

# Google Colab

Versión 2.0.0

10/2024

Guía de cátedra

Pensamiento computacional (90)  
Cátedra: Balbiano

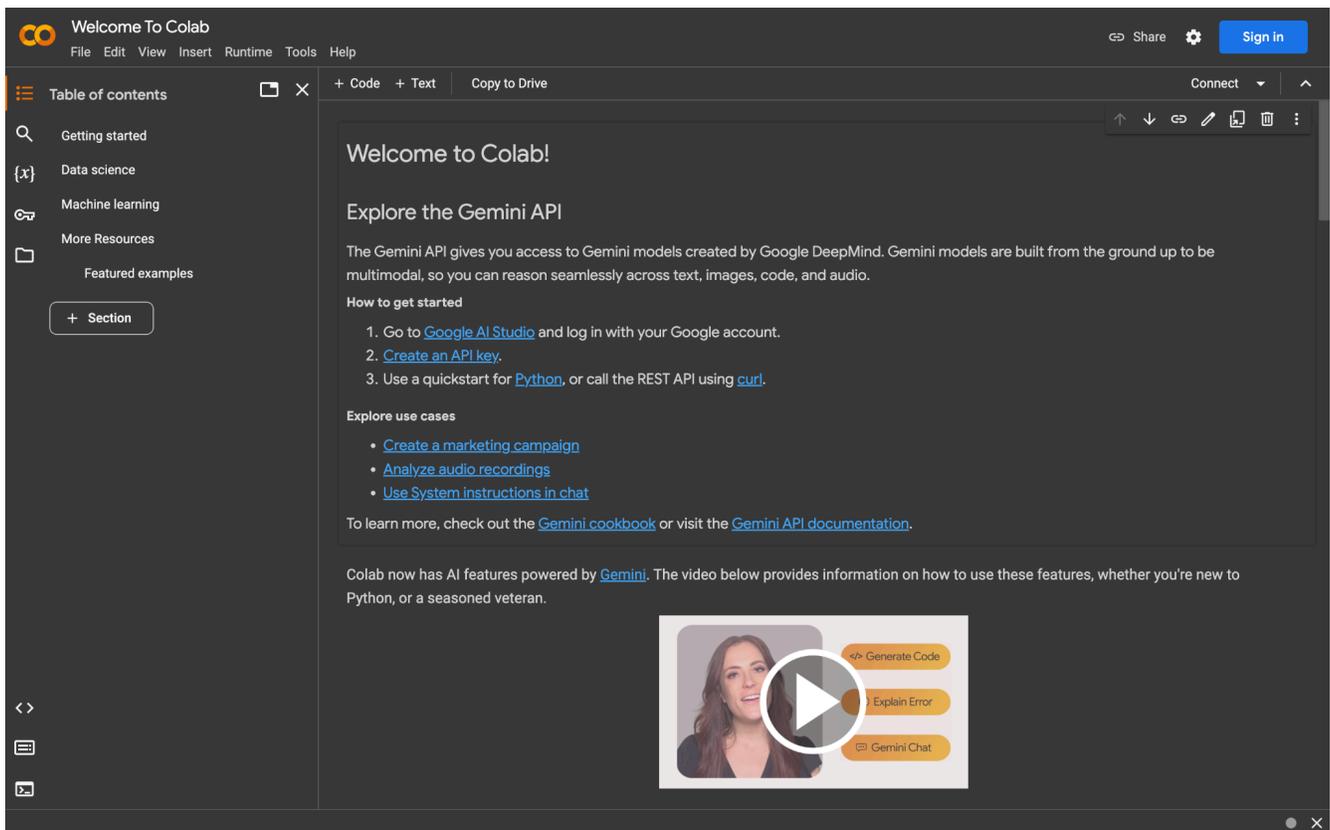
**.UBA XXI**

<b>Google Colab.....</b>	<b>3</b>
Crear un notebook.....	3
Cómo ejecutar el código que escribimos?.....	4
¿Cómo agrego más celdas al notebook?.....	5
¿Cómo ejecutar varias celdas de un mismo notebook?.....	5
Ubicando el notebook en mi drive.....	6
Archivos en Colab.....	7
Conectar nuestro google drive a colab.....	7
Crear archivos en colab.....	8
Leer un archivo desde colab.....	11
Conclusiones.....	13

# Google Colab

Google Colab es una plataforma gratuita que permite escribir y ejecutar código en Python directamente en el navegador.

