



Uso de los materiales didácticos mediante

GOOGLE COLAB

¡Bienvenido al proyecto MACTI!

¿Qué es el proyecto MACTI? 3

¿Qué es Google Colab? 4

¿Dónde obtengo los materiales didácticos? 9

¿Cómo comienzo a usar los materiales didácticos? 11

¿Cómo cargar códigos dependientes en Google Colab? 15

¿Qué es el proyecto MACTI?

Macti surgió de un proyecto PAPIME¹ orientado en la creación de materiales didácticos para reforzar los conocimientos de materias como Cálculo, Álgebra Lineal, Análisis Numérico y Ecuaciones Diferenciales Ordinarias y Parciales. El objetivo principal es que los estudiantes que cursan estas materias tengan a la mano herramientas interactivas con las que puedan practicar lo aprendido en clase. Estas herramientas están construidas en el lenguaje *Python* y todo su entorno científico; algunas de ellas permiten modificar parámetros para entender con certeza cada concepto de las materias antes mencionadas. Las herramientas son de acceso libre² y abierto³; se puede acceder a ellas a través de un repositorio en Github. Están en formato *.ipynb*, es decir notebooks, por lo que se pueden usar mediante *Google Colab* o *Jupyter*.

1 1 PAPIME: Programa de Apoyo a Proyectos para Innovar y Mejorar la Educación

2 Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional

3 El código de las herramientas es visible y se puede modificar.

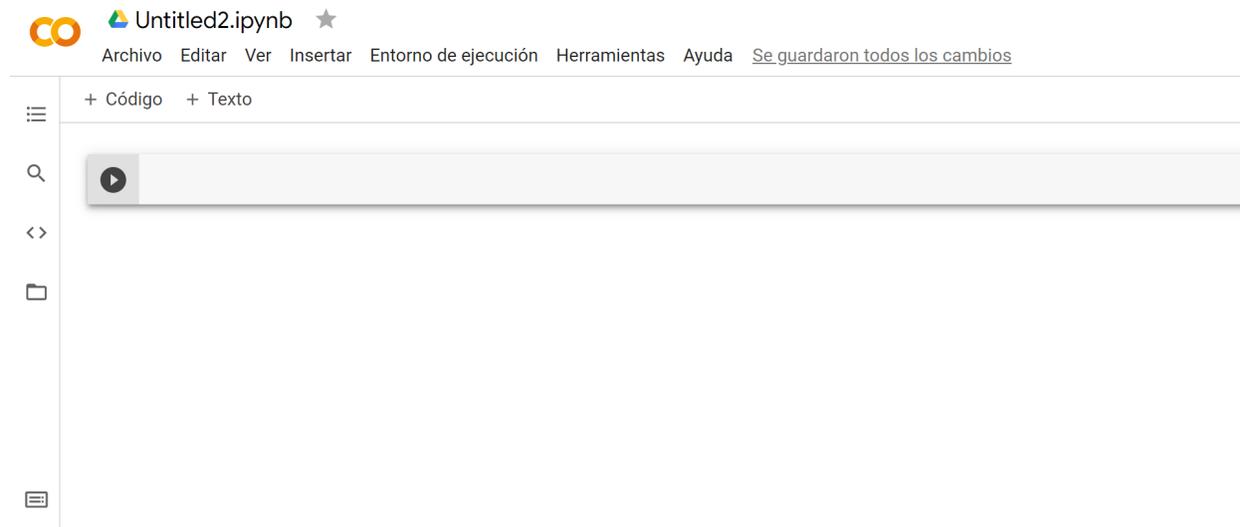
¿Qué es Google Colab?

Google Colab, también conocido como *Collaboratory*, es un entorno gratuito basado en *Jupyter notebook* que se ejecuta en los servidores de la nube de *Google*. No requiere instalación ni configuración de Python. Se tiene acceso gratuito a hardware CPU, GPU y TPU. Se puede compartir el código de manera similar que con *Google Drive*.

Para acceder a este servicio lo primero que se requiere es una cuenta en Gmail. Una vez que tienes la cuenta, hay que ingresar a ella y luego, en otra ventana del navegador, ir a la siguiente dirección <https://colab.research.google.com/>. Lo anterior te llevará a la página de inicio de *Collaboratory* y en ella encontrarás un excelente tutorial para iniciar. Adicionalmente, el siguiente [video](#) proporciona una introducción del uso de este ambiente.

