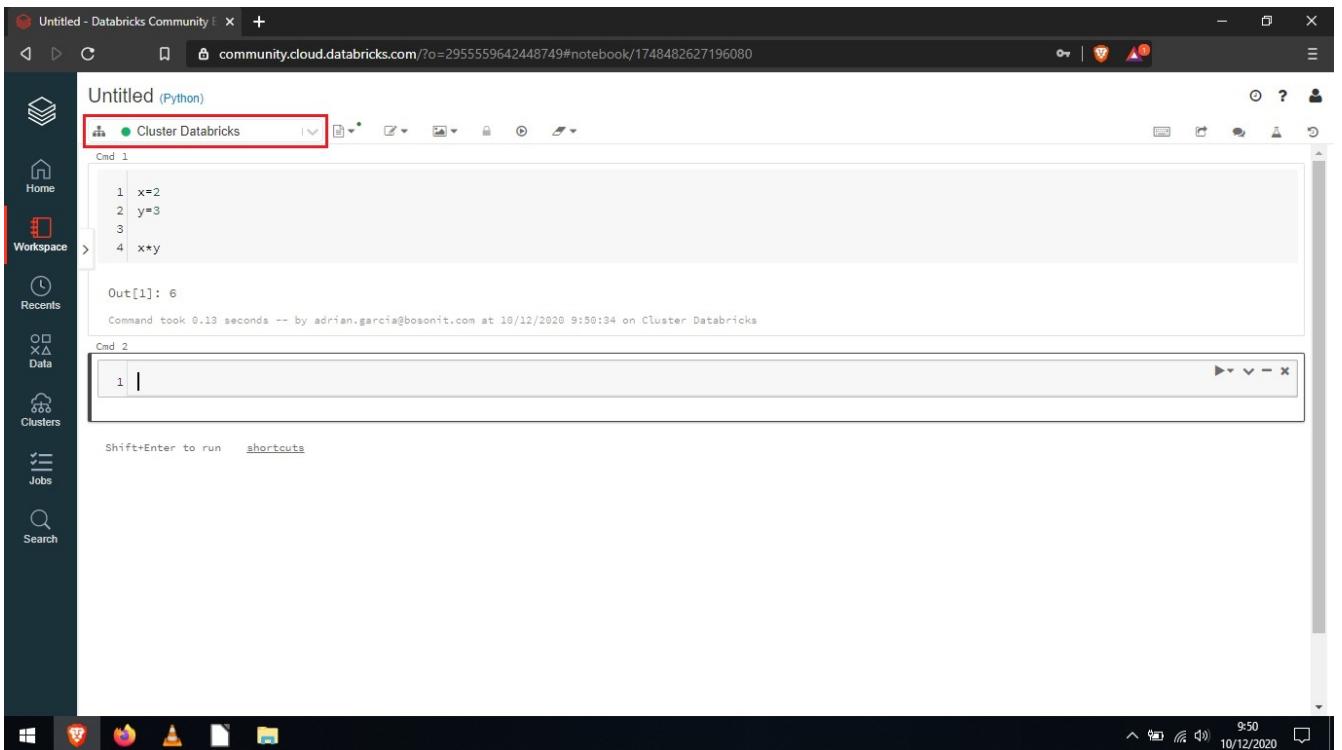
Entonces se nos abrirá el notebook de la siguiente forma:



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



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.



### 3.2. 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.

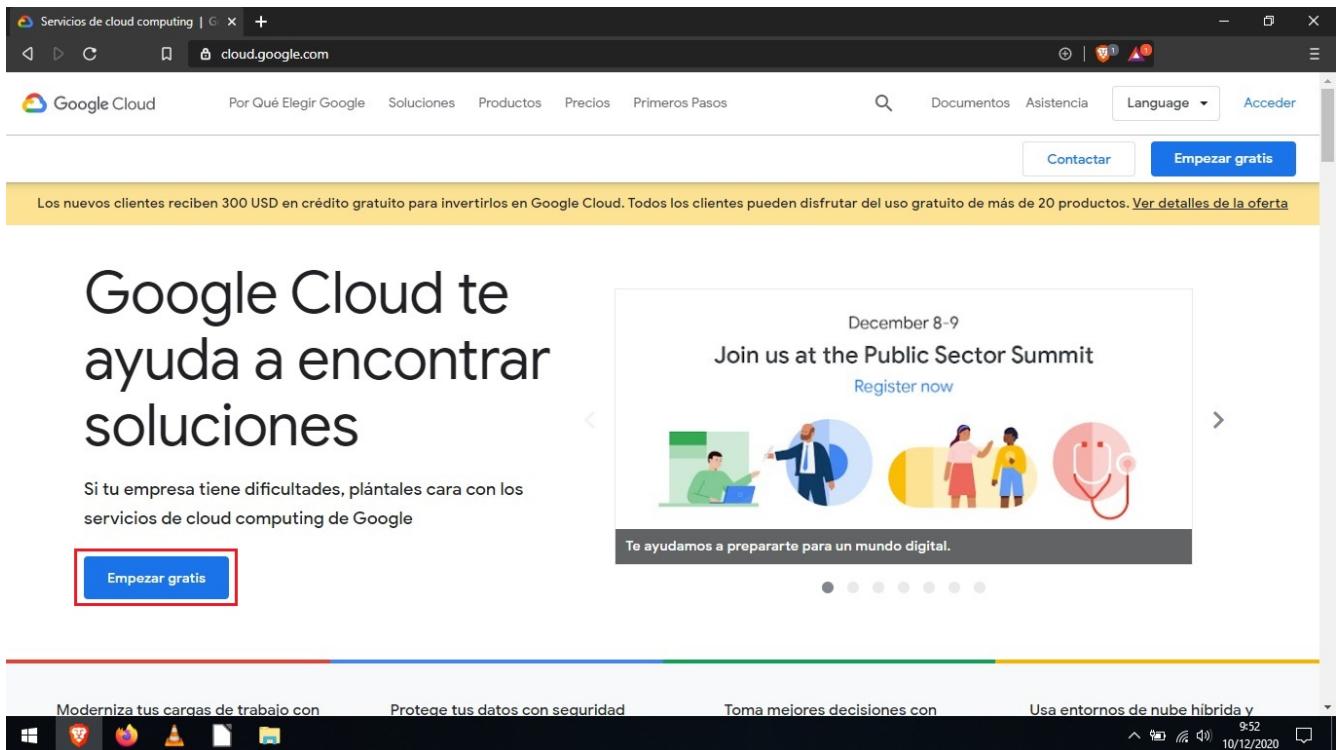
### 3.2.1. Registro

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\$.

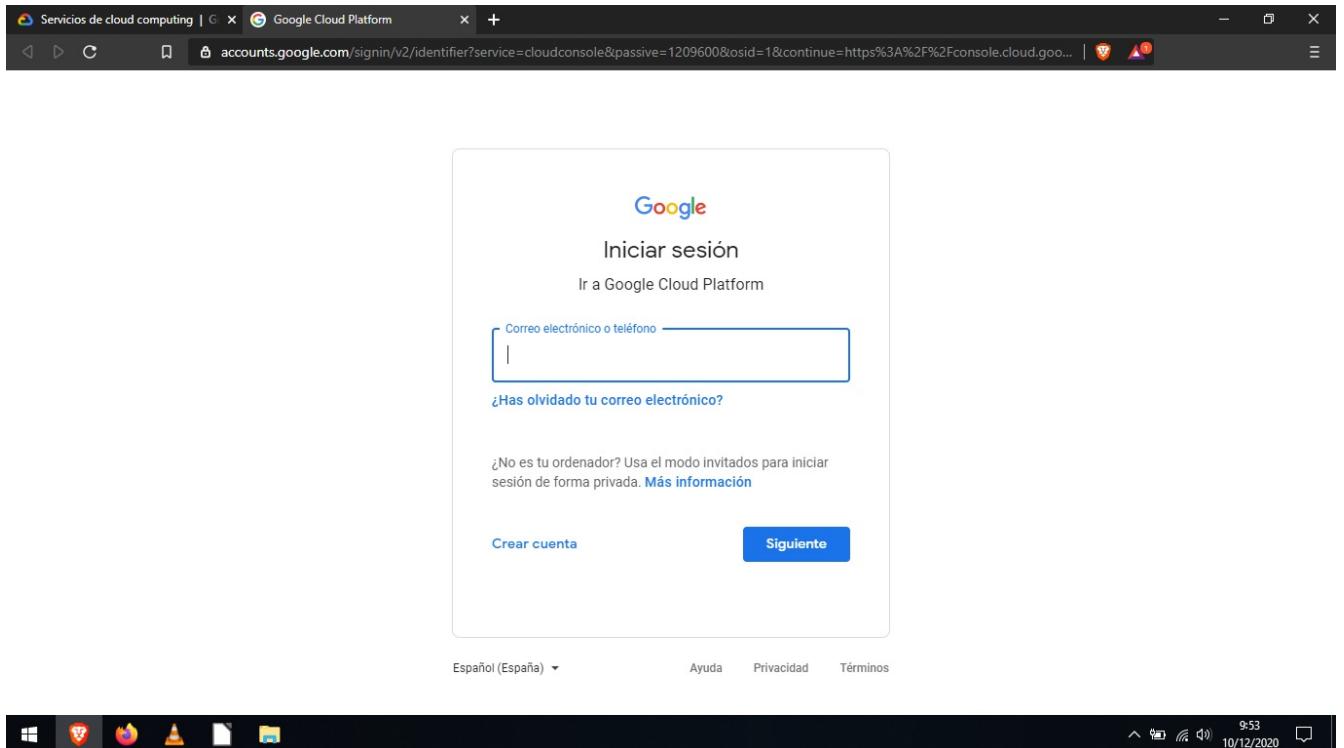
Lo primero que tendremos que hacer sera registrarnos si no tenemos cuenta de Google o crear una. Una vez la tengamos iremos a la siguiente pagina web:

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

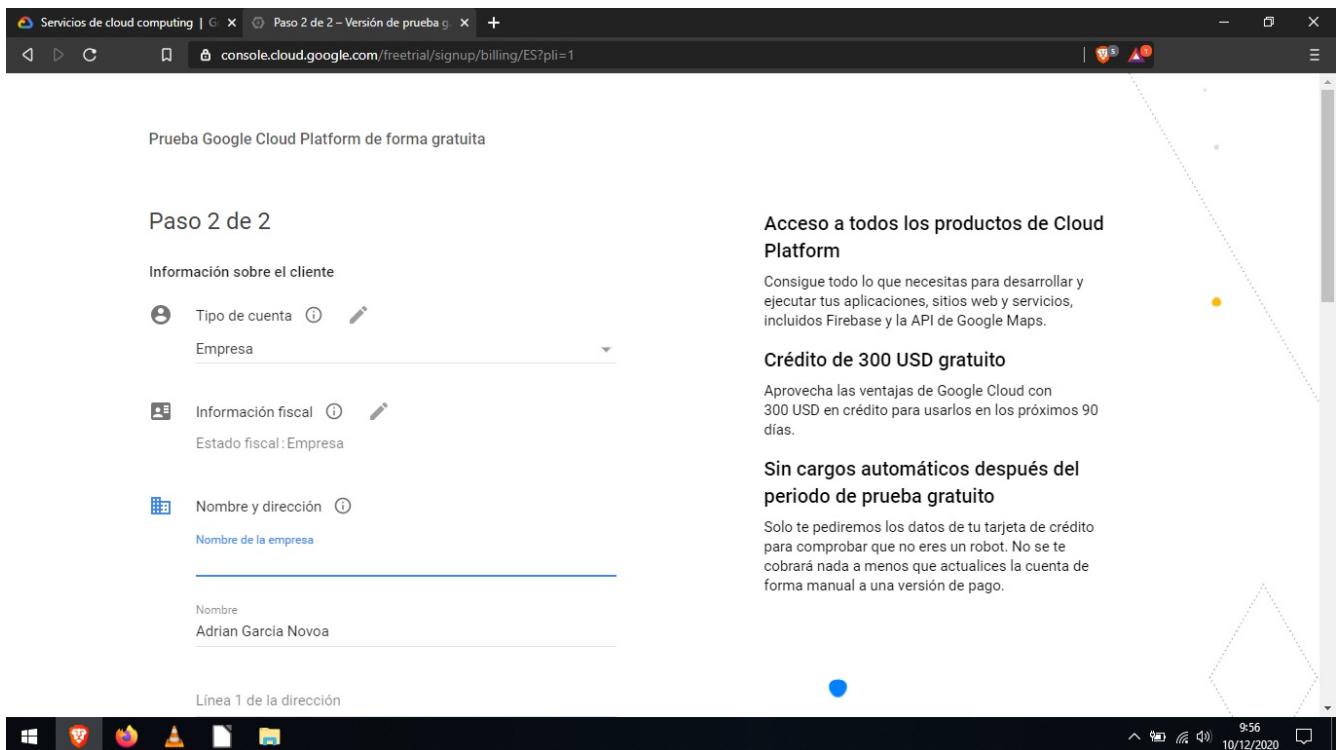
La pagina que nos aparecera sera la siguiente:



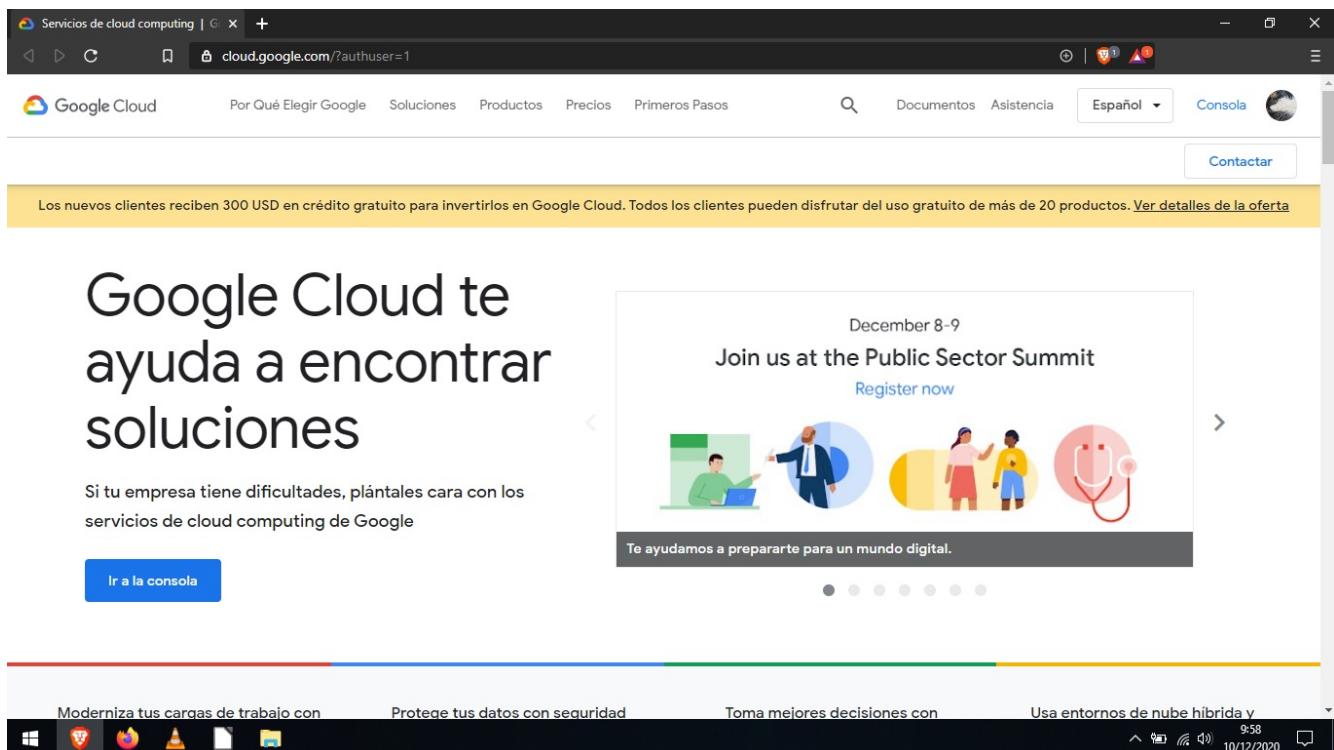
Como podemos observar en la parte superior, ya se nos ofrece el servicio gratuito de prueba. Le damos a 'Empezar gratis' y nos pedira que iniciemos sesion.



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.

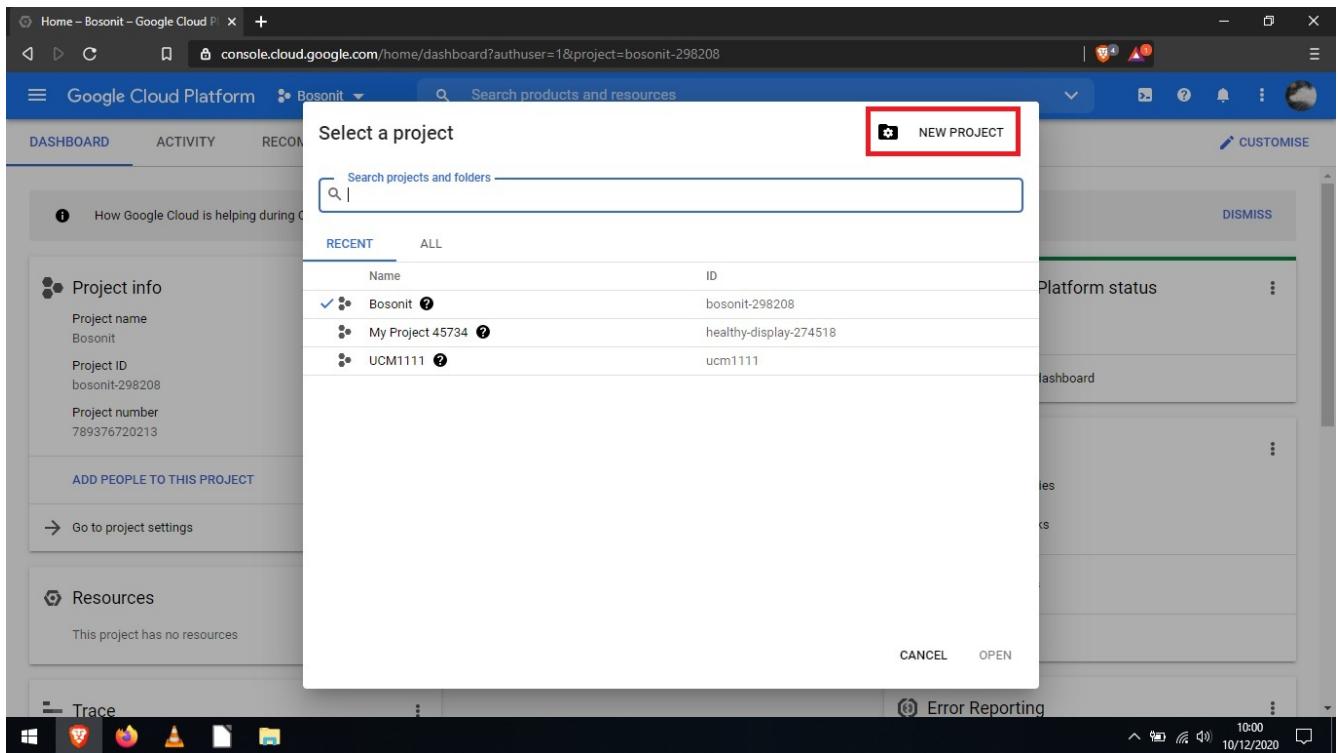


Una vez hemos completado los pasos y hemos iniciado sesion la web principal sera asi:

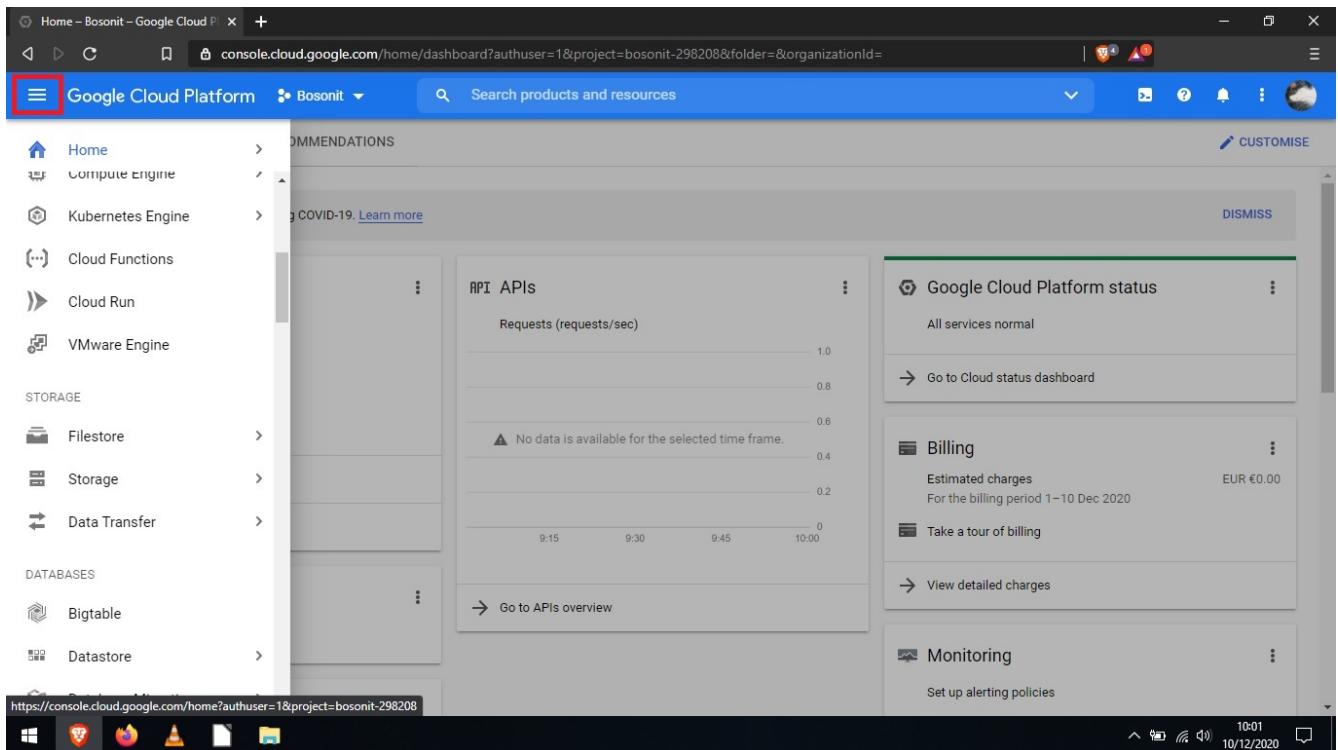


Donde ahora 'Empezar gratis' se ha convertido en 'Ir a la consola'. Hacemos click y nos lleva al Dashboard de la plataforma, donde tendremos que crear un proyecto haciendo click en 'Select project' en la parte superior:

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.



### 3.2.2. 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, there's a sidebar with various services: Home, Compute Engine, Kubernetes Engine, Cloud Functions, Cloud Run, VMware Engine, Storage, Data Transfer, Bigtable, and Datastore. The 'Storage' section is currently selected. A red box highlights the 'Pin' icon next to 'Storage'. A dropdown menu for 'Storage' is open, showing options: Browser (which is highlighted with a red box), Monitoring, and Settings. The main content area displays the 'API APIs' section with a chart showing requests per second over time. To the right, there are cards for Google Cloud Platform status (All services normal), Billing (Estimated charges: EUR €0.00), and Monitoring (Set up alerting policies). The URL in the browser bar is <https://console.cloud.google.com/storage?authuser=1&project=bosonit-298208&prefix=>.

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 Google Cloud Storage browser. At the top, there's a header with 'Storage browser - Storage - Bosonit' and a '+' button labeled '+ CREATE BUCKET'. Below the header, there's a sidebar with 'Storage', 'Browser', 'Monitoring', and 'Settings'. The main area is titled 'Storage browser' and shows a table with columns: Name (sorted by name), Created, Location type, Location, and Default storage class. A message says 'No rows to display'. On the right, there's an 'INFO PANEL' with tabs for 'PERMISSIONS' (selected) and 'LABELS'. It says 'Select a bucket' and 'Please select at least one resource'. At the bottom, there's a call-to-action: 'Store and retrieve your data' with the text 'Get started by creating a bucket – a container where you can organise and control access to your data and files in Cloud Storage.' The URL in the browser bar is <https://console.cloud.google.com/storage/browser?authuser=1&project=bosonit-298208&prefix=>.

