
Breve introducción a Google Colab

Noe Ferrero - 2021

1

I. Introducción

“Google Colab” es una herramienta para escribir y ejecutar código Python en la nube de Google. También es posible incluir texto enriquecido, “links” e imágenes. En caso de necesitar altas prestaciones de cómputo, el entorno permite configurar algunas propiedades del equipo sobre el que se ejecuta el código.

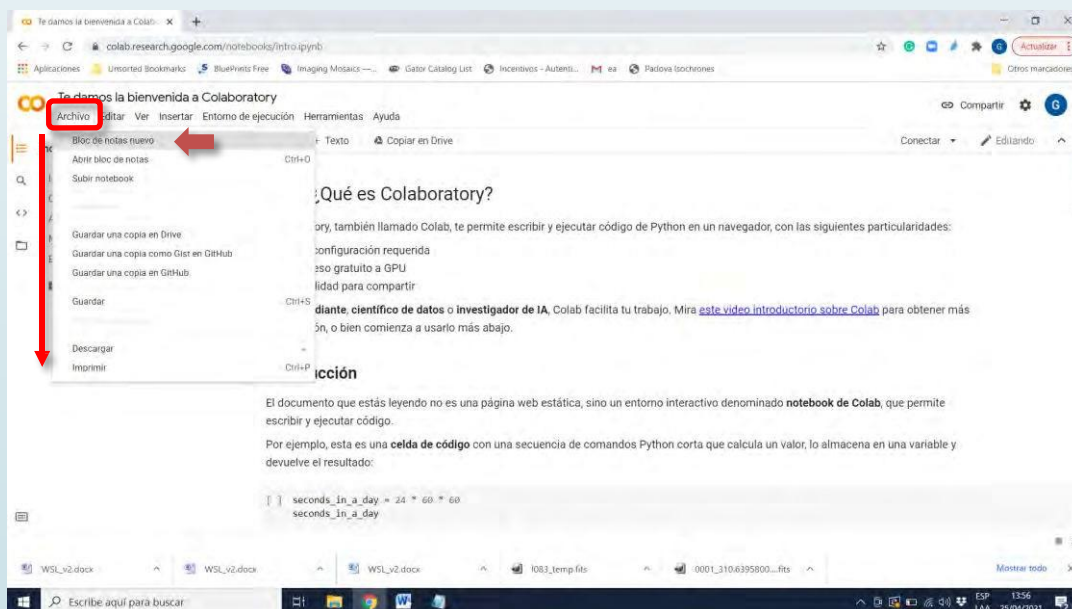
En definitiva, el uso de “Google Colab” permite disponer de un entorno para llevar a cabo tareas que serían difíciles de realizar en un equipo personal. Por otro lado, siguiendo la idea de “Drive”, “Google Colab” brinda la opción de compartir los códigos realizados lo que es ideal para trabajos en equipo.

II. Elementos Básicos

1. Acceso a “Google Colab”

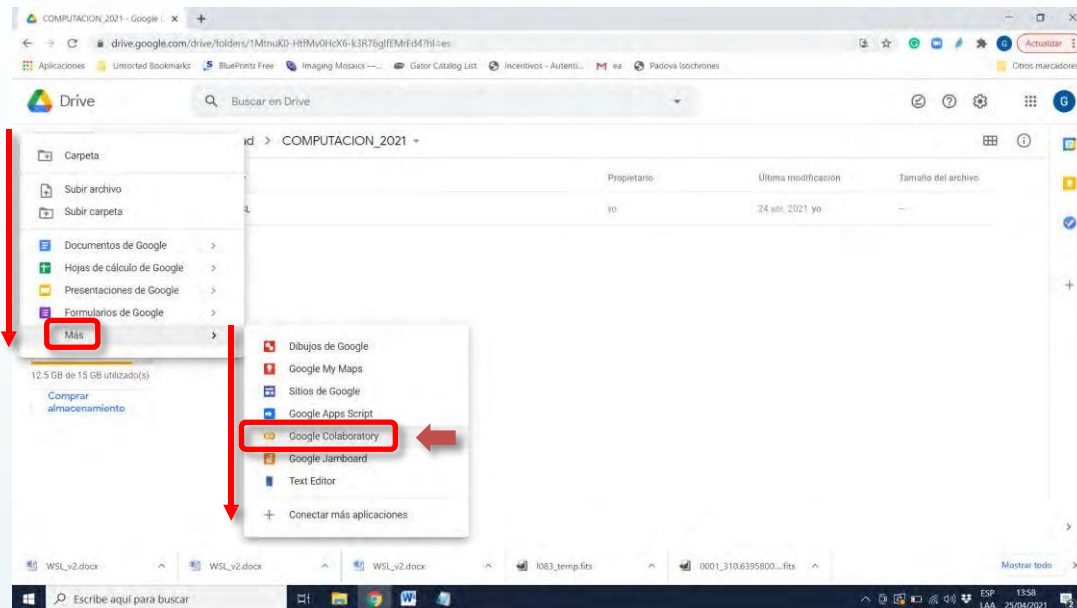
Existen algunas variantes para acceder a “Google Colab”, estas son:

Opción A: Ir al sitio de “Google Colab” (<https://colab.research.google.com/>), ir a “Archivo” (arriba a la izquierda) y elegir la opción “Block de notas nuevo”



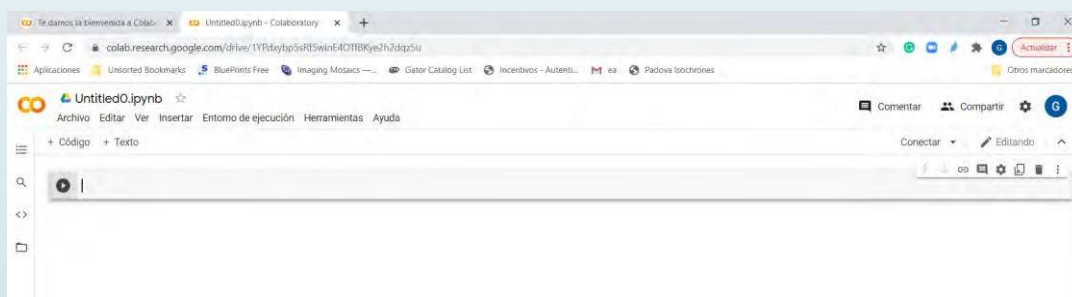
Opción B: Ir a “Google Drive” (<https://drive.google.com/drive/my-drive?hl=es>), hacer “click” en “Nuevo” (arriba a la izquierda), elegir la opción “Mas” (la última del menú) y seleccionar “Google Colaboratory”.

2



2. Notebook

Independiente de la opción adoptada para acceder a “Google Colab”, al final se abrirá una nueva pestaña en el navegador y en ella se encontrará el entorno de trabajo.

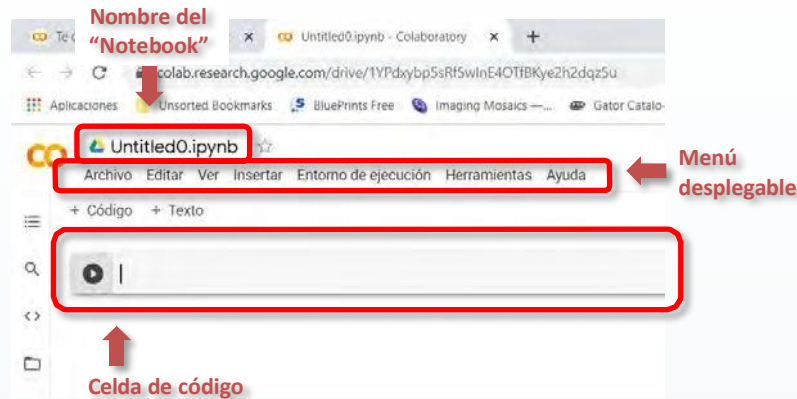


Este entorno se denomina “Notebook” y puede ser almacenado como un archivo tanto en “Drive” como en el equipo local.

