



Uso de los materiales didácticos mediante

JUPYTER

¡Bienvenido al proyecto MACTI!

¿Qué es el proyecto MACTI?

3

¿Qué es Jupyter Notebook?

4

¿Cómo se instala Jupyter Notebook?

5

¿Dónde obtengo los materiales didácticos?

11

¿Cómo comienzo a usar los materiales didácticos?

13

¿Cómo cargar códigos dependientes en Jupyter

17

Notebook?

¿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. Éstas 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 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 Jupyter Notebook?

Jupyter Notebook es una aplicación lanzada en 2015. Permite la creación y el intercambio de documentos web en formato JSON , que siguen un esquema versionado y una lista ordenada de celdas de entrada / salida. Estas celdas ofrecen espacio para código, texto, fórmulas y ecuaciones matemáticas o contenido multimedia, entre otras cosas.

La aplicación se puede iniciar con todos los navegadores habituales, se puede utilizar para procesar su portátil si el servidor Jupyter Notebook también está **instalado**. Los documentos de Jupyter creados se pueden exportar como documentos HTML, PDF, Markdown o Python. Alternativamente, se pueden compartir con otros usuarios a través del correo electrónico, Dropbox, GitHub o el propio visor de Jupyter Notebook.

Se incluirá una breve introducción al uso de Jupyter Notebook y adicionalmente, el siguiente [video](#) proporciona una introducción del uso de este ambiente, así como un recurso para la instalación de [Jupyter](#).

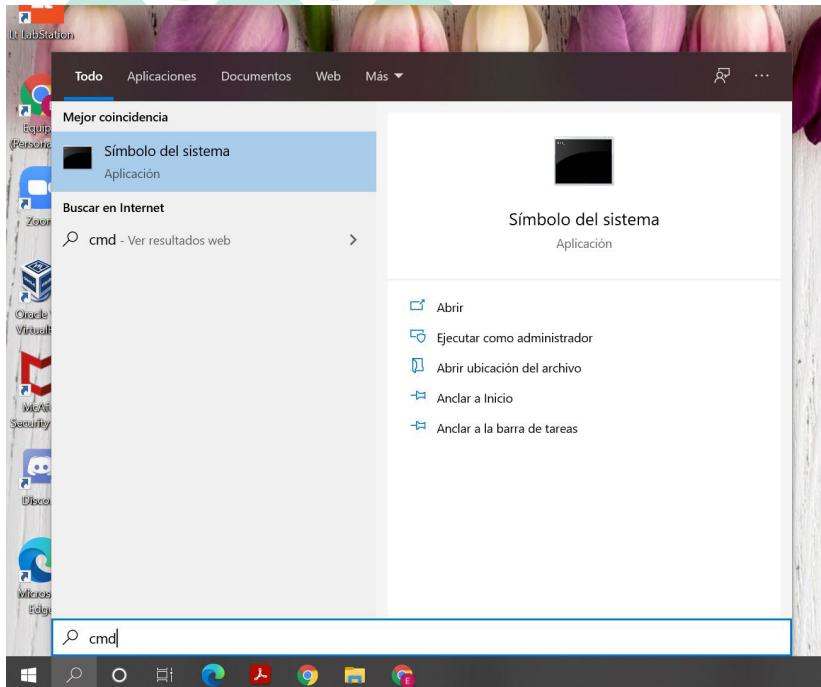
¿Cómo se instala Jupyter Notebook?

Se abordará una breve descripción sobre la instalación de Jupyter Notebook en los sistemas Windows y Linux.

Windows

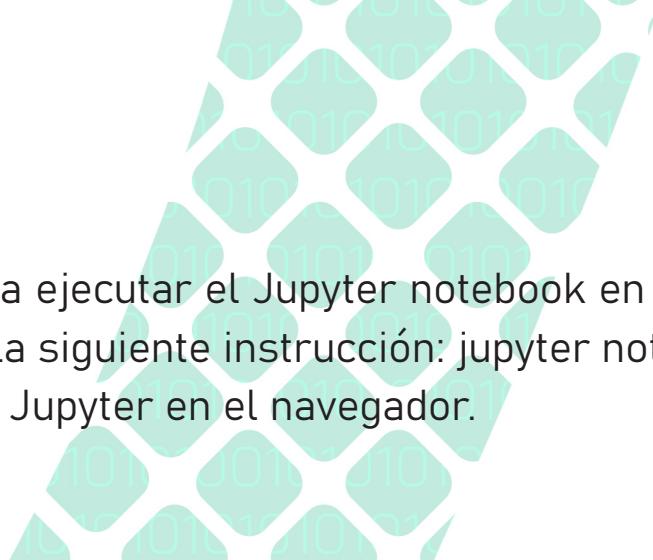
Es necesario que su computadora tenga instalado Python, y este se encuentre agregado al PATH de Windows. Además, para la instalación es necesario que tenga acceso a Internet.

Se debe abrir consola de comandos de Windows. En la barra de búsqueda de Windows, escribir cmd y seleccionar Símbolo del sistema.



