

October 16, 2021

## 1 Fundamentos de Programación

### 1.1 PEC 4 - Enunciado

En este Notebook se encontraréis el conjunto de actividades evaluables como PEC de la asignatura. Veréis que cada una de ellas tiene asociada una puntuación, que indica el peso que tiene la actividad sobre la nota final de la PEC. Adicionalmente, hay un ejercicio opcional, que no tiene puntuación dentro de la PEC, pero que se valora al final del semestre de cara a conceder las matrículas de honor y redondear las notas finales. Podréis sacar la máxima nota de la PAC sin necesidad de hacer este ejercicio. El objetivo de este ejercicio es que sirva como pequeño reto para los estudiantes que quieran profundizar en el contenido de la asignatura.

Veréis que todas las actividades de la PEC tienen una etiqueta, que indica los recursos necesarios para llevarla a cabo. Hay tres posibles etiquetas:

- **NM Sólo materiales:** las herramientas necesarias para realizar la actividad se pueden encontrar en los materiales de la asignatura.
- **EG Consulta externa guiada:** la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, pero el enunciado contiene indicaciones de dónde o cómo encontrar la información adicional necesaria para resolver la actividad.
- **EI Consulta externa independiente:** la actividad puede requerir hacer uso de herramientas que no se encuentran en los materiales de la asignatura, y el enunciado puede no incluir la descripción de dónde o cómo encontrar esta información adicional. Será necesario que el estudiante busque esta información utilizando los recursos que se han explicado en la asignatura.

Es importante notar que estas etiquetas no indican el nivel de dificultad del ejercicio, sino únicamente la necesidad de consulta de documentación externa para su resolución. Además, recordad que las **etiquetas son informativas**, pero podréis consultar referencias externas en cualquier momento (aunque no se indique explícitamente) o puede ser que podáis hacer una actividad sin consultar ningún tipo de documentación. Por ejemplo, para resolver una actividad que sólo requiera los materiales de la asignatura, podéis consultar referencias externas si queréis, ya sea tanto para ayudaros en la resolución como para ampliar el conocimiento!

En cuanto a la consulta de documentación externa en la resolución de los ejercicios, recordad **citar siempre la bibliografía utilizada** para resolver cada actividad.

## 1.2 Ejercicios para la PEC

### NumPy - Ejercicios

#### 1.2.1 Ejercicio 1

Crea tres matrices diferentes 3x3 y muéstralas por pantalla. Verifica que la propiedad asociativa del producto entre matrices elemento a elemento se cumple con las matrices creadas anteriormente. El producto entre matrices elemento a elemento se realiza mediante el operador `*`.

**Nota:** Podéis consultar el siguiente [link](#) para saber en qué consiste la propiedad asociativa del producto.

(0.5 puntos) NM

```
[ ]: # Respuesta
```

#### 1.2.2 Ejercicio 2

Crea un array de NumPy 1D de 10 elementos aleatorios enteros de 0 a 100. A continuación, pon en negativo todos los elementos impares y muestra el array resultante por pantalla.

**Nota:** Para resolver este ejercicio pueden ser de utilidad las funciones [randint](#) y [where](#).

(0.5 puntos) EG

```
[ ]: # Respuesta
```

#### 1.2.3 Ejercicio 3

Considera la matriz `[[1, 2, 3], [4, 5, 6],[7,8,9]]`. Transfórmala en una matriz 1D considerando el orden por columnas.

**Nota:** Para resolver este ejercicio puede ser de utilidad la función [ravel](#).

(0.5 puntos) EG

```
[ ]: # Respuesta
```

#### 1.2.4 Ejercicio 4

Definid una función que calcule y muestre por pantalla el producto matricial de dos matrices A y B mediante la función de NumPy `dot()`. La función deberá realizar lo siguiente:

- Primero comprobar si las matrices tienen las dimensiones adecuadas para realizar el producto matricial.
- Segundo realizar el producto matricial de dos matrices A y B. Mostrar el resultado por pantalla.

Si las matrices no tuvieran las dimensiones adecuadas, la función deberá mostrar un mensaje de alerta por pantalla y no realizar el producto matricial. A continuación, poned un par de ejemplos para validar la función en las dos casuísticas posibles.

(1 punto) NM

```
[ ]: # Respuesta
```

### 1.2.5 Ejercicio 5

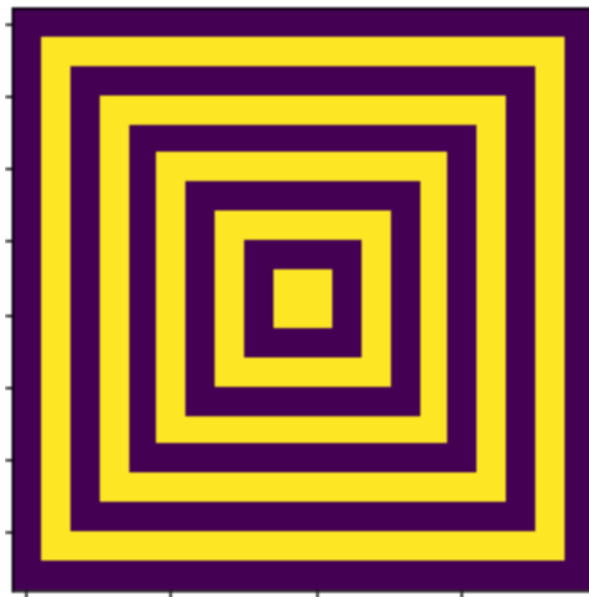
Crea una matriz, M, de tamaño 5x5 con números enteros aleatorios de 1 al 100 sin repetición. Modificad la matriz anterior de tal forma que  $M[i, j] = (3*j+i)^2$  para todo valor múltiple de 3, y para el resto de posiciones  $M[i, j] = M[j, i]$  donde  $i, j$  corresponden a los índices de las filas y columnas.