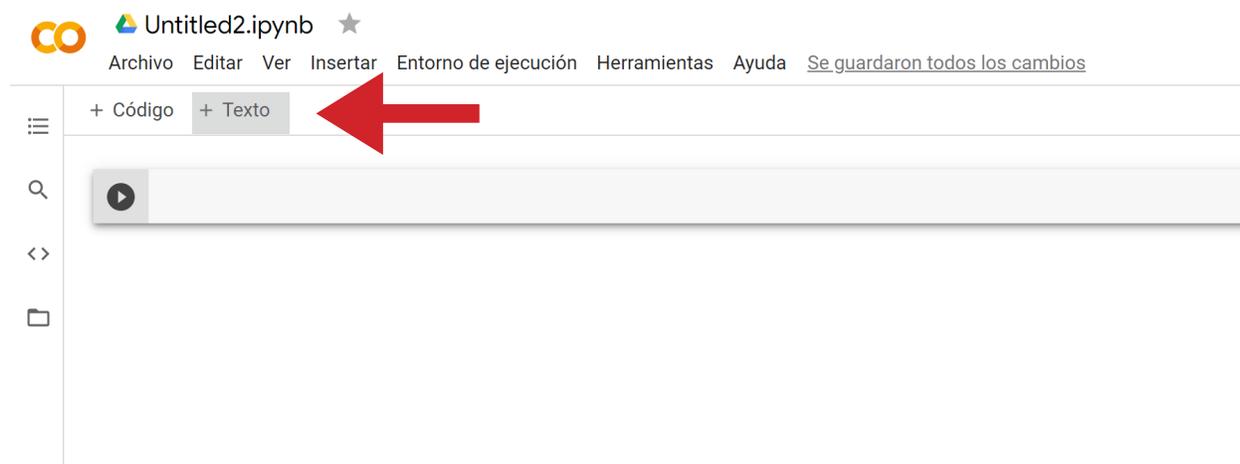
Para introducirnos un poco más en el mundo de *Google Colab* identifiquemos los aspectos básicos de una notebook en Colab.

1. Cuando se inicia una notebook obtendremos una vista como la siguiente:

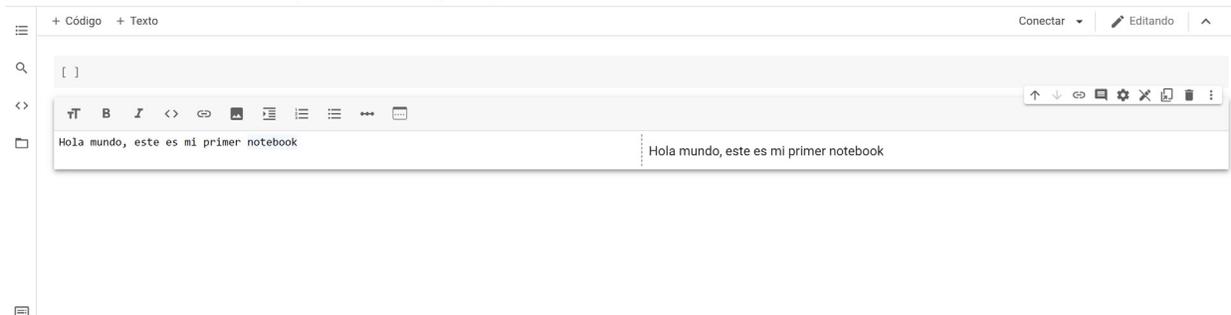


2. En *Google Colab*, así como en *Jupyter Notebook*, existen dos tipos de celdas: Las enfocadas en el texto y las enfocadas en el código. Como bien lo describe su nombre la primera es útil para la redacción de texto incluyendo *HTML* y *LaTeX*, mientras que la segunda está enfocada a la ejecución de código en el lenguaje de programación *Python*.

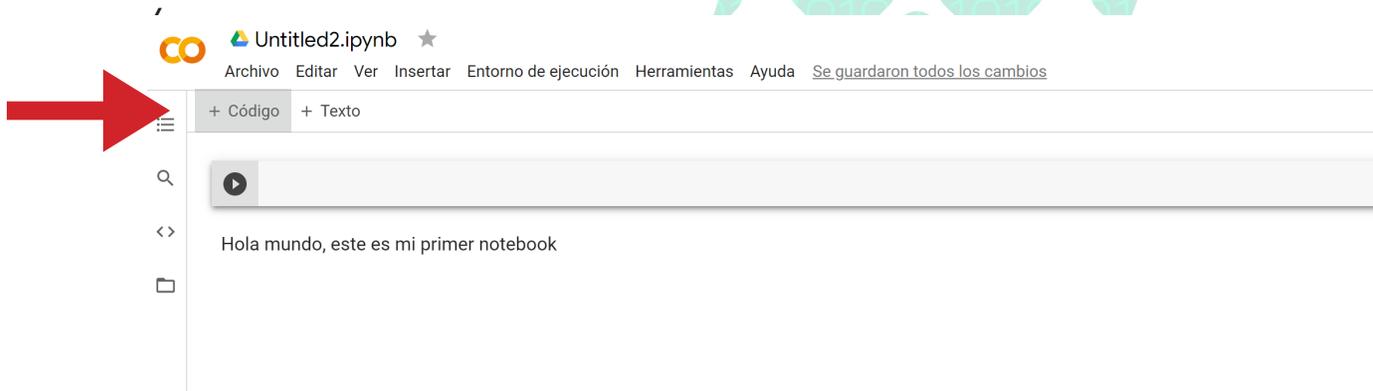
3. Para agregar una celda de texto será necesario dar clic en el siguiente botón.