En la línea de comandos escribir la siguiente instrucción:

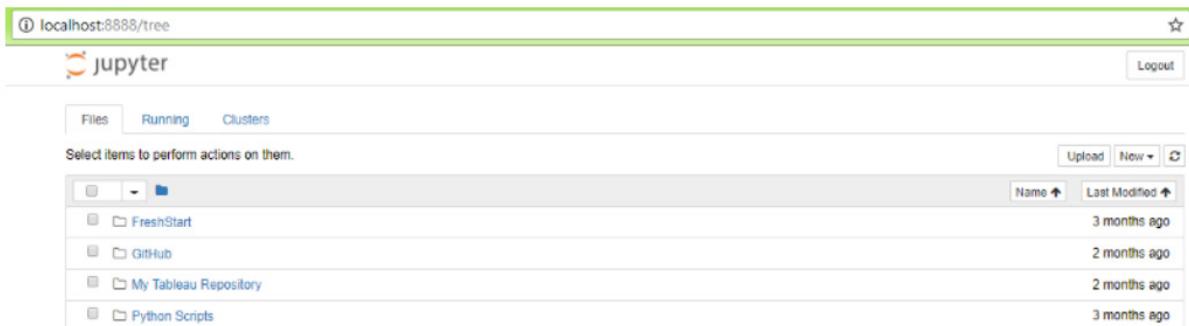
`pip install jupyter` y presionar *Enter*. Python descargará e instalará las librerías necesarias para utilizar jupyter



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18363.1379]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Cienciae1mx>pip install jupyter
```

Posteriormente para ejecutar el Jupyter notebook en la línea de comandos escribir la siguiente instrucción: jupyter notebook con la cual se ejecutará Jupyter en el navegador.



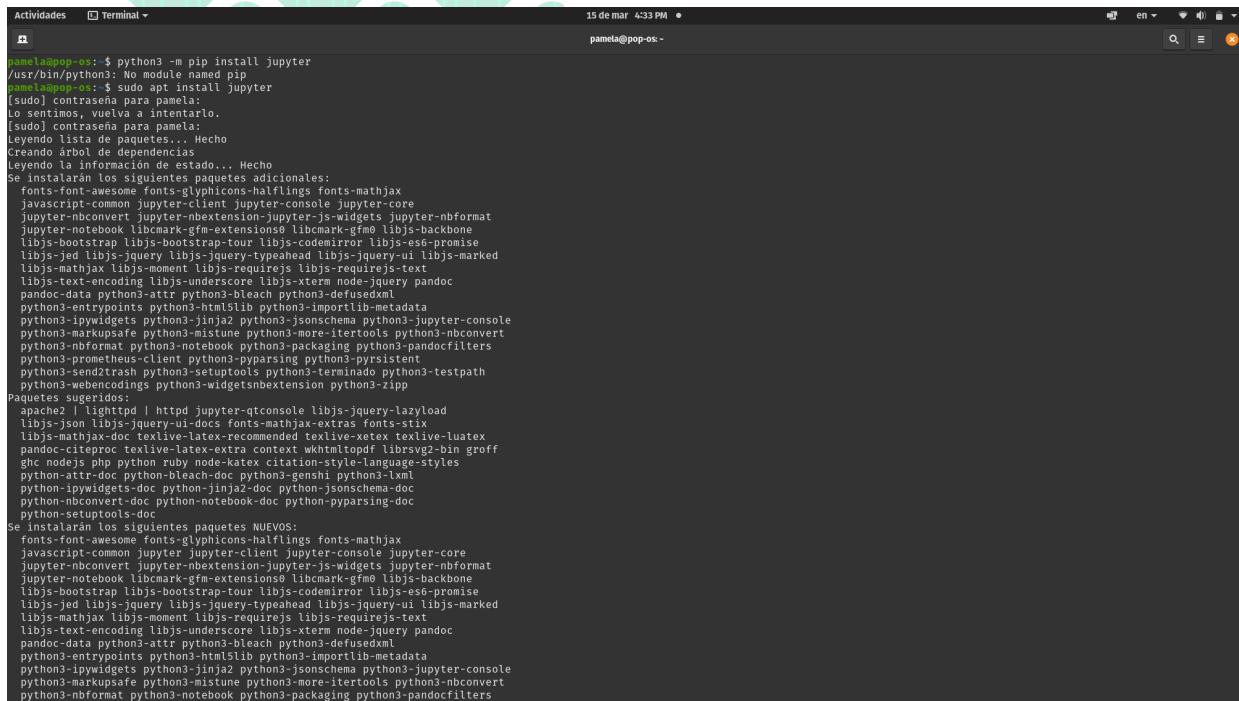
The screenshot shows a web browser window with the following details:

- Title Bar:** Shows the URL `localhost:8888/tree` and the word "jupyter".
- Header:** Includes a "Logout" button.
- Navigation:** Buttons for "Files", "Running", and "Clusters".
- Actions:** Buttons for "Upload", "New", and "Create".
- File List:** A table showing the contents of the current directory:

| | Name | Last Modified |
|---|-----------------------|---------------|
| 📁 | FreshStart | 3 months ago |
| 📁 | GitHub | 2 months ago |
| 📁 | My Tableau Repository | 2 months ago |
| 📁 | Python Scripts | 3 months ago |

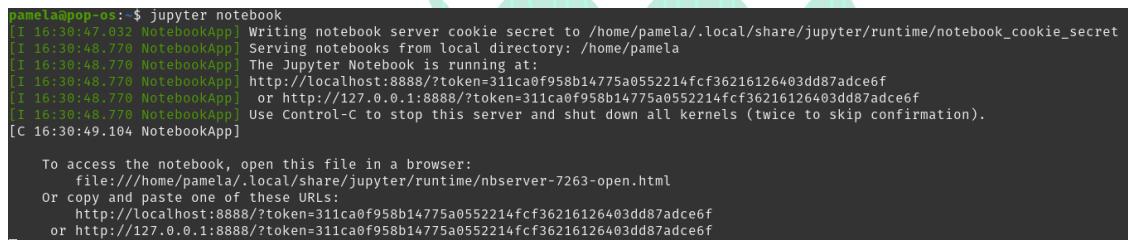
Linux

En Linux bastará con ejecutar el siguiente comando en el bash
sudo apt **install jupyter** en ocasiones es necesario realizar algunas actualizaciones en Linux por lo que se recomienda seguir en siguiente [tutorial](#).