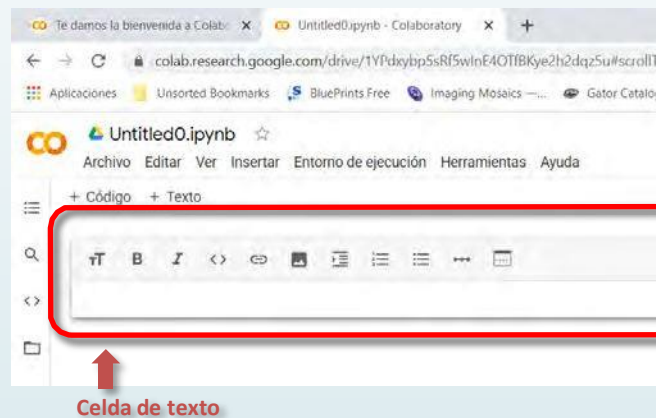
La estructura del “Notebook” es la siguiente:

- En la parte superior se indica el nombre del archivo correspondiente. La extensión del archivo será “.ipynb” (“Interactive Python NoteBook”)
- Debajo se encuentra un menú desplegable con varias opciones
- Luego se halla una “celda” que es el lugar donde se puede escribir el texto o el código deseado. Una “celda” es la unidad mínima de ejecución dentro del “Notebook”. Usualmente no se utiliza solo una “celda”, sino una sucesión de “celdas”.

3



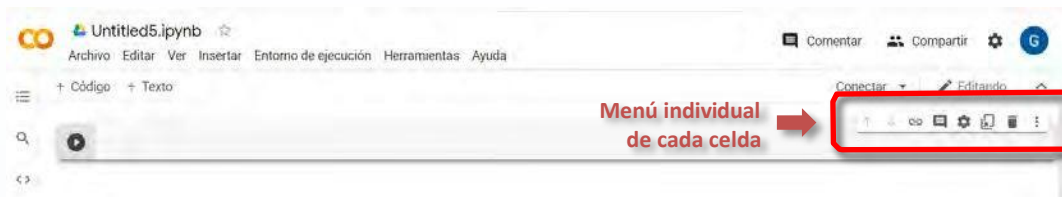
En una “celda” es posible escribir texto o código Python. No obstante, la “celda” presentada en el archivo creado es una “celda de código” y solo admite la escritura de código Python. Para poder escribir texto es necesario transformarla en una “celda de texto” localizando el cursor en ella y utilizando las teclas “**Ctrl+M M**” (con “**Ctrl+M Y**” es posible volver a una “celda de código”).



Se pueden indicar entonces las siguientes características de las “celdas”:

- Luego de escribir texto o código en la “celda” (según corresponda), ella se puede ejecutar con el ícono pulsando las teclas “**Shift+Enter**”.
- En el caso de tratarse de una “celda de texto”, se admiten comandos básicos y fórmulas de “Latex”. También se pueden insertar “links” e imágenes
- En el caso de una “celda de código”, al ejecutarla, aparecerá eventual resultado debajo de la “celda”.
- En ambos tipos de celdas, al utilizar las teclas “**Shift+Enter**”, se produce su ejecución y se crea a continuación una nueva “celda de código”.
- Al situarse el cursor en una celda (de código o de texto) aparece un menú en su ángulo superior derecho que brinda diversas opciones

4



Las “celdas de texto” intercaladas entre las “celdas de código” permiten ir documentando el código que se está desarrollando, permitiendo su mejor comprensión. A continuación se presenta un ejemplo simple de un “Notebook” previo a su ejecución y conteniendo ambos tipos de “celdas”:

Carga de paquetes

- Numpy
- Matplotlib

```
[ ]: import numpy as np
import matplotlib.pyplot as plt
```

Estadística básica

- Definición de un conjunto de números al azar con distribución uniforme
 x_1, x_2, \dots, x_N
- Valor medio o promedio

$$prom = \frac{\sum_{i=1}^N x_i}{N} = \frac{x_1 + x_2 + \dots + x_N}{N}$$
- Desviación estándar

$$stdev = \sqrt{\frac{\sum_{i=1}^N (x_i - prom)^2}{N - 1}}$$

```
[ ]: # -----
# Estadística básica con Numpy
# -----

# Conjunto de N valores con distribución uniforme entre 0 y A
N = 1000
A = 50
x = A * np.random.rand(N)

# Cálculos estadísticos
prom = np.mean(x)
stdev = np.std(x)
x_min = np.min(x)
x_max = np.max(x)

[ ]: # -----
# Salida de resultados
# -----

print('-----')
print('N =', N)
print('promedio =', prom)
print('desviación estándar =', stdev)
print('valor mínimo =', x_min)
print('valor máximo =', x_max)
print('-----')
```

