

Manual *macti_lib* y *NBGrader*

...

MeIA 2023

Descripción general

En este manual se dará una breve introducción al uso de *macti_lib* para la evaluación de celdas de código dentro de un *Jupyter Notebook*.

Y de igual manera se mostrará cómo vincular el uso de *macti_lib* con *NBGrader*, esta última herramienta ayuda en el proceso de la evaluación del conocimiento adquirido mediante *Jupyter Notebook*.

macti_lib :

Biblioteca desarrollada bajo el proyecto MACTI, que permite al usuario automatizar la evaluación.

¿Cómo usar *macti_lib*?

Estructura adecuada

Cada notebook contiene un número de preguntas, por lo tanto es necesario tener una carpeta en el cuál se almacenen las respuestas.

Jupyter Notebook

Para usar la biblioteca *macti_lib* lo primero que necesitamos es un *Jupyter Notebook* que desarrolle algún tema.

Este *notebook* debe contener al menos una celda de código.

Celdas de evaluación

Una vez que se tienen los 2 elementos anteriores, lo siguiente es modificar el notebook inicial, de tal manera que ahora contenga celdas que evalúen el conocimiento adquirido.

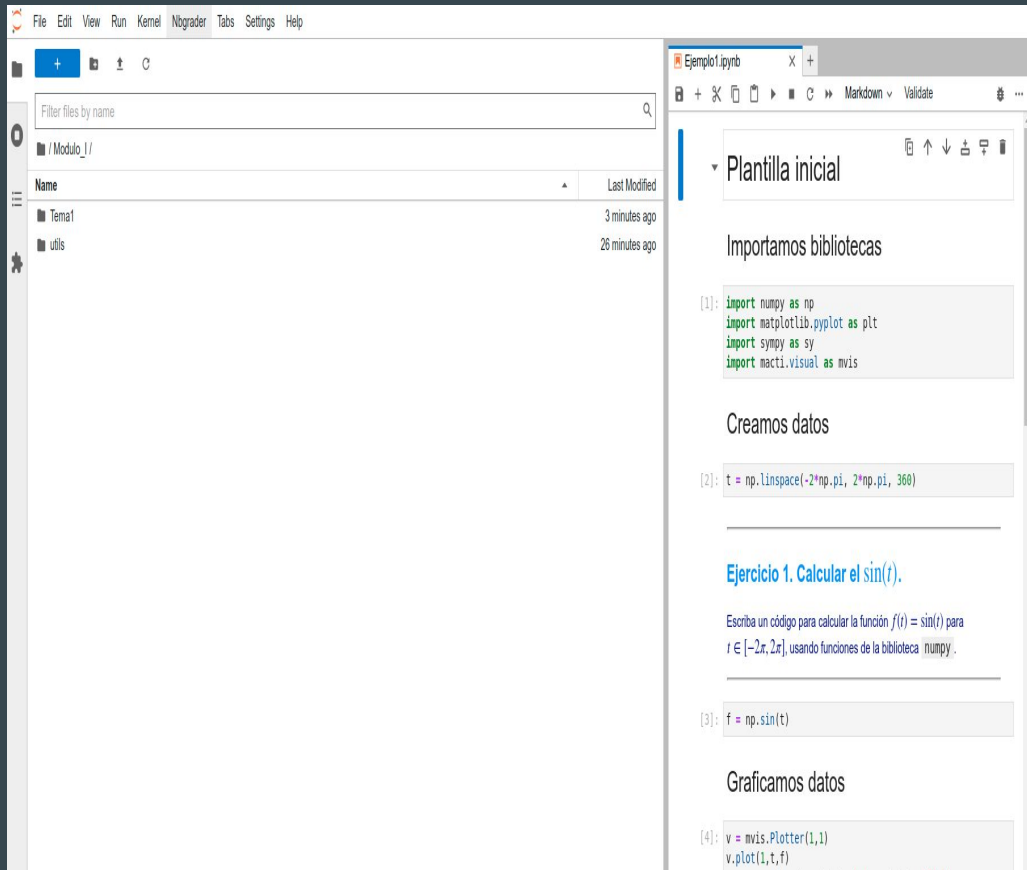
Estructura

Carpetas y subcarpetas

Se sugiere que las carpetas donde se almacenen los *notebooks* tengan la siguiente estructura.

Estructura:

- ***Nombre_Curso_Iniciales***: carpeta que contiene todos los temas del curso
- ***Tema***: carpeta que contiene notebooks por tema.
- ***utils***: contenido varios.



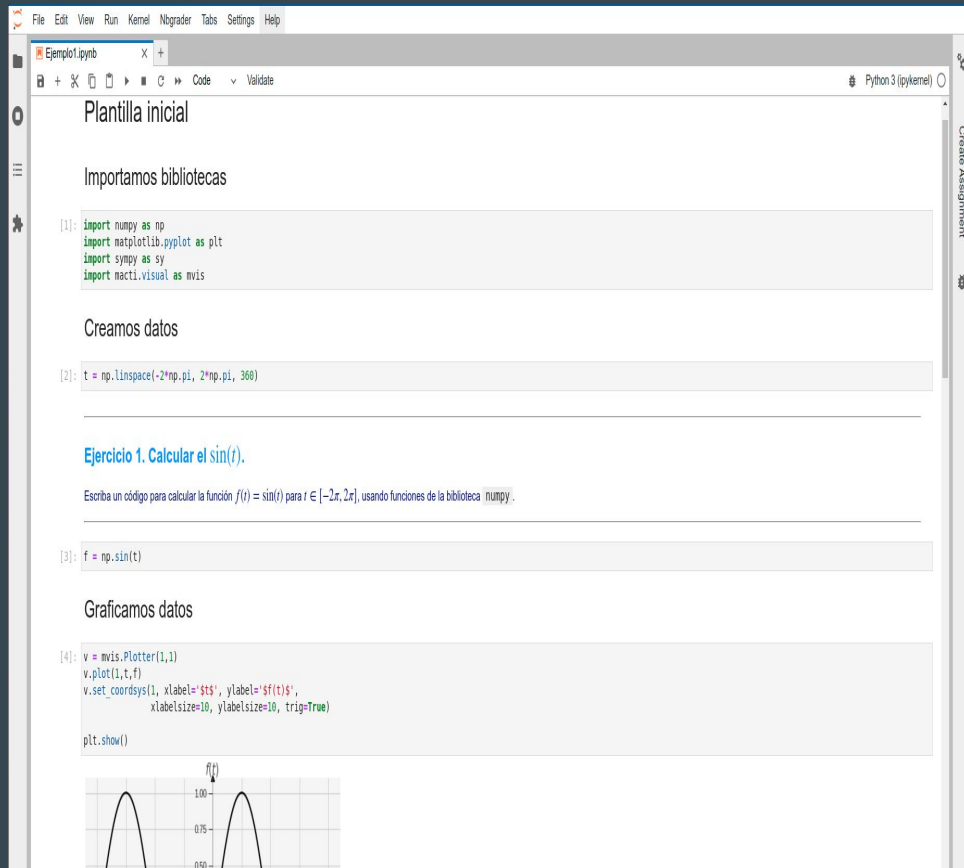
Jupyter Notebook

Ejemplo01.ipynb

Supongamos que ya se tiene un notebook que cumple con los requisitos para el uso de *macti_lib*.

Requisitos:

- celdas de código.
- conocimiento a evaluar.



Incorporación de *macti_lib*

Tomando cómo base el notebook *Ejemplo01.ipynb* se crea un nuevo *notebook* llamado *Ejemplo02.ipynb*, y el siguiente paso consiste en generar el archivo de respuestas que será usado posteriormente para la evaluación del conocimiento.

Para tal propósito es necesario importar la biblioteca *macti_lib* en el encabezado del notebook y es necesario indicar un par de parámetros, de la siguiente forma.