```
pamelapop-os:~$ python3 -m pip install jupyter
/usr/bin/python3: No module named pip
pamelapop-os:~$ sudo apt install jupyter
[sudo] contraseña para pamela:
Lo sentimos, vuelva a intentarlo.
[sudo] contraseña para pamela:
Leyendo lista de paquetes... Hecho
Creado árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
fonts-font-awesome fonts-glyphicons-halflings fonts-mathjax
javascript-common jupyter-client jupyter-console jupyter-core
jupyter-nbconvert jupyter-nbextension jupyter-js-widgets jupyter-nbformat
jupyter-notebook libmark-gfm-extensions8 libmark-gfm libjs-backbone
libjs-busy.js libjs-jquery libjs-jquery-typeahead libjs-es6-promise
libjs-es6-promise libjs-jquery libjs-jquery-typeahead libjs-es6-promise libjs-marked
libjs-mathjax libjs-moment libjs-requirejs libjs-xterm libjs-query pandoc
pandoc-data python3-attr python3-bleach python3-defusedxml
python3-EntryPoint python3-html5lib python3-importlib-metadata
python3-ipypwidgets python3-jinja2 python3-jsonschema python3-jupyter-console
python3-markupsafe python3-mistune python3-more-itertools python3-nbconvert
python3-nbformat python3-notebook python3-packaging python3-pandocfilters
python3-prometheus-client python3-pyparsing python3-persistent
python3-send2trash python3-setuputils python3-terminal python3-testpath
python3-webencodings python3-wget nbextension python3-zip
Paquetes sugeridos:
jupyter nbconvert | http://jupyter-stacks.s3.amazonaws.com/libjs_jquery_lazyload
libjs-json libjs-jquery-ui fonts-mathjax-extras fonts-stix
libjs-mathjax-doc texlive-latex-recommended texlive-xetex texlive-luatex
pandoc-citeproc texlive-latex-extra-context wkhtmltopdf librsvg2-bin groff
ghc nodejs php python ruby node-katex citation-style-language-styles
python-attr-doc python-bleach-doc python3-genshi python3-lxml
python-ipypwidgets-doc python-jinja2-doc python-jsonschema-doc
python-nbconvert-doc python-notebook-doc python-pyparsing-doc
python-setuptools-doc
Se instalarán los siguientes paquetes NUEVOS:
fonts-font-awesome fonts-glyphicons-halflings fonts-mathjax
javascript-common jupyter-client jupyter-console jupyter-core
jupyter-nbconvert jupyter-nbextension jupyter-js-widgets jupyter-nbformat
jupyter-notebook libmark-gfm-extensions8 libmark-gfm libjs-backbone
libjs-bootstrap libjs-bootstrap-tour libjs-codemirror libjs-es6-promise
libjs-jed libjs-jquery libjs-jquery-typeahead libjs-jquery-ui libjs-marked
libjs-mathjax libjs-moment libjs-requirejs libjs-query pandoc
pandoc-data python3-attr python3-bleach python3-defusedxml
python3-EntryPoint python3-html5lib python3-importlib-metadata
python3-ipypwidgets python3-jinja2 python3-jsonschema python3-jupyter-console
python3-markupsafe python3-mistune python3-more-itertools python3-nbconvert
python3-nbformat python3-notebook python3-packaging python3-pandocfilters
```

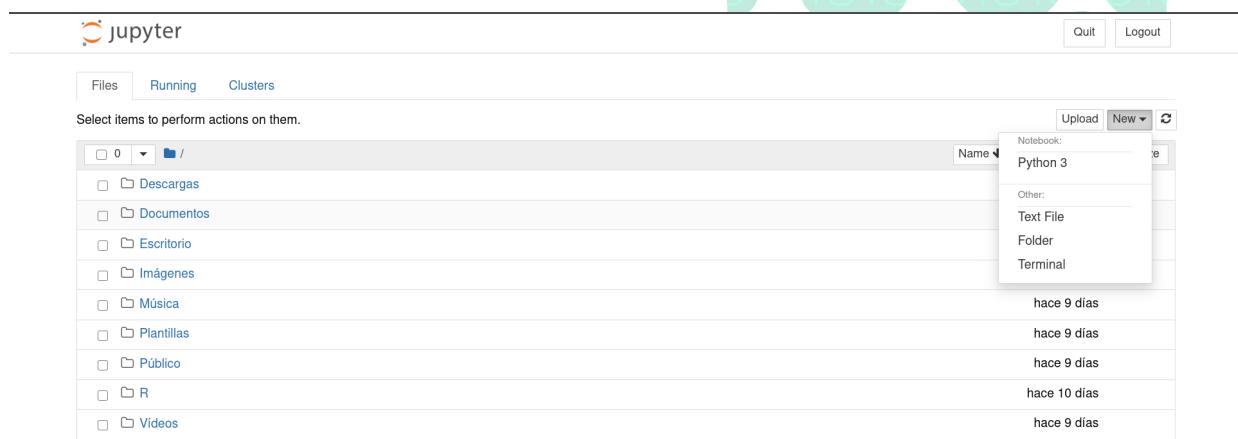
Para finalizar solo será necesario escribir en el bash **jupyter notebook** y se ejecutará en el navegador el inicio de Jupyter Notebook.