Gráfico con Matplotlib

Histograma de los datos junto con su valor medio y desviación estándar

5

```
[ ] # -----
# Histograma del conjunto de datos
#
# Creación de una figura (fig)
fig = plt.figure(figsize=(8,5))

# Creación de un gráfico (ax)
ax = fig.add_subplot(111)

# Identificación de los ejes del gráfico
ax.set_xlabel('x')
ax.set_ylabel('N')

# Histograma y cálculo de su valor máximo
histogram = ax.hist(x)
histo_max = np.max(histogram[0])

# Líneas del valor promedio y de la desviación estándar
ax.plot([prom,prom],[0,histo_max], lw=3, color='yellow', linestyle='--')
ax.plot([prom-stdev,prom-stdev],[0,histo_max], lw=3, color='yellow', linestyle='--')
ax.plot([prom+stdev,prom+stdev],[0,histo_max], lw=3, color='yellow', linestyle='--')

# Mostrar la figura
plt.show()
```

Se nota que las “celdas de texto” indican en su parte derecha el futuro resultado que ellas producirán al ser ejecutadas

Finalmente, una vez ejecutadas todas las celdas, el resultado obtenido es:

Carga de paquetes

- Numpy
- Matplotlib

```
[1] import numpy as np
import matplotlib.pyplot as plt
```

Estadística básica

- Definición de un conjunto de números al azar con distribución uniforme
 x_1, x_2, \dots, x_N
- Valor medio o promedio

$$prom = \frac{\sum_{i=1}^N x_i}{N} = \frac{x_1 + x_2 + \dots + x_N}{N}$$
- Desviación estándar

$$stdev = \sqrt{\frac{\sum_{i=1}^N (x_i - prom)^2}{N - 1}}$$

[2] Estadística básica con Numpy

```
# Conjunto de N valores con distribución uniforme entre 0 y A
N = 1000
A = 50
x = A * np.random.rand(N)

# Cálculos estadísticos
prom = np.mean(x)
stdev = np.std(x)
x_min = np.min(x)
x_max = np.max(x)
```