Importamos bibliotecas

```
import numpy as np
import matplotlib.pyplot as plt
import sympy as sy

import macti.visual as mvis
from macti.evaluation import FileAnswer
file_answer = FileAnswer('Modulo_I_MAPL', 'Tema1')
```

Parámetros *macti_lib*

Modulo_I_MAPL

Este parámetro hace referencia al nombre del curso al que pertenece este *notebook*.

Es muy importante por que en esta carpeta se ubica *utils* y ahí se almacenan las respuestas.

Tema1

Hace referencia al tema que será evaluado usando *macti_lib*.

Es importante notar que por cada tema puede haber más de un *notebook* a ser evaluado.

FileAnswer

Estos datos son usados por la clase *FileAnswer* para generar el archivo de respuestas de este *notebook*.

file_answer

El objeto *file_answer* generado al momento de importar *macti_lib* nos permite almacenar las respuestas en la ruta adecuada.

Supongamos que se le pide al alumno graficar la función $\sin(t)$ entre -2π y 2π , tal cómo se muestra en la siguiente imagen.

Creamos datos

```
[ ]: t = np.linspace(-2 * np.pi, 2 * np.pi, 360)
```

Ejercicio 1. Calcular el $\sin(t)$.

Escriba un código para calcular la función $f(t) = \sin(t)$ para $t \in [-2\pi, 2\pi]$, usando funciones de la biblioteca `numpy`.

```
[ ]: f = np.sin(t)
```

```
[ ]: file_answer.write('1', f, 'Revisa el uso correcto de la función np.sin()')
```

file_answer

En la sección “Creamos datos” se generan los elementos necesarios para que el alumno conteste y para graficar en caso de ser necesario.

A priori el profesor ya conoce la respuesta por lo que basta generarla con la llamada a la función $f = np.sin(t)$. Finalmente se le indica a *file_answer* **cuál es la respuesta correcta y un *hint* en caso de que el alumno no responda correctamente.**

Creamos datos

```
[ ]: t = np.linspace(-2 * np.pi, 2 * np.pi, 360)
```

Ejercicio 1. Calcular el $\sin(t)$.

Escriba un código para calcular la función $f(t) = \sin(t)$ para $t \in [-2\pi, 2\pi]$, usando funciones de la biblioteca `numpy`.

```
[ ]: f = np.sin(t)
```

```
[ ]: file_answer.write('1', f, 'Revisa el uso correcto de la función np.sin()')
```

¿Qué tipo de datos soporta *macti_lib*?

Datos Numéricos

Todo aquel valor numérico (tipos primitivos de python, números, booleanos, complejos, etc.).

Así como listas, vectores, matrices.

Cálculo Simbólico

Toda aquella respuesta que requiera hacer uso de la biblioteca *SymPy* está soportada por *macti_lib*.

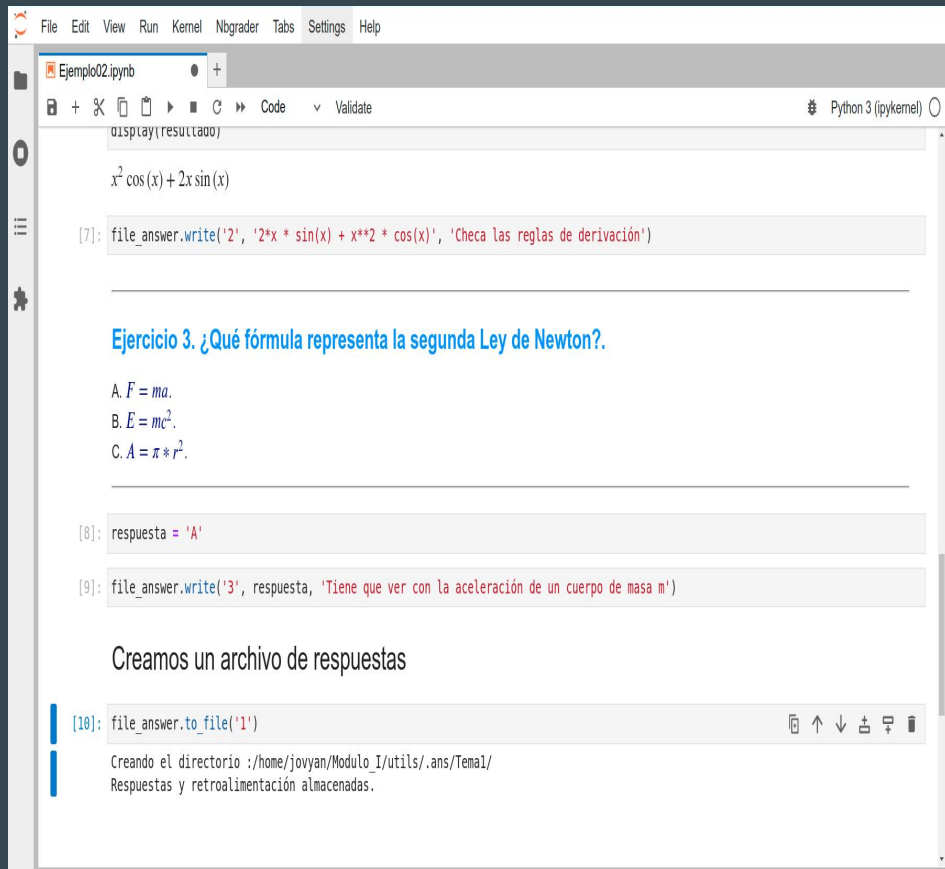
Caracteres

En caso de que la pregunta sea de opción múltiple, *macti_lib* acepta todo tipo de caracteres alfanuméricos para validar la respuesta del alumno.

Jupyter con respuestas

Ejemplo02.ipynb

Una vez que se generó el *notebook* con las respuestas y que cada una de estas respuestas se almacenó de manera temporal, es necesario “vaciar” las respuestas en un archivo dentro de la sesión de *JupyterLab*.



The screenshot shows a JupyterLab notebook titled 'Ejemplo02.ipynb'. The interface includes a menu bar (File, Edit, View, Run, Kernel, Nbgrader, Tabs, Settings, Help) and a toolbar with icons for file operations and execution. The notebook content is as follows:

```
display(resultado)
```

$$x^2 \cos(x) + 2x \sin(x)$$

```
[7]: file_answer.write('2', '2*x * sin(x) + x**2 * cos(x)', 'Checa las reglas de derivación')
```

Ejercicio 3. ¿Qué fórmula representa la segunda Ley de Newton?

A. $F = ma$.
B. $E = mc^2$.
C. $A = \pi * r^2$.

```
[8]: respuesta = 'A'
```

```
[9]: file_answer.write('3', respuesta, 'Tiene que ver con la aceleración de un cuerpo de masa m')
```

Creamos un archivo de respuestas

```
[10]: file_answer.to_file('1')
```

Creando el directorio :/home/jovyan/Modulo_I/Utils/.ans/Tema1/
Respuestas y retroalimentación almacenadas.

file_answer

Es muy importante notar la llamada a la función *file_answer.to_file(1)*.

Está llamada lo que hace es almacenar las respuestas correspondientes al quizz 1 del Tema 1 en la ruta */user/local/share/nbgrader/exchange/Modulo_I/utils/.ans/Tema1/*

Este archivo de respuestas está **oculto** y además en una **ubicación compartida** dentro del servidor de *MACTI*.

Creamos un archivo de respuestas

```
file_answer.to_file('1')
```



El directorio `:/usr/local/share/nbgrader/exchange/Modulo_I_MAPL/utils/.ans/Tema1/` ya existe
Respuestas y retroalimentación almacenadas.

NBGrader:

biblioteca cuya finalidad es automatizar en la medida de lo posible el proceso de evaluación y retroalimentación.

macti_lib y *NBGrader*

Vale la pena mencionar que a pesar de que *macti_lib* y *NBGrader* son completamente compatibles y su potencial radica en la combinación de ambas, no es obligatorio hacer uso de ambas.

macti_lib

Biblioteca desarrollada bajo el proyecto MACTI que nos ayuda a almacenar las respuestas y evaluar de manera automática las preguntas que se le muestren a los alumnos.