```
pamelapop-os:~$ jupyter notebook
[1 16:30:47.032 NotebookApp] Writing notebook server cookie secret to /home/pamela/.local/share/jupyter/runtime/notebook_cookie_secret
[1 16:30:48.770 NotebookApp] Serving notebooks from local directory: /home/pamela
[1 16:30:48.770 NotebookApp] The Jupyter Notebook is running at:
[1 16:30:48.770 NotebookApp] http://localhost:8888/?token=311ca0f958b14775a0552214fcf36216126403dd87adce6f
[1 16:30:48.770 NotebookApp] or http://127.0.0.1:8888/?token=311ca0f958b14775a0552214fcf36216126403dd87adce6f
[1 16:30:48.770 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:30:49.104 NotebookApp]

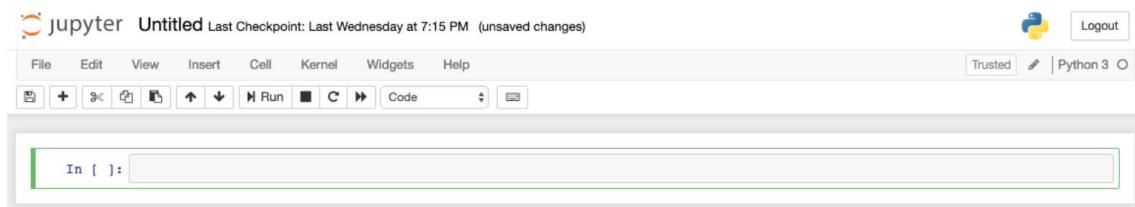
To access the notebook, open this file in a browser:
  file:///home/pamela/.local/share/jupyter/runtime/nbserver-7263-open.html
Or copy and paste one of these URLs:
  http://localhost:8888/?token=311ca0f958b14775a0552214fcf36216126403dd87adce6f
  or http://127.0.0.1:8888/?token=311ca0f958b14775a0552214fcf36216126403dd87adce6f
```

Para abrir un nuevo cuaderno de Jupyter bastará con presionar el botón new y seleccionar Python 3 para poder ejecutar el notebook con este lenguaje de programación. Es importante notar que Jupyter es compatible para utilizar otros lenguajes de programación como Julia.

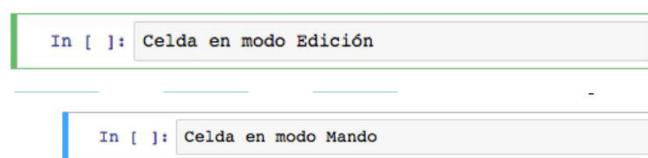


Para introducirnos un poco más en el mundo de *Jupyter Notebooks* identifiquemos los aspectos básicos de una notebook.

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



2. Existen dos tipos de celdas en *Jupyter Notebook*: 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*. Se pueden diferenciar por el color de las celdas, para aquellas enfocadas a código se observarán de color verde y aquellas enfocadas a texto se observarán de color azul.



3. Para agregar una celda es necesario dar clic en el siguiente botón.



jupyter Untitled Last Checkpoint: Last Wednesday at 7:15 PM (unsaved changes)

Logout



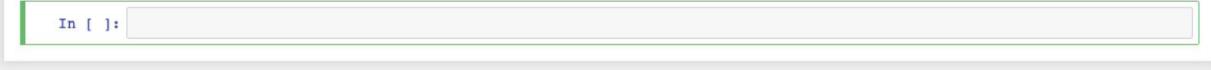
4. Debes indicar qué tipo de celda deseas utilizar en la selección, para obtener una celda para ejecutar código debes utilizar la opción *Code* y para obtener una celda de texto la opción *Markdown*.

jupyter Untitled Last Checkpoint: Last Wednesday at 7:15 PM (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3



Posteriormente bastará con escribir el texto que deseas redactar en caso de tener una celda para Markdown, en este tipo de celdas es posible redactar texto con *HTML*, *LATEX*, figuras, videos o texto normal.

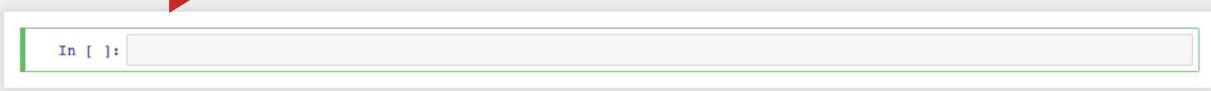