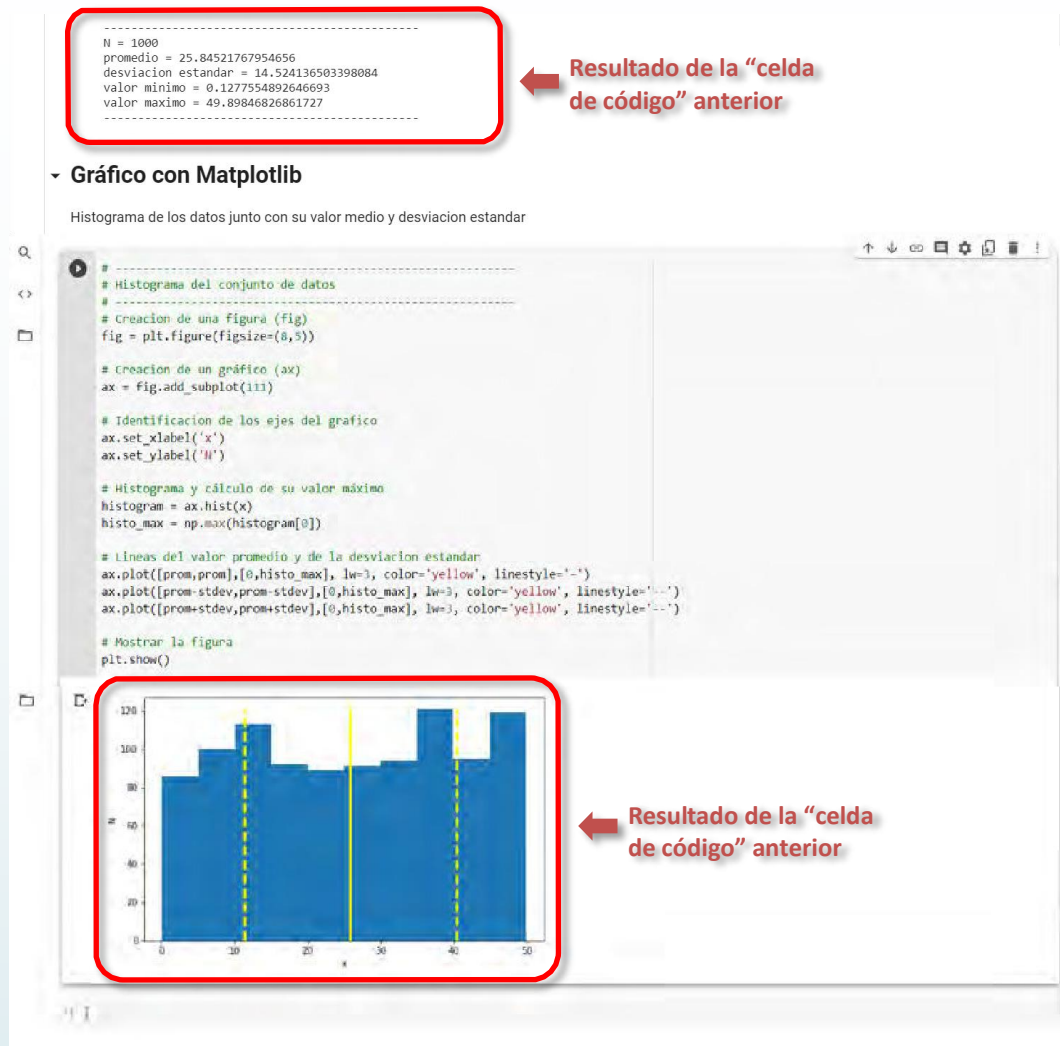
[3] Salida de resultados

```
print('-----')
print('N =', N)
print('promedio =', prom)
print('desviación estándar =', stdev)
print('valor mínimo =', x_min)
print('valor máximo =', x_max)
print('-----')
```

Resultado de la “celda de texto”

En este caso la “celda de código” anterior solo realiza cálculos y no produce ningún resultado

6



O sea que en el lugar de las “celdas de texto” solo se presenta el texto enriquecido y debajo de las “celdas de código” aparecen los respectivos resultados, si es que ellas producen alguno

III. Entorno de ejecución

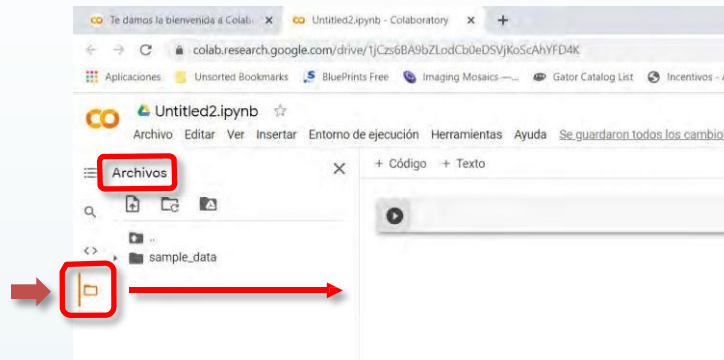
“Google Colab” permite ejecutar código Python (versión 3) sobre diferentes configuraciones del equipo, o sea que éste sólo utilice CPU que posea además algún acelerador (utilizando GPU o TPU).

Además, cabe notar que existe un límite de tiempo de uso del servicio de “Google Colab” (unas 12 horas corridas) y de almacenamiento en RAM.

Sin embargo, tango el uso de aceleradores como los límites indicados solo se tornan relevantes para el caso de códigos muy complejos. No es necesario preocuparse por ellos para el caso de desarrollar códigos simples.