NBGrader

Framework que integra a *JupyterLab* elementos tanto gráficos como técnicos que nos permiten automatizar el proceso de enseñanza-aprendizaje.

Uso de la clase *Quizz*

Tomando cómo base el notebook *Ejemplo02.ipynb* se crea un nuevo *notebook* llamado *Ejemplo03.ipynb* y el siguiente paso consiste en invocar el archivo de respuestas creado previamente para validar las respuestas del alumno.

Los parámetros de esta clase indican: número de *quizz*, nombre del curso, y tema. Estos parámetros son usados para acceder al archivo de respuestas creado previamente.

Importamos bibliotecas

```
import numpy as np
import matplotlib.pyplot as plt
import sympy as sy

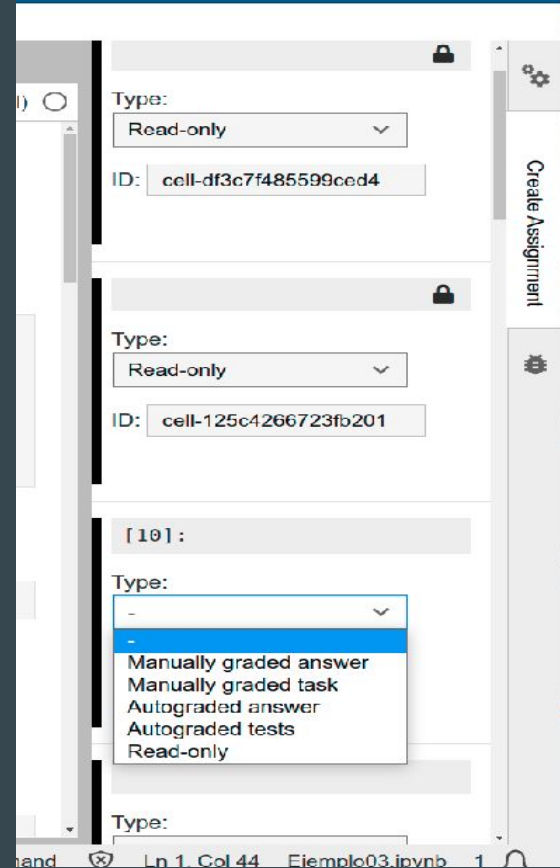
import macti.visual as mvis
from macti.evaluation import Quizz
quizz = Quizz('1', 'Modulo_I_MAPL', 'Tema1')
```



GUI NBGrader

Elementos gráficos

Del lado derecho de la pantalla de *JupyterLab* podemos notar un botón que dice “*Create Assignment*” y al presionarlo veremos que se despliega un listado de todas las celdas que compone al notebook y por cada celda podemos elegir varias opciones que se detallan a continuación.



Tipos de modificador de celda *NBGrader*

read only

Toda celda que tenga este modificador asociado no podrá ser modificada por el usuario final, una vez que haya sido procesada con *NBGrader*.

Autograded answer

Uso obligatorio con celdas de código que puede ser editada y ejecutada por el alumno.

Autograded test

Celda de código que sólo puede ser ejecutada y que contiene la prueba automatizada que debe pasar el alumno.

Manually graded answer

Celda que puede ser editada por el alumno y que contiene una respuesta que requiere evaluación manual. Puede ser de texto o de código.

Manually graded task

Celda que contiene la evaluación de la celda previa.

Uso del objeto *quizz* y *NBGrader*

En esta imagen podemos ver que las “porciones” de código que requieran ser eliminadas para que el alumno las complete (una vez procesado el *notebook*) deben estar entre los comentarios `### BEGIN SOLUTION` y `### END SOLUTION`.

Y por otro lado la celda que realiza la autoevaluación hace uso del objeto *quizz* y mediante la función `eval_numeric('l', f)` se indica que se requiere hacer uso de la respuesta `l` (almacenada previamente) y debe ser contrastada contra la respuesta del alumno `f`.

The screenshot displays a Jupyter Notebook interface with two main panels. The left panel shows the notebook content, and the right panel shows the NBGrader evaluation interface.

Left Panel (Notebook Content):

- Ejercicio 1. Calcular el $\sin(t)$.**
- Instruction: "Escriba un código para calcular la función $f(t) = \sin(t)$ para $t \in [-2\pi, 2\pi]$, usando funciones de la biblioteca `numpy`."
- Cell [12]:

```
### BEGIN SOLUTION
f = np.sin(t)
### END SOLUTION
```
- Cell [13]:

```
quizz.eval_numeric('l', f)
```
- Output of Cell [13]:

```
Tu resultado es correcto.
```
- Text below the output: "Graficamos datos"

Right Panel (NBGrader Evaluation Interface):

- Top Section:** Type: Read-only, ID: cell-fedb644628cb60cd
- Bottom Section:** Type: Autograded answer, ID: cell-b40331105712505e

Jupyter autoevaluado

Ejemplo03.ipynb

En esta *notebook* podemos ver varios elementos:

- procesamiento con *NBGrader*.
- evaluación mediante objeto *quizz*.
- tipos de celdas.

The screenshot displays a Jupyter Notebook titled 'Ejemplo03.ipynb' running on Python 3 (ipykernel). The notebook contains several code cells and a quiz evaluation section.

Code Cell [11]:

```
Creemos datos
```

```
[11]: t = np.linspace(-2 * np.pi, 2 * np.pi, 360)
```

Ejercicio 1. Calcular el $\sin(t)$.

Escriba un código para calcular la función $f(t) = \sin(t)$ para $t \in [-2\pi, 2\pi]$, usando funciones de la biblioteca `numpy`.

Code Cell [12]:

```
## BEGIN SOLUTION
f = np.sin(t)
## END SOLUTION
```

Code Cell [13]:

```
quizz.eval_numeric('1', f)
```

Tu resultado es correcto.

Code Cell [14]:

```
v = mvis.Plotter(1,1)
v.plot(1,t,f)
v.set_coordsys(1, xlabel='t', ylabel='f(t)',
               xlabelsize=10, ylabelsize=10, trig=True)
plt.show()
```

The plot shows a sine wave $f(t)$ versus t . The x-axis is labeled t and the y-axis is labeled $f(t)$. The plot is titled $f(t)$.

Quiz Evaluation:

The right sidebar shows the quiz evaluation results for three questions:

- [11]:** Type: Read-only, ID: cell-bf8ed569808fc26f
- [12]:** Type: Autograded answer, ID: cell-b40331105712505e
- [13]:** Type: Autograded tests, ID: cell-fca96a624845069, Points: 2

Creamos datos

```
[11]: t = np.linspace(-2 * np.pi, 2 * np.pi, 360)
```

Ejercicio 1. Calcular el $\sin(t)$.

Escriba un código para calcular la función $f(t) = \sin(t)$ para $t \in [-2\pi, 2\pi]$, usando funciones de la biblioteca `numpy`.

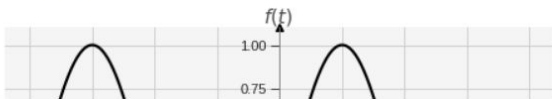
```
[12]: ## BEGIN SOLUTION  
f = np.sin(t)  
## END SOLUTION
```

```
[13]: quizz.eval_numeric('1', f)
```

Tu resultado es correcto.

Graficamos datos

```
[14]: v = mvis.Plotter(1,1)  
v.plot(1,t,f)  
v.set_coordsys(1, xlabel='$t$', ylabel='$f(t)$',  
              xlabelsize=10, ylabelsize=10, trig=True)  
plt.show()
```



[11]:

Type: Read-only

ID: cell-bf8ed569808fc26f

[12]:

Type: Autograded answer

ID: cell-b40331105712505e

[13]:

Type: Autograded tests

ID: cell-fca96a624845069

Points: 2

FileEditViewRunKernelNbgraderTabsSettingsHelp

Ejemplo03.ipynb

Python 3 (ipykernel)

Ejercicio 2. Derivada de una función.