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  GB  
\$0.026 per GB-month

Data retrieval size  GB  
Free

**Operations**

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

Class B operations  per month  
\$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

- europe-north1 (Finland)
- europe-west1 (Belgium)
- europe-west2 (London)
- europe-west3 (Frankfurt)
- europe-west4 (Netherlands)
- europe-west6 (Zurich)

**CONTINUE**

**Monthly cost estimate**

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

**Storage and retrieval**

Storage size  GB  
\$0.020 per GB-month

Data retrieval size  GB  
Free

**Operations**

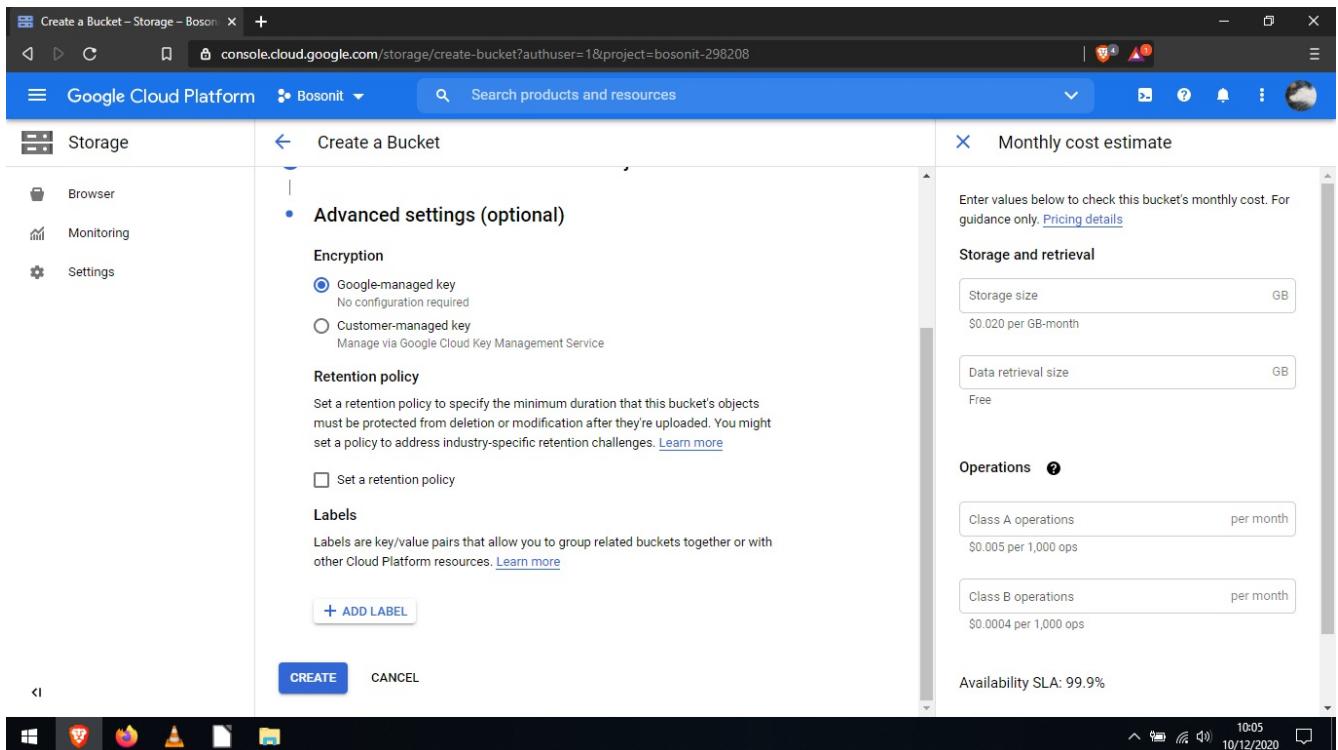
Class A operations  per month  
\$0.005 per 1,000 ops

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

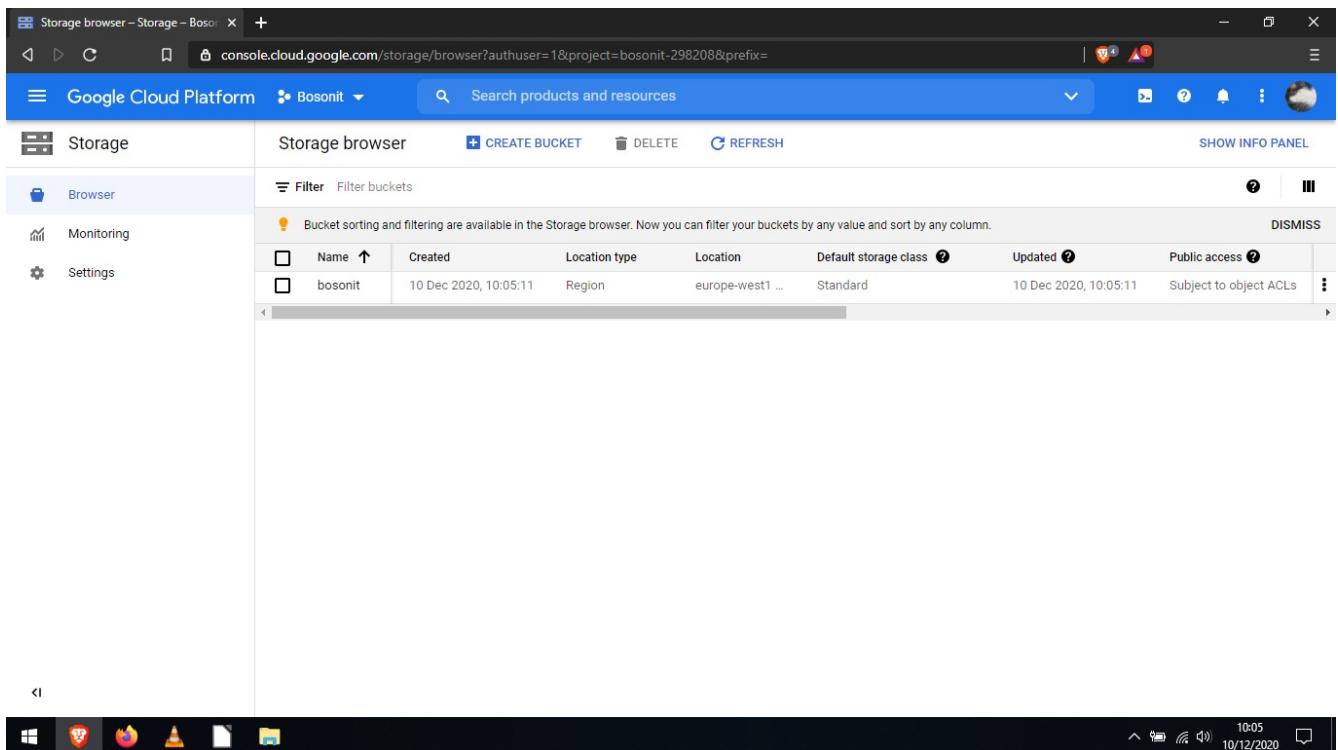
Availability SLA: 99.9%

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 explains that a storage class sets costs for storage, retrieval, and operations. Below this are sections for 'Choose how to control access to objects' and 'Advanced settings (optional)'. On the right, a 'Monthly cost estimate' panel provides details for a bucket with 0.020 GB of storage and free data retrieval. The bottom of the screen shows the Windows taskbar with icons for File Explorer, Task View, Task Manager, and others.

This screenshot shows the continuation of the 'Create a Bucket' wizard. The 'Name your bucket' step is now active, indicated by a checked checkbox. The rest of the steps from the previous screenshot ('Choose where to store your data', 'Choose a default storage class for your data', and 'Choose how to control access to objects') remain visible. The 'Access control' section is expanded, showing two options: 'Fine-grained' (selected) and 'Uniform'. The 'CONTINUE' button is visible at the bottom of the main panel. The right-hand 'Monthly cost estimate' panel remains the same. The Windows taskbar is visible at the bottom.



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



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 icons for 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. The table includes columns for Name, Created, Location type, Location, Default storage class, Updated, and Public access. A note at the top of the table says: 'Bucket sorting and filtering are available in the Storage browser. Now you can filter your buckets by any value and sort by any column.' There are also 'CREATE BUCKET', 'DELETE', and 'REFRESH' buttons at the top of the table.

### 3.2.3. 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 to the right, followed by the text 'Cloud Dataproc API' and 'Google'. Below this, it says '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 a heading 'Overview' and a paragraph: 'Manages Hadoop-based clusters and jobs on Google Cloud Platform.' It also has a 'About Google' section with text about Google's mission to organize the world's information. The 'Additional details' section includes links for 'Type: APIs & services', 'Last updated: 10/12/2019', and 'Service name: dataproc.googleapis.com'. The status bar at the bottom of the browser window shows the date as 10/12/2020 and the time as 10:15.

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:

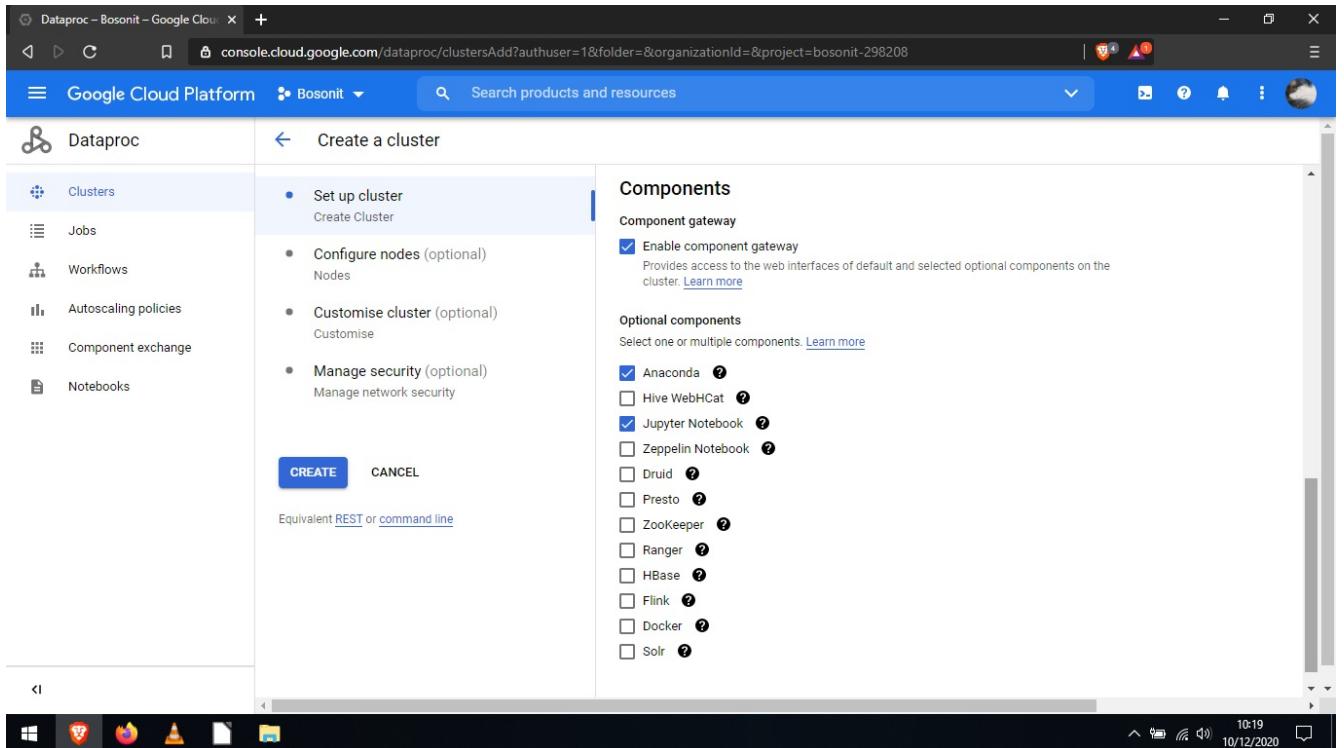
**Versioning**  
Use a custom image to load pre-installed packages. [Learn more](#)  
**Image Type and Version**  
1.3-debian10  
**Release date**  
First released on 8/16/2018.  
**Components**  
**Component gateway**  
 Enable component gateway  
Provides access to the web interfaces of default and 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.