IV. Acceso a los archivos de datos

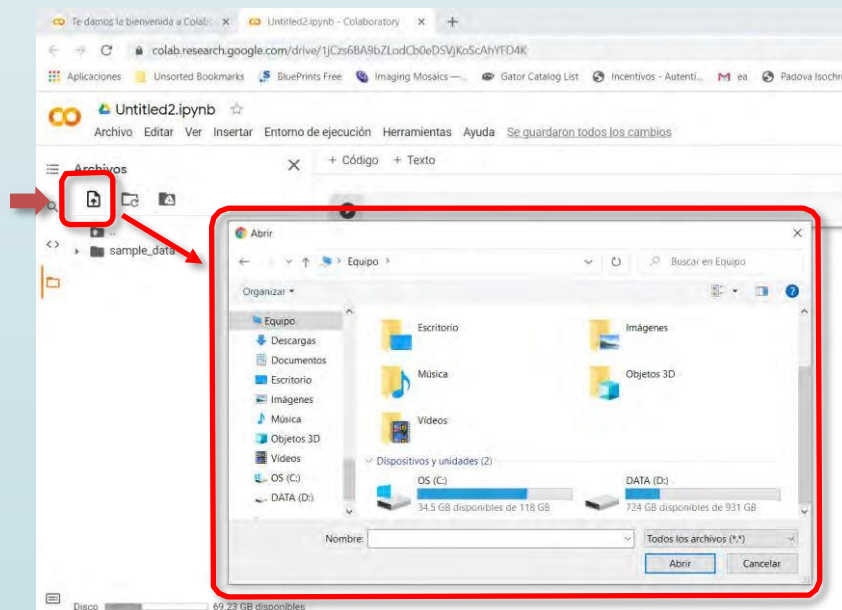
- 7 Los archivos de datos que a los que el código puede acceder se pueden visualizar haciendo “click” en el ícono de una carpeta que se encuentra a la izquierda de la pantalla, entonces se abrirá hacia la izquierda una pestaña denominada “Archivos”.



Existen diferentes formas para que los archivos deseados aparezcan en este listado. Ellas se indican a continuación:

Caso 1: Archivos de datos en el equipo local

En este caso es necesario subir los datos localizados en el equipo local a la nube de Google para que sean accesibles por el “Notebook”. Esto se puede llevar a cabo haciendo “click” en el primer ícono del margen superior izquierdo (flecha hacia arriba). Se abrirá entonces una ventana que permite navegar en el equipo local y seleccionar la carpeta o archivo deseado. Cabe notar que en este caso los datos subidos serán eliminados de la nube una vez finalizado el uso del “Notebook”,



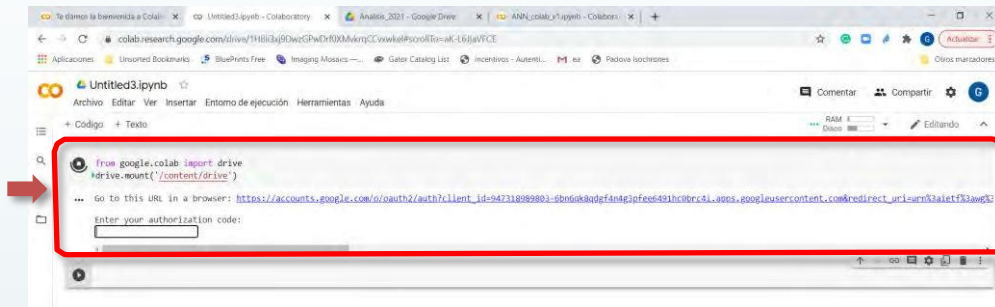
Caso 2: Archivos de datos en “Drive”

8

En este caso es necesario montar “Google Drive”. Esto se puede realizar de dos formas:

a) Indicando los siguientes comandos, usualmente, en la primer “celda de código” del “Notebook”.

```
from google.colab import drive
drive.mount('/content/drive')
```



Los pasos a seguir entonces son:

- Ejecutar la “celda” anterior, con lo que aparecerá un enlace
- Hacer “click” en dicho enlace, entonces se abrirá otra pestaña solicitando la validación de la cuenta de usuario de Google
- Validar la cuenta de Google y aparecerá un código de acceso
- Copiar (“Ctrl+C”) el código que aparece y pegarlo (“Ctrl+V”) en el recuadro que aparece debajo de la “celda” ejecutada en el “Notebook”
- Presionar “Enter” y entonces aparece el mensaje

Mounted at /content/drive

Indicando que el “Drive” del usuario se ha validado y se ha montado en el “Notebook”.