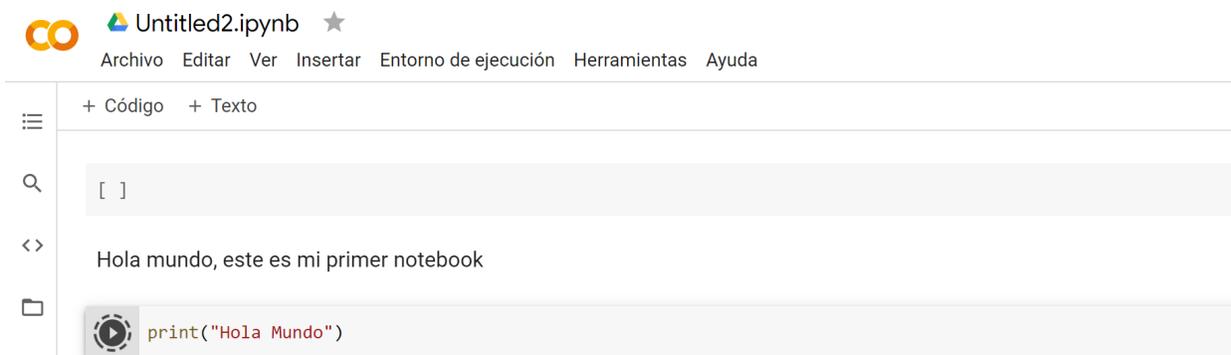
4. Posteriormente bastará con escribir el texto que deseas redactar, en este tipo de celdas cuentan con una barra de herramientas básica para modificar el texto como subrayar, letras negritas, viñetas , itálicas, etc. Cuando edites una celda, en el lado derecho podrás encontrar una vista previa del texto que estás editando. Un ejemplo es el siguiente:



5. Para agregar una celda de código será necesario dar clic en el botón siguiente:



6. La acción anterior insertará una celda en la cual podrás escribir y ejecutar código, de la siguiente manera:





Untitled2.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se guardaron todos los cambios](#)



+ Código + Texto



[]



Hola mundo, este es mi primer notebook



 `print("Hola Mundo")`

Hola Mundo

7. Al editar un código es posible que realices comentarios sobre tu código agregando el símbolo # al inicio de la línea, de esta forma al ejecutar el código, el entorno omitirá dicha línea dado que es una línea con fines informativos. Podrás identificar las líneas comentadas por que ellas tienen un color verde, a diferencia de los colores en otras líneas, las cuales serán ejecutadas.



Untitled2.ipynb ★

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se guardaron todos los cambios](#)



+ Código + Texto



[]



Hola mundo, este es mi primer notebook



[1] `print("Hola Mundo")`

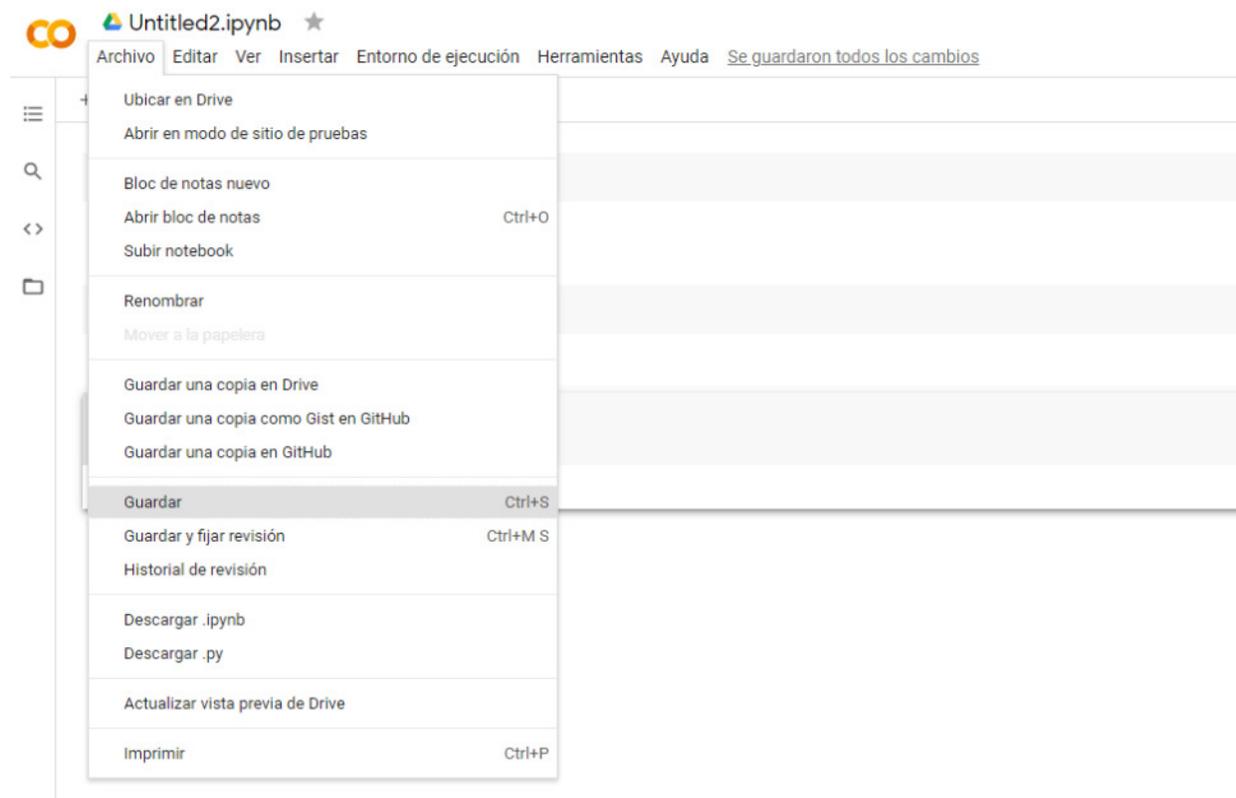
Hola Mundo

 `#Esta línea es un comentario`
`print("Esta línea no es un comentario")`

Esta línea no es un comentario



8. Google Colab cuenta con el guardado automático que algunos editores tienen implementados, pero si deseas asegurarte de guardar tus cambios puedes utilizar la combinación de teclas [CTRL + S] o hacer clic en el menú: *Archivo + Guardar*.