Cuando accedemos a la página de [Google Colab](#) se nos presenta una interfaz de bienvenida con algunas noticias y tutoriales para comenzar con el uso de la herramienta.

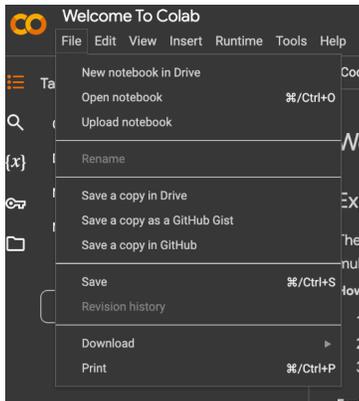


## Crear un notebook

Lo primero que haremos es crear un nuevo notebook. Un notebook es un entorno virtual interactivo donde podemos escribir y ejecutar código Python, así como agregar texto explicativo, imágenes, y otros elementos multimedia.

Para ello haremos click en **File > New notebook in Drive** ó **Archivo > Nuevo notebook en Drive**. Los notebooks se guardan en su google drive.

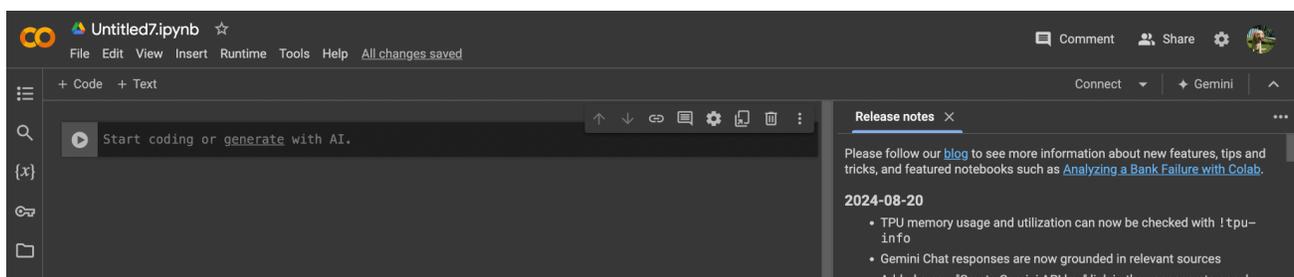
**Nota:** Si no tenemos una sesión de google iniciada nos pedirá que agreguemos una.



Una vez creado el notebook veremos la interfaz de desarrollo donde ya podremos empezar a trabajar con código sin ninguna instalación previa. En la esquina superior izquierda veremos el nombre default que google le da a los archivos( en general, Untitled.ipynb), podemos cambiarlo por el nombre que deseen. Recomendamos tener un notebook por sesión.

A la derecha veremos una ventana con los detalles de las características de la última versión de la plataforma, podemos cerrarla sin problemas.

En el recuadro que dice **Start coding or generate with AI.** es donde vamos a escribir nuestro código.



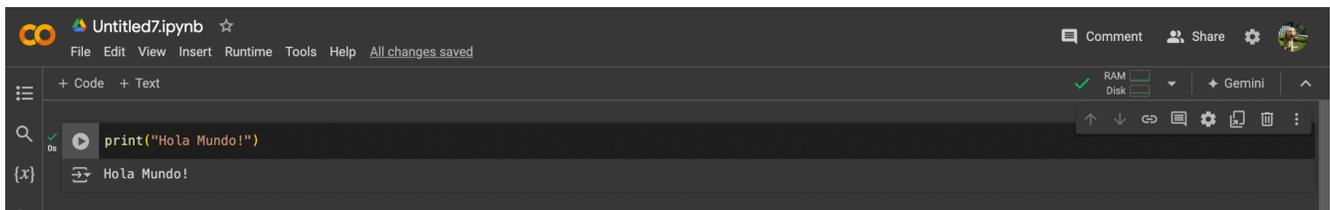
## Cómo ejecutar el código que escribimos?

Dentro de cada celda de código, como se denominan a las “cajitas” donde se puede escribir código python y ejecutarlo. Tenemos al costado un botón igual al de play

(). Es con ese botón, con el que podrán decirle a colab que ejecute el código que escribieron en esa celda y el resultado de la ejecución se muestra debajo de la misma.