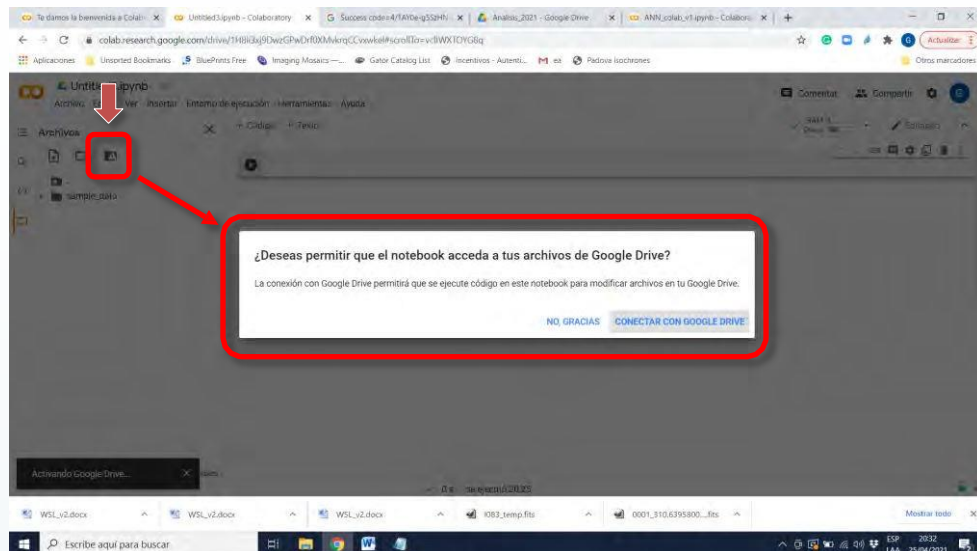
Para verificar que ha funcionado todo bien, es posible visualizar el contenido de “Drive” ejecutando en una “celda” sucesiva el código:

```
!ls /content/drive/MyDrive/
```

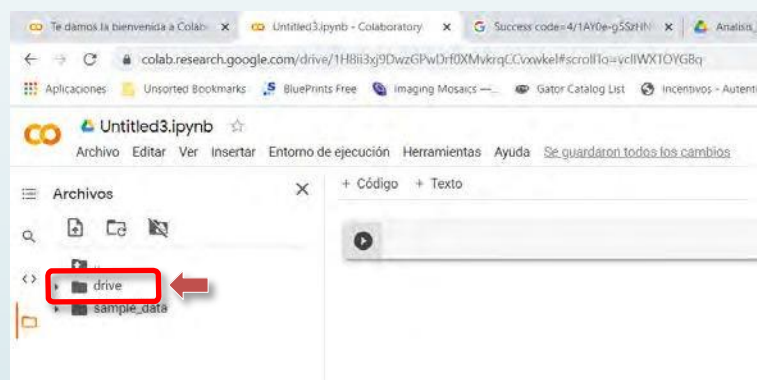
O sea, que se trata simplemente del comando “ls” de Linux luego del símbolo “!”. Otros comandos de Linux también se pueden utilizar de esta forma.

b) La otra forma de efectuar el mismo proceso es utilizar el ícono del margen superior izquierdo de la pestaña “Archivos”

9



A continuación se solicitará un permiso para acceder a los archivos de Drive. Luego de aceptar, el sistema demorará unos segundos en hacer la conexión y se aparecerá la carpeta “drive” en la pestaña “Archivos”



O sea, el caso b) es más simple, pero el caso a) queda automatizado en el código a desarrollar y solo requiere los pasos mencionados la primera vez que el mismo se ejecuta.

Caso 3: Archivos de datos en algún repositorio

En este caso es necesario traer los datos desde el repositorio y descompactarlos si es necesario. Este proceso se lleva a cabo con los comandos “**!wget**” y “**!unzip**” en la primer “celda”.

