

# FACULTAD CIENCIA Y TECNOLOGÍA UADER.



**CARRERA:** Lic. En Sistemas de la Información:

**DOCENTE:** Julian Escalante, Pedro Colla.

**AÑO:** 3er año Turno noche

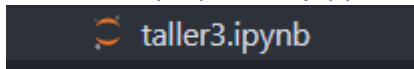
**ASIGNATURA:** Base de datos

**TEMA:** Modelo conceptual.

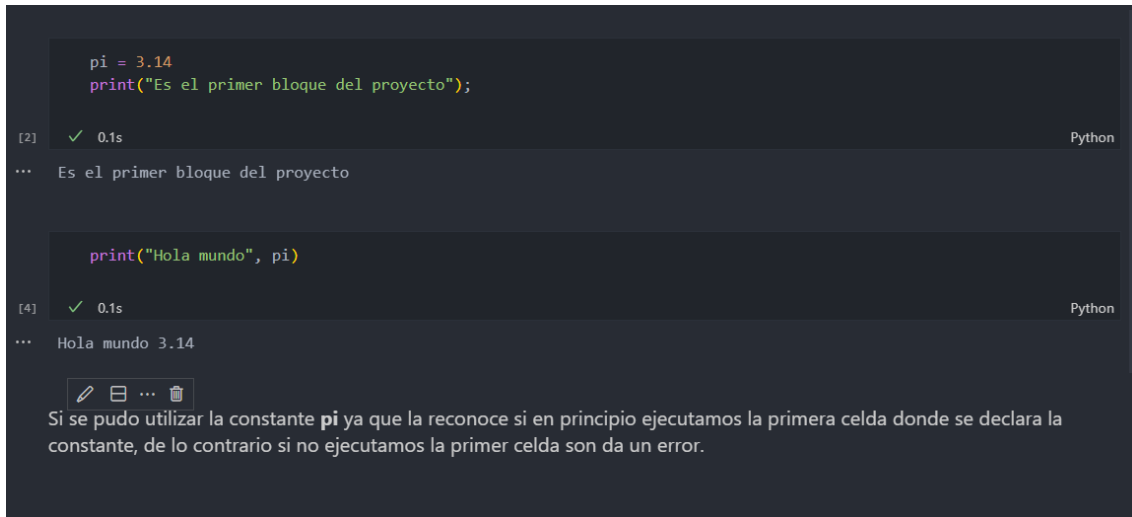
**ALUMNOS:** Shayen Sirk, Alex Rodríguez, Matías Racigh, jerónimo Perez.

## Parte 1 taller 3

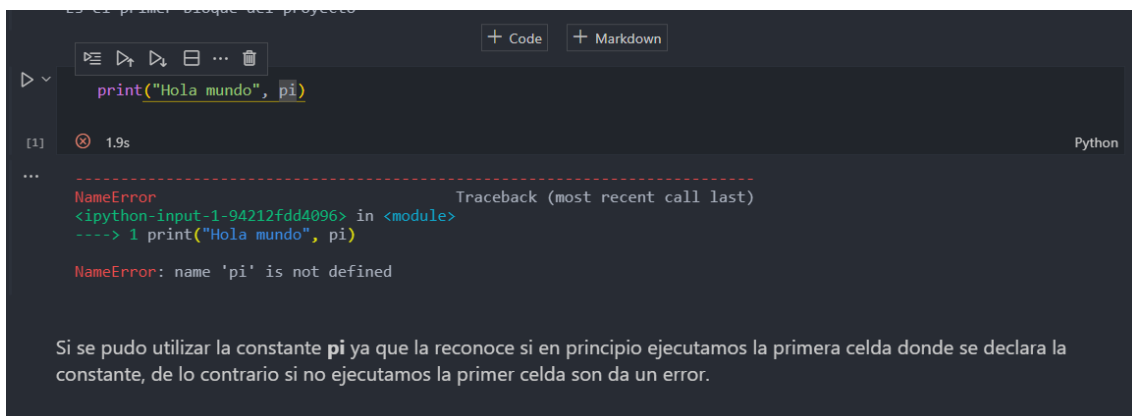
- 1) Creamos un proyecto en jupyter notebook .



- 2) Luego aplicamos los puntos, del punto 2 al 4.

This screenshot shows two code cells in a Jupyter Notebook. The first cell, labeled [2], contains the code `pi = 3.14` and `print("Es el primer bloque del proyecto");`. It executed successfully in 0.1s, and the output is "Es el primer bloque del proyecto". The second cell, labeled [4], contains the code `print("Hola mundo", pi)`. It also executed successfully in 0.1s, and the output is "Hola mundo 3.14". Below the code cells, there is a text box with a pencil icon and a trash icon, containing the text: "Si se pudo utilizar la constante **pi** ya que la reconoce si en principio ejecutamos la primera celda donde se declara la constante, de lo contrario si no ejecutamos la primer celda son da un error."