5. Al insertar una celda de código será posible ejecutar el código que desarrolles presionando el botón *Run* y obtendrás los resultados de la siguiente manera:

jupyter Untitled Last Checkpoint: Last Wednesday at 7:15 PM (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3





jupyter Untitled Last Checkpoint: hace un minuto (unsaved changes)

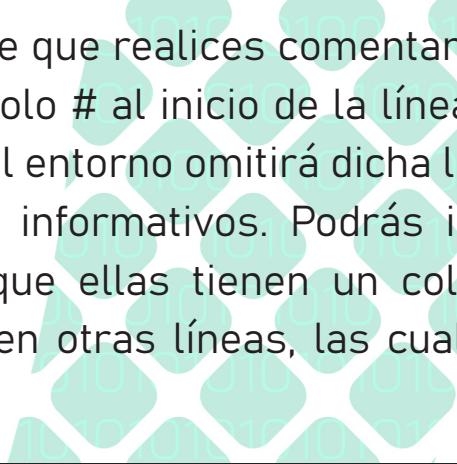
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

In [1]: `print("Hola mundo")`

Hola mundo

In []:

6. 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 porque ellas tienen un color verde, a diferencia de los colores en otras líneas, las cuales serán ejecutadas.



Actividades Navegador web Firefox 15 de mar 4:48 PM en Logout

Home Page - Select or create Untitled - Jupyter Notebook +

localhost:8888/notebooks/Untitled.ipynb?kernel_name=python3

150% ... ☰

jupyter Untitled Last Checkpoint: hace 4 minutos (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

In [1]: `print("Hola mundo")`

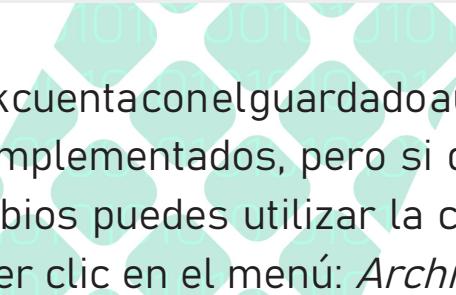
Hola mundo

In [2]: `#Esta linea es un comentario
print("Esta linea no es un comentario")`

Esta linea no es un comentario

In []:

7. Jupyter Notebook 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.



File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 Logout

New Notebook Open... ↑ ↓ Run Cell Widgets Help

Make a Copy... Save as... Rename... Save and Checkpoint Revert to Checkpoint Print Preview Download as Trusted Notebook Close and Halt

`int("Hola mundo")`

la mundo

`sta linea es un comentario
int("Esta linea no es un comentario")`

ta linea no es un comentario

¿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*.

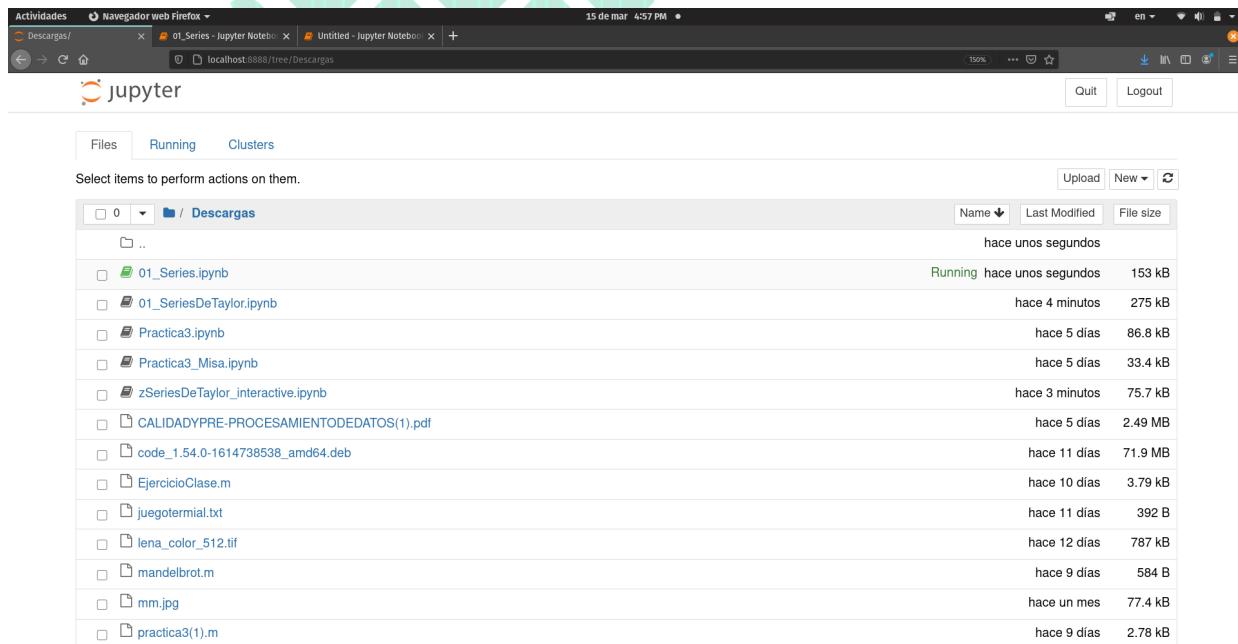
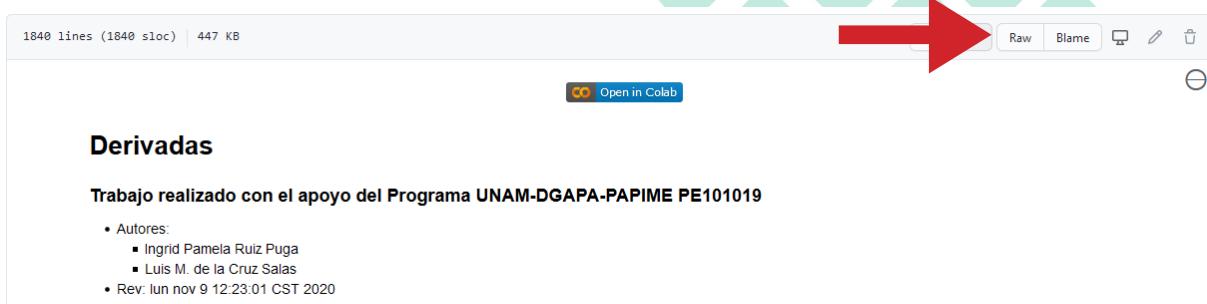


Figura 1. Repositorio Prometeo con los materiales didácticos del proyecto Macti.

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

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



1840 lines (1840 sloc) | 447 KB

Raw Blame

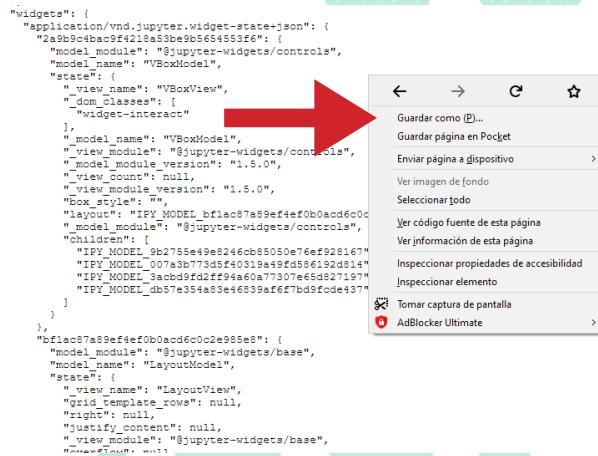
Open in Colab

Derivadas

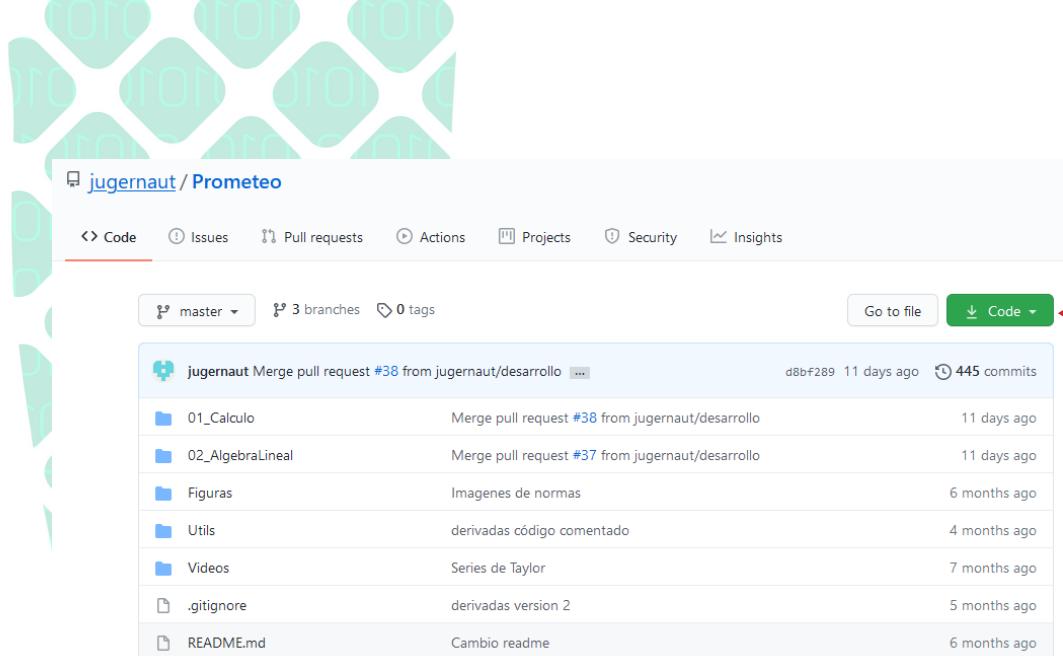
Trabajo realizado con el apoyo del Programa UNAM-DGAPA-PAPIME PE101019

- Autores:
 - Ingrid Pamela Ruiz Puga
 - Luis M. de la Cruz Salas
- Rev. lun nov 9 12:23:01 CST 2020

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 obtener el material del proyecto MACTI es realizando la descarga completa de todos los materiales. Se deberá ingresar al repositorio de "Prometeo" y elegir las opciones de **code** para posteriormente descargar el repositorio en una carpeta ZIP.