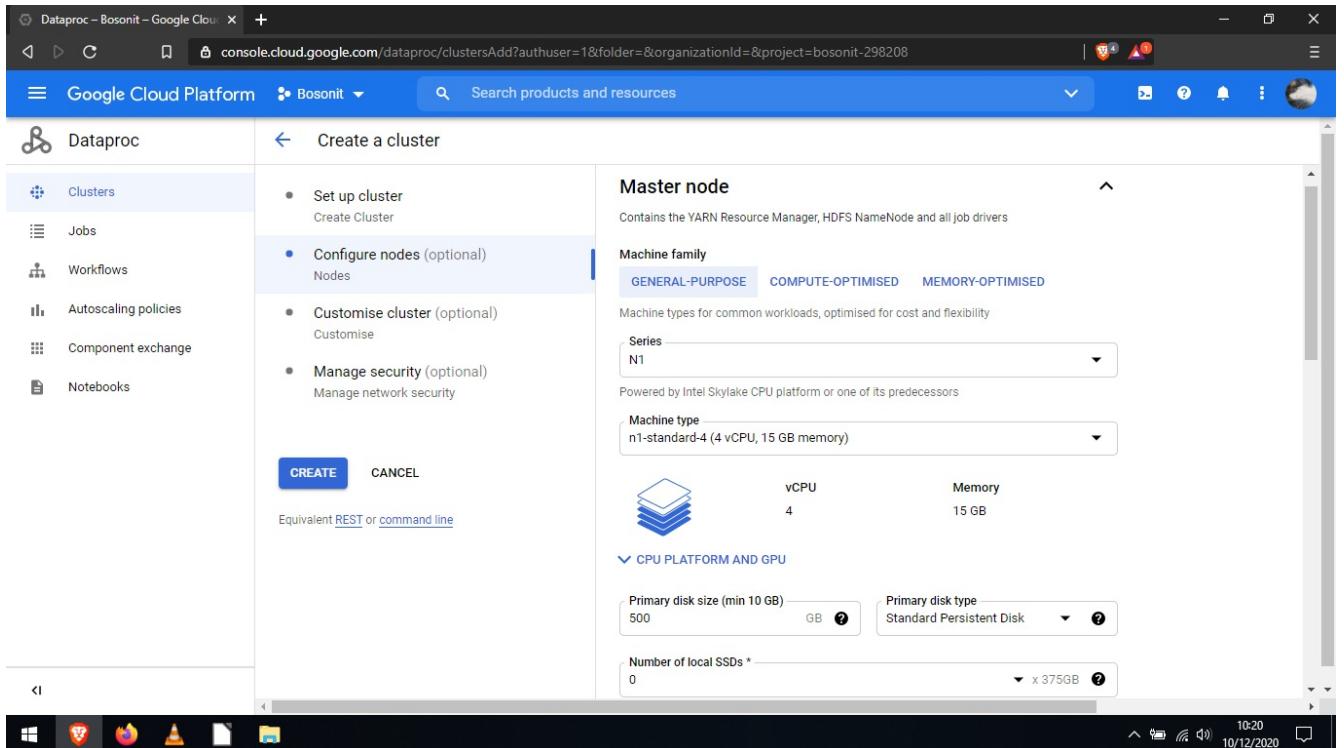
En nuestro caso recomendamos seleccionar la version 1.4 que incluye el sistema operativo Debian 10 con Hadoop 2.9 y Spark 2.4.

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

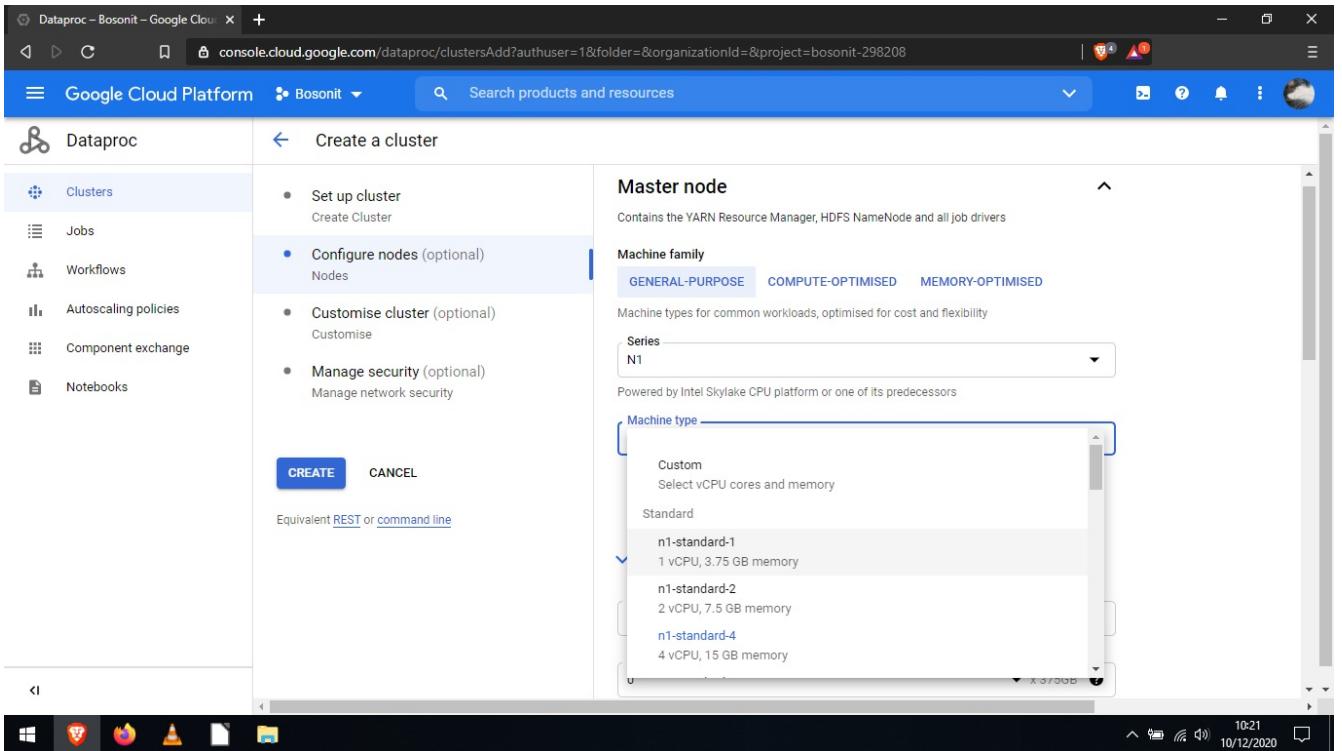
seleccionado Anaconda y Jupiter Notebook. Y no nos olvidemos de marcar 'Enable component gateway' para poder trabajar sin errores en el navegador.



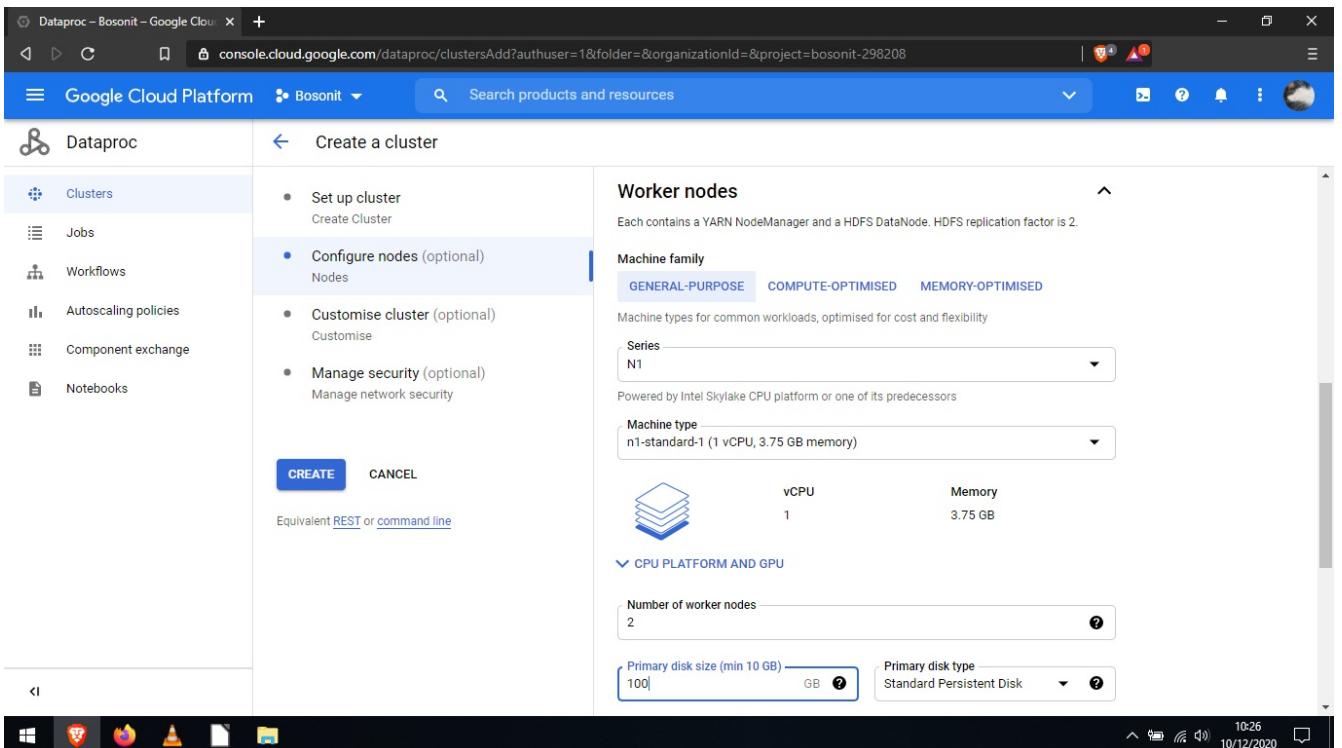
En el segundo apartado de la configuracion veremos y podremos modificar las caracteristicas del Master y los Workers:



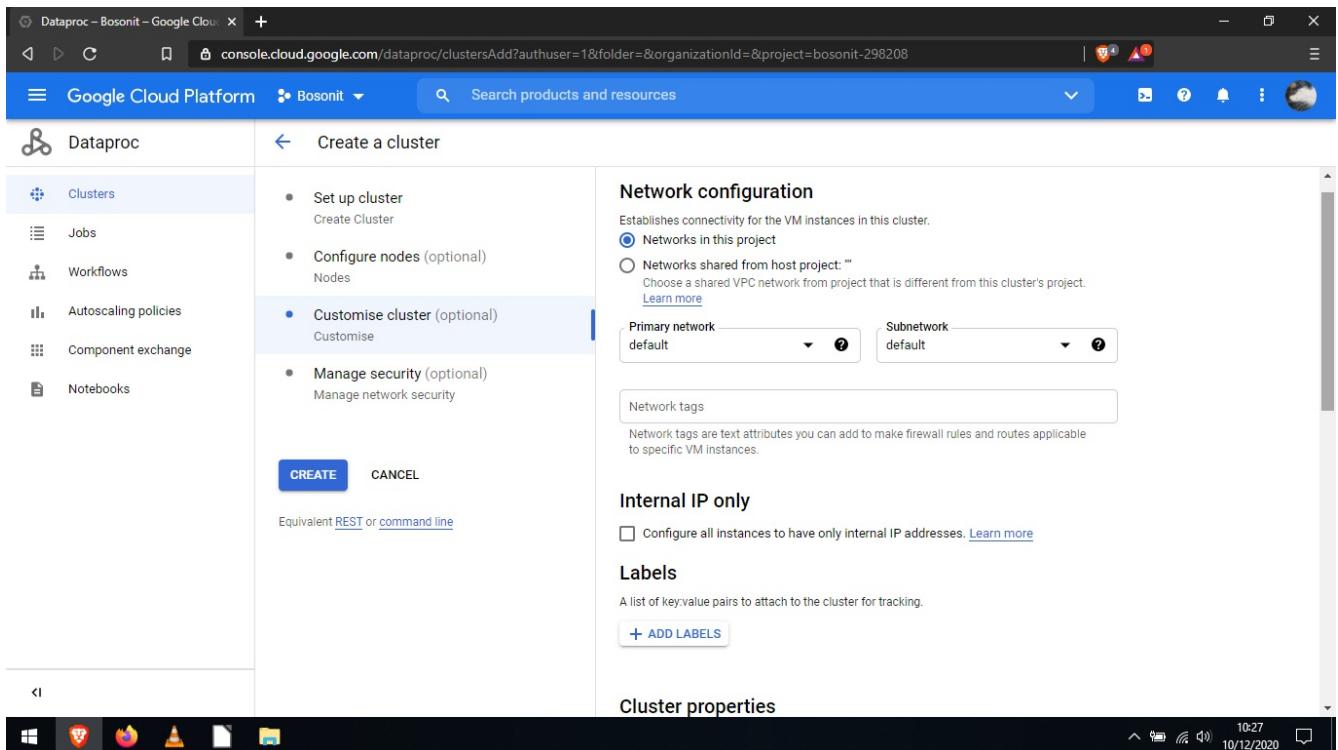
Para el Master vamos a seleccionar el mas pequeño ya que para nuestro aprendizaje no vamos a necesitar una gran cantidad de potencia de calculo y las caracteristicas por defecto en cuanto a tamaño de disco y tipo de disco:



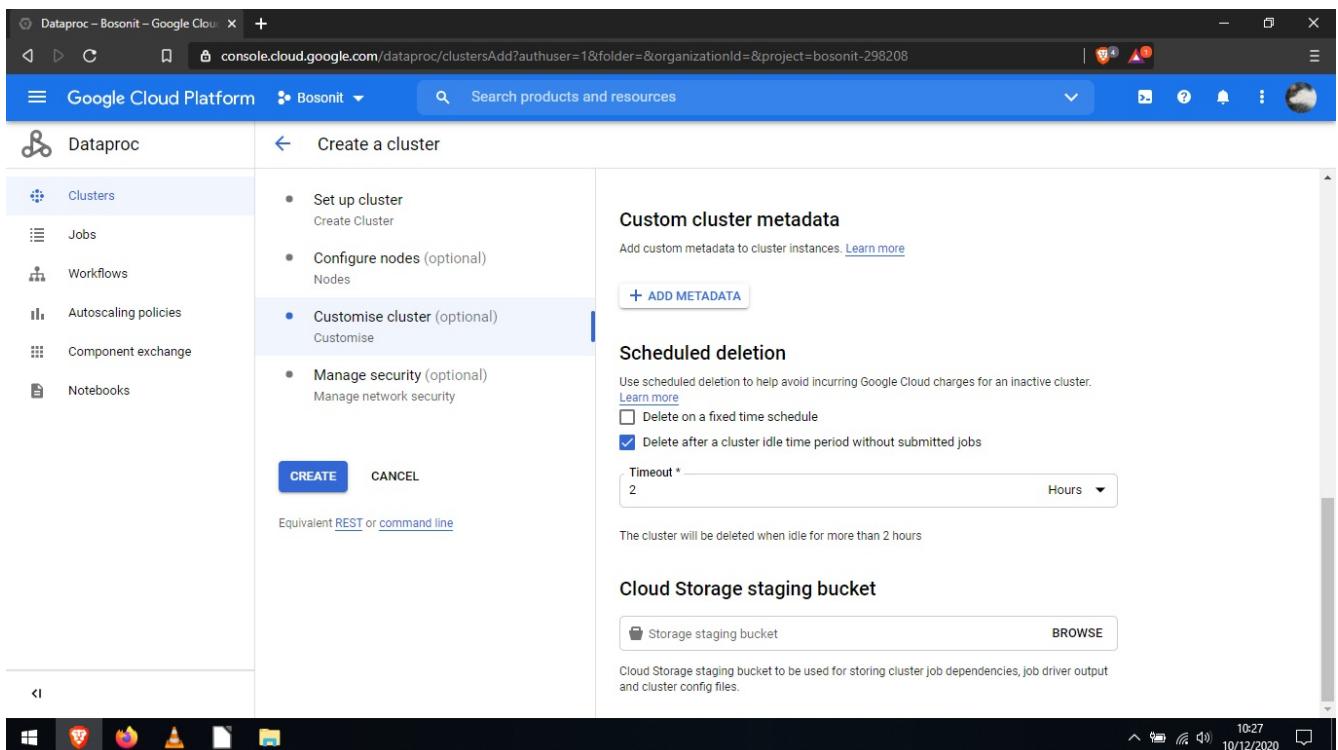
Si bajamos vemos la sección de Workers, donde podremos seleccionar cuantos queremos, la capacidad de cada uno de ellos y el tamaño de disco y tipo de disco. En mi caso como en el Master selecciono el de menos potencia y pongo el tamaño de los discos a 100. De todas formas se puede dejar por defecto sin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



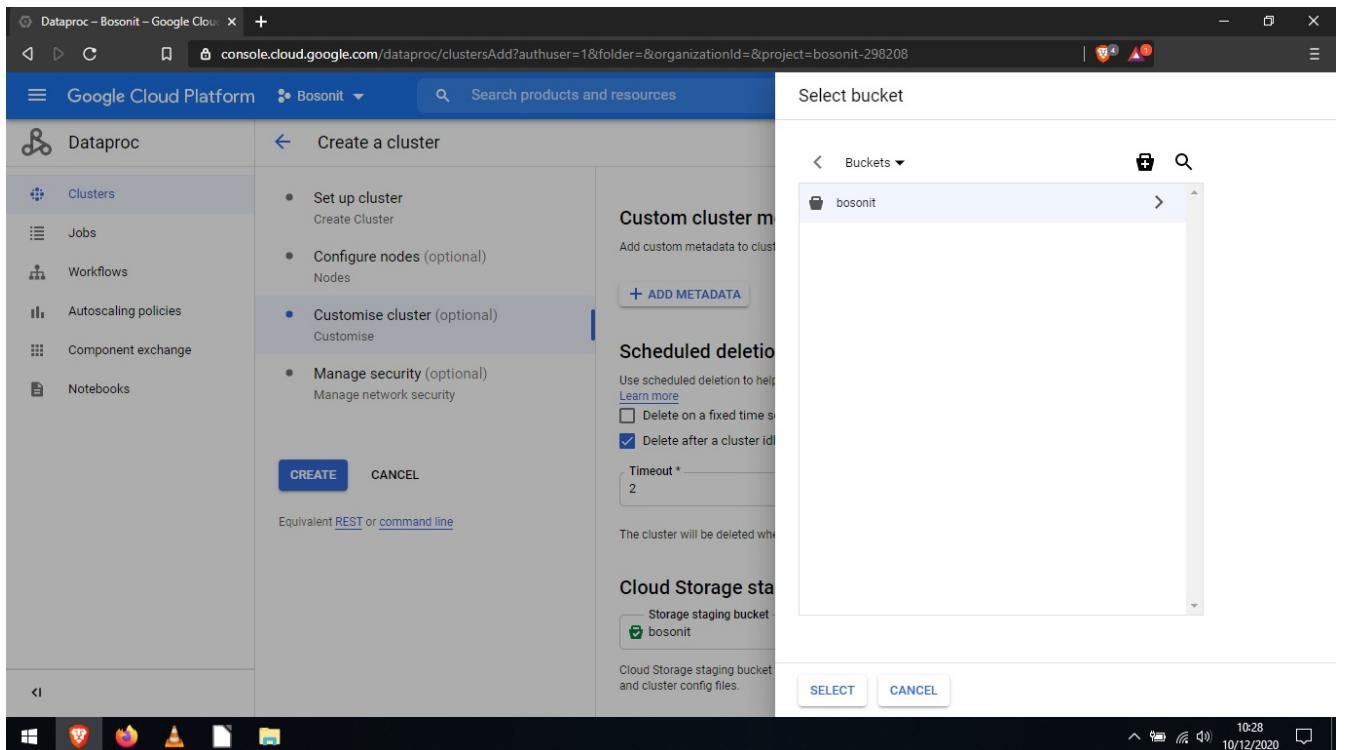
En la tercera sección vamos a dejar la primera parte de la siguiente manera por defecto:



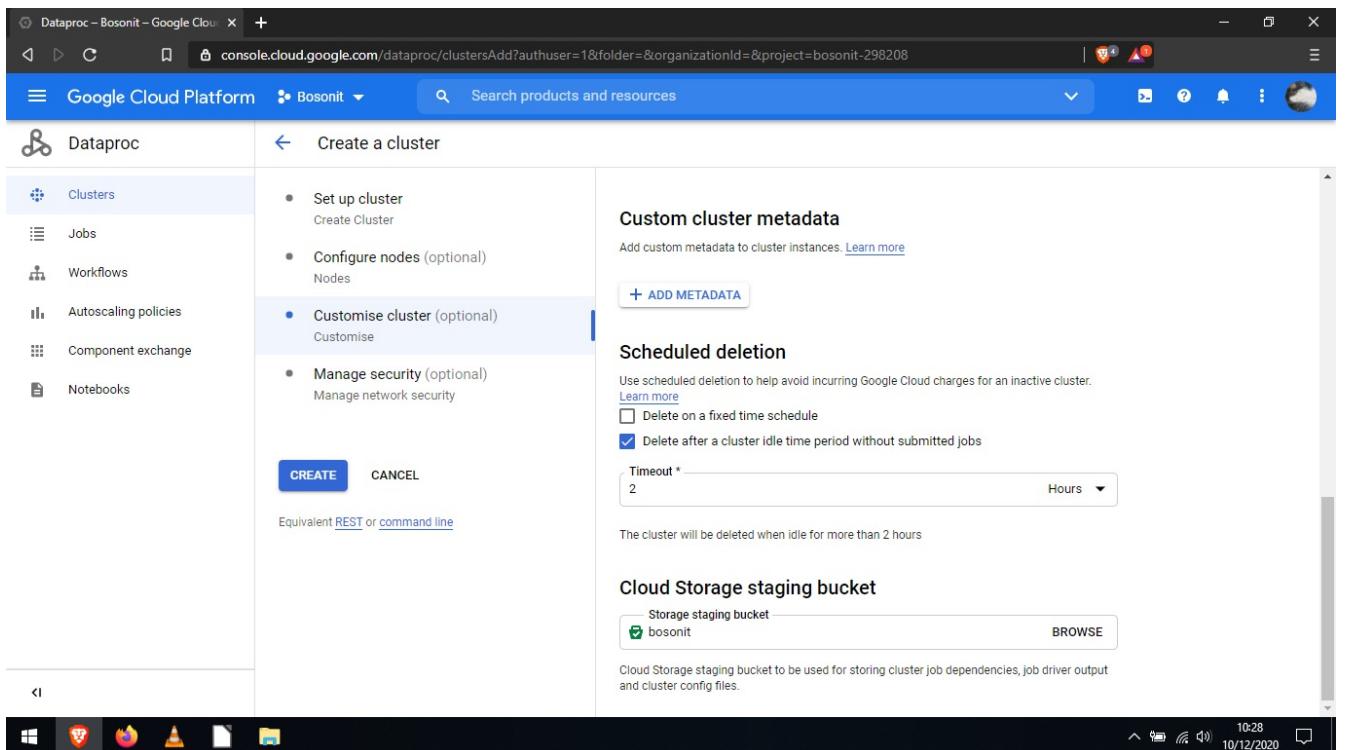
Y en la parte de abajo debemos fijarnos en activar la función de apagado tras un periodo de inactividad (2h en mi caso) para que el cluster se elimine cuando dejemos de usarlo, por si se nos olvida apagarlo.



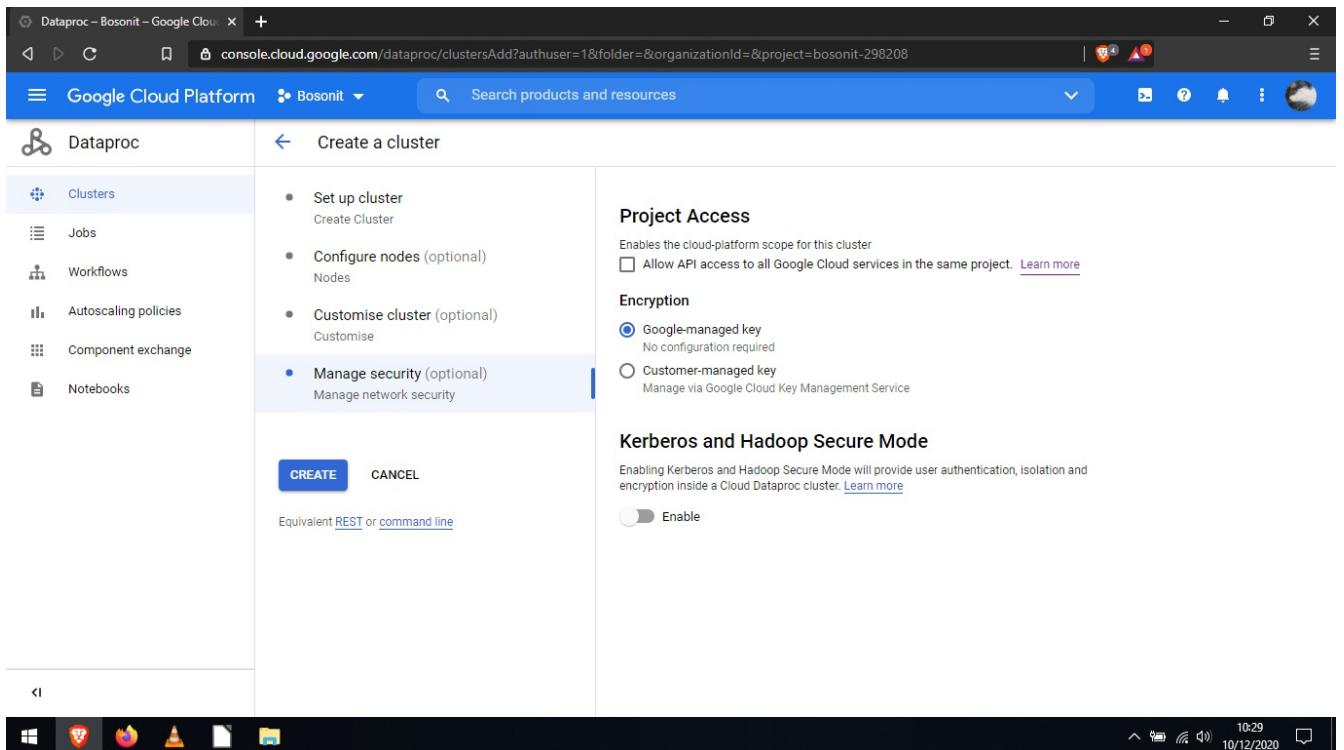
Pero aun mas importante es seleccionar 'Cloud Storage staging Bucket'. Aqui es donde seleccionaremos el Bucket que hemos creado en la sección anterior y donde guardaremos nuestros Dataset y Notebooks para que el cluster pueda trabajar con ellos y queden guardados aunque el cluster sea eliminado. Lo seleccionamos 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
europe-cluster-13cf	europe-west2	europe-west2-b	2	On	bosonit	10 Dec 2020, 10:29:37	Running