(1 punto) NM

```
[ ]: # Respuesta
```

### 1.2.6 Ejercicio 6

Crea una función que genere una matriz cuadrada con el siguiente patrón de 0 y 1, siendo 0 la capa inicial. La función debe de tener sólo un parámetro de entrada correspondiente al número de filas



o columnas de la matriz.

La función tiene que mostrar el resultado visualmente con la función `imshow` de Matplotlib que nos permite visualizar una imagen 2D. (1 punto)NM

```
[1]: # Respuesta
```

Matplotlib - Ejercicios

### 1.2.7 Ejercicio 7

Representa en un único gráfico las siguientes funciones  $f1(x)$  y  $f2(x)$ , evaluadas en el intervalo  $[-10, 10]$  y con paso (resolución) 0.01. Haz que los rangos de valores mostrados sean  $[-2, 2]$  para el eje X y  $[1, -3]$  por el eje Y para visualizar la figura que se obtiene de representar ambas funciones

a la vez. Adicionalmente, muestra la gráfica correspondiente a  $f_1(x)$  con color verde y la gráfica correspondiente a  $f_2(x)$  con color rojo. Se debe de mostrar la leyenda.

(1 punto) NM

```
[3]: # Definimos las funciones
def f1(x):
    return np.sqrt(1-np.power((abs(x)-1), 2))

def f2(x):
    return np.arccos(1-abs(x))-np.pi
```

```
[4]: # Respuesta
```

### 1.2.8 Ejercicio 8

Los 10 países en encabezar el medallero de los Juegos Olímpicos Tokyo 2020 fueron los siguientes:

Country	Gold	Silver	Bronze	Total
USA	39	41	33	113
China	38	32	18	88
Japan	27	14	17	58
Great Britain	22	21	22	65
ROC	20	28	23	71
Australia	17	7	22	46
Netherlands	10	12	14	36
France	10	12	11	33
Germany	10	11	16	37
Italy	10	10	20	40

Mostrad en un único gráfico de barras verticales el número de medallas totales obtenidas por país. Consideraciones que debéis de tener en cuenta para realizar el gráfico: - Cada una de las barras de los diferentes países tiene que incluir la información del número de medallas obtenidas de oro, plata y bronce apilando la información de cada metal y coloreando con un color diferente. - El eje de las X tiene que incluir las etiquetas con los nombres de los países. - Se debe incluir un texto con el número total de medallas para cada país en la parte superior de la barra. - Se debe incluir leyenda. - Se debe incluir el título de cada uno de los ejes. - El tamaño de la figura tiene que ser de 10 por 10.

**Nota** Para resolver este ejercicio podéis consultar la documentación oficial de [Matplotlib](#). También pueden ser de utilidad las funciones [text](#) o [annotate](#) para mostrar el número de medallas total. (1,5 puntos) EG

```
[ ]: # Respuesta
```

pandas - Ejercicios

### 1.2.9 Ejercicio 9

Carga los datos del archivo `cwurData`, que puedes encontrar en la carpeta `data`, en un `dataframe`. Este conjunto datos recoge diferentes rankings globales de las mejores universidades del mundo según diferentes parámetros hasta el 2015. Estos datos los encontraréis en la web [Kaggle](#).

Muestra por pantalla las variables del dataset, el número de filas y columnas.

(0.5 puntos) NM

```
[ ]: # Respuesta
```

### 1.2.10 Ejercicio 10

La variable *COUNTRY* contiene los países que conforman este ranking. Para cada país mostrad la media aritmética de la variable *Score*. ¿Qué país tiene en promedio el score más alto?

**Nota.** Para resolver este ejercicio puede ser de utilidad la función [aggregate](#) de pandas, la qual nos permite aplicar diferentes funciones a una columna en un groupby.

(0.5 puntos) EG

```
[ ]: # Respuesta
```

### 1.2.11 Ejercicio 11

Muestra las universidades de América del Norte que según el ranking de publicaciones se encuentran en la posición 500 o inferior en el 2015 y además, comparten la misma posición en el ranking de ocupación de alumnos *alumni\_employment*.

**Nota:** En el Notebook de teoría hemos visto operaciones básicas de filtrado. Para resolver este ejercicio necesitarás investigar como aplicar condiciones más complejas.

(1 punto) EG

```
[ ]: # Respuesta
```

### 1.2.12 Ejercicio 12

Crea una copia del dataframe original añadiendo una nueva columna en la posición 5 que nos indicará si el ranking según la calidad de la educación *quality\_of\_education* es alto, medio o bajo.

- **High:** si el ranking de calidad es superior a 100
- **Medium:** si el ranking de calidad está entre 100 y 300
- **Low:** si el ranking de calidad es inferior a 300

Muestra por pantalla aquellas filas que tiene una ranking según calidad de la educación bajo en Portugal.

(1 punto) EI

```
[ ]: # Respuesta
```

SciPy - Ejercicio

### 1.2.13 Ejercicio Opcional

Dado un conjunto de valores  $x$  e  $y$ , ajusta dichos valores mediante una interpolación lineal,  $f1(x)$ , y una interpolación cúbica,  $f2(x)$ . Dibuja los datos originales, la función  $f1(x)$  y la función  $f2(x)$  en el intervalo  $[0,4]$  y con un paso de resolución de 0.1. Se debe incluir la leyenda.

**Nota:** Podéis utilizar la librería [interpolate](#).

```
[ ]: # Definimos las funciones  
  
x = np.linspace(0, 4, 10)  
  
y = np.cos(np.power(x, 2)+3)
```

```
[ ]: # Respuesta
```

---

---