Calcule la derivada de la función $f(x) = x^2 \sin(x)$. Escriba su respuesta en formato `sympy`.

```
[16]: x = sy.Symbol('x')
```

```
[17]: ### BEGIN SOLUTION
      resultado = 2*x * sy.sin(x) + x*x * sy.cos(x)
      ### END SOLUTION
      display(resultado)
```

$x^2 \cos(x) + 2x \sin(x)$

```
[18]: quizz.eval_expression('2', resultado)
```

Tu respuesta:

$x^2 \cos(x) + 2x \sin(x)$

es correcta.

Ejercicio 3. ¿Qué fórmula representa la segunda Ley de Newton?

A. $F = ma$.
B. $E = mc^2$.
C. $A = \pi * r^2$.

```
[19]: quizz.eval_option('3', 'A')
```

Tu respuesta: A, es correcta.

ID: cell-1815c2d9ff664d90

[17]:

Type: Autograded answer

ID: cell-40dd497407e75cff

[18]:

Type: Autograded tests

ID: cell-449fbef003565962

Points: 0

Type: Read-only

ID: cell-fd2d6fda38ea2fd3

[19]:

Type: Autograded tests

ID: cell-ceec9aaa44e1a88d

Points: 0

Create Assignment

Respuesta correcta

En caso de que el alumno proporcione una respuesta correcta, la biblioteca *macti_lib* se encarga de indicar que el resultado es correcto y muestra un mensaje resaltando que esa respuesta es correcta.

Y *NBGrader* (en caso de que el notebook ya haya sido procesado) se encarga de asignar la cantidad de puntos que corresponden a la calificación de esta pregunta en particular.

Ejercicio 2. Derivada de una función.

Calcule la derivada de la función $f(x) = x^2 \sin(x)$. Escriba su respuesta en formato `sympy`.

```
[16]: x = sy.Symbol('x')

[17]: ### BEGIN SOLUTION
      resultado = 2*x * sy.sin(x) + x*x * sy.cos(x)
      ### END SOLUTION

      display(resultado)

      x2 cos (x) + 2x sin (x)

[18]: quizz.eval_expression('2', resultado)
```

Tu respuesta:
 $x^2 \cos(x) + 2x \sin(x)$
es correcta.

[17]:

Type: Autograded answer

ID: cell-40dd497407e75cff

[18]:

Type: Autograded tests

ID: cell-449fbef003565962

Points: 4

Type: Read-only

ID: cell-fd2d6fda38ea2fd3

Respuesta incorrecta

En caso de que el alumno proporcione una respuesta incorrecta, la biblioteca *macti_lib* se encarga de indicar que el resultado es incorrecto y proporciona una sugerencia, tomada del archivo de retroalimentación que se creó al momento de almacenar las respuestas.

NBGrader (en caso de que el notebook ya haya sido procesado) se encarga de asignar 0 puntos a la calificación de esta pregunta en particular.

The screenshot displays a Jupyter Notebook window titled 'Ejemplo03.ipynb'. The notebook content includes a question in Spanish: 'Ejercicio 3. ¿Qué fórmula representa la segunda Ley de Newton?.' with three multiple-choice options: A. $F = ma$, B. $E = mc^2$, and C. $A = \pi * r^2$. Below the options, a code cell [20:] contains the command `quizz.eval_option('3', 'n')`. The output of this cell shows a message: 'Tu respuesta: n, es INCORRECTA.' followed by a hint: 'Hint: Tiene que ver con la aceleración de un cuerpo de masa m'. Below the hint, a red traceback box displays an `AssertionError` message, indicating that the answer 'n' is incorrect. The traceback points to the `quizz.eval_option` method in the `macti/evaluation.py` file. To the right of the notebook, a sidebar shows the 'Autograded answer' and 'Autograded tests' sections, both with a score of 0 points.

Ejemplo03.ipynb

Python 3 (ipykernel)

Ejercicio 3. ¿Qué fórmula representa la segunda Ley de Newton?.

A. $F = ma$.

B. $E = mc^2$.

C. $A = \pi * r^2$.

[20]: `quizz.eval_option('3', 'n')`

Tu respuesta: n, es INCORRECTA.

Hint: Tiene que ver con la aceleración de un cuerpo de masa m

AssertionError

Traceback (most recent call last)

Cell In[20], line 1

----> 1 `quizz.eval_option('3', 'n')`

File ~/opt/conda/lib/python3.10/site-packages/macti/evaluation.py:124, in `Quizz.eval_option(self, enum, ans)`

121 print()

122 print(Fore.RESET + 80*'-')

--> 124 raise AssertionError

AssertionError:

ID: cell-1815c2d9ff664d90

[17]:

Type: Autograded answer

ID: cell-40dd497407e75cff

[18]:

Type: Autograded tests

ID: cell-449fbee003565962

Points: 0

Type: Read-only

ID: cell-fd2d6fda38ea2fd3

Procesamiento con *NBGrader*

Hasta este punto ya tenemos todo lo necesario para procesar nuestro material con *NBGrader*, sólo necesitamos lo siguiente.

Configuración *NBGrader*:

- Estructura.
- Archivos de procesamiento.

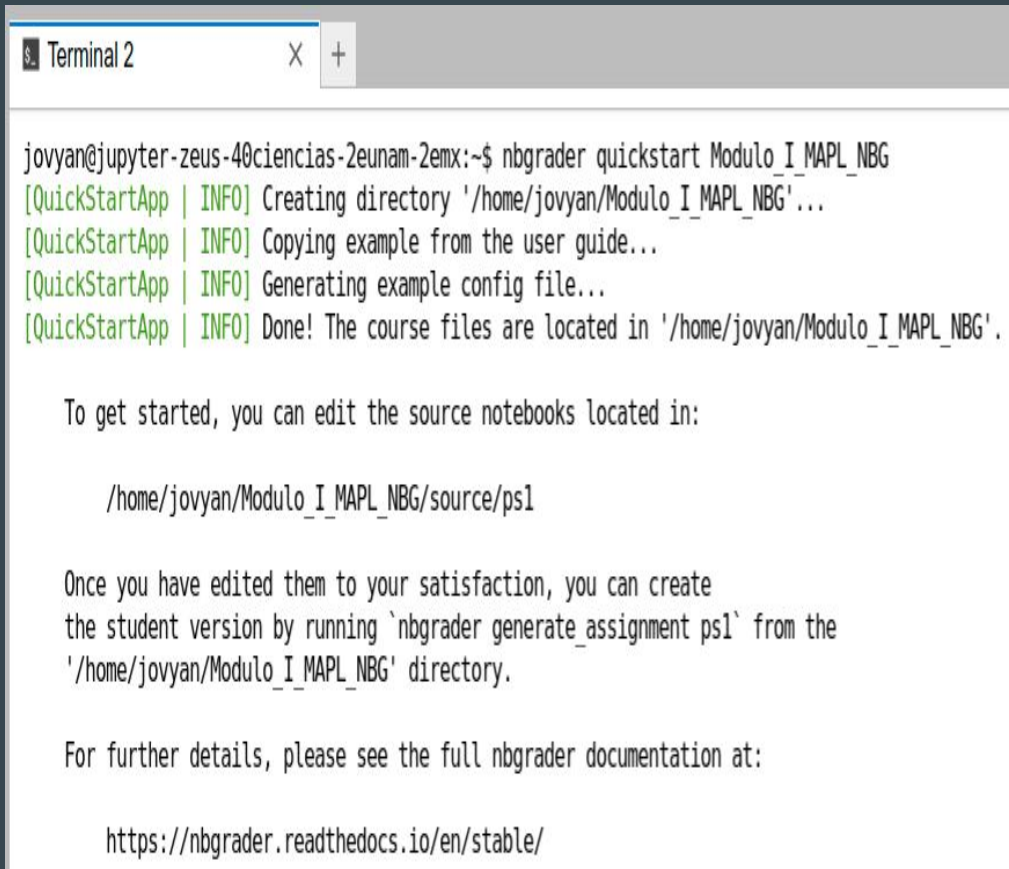
nbgrader quickstart

Inicializando

Desde una terminal y ubicados en la raíz de nuestra sesión en jupyterlab se ejecuta el comando:

nbgrader quickstart
Nombre_Curso_NGB

tal cómo se muestra en la imagen del lado derecho.