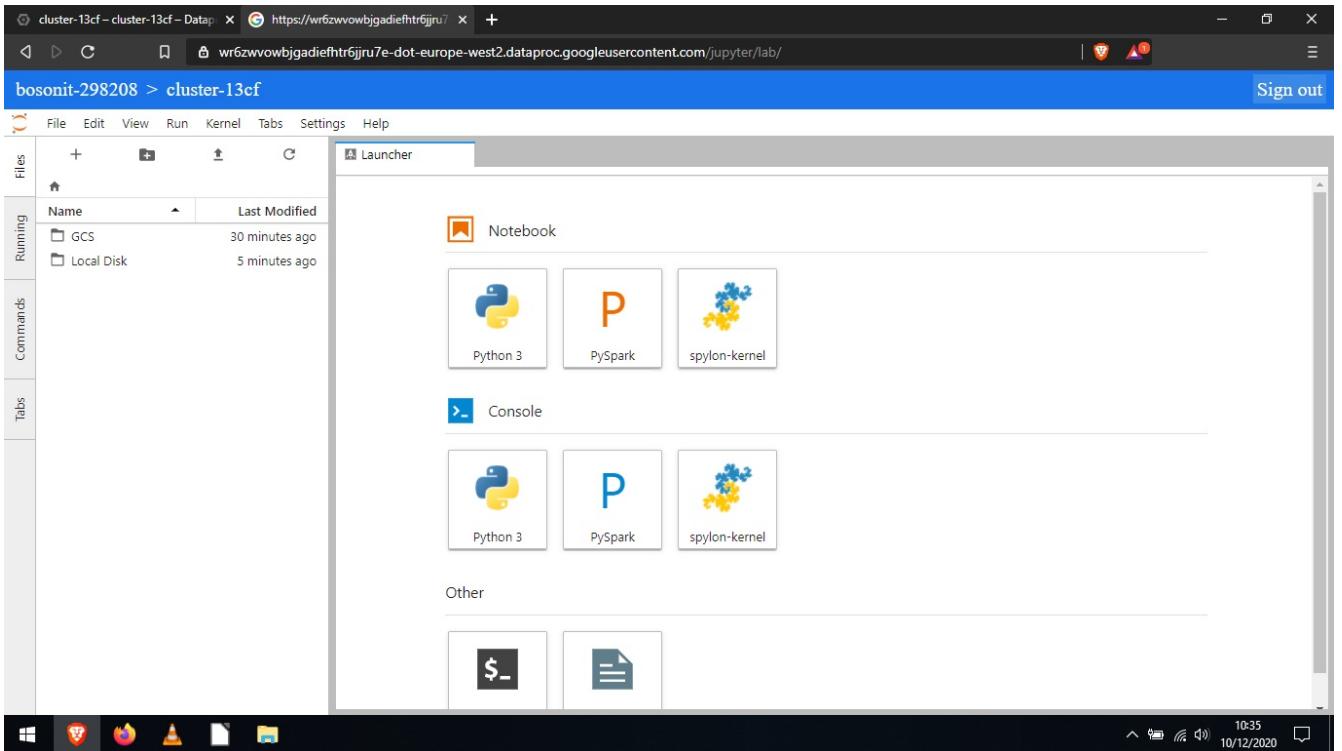
Si hacemos click en el nombre del Cluster nos llevara a la informacion general donde podremos ver la monitorizacion de los procesos y usos del cluster.

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 two charts: 'YARN memory' and 'YARN pending memory'. Both charts show memory usage from 9:45 to 10:30. The YARN memory chart shows values of 953.67 MB, 762.94 MB, 572.2 MB, 381.47 MB, and 190.73 MB. The YARN pending memory chart shows values of 953.67 MB, 762.94 MB, 572.2 MB, 381.47 MB, and 190.73 MB. A note at the top of the monitoring section 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.

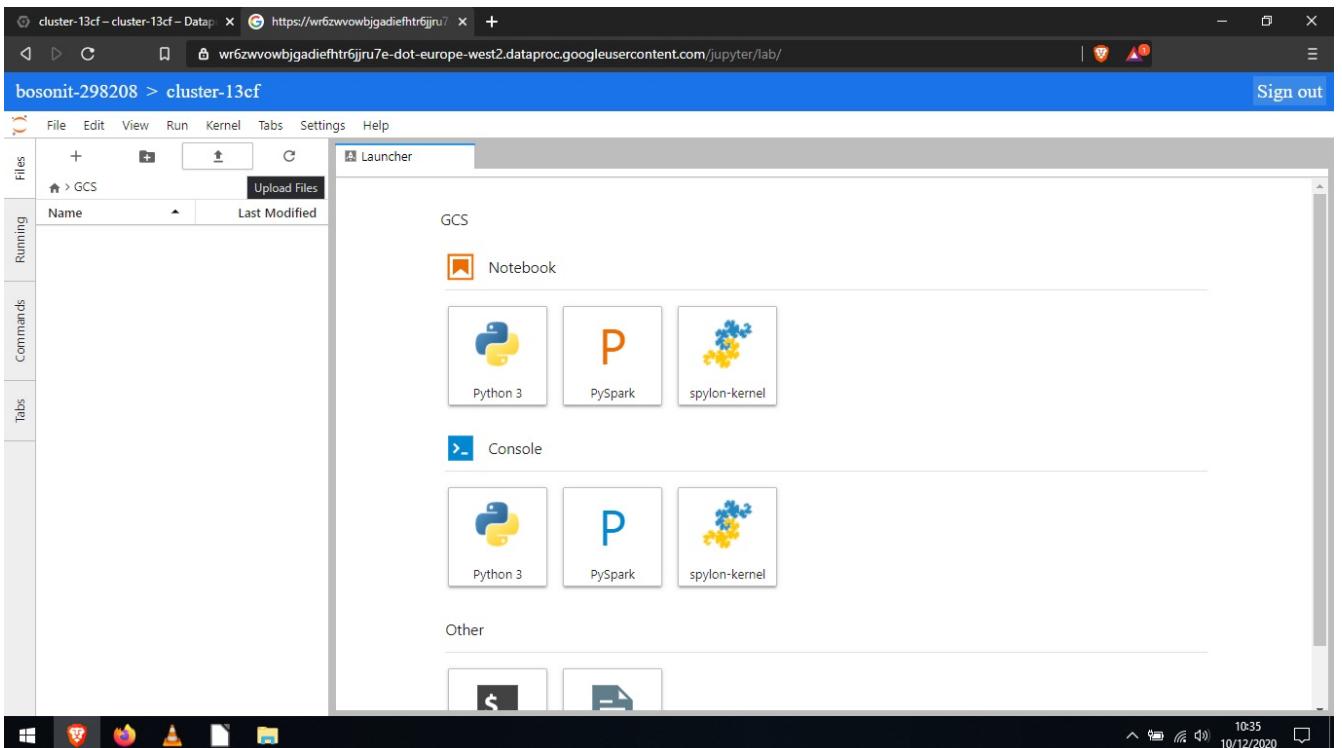
The screenshot shows the Google Cloud Platform DataProc cluster interfaces interface for 'cluster-13cf'. The left sidebar has 'Clusters' selected. The main area displays cluster details: Status (Running). Below this are tabs for MONITORING, JOBS, VM INSTANCES, CONFIGURATION, and WEB INTERFACES. The WEB INTERFACES tab is active, showing a list of 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 link. A note above the Component gateway section says: 'Create an SSH tunnel to connect to a web interface' and 'Provides access to the web interfaces of default and selected optional components on the cluster. [Learn more](#)'.

Hacemos click en 'JupyterLab' y veremos lo siguiente:

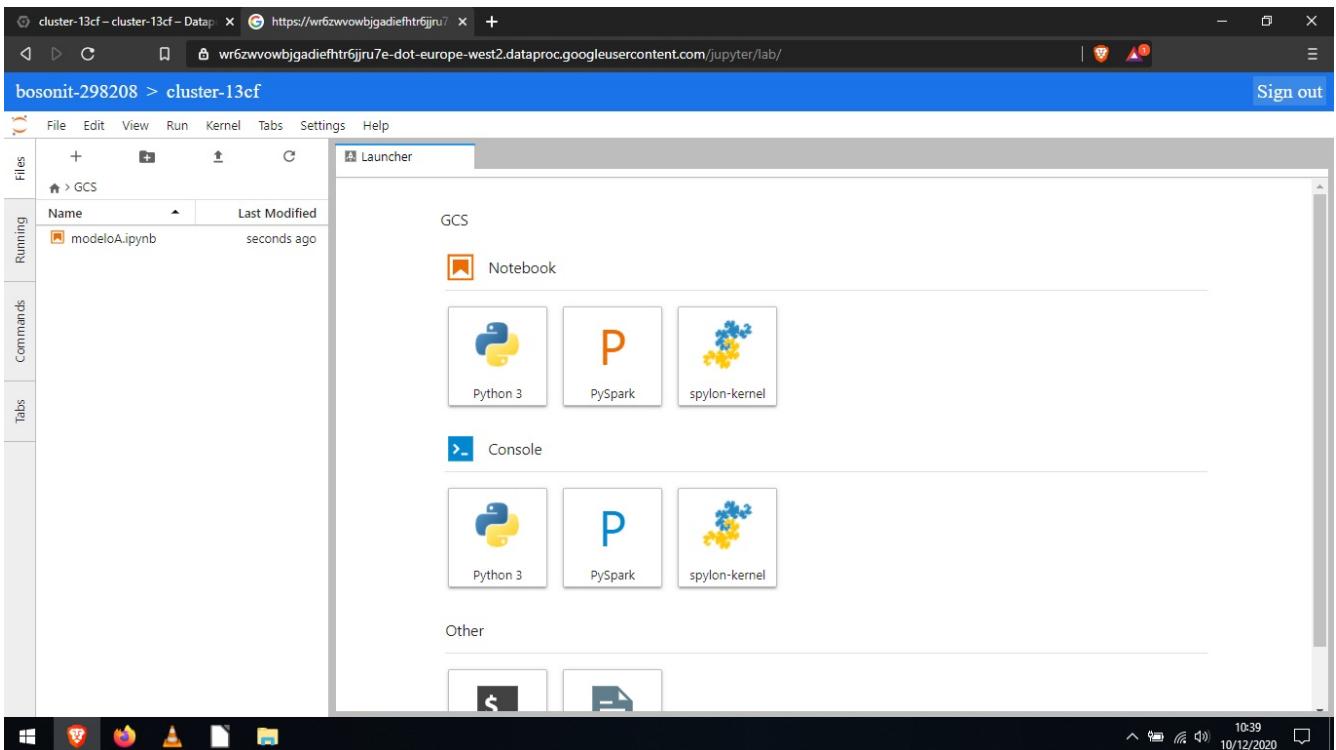


En la parte izquierda veremos dos carpetas, una GCS que es 'Google Cloud Storage' que se trata del almacenamiento permanente del que hablamos y Local que es el almacenamiento del propio cluster donde si entramos podremos ver los diferentes archivos del sistema operativo y demás del propio cluster. Esta 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:

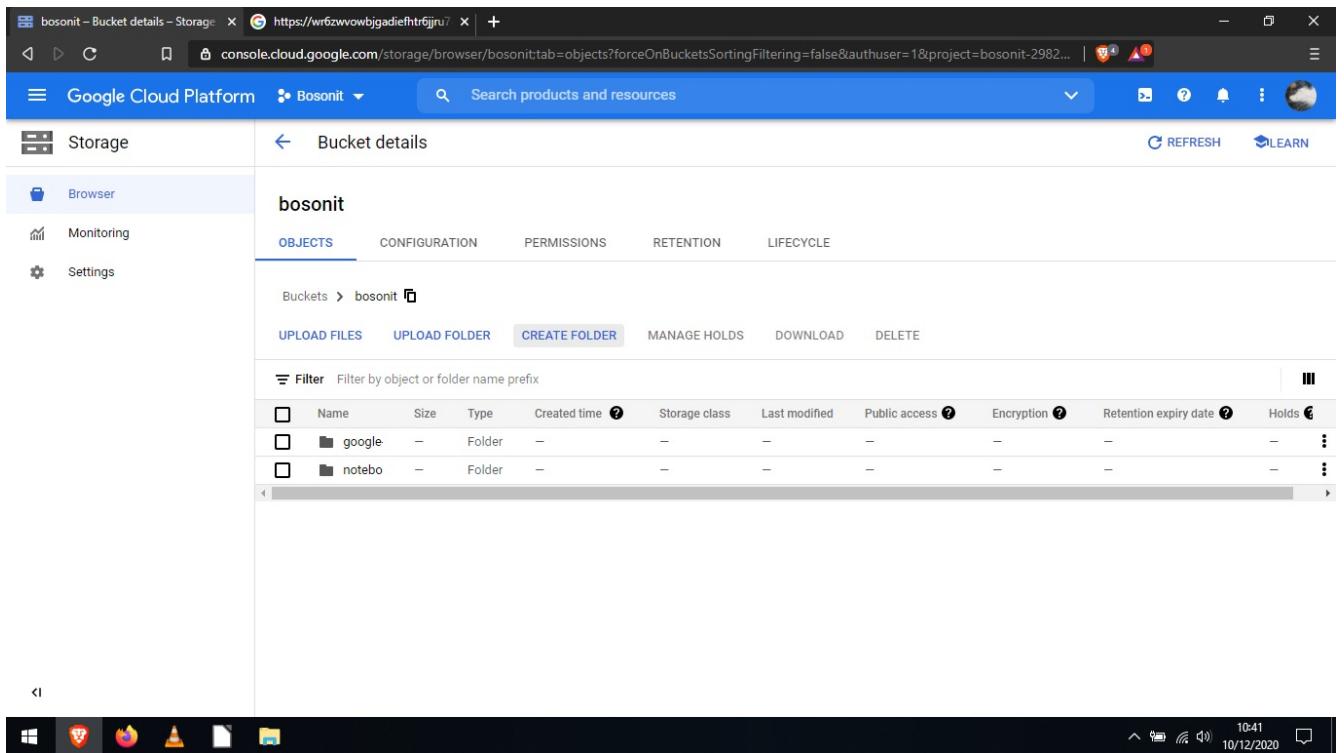


Este archivo quedara guardado de forma permanente en Storage hasta que lo eliminemos manualmente, independientemente de que el cluster este activo o no.

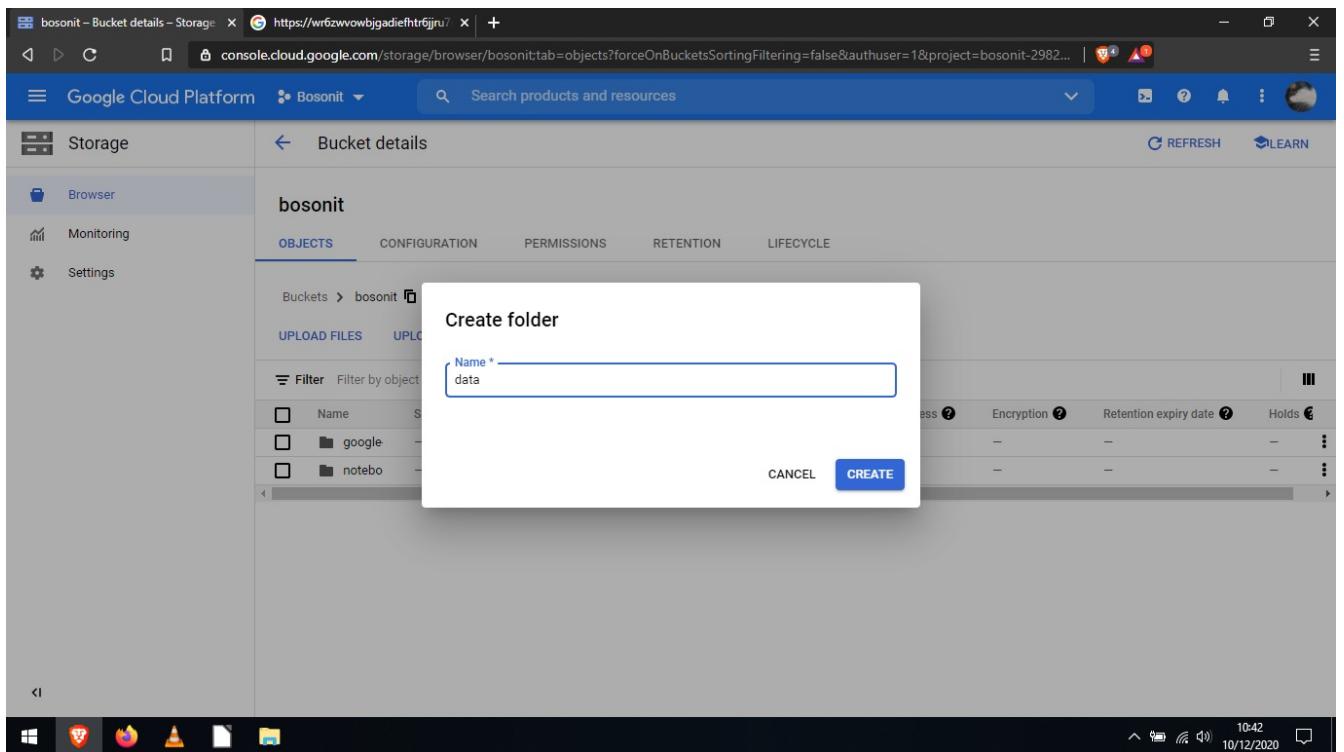
Ahora vamos a subir un Dataset para poder comprobar que todo funciona correctamente. Si intentamos subirlo desde aqui nos dira que solo podemos subir archivos con un tamaño maximo de 15Mb por lo tanto vamos a volver a la seccion 'Storage':

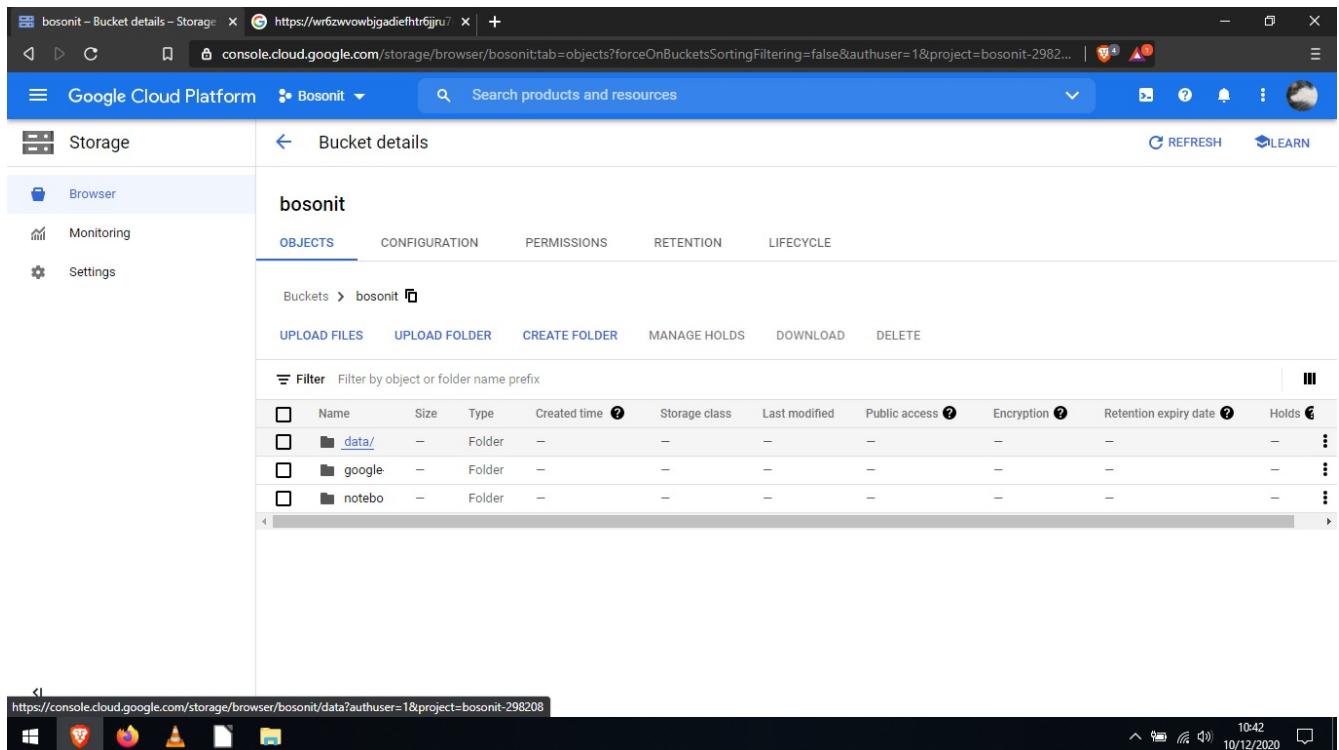
Name	Created	Location type	Location	Default storage class	Updated
bosonit	10 Dec 2020, 10:05:11	Region	europe-west1 ...	Standard	10 Dec 2 ...
dataproc-temp-europe-west2-789376720213-hg3o7q...	10 Dec 2020, 10:29:38	Region	europe-west2 ...	Standard	10 Dec 2 ...