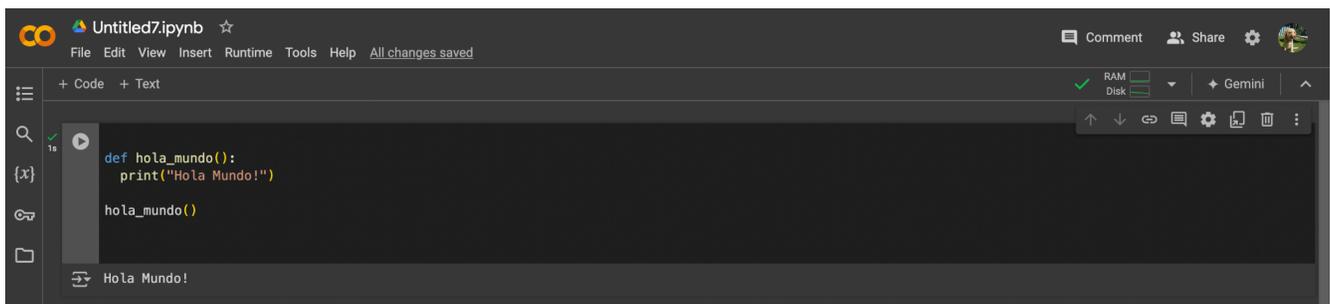
Ejemplo:

escribiremos el código que imprima el texto “Hola Mundo!”. Luego de hacer click en play ó “run cell” vemos el resultado:



The screenshot shows the Google Colab interface with a single code cell. The code is `print("Hola Mundo!")`. Below the code, the output is displayed as `Ho La Mundo!`. The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a toolbar with icons for code and text, and a status bar showing RAM and Disk usage.

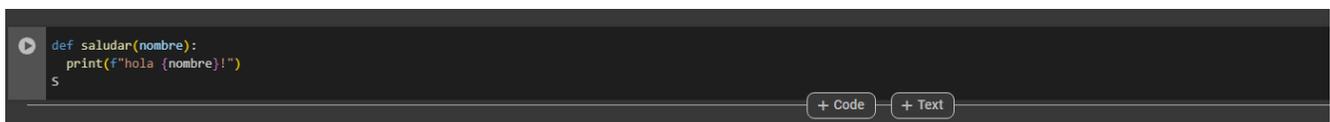
También podemos definir y llamar funciones para que se ejecuten.



The screenshot shows the Google Colab interface with a code cell containing a function definition and a call to the function. The code is `def hola_mundo(): print("Hola Mundo!")` followed by `hola_mundo()`. The output is `Ho La Mundo!`. The interface includes a menu bar, a toolbar, and a status bar.

## ¿Cómo agrego más celdas al notebook?

Posicionando el mouse sobre el borde inferior de la celda de código la plataforma nos da la opción de agregar más celdas, en caso que necesitemos. Por ejemplo, para agregar otro ejercicio de la sesión.



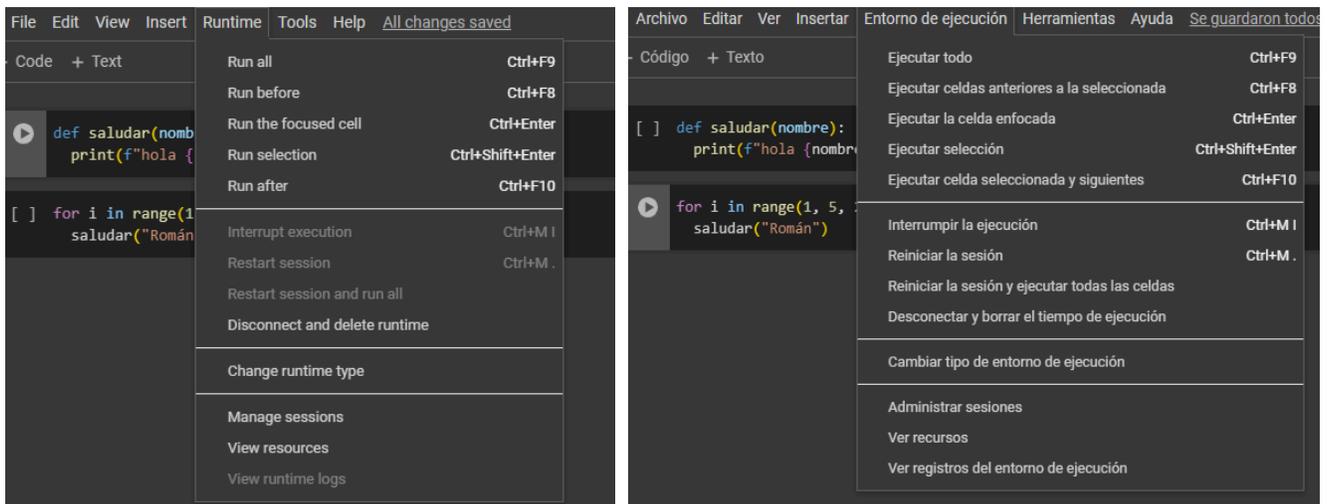
The screenshot shows the Google Colab interface with a code cell containing a function definition and a call to the function. The code is `def saludar(nombre): print(f'hola {nombre}!')` followed by `s`. Below the code, there are two buttons: `+ Code` and `+ Text`.