¿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 podrá visualizar una "vista previa" del notebook. En esta vista no podrás interactuar con el código, sin embargo una vez descargados los archivos que deseas revisar bastará con seleccionar al inicio de Jupyter la carpeta donde guardaste los archivos para poder abrirlos. Se abrirá en una nueva pestaña de tu navegador el archivo que visualizaras.

Actividades Navegador web Firefox • 15 de mar 4:57 PM •

Descargas/ 01_Series - Jupyter Notebook Untitled - Jupyter Notebook + localhost:8888/free/Descargas

jupyter

Files Running Clusters

Select items to perform actions on them.

Upload New

| | Name | Last Modified | File size |
|--|---------|--------------------|-----------|
| <input type="checkbox"/> 0 | .. | | |
| <input type="checkbox"/> 01_Series.ipynb | Running | hace unos segundos | 153 kB |
| <input type="checkbox"/> 01_SeriesDeTaylor.ipynb | | hace 4 minutos | 275 kB |
| <input type="checkbox"/> Practica3.ipynb | | hace 5 días | 86.8 kB |
| <input type="checkbox"/> Practica3_Misa.ipynb | | hace 5 días | 33.4 kB |
| <input type="checkbox"/> zSeriesDeTaylor_interactive.ipynb | | hace 3 minutos | 75.7 kB |
| <input type="checkbox"/> CALIDADYPRE-PROCESAMIENTODEDATOS(1).pdf | | hace 5 días | 2.49 MB |
| <input type="checkbox"/> code_1.54.0-1614738538_amd64.deb | | hace 11 días | 71.9 kB |
| <input type="checkbox"/> EjercicioClase.m | | hace 10 días | 3.79 kB |
| <input type="checkbox"/> juegotermial.txt | | hace 11 días | 392 B |
| <input type="checkbox"/> lena_color_512.tif | | hace 12 días | 787 kB |
| <input type="checkbox"/> mandelbrot.m | | hace 9 días | 584 B |
| <input type="checkbox"/> mm.jpg | | hace un mes | 77.4 kB |
| <input type="checkbox"/> practica3(1).m | | hace 9 días | 2.78 kB |

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

Debes ubicarte en la celda que deseas modificar y debes localizar el siguiente botón que dice Run, el cual se encuentra en la parte superior del navegador.

Actividades Navegador web Firefox • 15 de mar 5:10 PM •

Descargas/ 01_Series - Jupyter Notebook + localhost:8888/notebooks/Descargas/01_Series.ipynb

jupyter 01_Series Last Checkpoint: hace 14 minutos (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3

Run Code

$$\sum_{n=0}^N b_n - b_{n+1} = b_0 - b_{N+1}$$

Por ejemplo: $\sum_{k=1}^8 \left(\frac{1}{k} - \frac{1}{k+1} \right)$

Al realizar las sumas parciales de la serie anterior es posible observar que el resultado de las sumas parciales anteriores es igual a $\frac{1}{1} - \frac{1}{9} = 0.888\dots$