¿Dónde obtengo los materiales didácticos?

El proyecto Macti cuenta con un repositorio público en Github¹ al cual se puede acceder a través de la siguiente dirección:

<https://github.com/jugernaut/Prometeo>. Cuando accedas a esa dirección obtendrás algo como lo que se muestra en la figura 1. Para bajar el código a tu equipo solo tienes que hacer clic en el botón verde que dice Code y elegir Download ZIP. Posteriormente, puedes descomprimir el archivo y navegar por todas las carpetas usando *Jupyter Notebook*.

1 GitHub es una compañía sin fines de lucro que ofrece un servicio de hospedaje de repositorios de documentos (mayormente software). Permite el uso del sistema de control de versiones conocido como git de una manera sencilla.

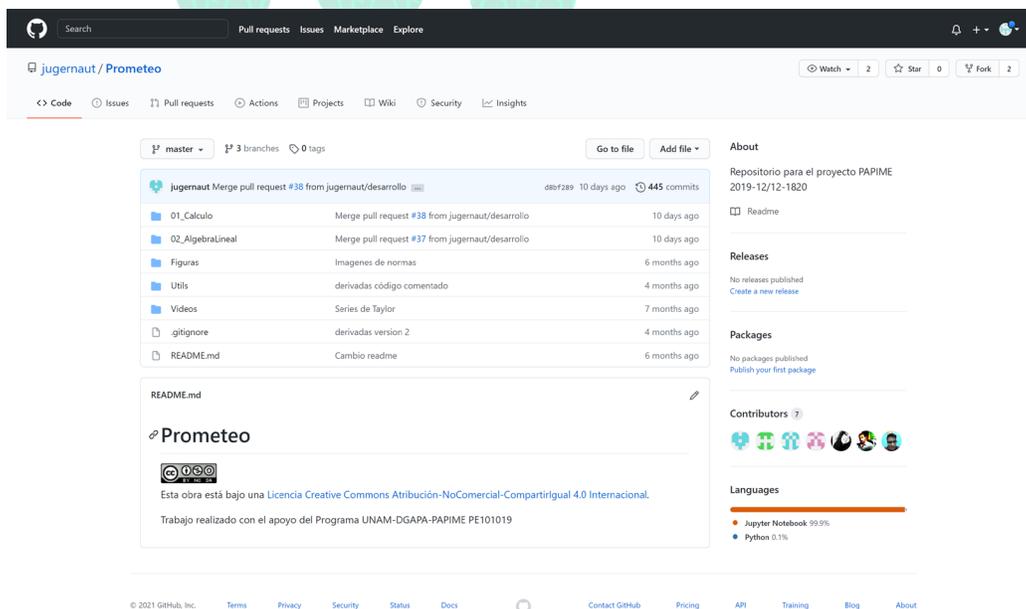
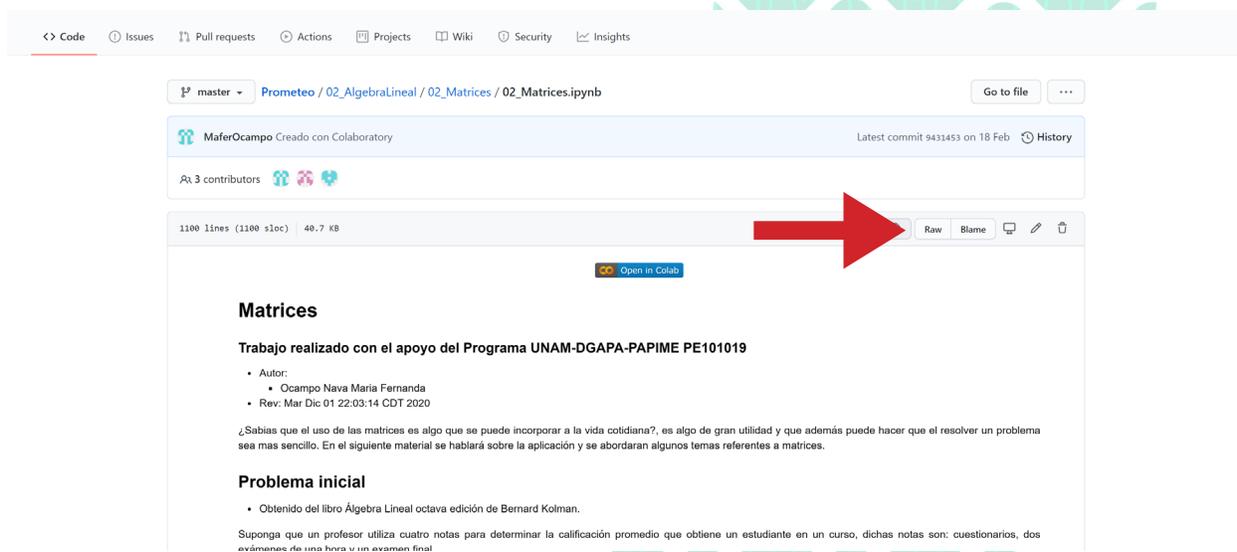