A terminal window titled "Terminal 2" with a close button (X) and a plus button (+). The terminal shows the execution of the command `nbgrader quickstart Modulo_I_MAPL_NBG`. The output consists of four lines of green text: `[QuickStartApp | INFO] Creating directory '/home/jovyan/Modulo_I_MAPL_NBG'...`, `[QuickStartApp | INFO] Copying example from the user guide...`, `[QuickStartApp | INFO] Generating example config file...`, and `[QuickStartApp | INFO] Done! The course files are located in '/home/jovyan/Modulo_I_MAPL_NBG'.`. Below the terminal output, there is instructional text: "To get started, you can edit the source notebooks located in:" followed by the path `/home/jovyan/Modulo_I_MAPL_NBG/source/ps1`. Then, it says: "Once you have edited them to your satisfaction, you can create the student version by running ``nbgrader generate_assignment ps1`` from the `'/home/jovyan/Modulo_I_MAPL_NBG'` directory." Finally, it says: "For further details, please see the full nbgrader documentation at:" followed by the URL `https://nbgrader.readthedocs.io/en/stable/`.

```
Terminal 2 X +
jovyan@jupyter-zeus-40ciencias-2eunam-2emx:~$ nbgrader quickstart Modulo_I_MAPL_NBG
[QuickStartApp | INFO] Creating directory '/home/jovyan/Modulo_I_MAPL_NBG'...
[QuickStartApp | INFO] Copying example from the user guide...
[QuickStartApp | INFO] Generating example config file...
[QuickStartApp | INFO] Done! The course files are located in '/home/jovyan/Modulo_I_MAPL_NBG'.

To get started, you can edit the source notebooks located in:

/home/jovyan/Modulo_I_MAPL_NBG/source/ps1

Once you have edited them to your satisfaction, you can create
the student version by running `nbgrader generate_assignment ps1` from the
'/home/jovyan/Modulo_I_MAPL_NBG' directory.

For further details, please see the full nbgrader documentation at:

https://nbgrader.readthedocs.io/en/stable/
```

quickstart

Al ejecutar este comando podemos ver que se generaron varios archivos pero los más importantes son:

source

Carpeta en la cuál se van a almacenar los *notebooks* que requieran ser evaluados por *nbgrader*.

nbgrader_config.py

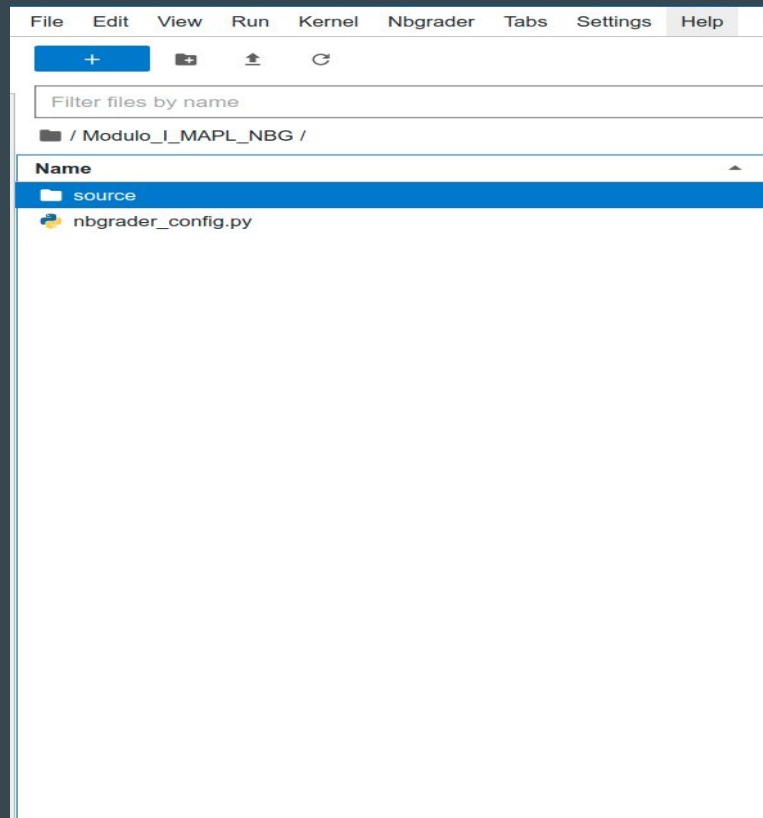
Archivo de configuración de *nbgrader*.

nbgrader_config.py

```
nbgrader_config.py  X  +

1  c = get_config()
2
3  #####
4  # Begin additions by nbgrader quickstart
5  #####
6
7  # You only need this if you are running nbgrader on a shared
8  # server set up.
9  c.CourseDirectory.course_id = "Modulo_I_MAPL_NBG"
10
11  c.IncludeHeaderFooter.header = "source/header.ipynb"
12
13  #####
14  # End additions by nbgrader quickstart
15  #####
16
17  # Configuration file for nbgrader-generate-config.
18
19  c = get_config()  #noqa
20
21  #-----
22  # Application(SingletonConfigurable) configuration
23  #-----
24  ## This is an application.
25
26  ## The date format used by logging formatters for %(asctime)s
27  # Default: '%Y-%m-%d %H:%M:%S'
28  # c.Application.log_datefmt = '%Y-%m-%d %H:%M:%S'
29
30  ## The Logging format template
31  # Default: '[(name)s]%(highlevel)s %(message)s'
32  # c.Application.log_format = '[(name)s]%(highlevel)s %(message)s'
33
34  ## Set the log level by value or name.
35  # Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITICAL']
36  # Default: 30
37  # c.Application.log_level = 30
38
39  ## Configure additional log handlers.
40  #
41  # The default stderr log handler is configured by the log_level, log_datefmt
```