```
In [2]: # Calculo de la serie telescopica
def seriatelescopica(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

seriatelescopica(1,8)
Out[2]: 0.8888888888888888
```

Dado que la serie del ejemplo 3 es una serie telescopica, usando la fórmula obtenemos que

$$\sum_{k=1}^N \left(\frac{1}{k} - \frac{1}{k+1} \right) = 1 - \frac{1}{N+1},$$

y más aún, convergería a 1 para N muy grande. Esto último se puede comprobar calculando el límite cuando N tiende a ∞ de la última expresión.

Primero observaremos la gráfica de la serie para posteriormente calcular el límite.

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

```
In [2]: # 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.

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.

```
In [2]: # 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)
```

Out[2]: 0.888888888888888

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:

```
In [2]: # 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)
```

A la cual se le desea realizar una modificación para calcular una serie telescopica 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:

```
In [2]: # 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(1,15)
```

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

```
In [3]: # 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(1,15)
```

Out[3]: 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 telescopica; posteriormente, se tiene una segunda celda para graficar la serie telescopica 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.

A screenshot of a Firefox browser window showing a Jupyter Notebook interface. The title bar says "jupyter 01_Series Last Checkpoint: hace 22 minutos (unsaved changes)". The notebook has two code cells:

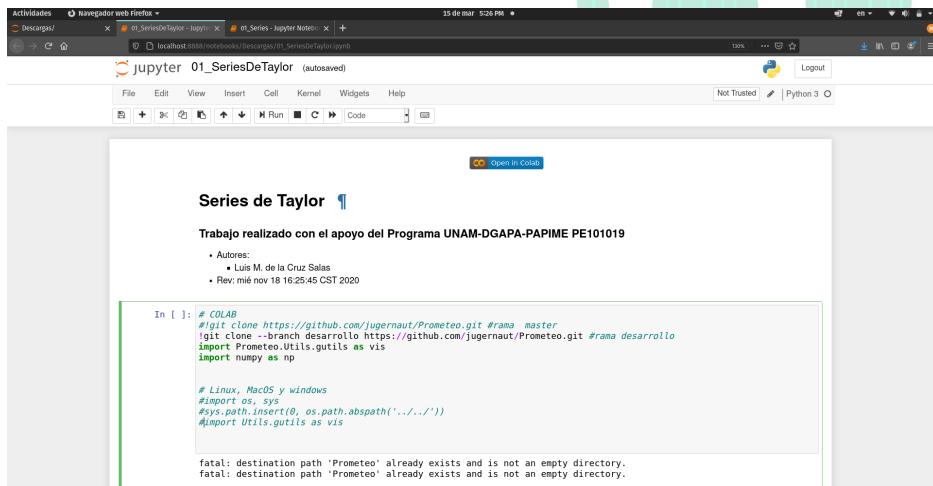
```
In [4]: def seriatelescopiacompleta(inicio,final):
    lista = []
    r = 0
    for x in range(inicio, final+1):
        r += (1/x)-(1/(x+1))
        lista.append(r)
    return lista

In [5]: x4 = np.arange(1, 9, 1)
y4 = seriatelescopiacompleta(1,8)
par = [{ 'title': 'Serie telescopica',
          ' xlabel': 'sns',
          ' ylabel': '$\$\$',
          ' ylim': (0,1)}
        ]
graf = vis.planoCartesiano(par=par)
graf.plot(x=[1,8],y=[seriatelescopica(1,8),seriatelescopica(1,8)],
          par={ 'c': 'green', 'linestyle': '--', 'lw':2, 'label':r'Función constante = 0.88...'})
graf.scatter(x=x4, y=y4,
            par={ 'label':r'Serie: $\sum_{k=1}^N \left( \frac{1}{k} - \frac{1}{k+1} \right)$', 'lw':1, 'color': 'red',
                  'loc': 'lower right'})
graf.show()
```