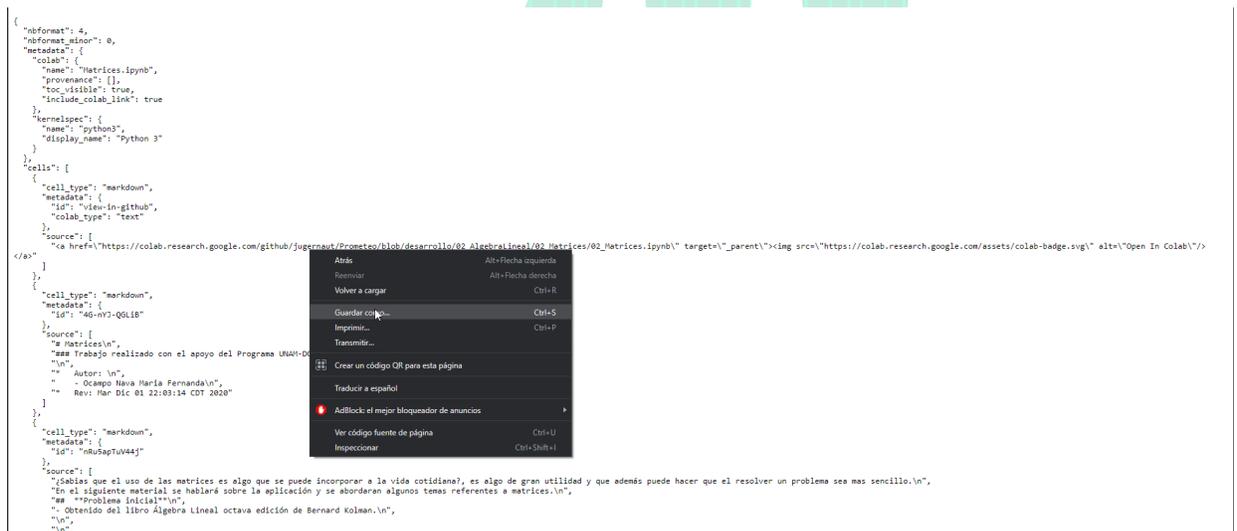
Figura 1. Repositorio Prometeo con los materiales didácticos del proyecto Macti.

También puedes descargar un archivo individual del repositorio. Para ello debes realizar lo siguiente:

1. Elegir el archivo que deseas descargar.
2. Abrir el archivo para visualizarlo.
3. Dar clic en Raw



4. Dar clic derecho y elegir la opción guardar como.
5. Elige la carpeta en tu PC donde deseas guardar el archivo.



Otra manera de hacer uso de estos materiales es mediante Google Colab y eso lo explicaremos a continuación.

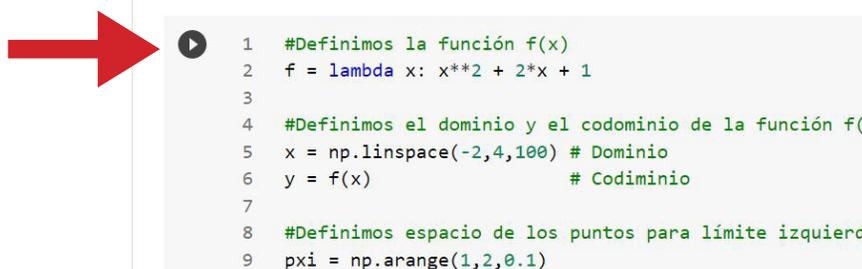
¿Cómo comienzo a usar los materiales didácticos?

1. En el repositorio [Prometeo](#) del proyecto MACTI encontrarás dos carpetas 01_Calculo y 02_AlgebraLineal. Elige la que sea de tu interés para posteriormente seleccionar un tema dicha carpeta. Cada tema tiene uno o varios notebooks en los cuales se explican conceptos de la materia seleccionada usando implementaciones en Python.
2. En el sitio de Github podrás visualizar una “vista previa” del notebook. En esta vista no podrás interactuar con el código, sin embargo podrás notar que al principio de cada notebook hay un botón que dice Open in Colab; si haces clic en dicho botón obtendrás una nueva pestaña con el notebook abierto en Colab en donde si podrás interactuar.



3. Para ejecutar el código de una notebook en Google Colab, se debe realizar lo siguiente:

-En la celda que deseas modificar debes localizar el siguiente botón, el cual se encuentra en la parte superior izquierda de cada celda de código.



-Debajo la celda que seleccionaste para ejecutar podrás ver el resultado de la ejecución. Por ejemplo, consideremos que queremos ejecutar la siguiente celda.

```
▶ # Calculo de la serie telescopica
def serietelescopica(inicio,final):
    control = 0
    for x in range(inicio, final+1):
        control += (1/x)-(1/(x+1)) #Se define la serie del ejemplo anterior
    return control

serietelescopica(1,8)
```

Al dar clic en el botón antes mencionado la celda comenzará a ejecutarse.



```
▶ # Calculo de la serie telescopica
def serietelescopica(inicio,final):
    control = 0
    for x in range(inicio, final+1):
        control += (1/x)-(1/(x+1)) #Se define la serie del ejemplo anterior
    return control

serietelescopica(1,8)
```

Y lo podremos identificar mediante las líneas punteadas en el perímetro del botón.