- 5) Pudimos comprobar que se **reinicia** el kernel, con el ejemplo claro en el segundo bloque que no permite leer la constante del primer bloque de código, debido a que elimina las variables y contantes de los códigos.

This screenshot shows a code cell in a Jupyter Notebook that has failed to execute. The cell is labeled [1] and contains the code `print("Hola mundo", pi)`. The execution time is 1.9s, and it resulted in a `NameError`. The error message is: `NameError: name 'pi' is not defined`. Above the error message, it says "Traceback (most recent call last)". Below the error message, there is a text box with a pencil icon and a trash icon, containing the text: "Si se pudo utilizar la constante **pi** ya que la reconoce si en principio ejecutamos la primera celda donde se declara la constante, de lo contrario si no ejecutamos la primer celda son da un error."

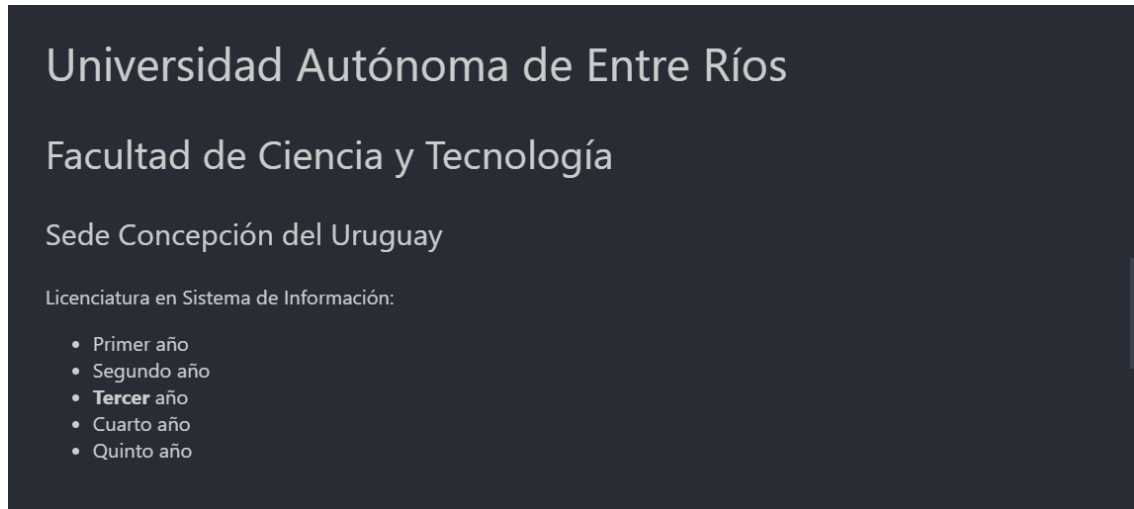
- 6) La función Restart nos permite reiniciar el kernel en jupyter notebook. EL kernel es el "motor computacional" que ejecuta el código contenido en un documento de Notebook. El kernel de ipython, ejecuta el código de python. Este núcleo es el encargado de realizar los cálculos y producir los resultados.

La función de restart & clear lo que hace es reiniciar el kernel pero a su vez borra los resultados de cada celda.

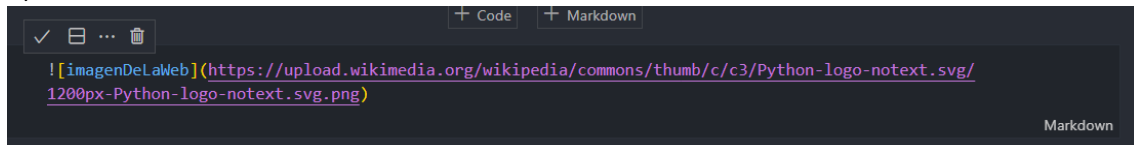
La función interrupt la función que cumple es de detener la ejecución del código, pero no elimina la variable que ha almacenado en la memoria, en este caso la constante.

- 7) La función que cumple los "In[numero]" nos indica el número de celda y la orden en la que se ejecuta cada bloque de código o de marckdown.

8) Salida del marckdown:



9)



Salida:



10)

```
✓ ☰ ... 🗑️  
![messi](messi.jpg)  
Markdown
```

Salida:



11) Para cambiar el orden de las celdas solo nos basta con arrastrar el fragmento de código o markdown al posición de deseemos. Y para combinarlas podemos hacer uso de la opción de “**unir con la celda de arriba o unir con la celda de abajo**” y ahí pudimos unir los fragmentos.