estructura



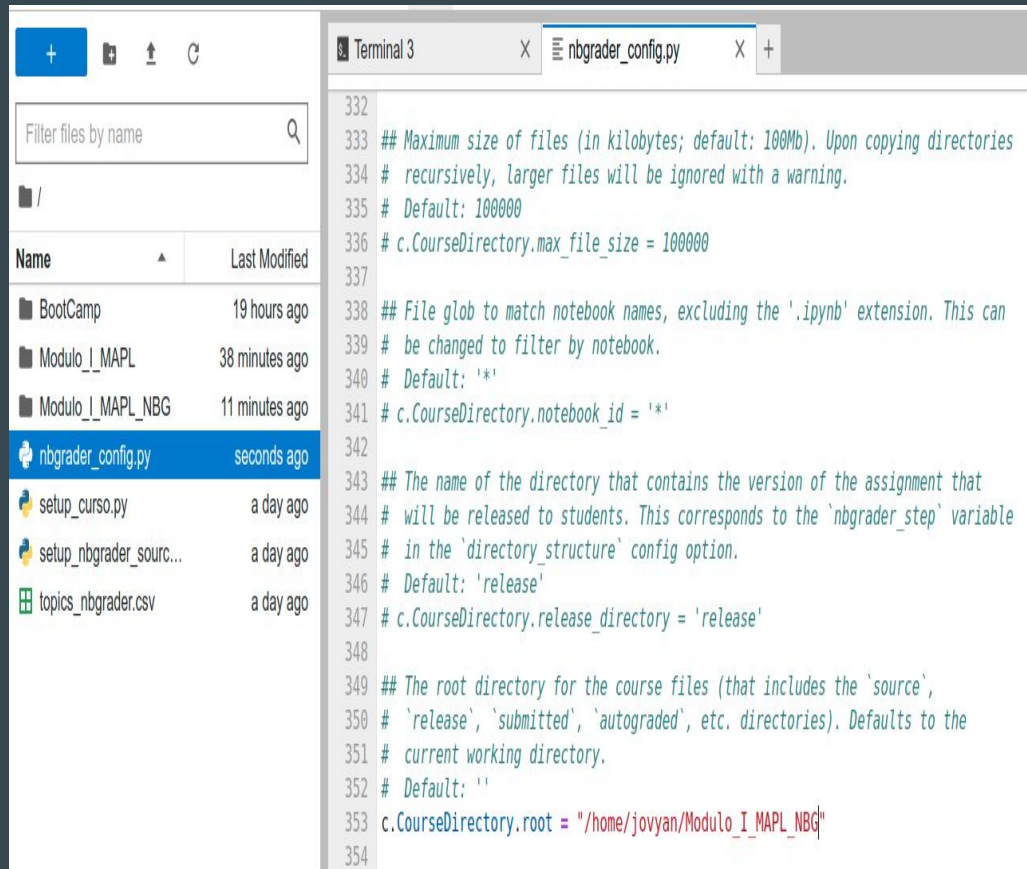
nbgrader_config.py

Copiando

Es necesario copiar y pegar el archivo `nbgrader_config.py` en la raíz de nuestra sesión y cambiar el contenido de la línea 353 con los datos.

```
c.CourseDirectory.root =  
"/home/jovyan/Modulo_I_MAPL_NBG"
```

Para que ahora se vea cómo se muestra en la imagen.

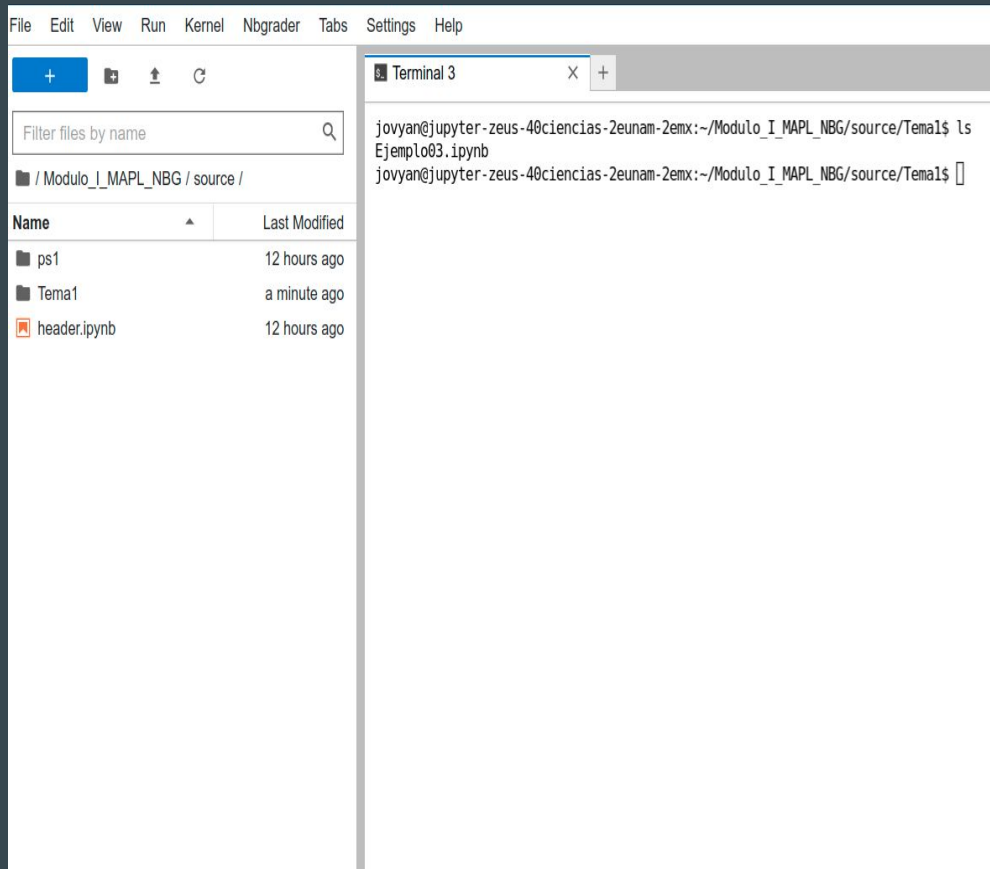


source

Fuentes

Dentro de la carpeta *source* vamos a poner todos los temas (con sus respectivos *notebooks*) que vayan a ser procesados con *nbgrader*.

Para este ejemplo en particular basta con agregar la carpeta *Tema1* con el *notebook* *Ejemplo03.py*.



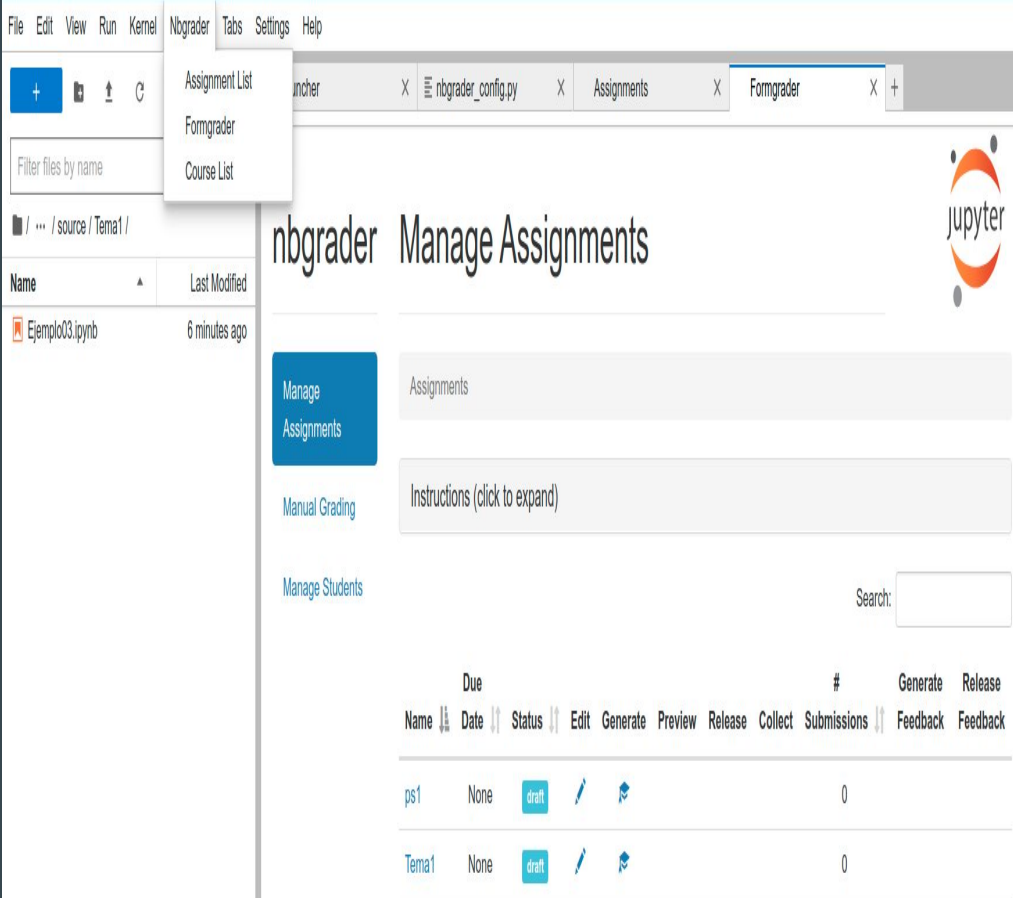
Form Grader

Menú NBGrader

Si todo resultó de manera correcta podemos ir al botón *NBGrader* en la parte superior de la pantalla y dar click en la opción *Form Grader*.