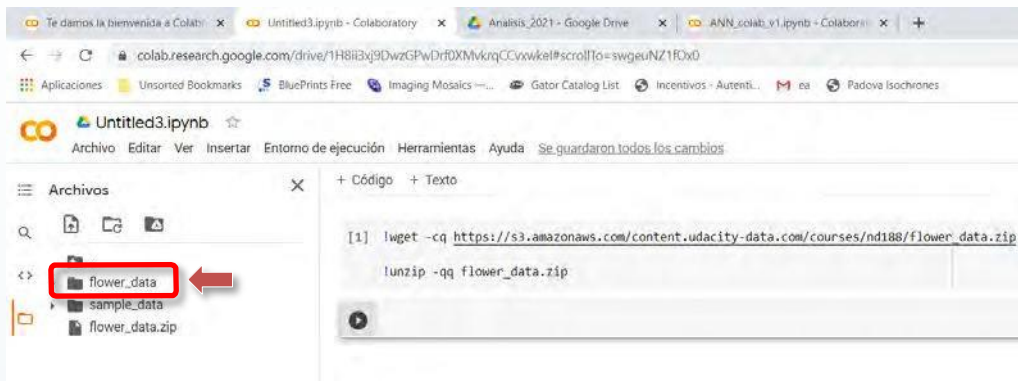
Por ejemplo:

```
!wget -cq https://s3.amazonaws.com/content.udacity-data.com/courses/nd188/flower_data.zip
```

```
!unzip -qq flower_data.zip
```

Al ejecutar esta “celda” se tendrán disponible los datos “flower_data” de Udacity.

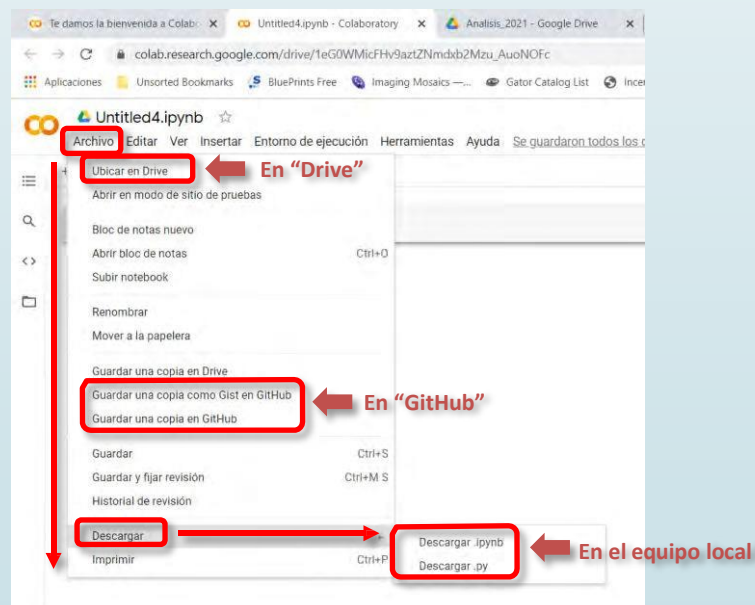
10



V. Guardando la “Notebook”

Una vez abierta la “Notebook” se crea un archivo con el nombre indicado en la parte superior. Ese nombre puede cambiarse simplemente haciendo “click” en ese lugar y escribiendo el nombre deseado.

Si bien la “Notebook” es guardada regularmente en forma automática por “Google Drive”, también es posible hacerlo manualmente utilizando con la opción “Archivo” el menú superior. Se desplegarán entonces varias opciones entre las que se encuentran la posibilidad de guardar el “Notebook” en “Drive”, en “GitHub” o descargarla en el equipo local. En este último caso puede ser en formato “.ipynb” o “.py”.



VI. Comentarios Finales

- 11 Como indica su título, el presente instructivo presenta solo los aspectos más generales para comenzar a trabajar con “Google Colab”. Por supuesto existen muchas posibilidades más y ellas pueden ser exploradas recorriendo los diferentes menús que brinda un “Notebook” o accediendo a distintas ayudas disponibles en Internet.

Referencias e información adicional

- Primeros pasos con Jupyter Notebook
<https://www.adictosaltrabajo.com/2018/01/18/primeros-pasos-con-jupyter-notebook/>
- Google Colab: Python y Machine Learning en la nube
<https://www.adictosaltrabajo.com/2019/06/04/google-colab-python-y-machine-learning-en-la-nube/>
- Getting Started With Google Colab
<https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c>
- Google Colab - Guía rápida
<https://www.hebergementwebs.com/tutorial-de-google-colab/google-colab-guia-rapida>

by  Noelia Ferrero