Entonces hacemos click en `+ Code`, y ya habrán creado una nueva celda de código.

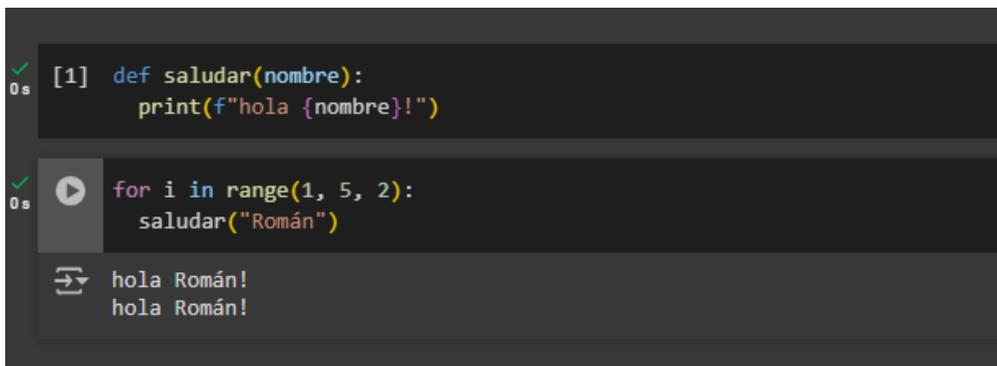
## ¿Cómo ejecutar varias celdas de un mismo notebook?

Cuando empezamos a tener distintas celdas de código, nos es un poco tedioso, tener que ejecutar las celdas una por una.

Es por ello que la herramienta nos da una opción para ejecutar todas las celdas de un notebook. Ello lo logramos yendo a **runtime > run all** ó **Entorno de ejecución > Ejecutar todo** .



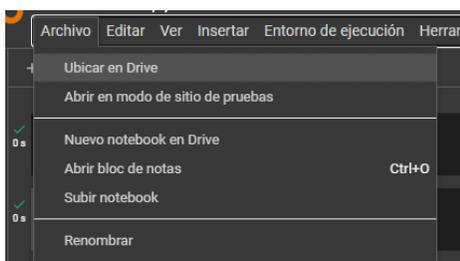
Luego de ejecutar todas las celdas veremos algo como:



## Ubicando el notebook en mi drive

Una vez que dejemos de usar el notebook, lo que se puede hacer es ubicar el notebook creado y guardarlo en algún lugar de drive que deseemos para que nos sea más fácil ubicarlo la próxima vez que lo necesitemos.

Para ello, vamos a **File > Locate in Drive** ó bien a **Archivo > Ubicar en Drive**.



Abrirá una nueva pestaña del navegador y estará pintada de celeste el archivo en cuestión.



Entonces a partir de allí, podrán crear una carpeta y mover el archivo a donde crean conveniente, etc.