Esto nos mostrará todos los temas que están listos para ser procesados con *NBGrader*.

En caso de no mostrar el contenido basta con reiniciar el server.



The screenshot shows the nbgrader web interface. The 'Nbgrader' menu is open, showing options: Assignment List, Formgrader, and Course List. The 'Formgrader' option is selected. The main content area is titled 'nbgrader Manage Assignments'. It features a sidebar with links: Manage Assignments (active), Manual Grading, and Manage Students. The main area has a search bar and a table of assignments.

Name	Due	Status	Generate	Preview	Release	Collect	Submissions	Generate Feedback	Release Feedback
ps1	None	draft					0		
Tema1	None	draft					0		

Form Grader

Opciones *Assignments*

Por cada tema dentro del menú de *Assignments* tenemos varias opciones:

- **Edit:** nos permite editar metadatos del tema.
- **Generate:** procesa todos los notebooks de ese tema con *nbgrader*.
- **Preview:** vemos la versión que se entrega al estudiante.
- **Release:** se libera el notebook.
- **Collect:** recolectamos los *notebooks*.

nbgrader

Manage Assignments

Manage Assignments

Manual Grading

Manage Students

Assignments

Instructions (click to expand)

Search:

	Due								#	Generate	Release
Name	Date	Status	Edit	Generate	Preview	Release	Collect	Submissions	Feedback	Feedback	
ps1	None	draft						0			
Tema1	None	draft						0			

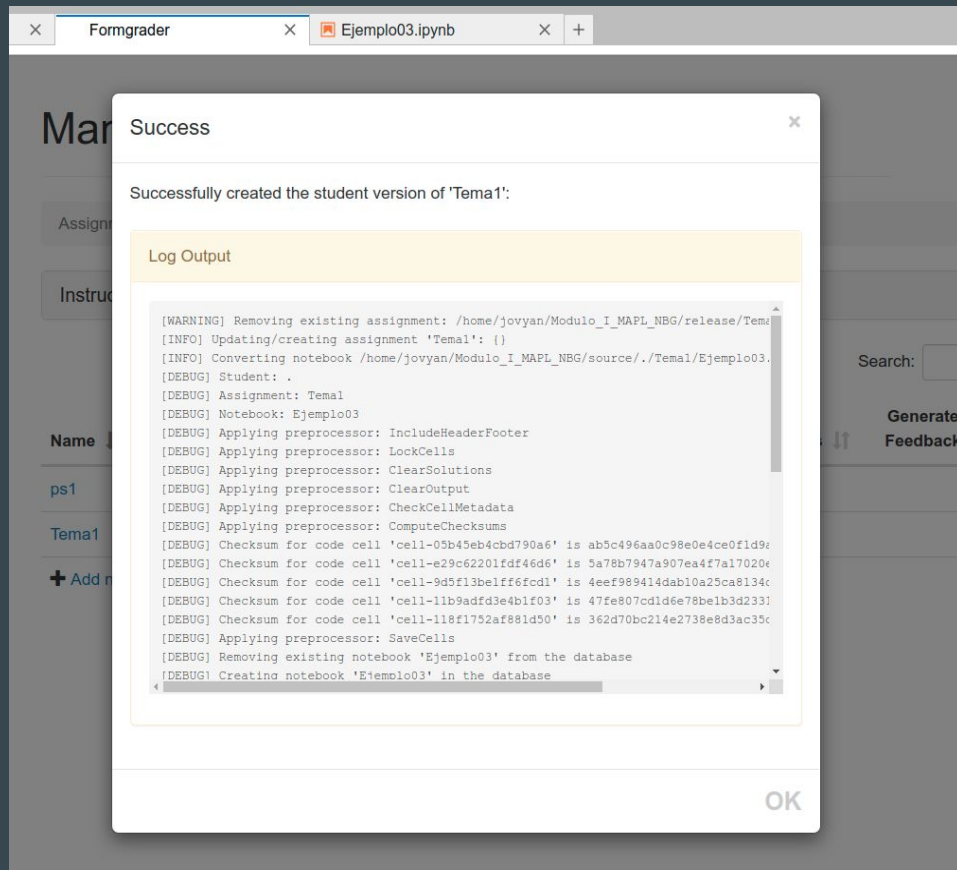
+ Add new assignment...

Generando *notebook*

Generate

Una vez en el menú de *asignments*, podemos dar click en la opción *Generate*.

Y en caso de que no haya errores al momento de ser generada la versión del estudiante, se habilitará la opción *Preview*, que nos mostrará la versión “mutilada”, es decir sin respuestas del notebook original (*Ejemplo03.py*)



Versión estudiante

Preview

Al dar click en la opción *Preview* (icono de lupa) se muestra en el explorador de archivos un la versión del estudiante del *notebook* original.

Este “nuevo” *notebook* se genera en la ruta:

Nombre_Curso_NBG/release

The screenshot displays the nbgrader web interface. On the left, a file explorer shows the path `/.../release/Tema1/` with a file named `Ejemplo03.ipynb` modified 4 minutes ago. On the right, the 'Manage Assignments' panel is active, showing a table of assignments and a sidebar with navigation links.

nbgrader Manage Assignments

Assignments

Instructions (click to expand)

Manual Grading

Manage Students

Name	Due	Date	Status	Edit	Generate	Preview
ps1	None	draft				
Tema1	None	draft				

+ Add new assignment...

Versión estudiante

Preview

Si abrimos este notebook, podemos notar varias cosas, pero las más relevantes son:

- Todas las secciones con `### BEGIN SOLUTION` y `### END SOLUTION` fueron sustituidas.
- Hay celdas que no pueden ser editadas, de acuerdo a cómo se creó este notebook.

```
Launcher x Formgrader x Ejemplo03.ipynb x Ejemplo03.ipynb x Python 3 (ipykernel)
```

Creamos datos

```
[ ]: t = np.linspace(-2 * np.pi, 2 * np.pi, 360)
```

Ejercicio 1. Calcular el $\sin(t)$.

Escriba un código para calcular la función $f(t) = \sin(t)$ para $t \in [-2\pi, 2\pi]$, usando funciones de la biblioteca `numpy`.

```
[3]: # YOUR CODE HERE
raise NotImplementedError()
```

NotImplementedError Traceback (most recent call last)
Cell In[3], line 2
1 # YOUR CODE HERE
----> 2 raise NotImplementedError()
NotImplementedError:

```
[ ]: quizz.eval_numeric('1', f)
```

Graficamos datos

```
[ ]: v = mvis.Plotter(1,1)
v.plot(1,t,f)
v.set_coordsys(1, xlabel='$t$', ylabel='$f(t)$',
               xlabelsize=10, ylabelsize=10, trig=True)

plt.show()
```


Liberar al estudiante

Release

Una vez que estamos seguros que el material ya puede ser liberado a los estudiantes, lo siguiente es presionar el icono *Release*.

Con esto lo que hacemos es poner al alcance del estudiante el material para que este lo pueda contestar.








Manage Assignments



Assignments

Instructions (click to expand)

Search:

Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback
ps1	None	draft						0		
Tema1	None	released						0		

+ Add new assignment...