Cuando la celda haya finalizado podremos observar nuestro resultado debajo del código ejecutado.

```
▶ # Calculo de la serie telescopica
def serietelescopica(inicio,final):
    control = 0
    for x in range(inicio, final+1):
        control += (1/x)-(1/(x+1)) #Se define la serie del ejemplo anterior
    return control

serietelescopica(1,8)
```



0.8888888888888888

5. Para modificar el código de las notebooks basta con seleccionar una celda, realizar los cambios que te gustaría intentar y ejecutar la celda, veamos un ejemplo:

-Retomando el ejemplo anterior, tenemos la siguiente celda:

```
▶ # Calculo de la serie telescopica
def serieteleoscopica(inicio,final):
    control = 0
    for x in range(inicio, final+1):
        control += (1/x)-(1/(x+1)) #Se define la serie del ejemplo anterior
    return control

serieteleoscopica(1,8)

0.8888888888888888
```

-A la cual se le desea realizar una modificación para calcular una serie telescópica de distintos índices, por ejemplo deseamos calcular la del 1 al 8. Por lo que realizaremos la modificación en la función de la siguiente manera:

```
▶ 1 # Calculo de la serie telescopica
2 def serieteleoscopica(inicio,final):
3     control = 0
4     for x in range(inicio, final+1):
5         control += (1/x)-(1/(x+1)) #Se define la serie del ejemplo anterior
6     return control
7
8 serieteleoscopica(1,15)

0.8888888888888888
```

-Y finalmente ejecutamos el código de la celda modificada como se ha visto en la sección anterior.

```
▶ 1 # Calculo de la serie telescopica
2 def serieteleoscopica(inicio,final):
3     control = 0
4     for x in range(inicio, final+1):
5         control += (1/x)-(1/(x+1)) #Se define la serie del ejemplo anterior
6     return control
7
8 serieteleoscopica(1,15)

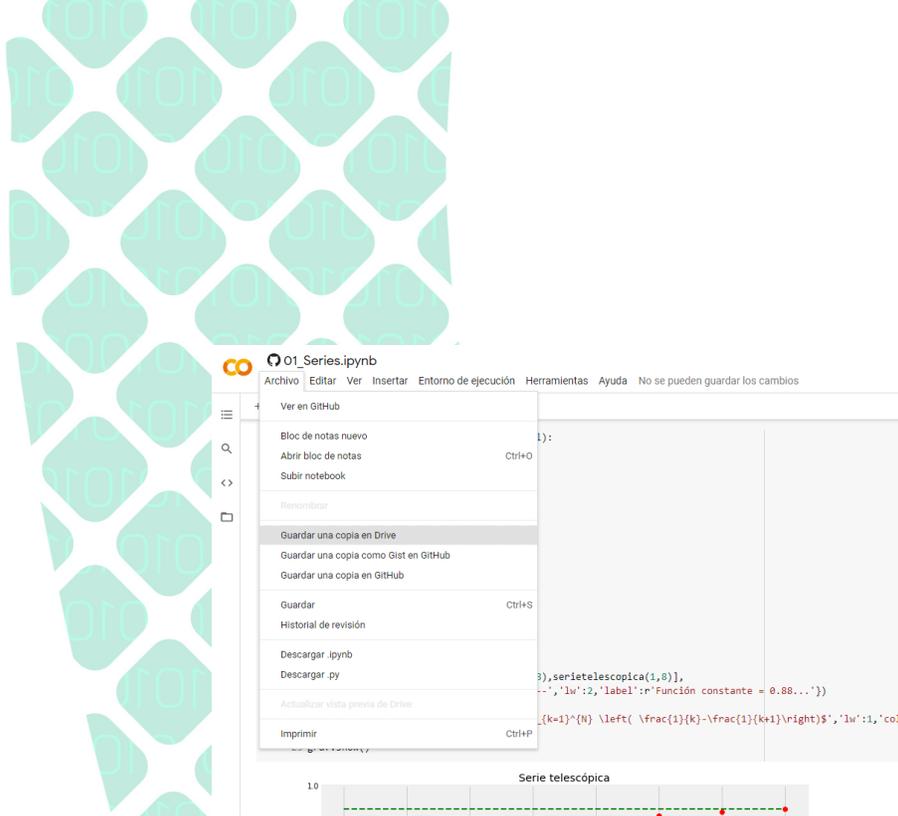
0.9374999999999998
```

Existen celdas de código que son dependientes de otras celdas desarrolladas anteriormente por lo que se recomienda ejecutar las celdas consecutivamente desde el inicio. Por ejemplo, en el siguiente código se tiene una primera celda que define una función para realizar el cálculo de una serie telescópica; posteriormente, se tiene una segunda celda para graficar la serie telescópica calculada en la primera celda; para evitar obtener un error, se deben ejecutar las celdas en orden, dado que la segunda celda depende de la primera.

```
[ ] 1 def serietelescopiacompleta(inicio,final):
2     lista = []
3     r = 0
4     for x in range(inicio, final+1):
5         r += (1/x)-(1/(x+1))
6         lista.append(r)
7     return lista
8
9 x4 = np.arange(1, 9, 1)
10 y4 = serietelescopiacompleta(1,8)
11 par = [{'title':'Serie telescópica',
12         'xlabel':'$n$',
13         'ylabel':'$$$',
14         'ylim':(0,1)}]
15
16
17 graf = vis.planoCartesiano(par=par)
18 graf.plot(x=[1,8],y=[serietelescopica(1,8),serietelescopica(1,8)],
19          par={'c':'green','linestyle':'--','lw':2,'label':r'Función constante = 0.88...'})
20 graf.scatter(x=x4, y=y4,
21             par={'label':r'Serie: $\sum_{k=1}^N \left( \frac{1}{k} - \frac{1}{k+1} \right)$', 'lw':1, 'color':'red', 'zorden':5})
22 graf.legend(par={'loc':'lower right'})
23 graf.show()
```



Si te interesa guardar una versión de una de las notebooks en tu propio Google Drive, tendrás que ir al menú **Archivo > Guardar una copia en Drive**.