## Archivos en Colab

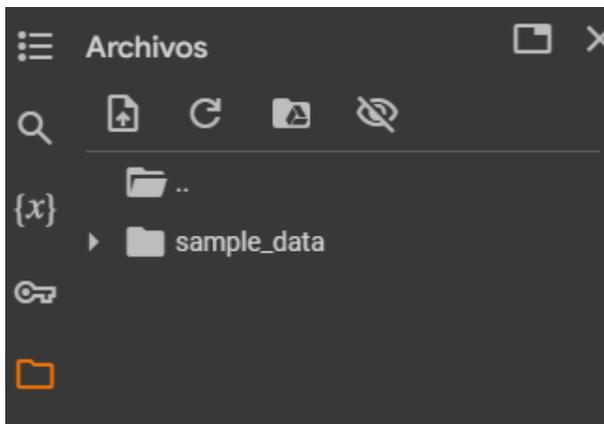
Buenas! En esta sección veremos cómo conectar nuestro google drive con Colab. Crearemos y leeremos archivos que se generarán desde Colab para poder trabajar con archivos con esta herramienta.

### Conectar nuestro google drive a colab

En primera instancia, lo que haremos es conectar nuestro archivo de colab con nuestro Google Drive para que sea posible la persistencia (guardado permanente) de nuestros archivos generados y que la comunicación entre nuestros archivos generados y colab sea más fácil.

Para ello, dentro de nuestro archivo colab donde queramos trabajar, le damos click en : 

Esto nos abrirá una pestaña como la siguiente:



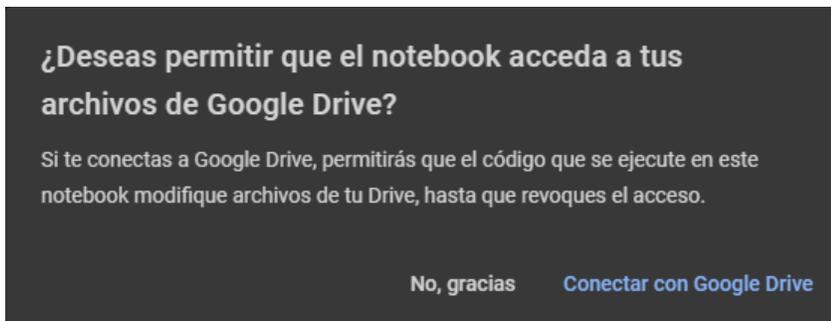
Tenemos distintas opciones, como por ejemplo, subir un archivo, refrescar, etc.

Nosotros nos centraremos en permitirle a colab, trabajar con nuestros archivos de drive.

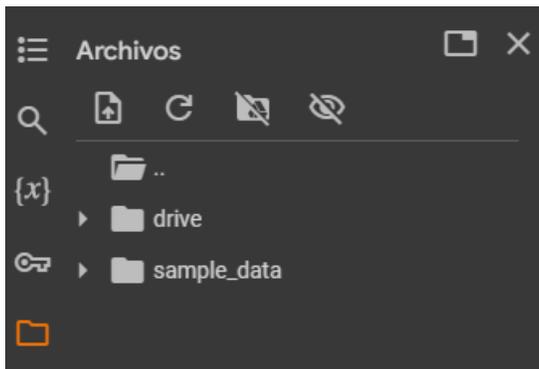
Para ello clickeamos en :



Nos aparecerá un mensaje como el siguiente, y clickean en “Conectar con Google Drive”:



Aparecerá ahora en nuestro listado de archivos y carpetas, una carpeta llamada **drive**:



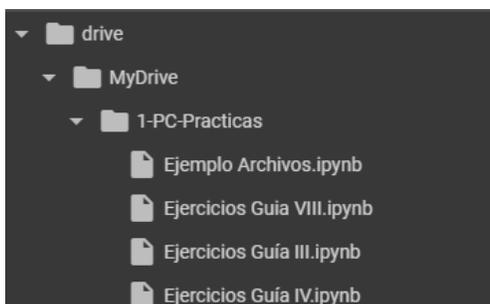
Y listo! ¡Ya tenemos nuestra conexión con drive para ponernos manos a la obra!

**Nota:** Una vez conectado con drive, no será necesario realizar estos pasos nuevamente. Es decir, cada vez que ingreses al mismo colab, tendrás permisos activados para trabajar con tus archivos de drive.

