

# Tarea 07

fl.gomez10 at uniandes.edu.co

23 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 `hw07-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`

## 1. Ejercicio 1 (30 puntos) Trabajo en Casa

Cree un notebook llamado `ejercicio01.ipynb`. En la primera celda puede incluir `%pylab inline` para cargar `numpy` y `matplotlib` de una vez.

### 1.1. A (6pts)

Graficar la función coseno

- (2 pts) Cree un array unidimensional `a` que tenga 30 números desde  $-2\pi$  hasta  $2\pi$  igualmente distanciados usando `np.linspace()`
- (2 pts) Cree un array unidimensional `b` que sea el coseno de `a`. Créelo directamente operando sobre el array `a` como un todo, no elemento por elemento.
- (2 pts) Grafique `b` vs. `a`.

### 1.2. B (14 pts)

Crear un array de  $8 \times 8$  que tenga el patrón del tablero de ajedrez.

- (2pts) Cree un array de ceros de  $8 \times 8$  usando `np.zeros()`.
- (4pts) Con un doble `for` (uno para barrer filas y otro para barrer columnas), recorra el array y coloque unos cada tanto siguiendo el patrón del tablero de ajedrez intercalando unos y ceros en ambos ejes (0 y 1).

---

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

- (4pts) Grafique usando `plt.imshow`.
- (4pts) Cree e imprima un array de  $8 \times 8 \times 4$  con un patrón de ajedrez 3D. Esto es, si uno se desplaza en cualquier eje (0, 1 o 2) va a encontrar intercalados unos y ceros.

### 1.3. C (10 pts)

Cree una variable  $N = 4$ . Cree un array de  $N \times N$  donde la matriz diagonal superior sean ceros, la diagonal sean unos y la matriz diagonal inferior se llene incrementando del siguiente modo:

$$\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ 5 & 6 & 7 & 1 \end{array} \quad (1)$$

Debe funcionar bien para los casos  $N = 0, 1, \dots, 10$ . Con esto se calificará. (10 pts). Puede empezar creando un array de ceros usando `np.zeros()`

## 2. Ejercicio 2 (30 pts)

En un notebook llamado `ejercicio02.ipynb` copie el siguiente fragmento de código.

```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3 import numpy as np
4
5 img=mpimg.imread('https://github.com/ComputoCienciasUniandes/
6   FISI2026-201910/raw/master/Talleres/Grupo_1/sorpresahubble.png
7   ')
8
9 imgplot = plt.imshow(img)
```

Listing 1: grafica-cos.py

- (10 pts) Escriba en una celda la forma (shape) de `img`. ¿Es un solo array? ¿Son varias capas? ¿Qué representa cada capa?
- (10 pts) Cree una variable  $k > 2,0$ . Con esta cree un nuevo array de la forma `img**k`. Grafique con `plt.imshow()`. En otra celda indique qué tipo de operación se está realizando.
- (10 pts) Aumente el valor de  $k$  hasta poder ver claramente la imagen. ¿Cuál es el valor crítico de “ $k$ ” que le permite ver el resultado?

Guarde el notebook con el resultado.

### 3. Ejercicio 3 (40 pts) Suavizado de imágenes

Este ejercicio es una breve introducción al procesamiento de imágenes desde un Jupyter Notebook. Se quiere suavizar una imagen (en escala de grises). Se puede trabajar con la imagen de wikipedia <sup>2</sup> o con esta versión de la [imagen con ruido](#) <sup>3</sup>.

Puede iniciar el notebook con:

```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimage
3 import numpy as np
4
5 image = mpimage.imread("https://github.com/ComputoCienciasUniandes/
    FISI2026-201910/raw/master/Talleres/Grupo_2/grayscale_w_noise.
    png")
6 plt.imshow(image)
```

El algoritmo debe funcionar de este modo:

- Se genera un nuevo array de ceros del mismo tamaño que la imagen original.
- Para un píxel en la posición  $[i, j]$  se suma la intensidad de sus ocho vecinos cercanos.
- (10 pts) Se promedia junto con la intensidad del píxel en cuestión.
- (10 pts) El resultado se almacena en la posición  $i, j$  del nuevo array.
- (10 pts) Se barren las filas y las columnas. Se pueden omitir la primera y la última, o se puede escribir el algoritmo para que calcule de forma distinta el promedio en las celdas de los bordes y las esquinas.
- (10 pts) Graficar con `imshow` la imagen original y el nuevo array.

---

<sup>2</sup>[https://upload.wikimedia.org/wikipedia/commons/f/fa/Grayscale\\_8bits\\_palette\\_sample\\_image.png](https://upload.wikimedia.org/wikipedia/commons/f/fa/Grayscale_8bits_palette_sample_image.png)

<sup>3</sup>[https://github.com/ComputoCienciasUniandes/FISI2026-201910/raw/master/Talleres/Grupo\\_2/grayscale\\_w\\_noise.png](https://github.com/ComputoCienciasUniandes/FISI2026-201910/raw/master/Talleres/Grupo_2/grayscale_w_noise.png)