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

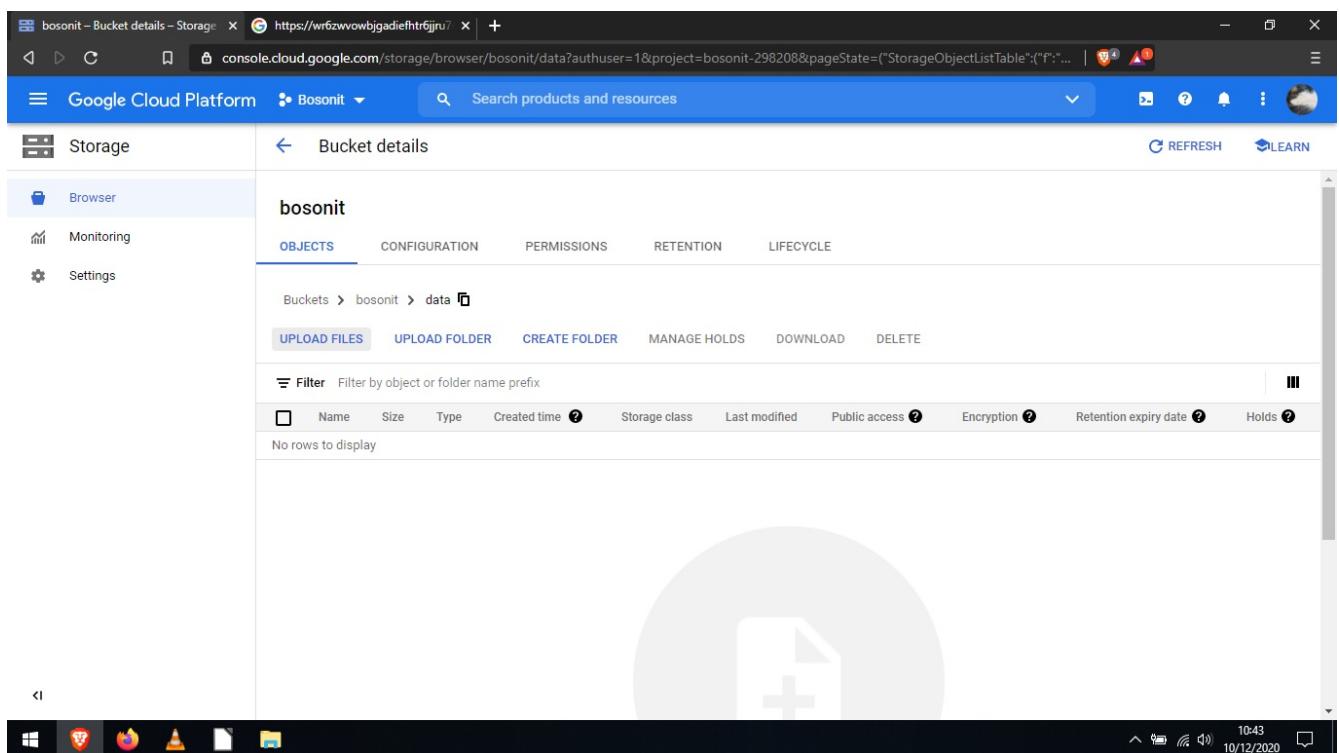


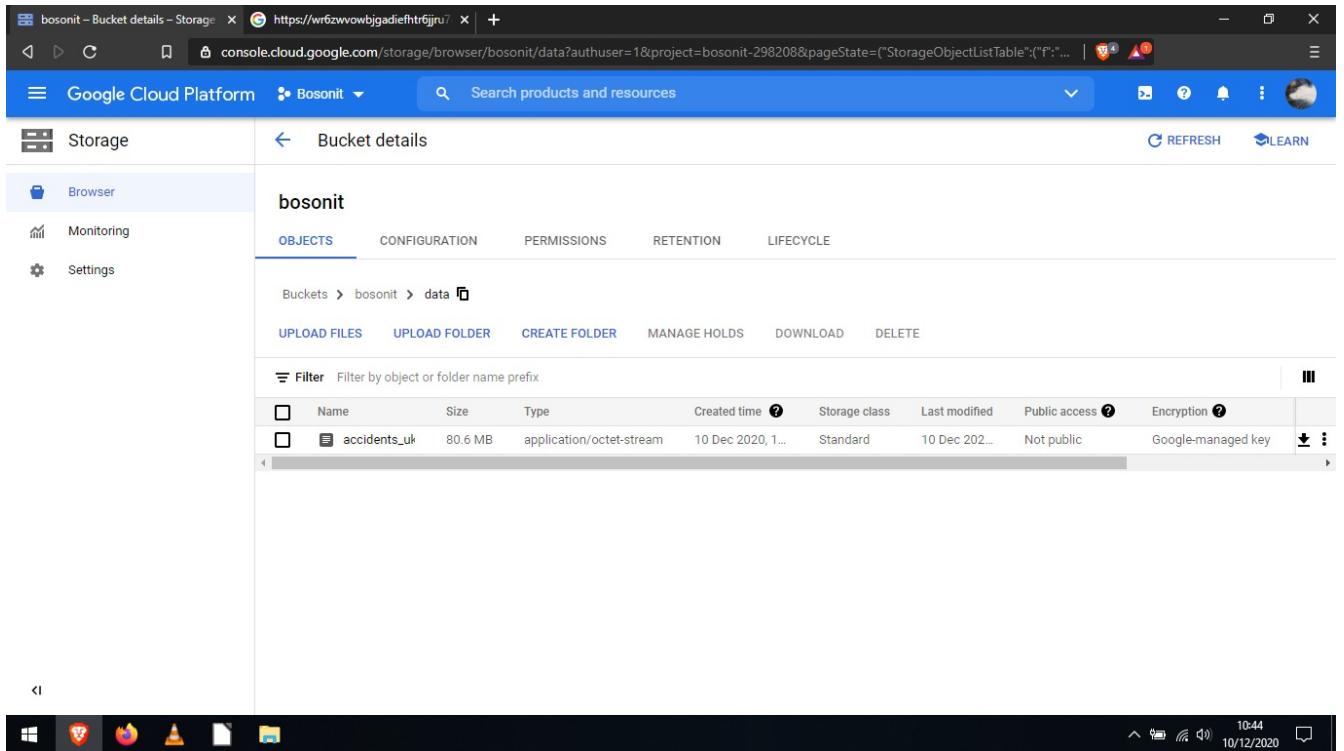
Crearemos una carpeta dandole a 'Create Folder' y le pondremos de nombre data:



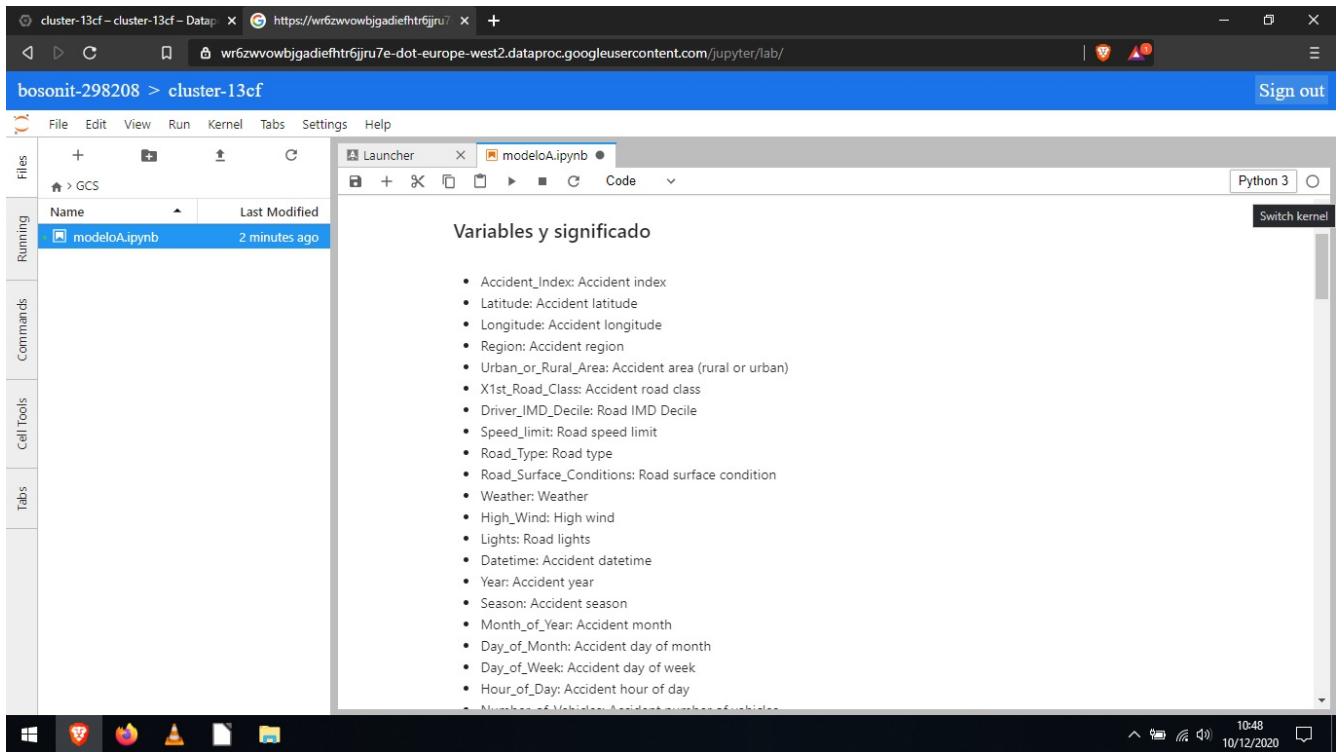


Entramos en ella y haciendo click en Upload subiremos nuestro dataset:

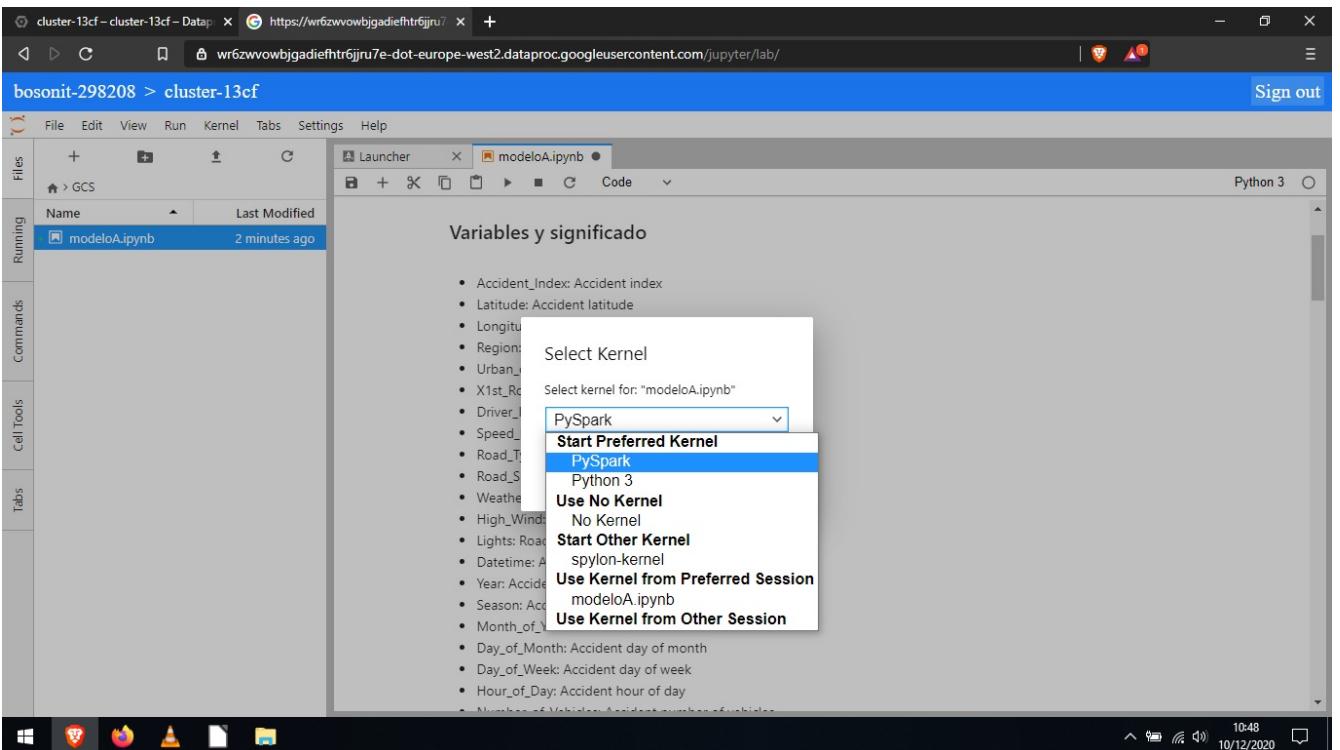




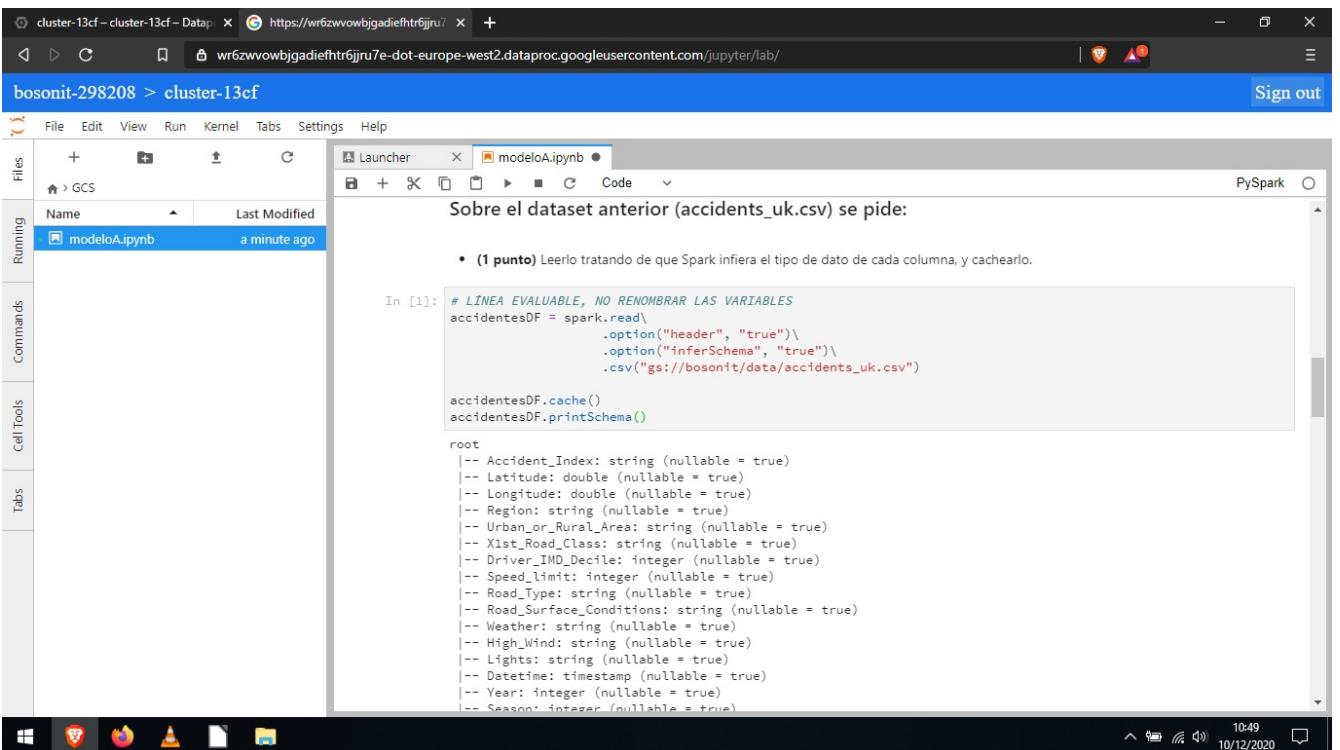
Ahora si volvemos a nuestro Notebook anterior y entramos en el:



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.