## Crear archivos en colab

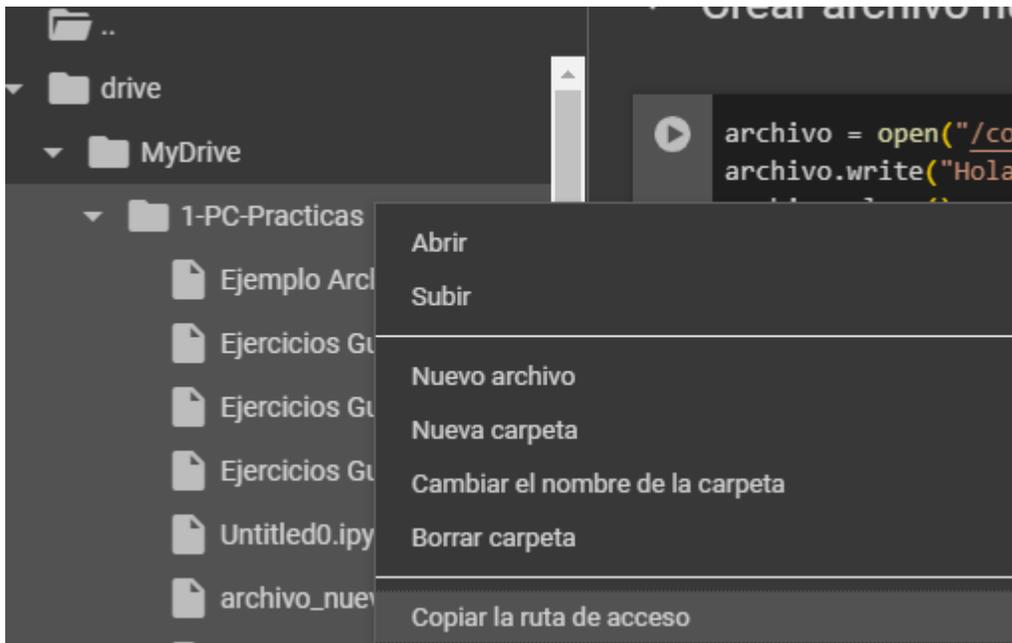
Para comenzar, crearemos nuestro primer archivo con Colab y lo almacenaremos en nuestro Drive.

En este ejemplo, tenemos una carpeta creada en drive, llamada “1-PRACTICAS-PC” donde se guardan ejercicios hechos en Colab en esa carpeta.



Para crear el archivo, la clave está en saber en qué ubicación se encuentra nuestro colab, para que al momento de generar el archivo, se genere en un lugar conocido, en este caso, en la misma carpeta donde guardo mi colab.

Para conocer esa dirección simplemente posicionamos el mouse sobre la carpeta y con clic derecho seleccionamos la opción: "copiar ruta de acceso".



Listo! Es momento de codear.

En nuestro bloque de código, les dejo un script para generar el primer archivo:

```
archivo = open("/content/drive/MyDrive/PC-Practicas/nuevo_archivo.txt", "w")
archivo.write("Hola Mundo!")
archivo.close()
```

En `open()`, como sabemos, tenemos 2 parámetros principales para decidir. El primero es una ruta que utilizaremos en este caso para crear nuestro archivo, y el segundo parámetro para decidir en qué modo se establece nuestro archivo, en nuestro caso en modo escritura ("w").

Entonces para el primer parámetro pegamos la ruta de acceso que nos copiamos previamente, en este caso:

```
/content/drive/MyDrive/1-PC-Practicas
```

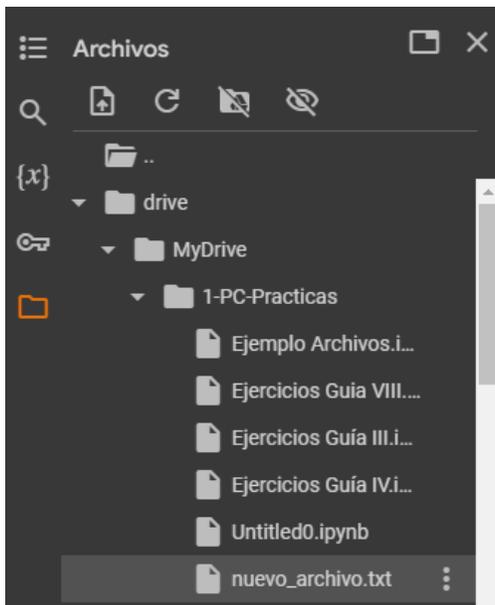
y le agregamos <una barra: /> + <el nombre de archivo que tendrá>, en este caso “/nuevo\_archivo.txt”, quedando la dirección:

```
/content/drive/MyDrive/1-PC-Practicas/nuevo_archivo.txt
```

Ahora corremos el script con ambos parámetros de open configurados y listo!

