

# Tarea 06

fl.gomez10 at uniandes.edu.co

9 de marzo de 2019

Horario de atención: Principalmente de 2:00pm a 5:00pm en la oficina i-109. También se pueden enviar dudas al correo electrónico.

Entregar la carpeta de trabajo en un archivo comprimido `hw06-username.tar` antes de finalizar la clase.

Trabaje iniciando sesión en la máquina virtual en línea [mybinder.org/](https://mybinder.org/)<sup>1</sup>. Aparte, cree un archivo de texto llamado `bitacora.txt`

## Ejercicio 1 (30 puntos) Trabajo en Casa

### a (10 pts): Módulo Desconocido.

- Abra un Notebook de Python 3. Guárdelo como `ejercicio01a.ipynb`
- En la primera celda importe el paquete `imageio`.
- Ejecute la celda con `shift + enter`
- ¿Qué resultado arroja la ejecución de esta celda? Guarde la salida (Esta puede ser una advertencia o un error) en la bitácora.
- Guarde este notebook. `ctrl + s`

### b (20 pts): Instalación de un Módulo con PIP

- Cierre el notebook anterior. Abra una terminal e instale con el comando `pip` el paquete `imageio`, del mismo modo que en el [vídeo \(minuto 4:08\)](#) se instala `jupyter`.
- Guarde en la bitácora la información que arroja el proceso de instalación.
- Abra un Notebook de Python 3.
- En la primera celda importe el paquete `imageio`.
- Ejecute la celda con `shift + enter`

---

<sup>1</sup><https://mybinder.org/v2/gh/ComputoCienciasUniandes/FISI2026-201910/master?urlpath=lab>

- ¿Qué resultado arroja la ejecución de esta celda? Guarde la salida (Esta puede ser una advertencia o un error) en la bitácora. Comente este resultado.
- Guarde este notebook como `ejercicio01b.ipynb`

## Ejercicio 2 (40 puntos) Trabajo en Casa

### a (10 pts)

Guarde el siguiente código en un archivo llamado `grafica_cos.py`.

```

1 import matplotlib.pyplot as plt
2 import math
3
4 t = 0
5 n = 0
6
7 A = list( range(0,100) )
8 x = []
9 for elemento in A:
10     x.append( t + elemento * 2 * 3.1416 / 100 )
11
12 y = []
13 for elemento in x:
14     y.append( math.cos(elemento) )
15
16 plt.subplot(111)
17 plt.axis("equal")
18 plt.plot(x,y)
19 plt.title( 't = {}'.format(t) )
20 plt.savefig( '{}.png'.format(n) )
21 plt.close()

```

Listing 1: `grafica-cos.py`

Este código genera cien números igualmente espaciados entre 0 y  $2\pi$ , le suma un número “t”, luego calcula  $y = \cos(x + t)$ , grafica y guarda como `n.png`.

- Corra el código una vez para comprobar su funcionamiento.
- Cambie `t` y `n` (entero) para comprobar su funcionamiento.
- Modifique el código para que sea una función que tenga dos argumentos de entrada: `t` y `n`. Con esto, al llamar la `funcion(t,n)` esta genera la gráfica y la guarda.
- Guarde este archivo.

### b (20 pts)

Cree un nuevo Notebook llamado `ejercicio02b.ipynb`.

- Importe el módulo `grafica.cos` que creó en el numeral anterior.<sup>2</sup>
- Llame la función que creó en el numeral anterior para comprobar que funciona.
- Inicie un contador  $n = 0$ , la variable  $t = 0$  y un paso de tiempo `DeltaT=0.2`
- Cree un ciclo `while` que termine cuando  $t > 6,29$
- en este ciclo llame la función que grafica con los argumentos `t` y `n`.
- aumente `t += deltat`.
- aumente `n += 1`

Ejecute la celda. Esto debería generar cerca de 30 archivos de imagen. Guarde el Notebook.

### c (10 pts)

En este ejercicio se va a hacer una “película.gif” con las imágenes que se crearon en el numeral anterior.

- Cree un nuevo notebook llamado `ejercicio02c.ipynb`.
- Cree una lista con los nombres de las imágenes como strings. Llámela `filenames`.
- Use una estructura de código similar a la propuesta para que el módulo `imageio` genere un nuevo archivo apilando las imágenes.

```

1 import imageio
2
3 filenames = []  ## Lista con los nombres de los archivos
4
5 with imageio.get_writer('pelicula.gif', mode='I') as writer:
6     for filename in filenames:
7         image = imageio.imread(filename)
8         writer.append_data(image)

```

Listing 2: ejercicio02x.ipynb

## Ejercicio 3 (En Clase)

Este ejercicio se define en clase.

---

<sup>2</sup>Pilas con el nombre del módulo. Si lo llaman “grafica-cos” el programa entenderá que hay que realizar una resta “grafica” menos “cos”.