¿Cómo cargar códigos dependientes en Jupyter Notebook?

Se presenta el método para utilizar notebooks dependientes de otros archivos o recursos. Una vez descargado todo el repositorio del proyecto bastará con realizar el siguiente cambio en la celda del notebook dependiente. Es importante considerar que los notebook utilizan scripts de clases realizadas en el proyecto por lo que se enfatiza en realizar la descarga del repositorio completo.

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



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:

```
In [ ]: # COLAB
#git clone https://github.com/jugernaut/Prometeo.git #rama master
#git clone --branch desarrollo https://github.com/jugernaut/Prometeo.git #rama desarrollo
#import Prometeo.Utils.gutils as vis
#import numpy as np

# Linux, MacOS y windows
import os, sys
sys.path.insert(0, os.path.abspath('../..'))
import Utils.gutils as vis
```

Asegurandote de tener comentada la sección del código que corresponde al Google Colab, así como de colocar la ruta donde se encuentran los archivos descargados, de esta manera el notebook podrá acceder a los archivos que necesitas, 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:

Herramienta interactiva

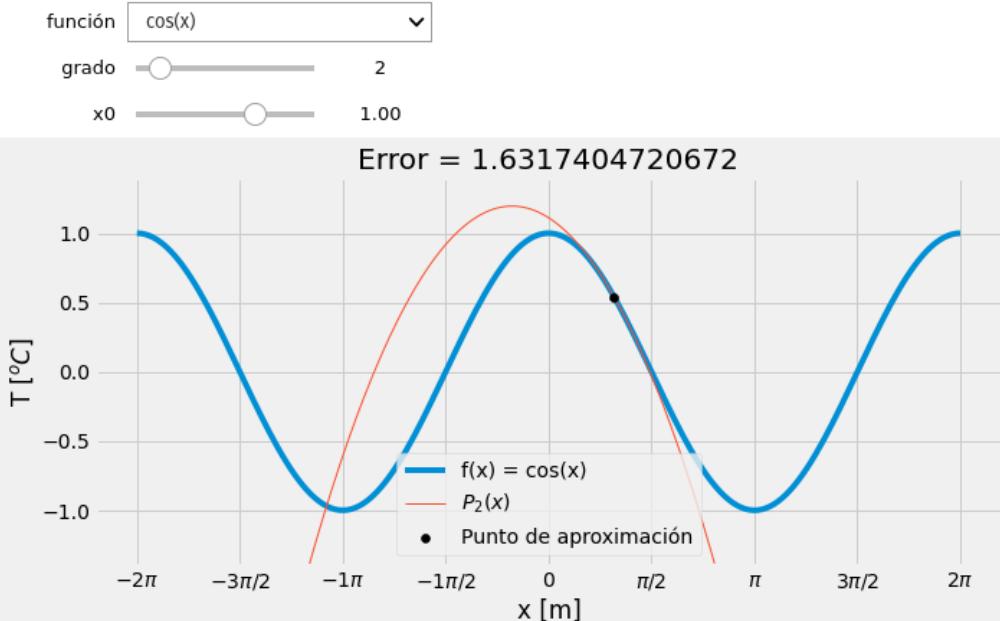
La siguiente herramienta permite evaluar de manera interactiva diferentes series de Taylor para algunas funciones. Puede elegir la función, el grado del polinomio de aproximación y el lugar donde se desea aproximar y observe el resultado.

```
In [ ]: # Linux, MacOS y Windows
#%run "./zSeriesDeTaylor_interactive.ipynb"

# Google Coolab
#%run "/content/Prometeo/01_Calculo/07_Series_de_Taylor/zSeriesDeTaylor_interactive.ipynb"
```

Como se dice en las instrucciones de esta herramienta, para poder ejecutarla quitaremos el comentario de la línea 2 y ejecutaremos:

```
3 # Linux, MacOS y Windows
4 %run "./zSeriesDeTaylor_interactive.ipynb"
5
6 # Google Coolab
7 #%run "/content/Prometeo/01_Calculo/07_Series_de_Taylor/zSeriesDeTaylor_interactive.ipynb"
```



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