Tenemos nuestro primer archivo generado desde Colab.

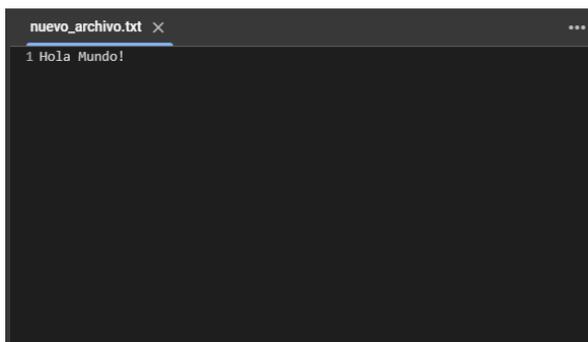
Para verificar que se haya generado, vamos a archivos en colab y vemos que esté donde los guardamos.



Efectivamente, refrescamos y vemos que se generó el archivo correctamente.

**Nota:** Si ven que no se actualiza y no figura el archivo, pueden dar clic en refrescar  para actualizar la vista de archivos y que aparezca nuestro archivo generado.

Para ver el contenido del archivo y verificar que tiene el contenido que le dimos se realiza un doble clic sobre el archivo, se abrirá una ventana como la siguiente:



## Leer un archivo desde colab

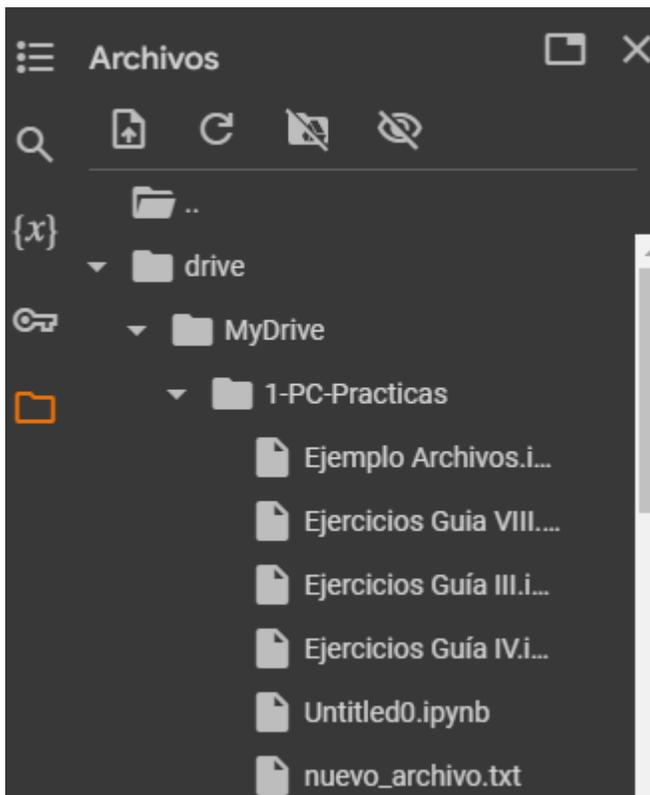
Para comenzar, tendremos que tener nuestro/s archivo/s que queramos leer dentro de nuestro Google Drive, ya sea habiendo generado el archivo como vimos en la sección anterior ó bien subiendo manualmente un archivo que queramos leer a nuestro drive.

Nuestra recomendación es que tengan el archivo en la misma carpeta donde tienen sus colabs, por una cuestión de organización, pero pueden guardarlo en cualquier lugar dentro de drive.

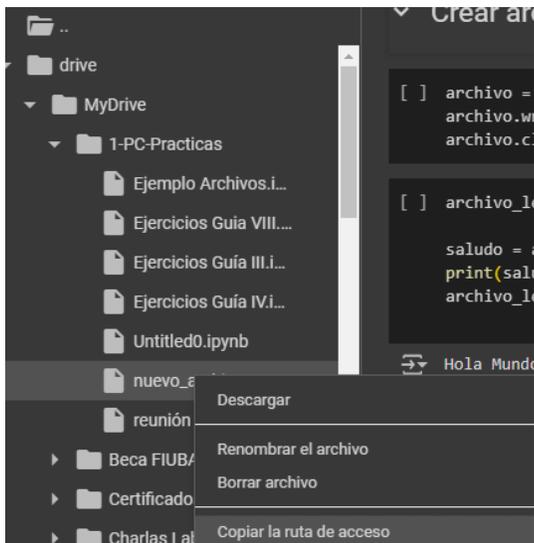
**Nota:** Antes de comenzar deben tener realizado el paso: [Conectar nuestro google drive a colab.](#)

En este ejemplo vamos a leer el archivo que generamos previamente llamado “nuevo\_archivo.txt” y que se encuentra en el drive.