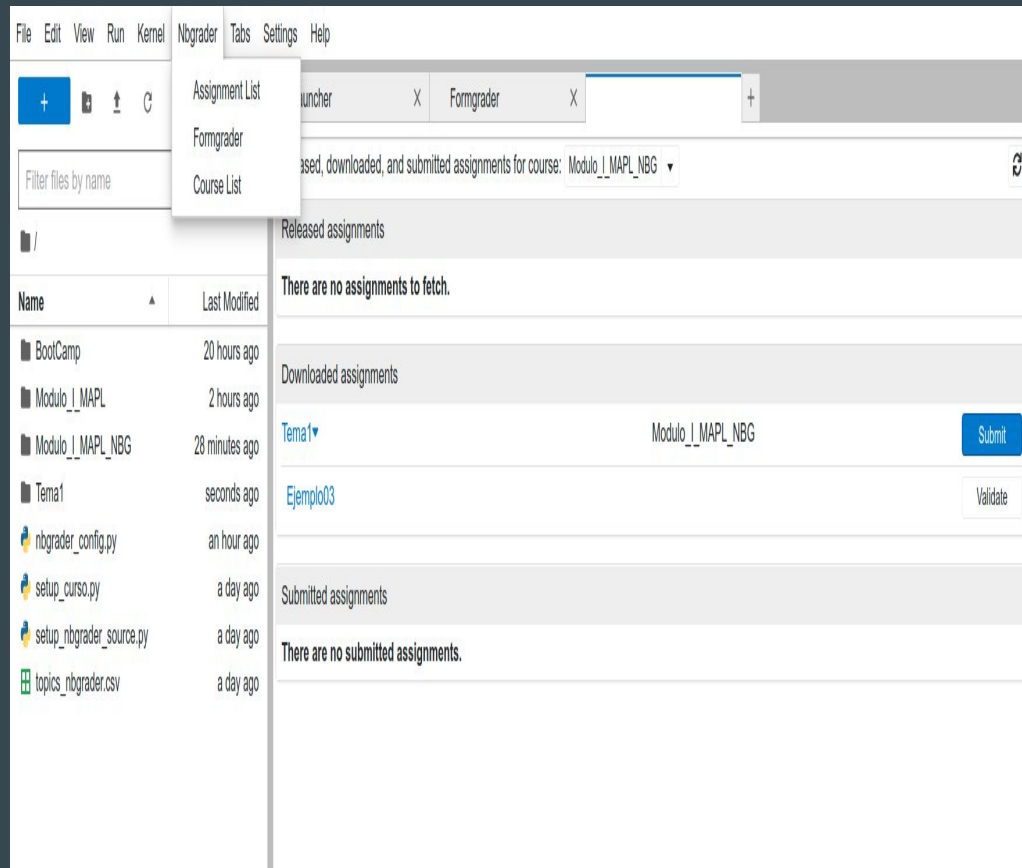
Tomar por el estudiante

Fetch

Ya que se liberó el material, el alumno puede ir al menú *NBGrader->Assignment List* y tomar el material

Al hacer esto se genera en el /home del alumno una carpeta por cada tema que haya tomado (*fetch*).

Una vez respondido el notebook, el alumno puede hacer *Submit*.

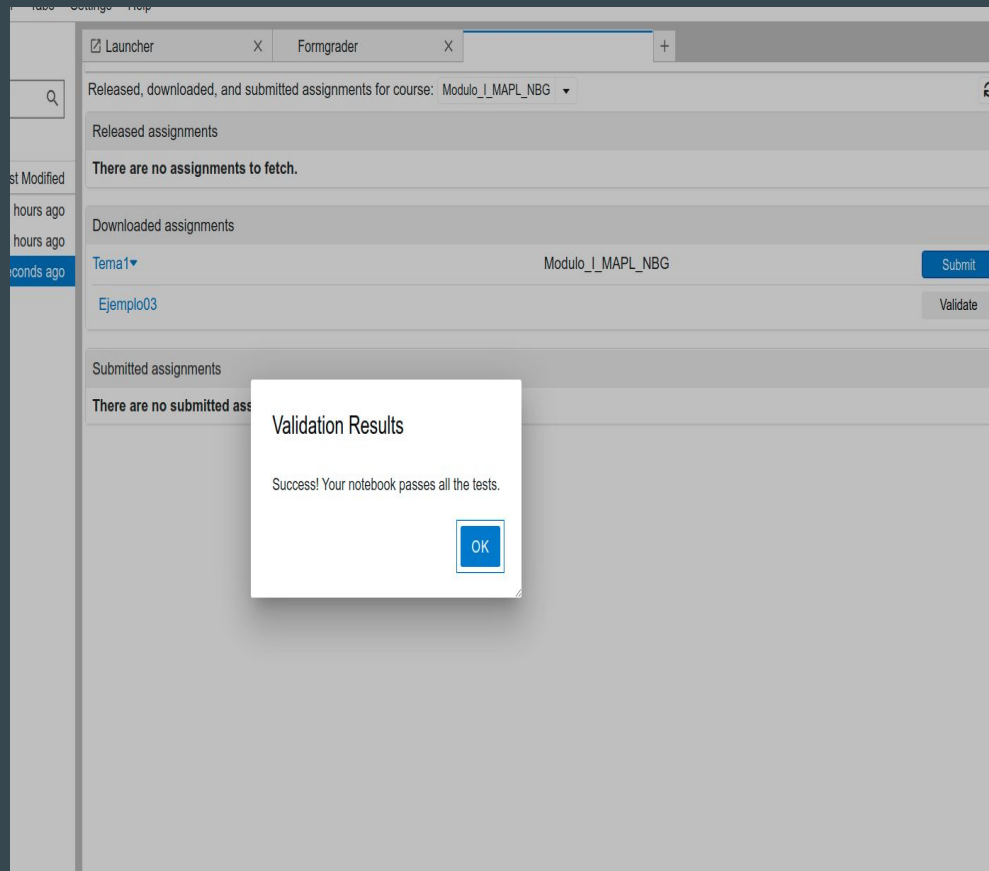


Aprobado

Pass

Si el alumno responde de manera correcta todo el *notebook*, *NBGrader* mostrará el siguiente mensaje, indicando que sus respuestas pasaron la prueba y que puede enviar sus respuestas al profesor.

En caso contrario *NBGrader* se encarga de mostrar que el *notebook* no pasa la prueba.



Retroalimentación

Feedback

Ya que el alumno se encargó de enviar sus respuestas, se le habilita un nuevo listado con los temas que ha respondido y con la posibilidad de obtener retroalimentación.

La retroalimentación la proporciona el profesor una vez que ha revisado la entrega del alumno.

Launcher

Formgrader

Released, downloaded, and submitted assignments for course: Modulo_I_MAPL_NBG

Released assignments

There are no assignments to fetch.

Downloaded assignments

Tema1

Modulo_I_MAPL_NBG

Submit

Submitted assignments

Tema1

Modulo_I_MAPL_NBG

Fetch Feedback

2023-05-09 14:16:10.063057 UTC

De parte del profesor

Finalmente

De parte del profesor sólo resta recolectar las entregas de los alumnos, autoevaluar y dejar retroalimentación en caso de ser necesario.

La retroalimentación la proporciona el profesor una vez que ha revisado la entrega del alumno.

Search: <input type="text"/>											
Name	Due Date	Status	Edit	Generate	Preview	Release	Collect	# Submissions	Generate Feedback	Release Feedback	
ps1	None	draft						0			
Tema1	None	released						1			

Instructions (click to expand)											
Search: <input type="text"/>											
Student Name	Student ID	Timestamp	Status	Score	Autograde	Generate Feedback	Release Feedback				
None, None	zeus@ciencias.unam.mx	2023-05-09 14:16:10 UTC	graded	5 / 5							

Ejercicio 1. Calcular el $\sin(t)$.

Escriba un código para calcular la función $f(t) = \sin(t)$ para $t \in [-2\pi, 2\pi]$, usando funciones de la biblioteca `numpy`.

In [4]:

Student's answer

```
# YOUR CODE HERE
f = np.sin(t)
```

!!!Muy bien!!!

In [5]:

Scripts

setup_curso.py

Este script nos ayuda a generar la estructura sugerida para comenzar a desarrollar nuestros materiales.

setup_nbgrader_source.py

Genera los archivos de configuración y estructura necesaria para procesar los notebooks con *NBGrader*.

topics_nbrgader.csv

Mediante este csv le hacemos saber al script *setup_nbgrader_source.py* cuales notebooks van a ser procesados con *nbgrader*.

Muchas gracias a
nombre del proyecto
MACTI