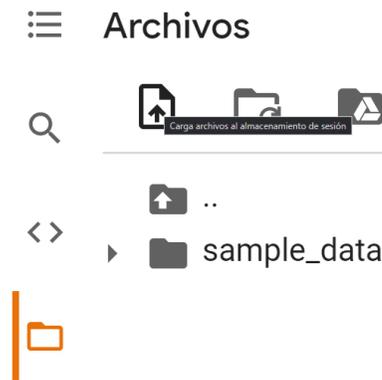


¿Cómo cargar códigos dependientes en Google Colab?

Se presentan dos opciones para poder cargar al ambiente de Google Colab códigos, archivos o programas de los cuales depende algún notebook.

Método 1: Cargar archivos desde tu computadora.

En caso de tener los archivos descargados en su PC, solo necesitarás realizar lo siguiente:



Seleccionar la carpeta de archivos ubicada en la barra lateral izquierda de colab  y presionar el icono de archivo  dicho botón abrirá tu navegador de archivos para que puedas seleccionar el archivo deseado. Posteriormente el código estará en el ambiente de Google Colab y podrás usarlo en todos los códigos dependientes.

Método 2: Clonar un repositorio

Para este método será necesario que el archivo deseado ajeno al notebook, se encuentre en un repositorio de Github. Necesitas ejecutar un código como el que se muestra en la figura siguiente; debes introducir el enlace del repositorio con su respectivo nombre y usuario:

```
1 !git clone https://github.com/jugernaut/Prometeo.git  
2
```

Al ejecutar esta celda Google Colab clonará el repositorio para que los archivos deseados puedan ser utilizados y mostrará algo similar a lo siguiente:

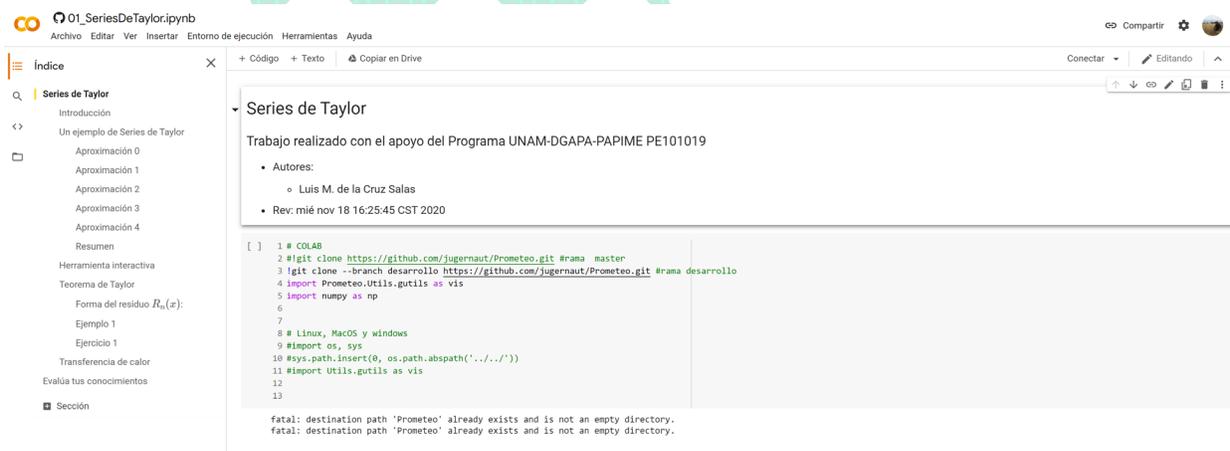
```
Cloning into 'Prometeo'...  
remote: Enumerating objects: 242, done.  
remote: Counting objects: 100% (242/242), done.  
remote: Compressing objects: 100% (239/239), done.  
remote: Total 2307 (delta 127), reused 0 (delta 0), pack-reused 2065  
Receiving objects: 100% (2307/2307), 14.78 MiB | 25.39 MiB/s, done.  
Resolving deltas: 100% (1452/1452), done.
```

Podemos ahora importar los archivos existentes en el repositorio que se acaba de clonar de la siguiente manera:

```
[ ] 1 import os
    2 os.listdir("Prometeo")
```

```
['README.md',
 '01_Calculo',
 '.git',
 '.gitignore',
 'Videos',
 '02_AlgebraLineal',
 'Utils',
 'Figuras']
```

Para ejemplificar lo anterior, abriremos el notebook 01_SeriesDeTaylor.ipynb que se encuentra en la carpeta 01_Calculo, obtendremos lo siguiente:



01_SeriesDeTaylor.ipynb

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

Compartir

Índice

Series de Taylor

- Introducción
- Un ejemplo de Series de Taylor
 - Aproximación 0
 - Aproximación 1
 - Aproximación 2
 - Aproximación 3
 - Aproximación 4
- Resumen
- Herramienta interactiva
- Teorema de Taylor
- Forma del residuo $R_n(x)$:
- Ejemplo 1
- Ejercicio 1
- Transferencia de calor
- Evalúa tus conocimientos

Sección

Series de Taylor

Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME PE101019

- Autores:
 - Luis M. de la Cruz Salas
- Rev: mié nov 18 16:25:45 CST 2020

```
[ ] 1 # COLAB
    2 #!git clone https://github.com/jugernaut/Prometeo.git #rama master
    3 #!git clone --branch desarrollo https://github.com/jugernaut/Prometeo.git #rama desarrollo
    4 import Prometeo.Utils.guttils as vis
    5 import numpy as np
    6
    7
    8 # Linux, MacOS y windows
    9 #import os, sys
   10 #sys.path.insert(0, os.path.abspath('../..'))
   11 #import Utils.guttils as vis
   12
   13
```

fatal: destination path 'Prometeo' already exists and is not an empty directory.
fatal: destination path 'Prometeo' already exists and is not an empty directory.

Esta notebook tiene dependencias con otros archivos así como con otros programas realizados para el proyecto Macti. Para que esta notebook funcione correctamente, se debe ejecutar el código de la siguiente celda:

```

1 # COLAB
2 #!git clone https://github.com/jugernaut/Prometeo.git #rama master
3 !git clone --branch desarrollo https://github.com/jugernaut/Prometeo.git #rama desarrollo
4 import Prometeo.Utills.gutils as vis
5 import numpy as np
6
7
8 # Linux, MacOS y windows
9 #import os, sys
10 #sys.path.insert(0, os.path.abspath('../..'))
11 #import Utills.gutils as vis
12
13
Cloning into 'Prometeo'...
remote: Enumerating objects: 242, done.
remote: Counting objects: 100% (242/242), done.
remote: Compressing objects: 100% (239/239), done.
remote: Total 2307 (delta 127), reused 0 (delta 0), pack-reused 2065
Receiving objects: 100% (2307/2307), 14.78 MiB | 27.56 MiB/s, done.
Resolving deltas: 100% (1452/1452), done.

```

De esta manera el repositorio completo se ha clonado en la sesión interactiva actual, y ya es posible ejecutar esta notebook que depende de otros archivos.

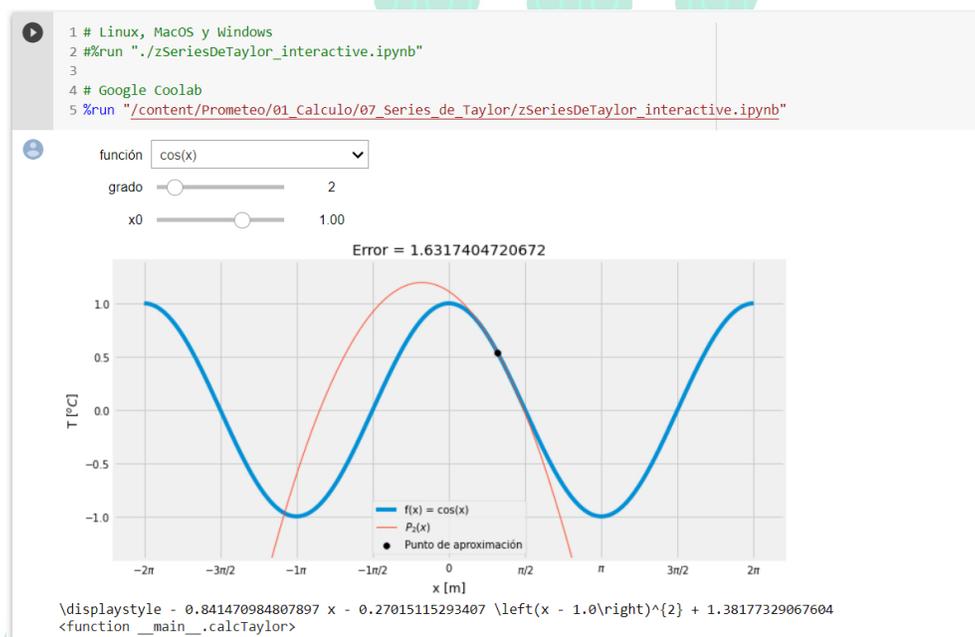
En esta misma notebook, se utiliza una herramienta, la cual se ejecuta como se muestra en la siguiente figura:

```

[ ] 1 # Linux, MacOS y Windows
2 #%run "./zSeriesDeTaylor_interactive.ipynb"
3
4 # Google Coolab
5 #%run "/content/Prometeo/01_Calculo/07_Series_de_Taylor/zSeriesDeTaylor_interactive.ipynb"

```

Como se dice en las instrucciones de esta herramienta, para poder ejecutarla en Google Colab, quitaremos el comentario de la línea 5 y ejecutamos:



TRABAJO REALIZADO CON EL APOYO DEL
PROGRAMA
UNAM-DGAPA-PAPIME PE101019



Modelación computacional en las ciencias y las
ingenierías como apoyo en el proceso
enseñanza-aprendizaje