Para leerlo, primero buscamos en qué dirección se encuentra nuestro archivo:



Para realizar la apertura del archivo tenemos que conocer el path ó también llamada dirección de ubicación del archivo. Para ello, hacemos clic derecho sobre el archivo que queremos leer y nos copiamos la dirección de ubicación(Drive la llama “ruta de acceso”):



Ahora podemos abrir un bloque de código y ponernos a trabajar con el archivo:

```
▼ Lectura de archivo en Drive - Ejemplo

# abro el archivo
archivo_leido = open("/content/drive/MyDrive/1-PC-Practicas/nuevo_archivo.txt", "r")

# proceso el archivo
saludo = archivo_leido.readline()
print(saludo)

# cierro el archivo, para no ocupar lugar en memoria.
archivo_leido.close()
```

En el `open()`, debemos colocar en principio dos flags:

- Primero para indicar la dirección donde se encuentra el archivo en este caso a leer,
- Como segundo parámetro el modo en que se abre el archivo, en este caso en modo lectura.

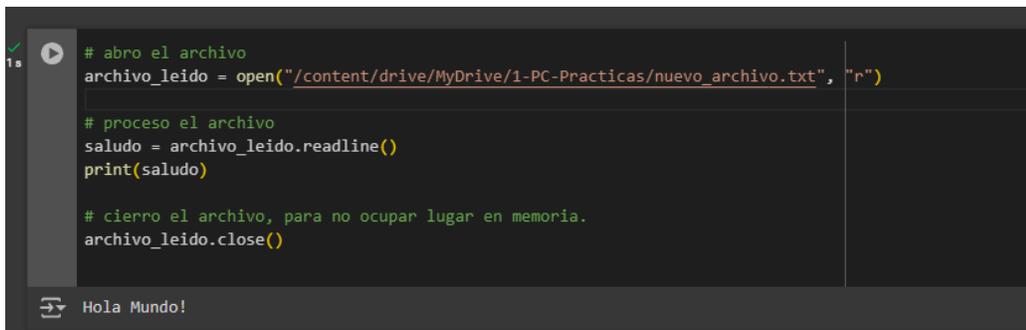
**Nota:** este último parámetro para el caso de la lectura se puede omitir, ya que por defecto, `open()` abre el archivo en modo lectura.

Entonces pegamos la dirección de nuestro archivo en el primer parámetro y como segundo parámetro, "r" (viene de *read* del inglés) para indicar modo lectura.

Luego, una vez abierto el archivo y almacenado en una variable, podemos comenzar a procesar el archivo de la manera que queramos.

En este caso, simplemente se lee la primera línea del archivo y la muestro por pantalla.

Vemos como queda una vez ejecutada la instrucción:



```
# abro el archivo
archivo_leido = open("/content/drive/MyDrive/1-PC-Practicas/nuevo_archivo.txt", "r")

# proceso el archivo
saludo = archivo_leido.readline()
print(saludo)

# cierro el archivo, para no ocupar lugar en memoria.
archivo_leido.close()
```

Hola Mundo!

Finalmente cierro el archivo con la función `close()` para avisarle al programa que ya no se va a usar el archivo y que libere recursos (memoria ram) para que el sistema operativo pueda usar esa memoria para realizar otras tareas.

Listo! Ya tienen la idea inicial para que puedan leer y escribir sobre archivos.

Les dejamos como tarea, basándose en estas ideas, probar de agregar datos a un archivo que ya contiene datos ó leer y escribir sobre un mismo archivo. Cualquier duda nos consultan en el foro.

## Conclusiones

La herramienta tiene muchas más características que irán descubriendo a medida que vayan utilizando la herramienta. Espero que les sirva.

**Nota:** Esta herramienta también la utilizaremos para lo que resta del curso.