



**Python 3**

Práctica:

## 6. Programación orientada a objetos

Carolina Mañoso, Ángel P. de Madrid y Miguel Romero

# Enunciado

- ◆ En este tema le proponemos realizar dos ejercicios.
- ◆ El primero es más sencillo y es obligatorio.
- ◆ El segundo es opcional. Realícelo sólo si ha sido capaz de realizar el primero sin excesiva dificultad. Sirve para poner en práctica el concepto de *herencia* en POO. Además, practicará con *bucles*, *condicionales* y manejo de *listas*. Es una buena manera de poner a prueba los conocimientos que ha adquirido hasta ahora.
  - En este caso, entregue en la plataforma los dos ficheros `.py` como un único archivo comprimido.



# Ejercicio 1: Países

## ◆ Defina:

- Clase `País` con los siguientes atributos: `nombre`, `población` y `área`.
- Dos métodos: `más grande que` y `densidad de población`.
  - ◆ `más grande que` : Devuelve `True` si este país es más grande que otro (que pasamos por parámetro).
  - ◆ `densidad de población` : Devuelve la densidad de población (población dividida por superficie).

## ◆ Realice un programa en Python que compare las superficies de España y Francia y calcule e imprima las densidades de población de ambos.

## ◆ Datos:

- España: 46.770.000 habitantes y 504.645 km<sup>2</sup>.
- Francia: 66.030.000 habitantes y 640.679 km<sup>2</sup>.
- Si lo desea, busque y añada datos adicionales de otros países...

## Ejercicio 2: Universidad (1/6)

- ◆ En este ejercicio realizará un programa en Python que asigne e imprima en pantalla las asignaturas, profesores y estudiantes de una pequeña “universidad”.
- ◆ Defina:
  - Clase superior `Miembro`.
    - ◆ Con atributos : `nombre`, `edad` y `dni`.
  - Dos clases que heredan de ella: `Profesor` y `Estudiante`.
    - ◆ `Profesor` tiene, adicionalmente:
      - Atributo `número de registro`.
      - Atributo `asignaturas imparte`: Lista, inicialmente vacía, con la relación de asignaturas que imparte el profesor.
      - Método `añade docencia`. Añade un elemento a la lista de asignaturas que imparte el profesor.
      - Método `imprime docencia`. Imprime la lista de asignaturas que imparte un profesor, junto con la relación de estudiantes matriculados que hay en cada una de ellas.
    - ◆ `Estudiante` tiene, adicionalmente:
      - Atributo `número de estudiante`.

# Ejercicio 2: Universidad (2/6)

## ■ Clase `Asignatura`:

- ♦ Atributos `nombre` y `código`.
- ♦ Atributo `estudiantes`: Lista, inicialmente vacía, con la relación de estudiantes matriculados en ella.
- ♦ Método `añade_estudiante`: Añade un estudiante matriculado a la lista de la asignatura.
- ♦ Método `imprime_listado`: Imprime la lista de estudiantes matriculados en una asignatura.

## Ejercicio 2: Universidad (3/6)

◆ Utilice *exclusivamente* los siguientes datos:

◆ Profesores:

Nombre	Edad	DNI	NReg
Luis	50	34567	5001
Pepe	37	65432	5010

◆ Estudiantes:

Nombre	Edad	DNI	NEst
Jorgito	20	56678	1001
Juanito	19	44444	1002
Jaimito	19	22334	1005

◆ Asignaturas:

Nombre	Código	Profesor
Matemáticas	5	Luis
Física	7	Luis
Latín	13	Pepe
Historia	19	Pepe
Filosofía	36	Pepe

# Ejercicio 2: Universidad (4/6)

- ◆ Relación de matrículas:

- ◆ Matemáticas:

- Jorgito

- ◆ Física:

- Juanito

- Jaimito

- ◆ Latín:

- Jorgito

- Jaimito

- ◆ Historia:

- Juanito

- Jaimito

- ◆ Filosofía:

- Jaimito

## Ejercicio 2: Universidad (5/6)

### ◆ Ejemplos de uso:

- Crear un objeto que corresponda al profesor Pepe:

```
pepe = Profesor("Pepe", 37, 65432, 5010)
```

- Crear un objeto que corresponda al estudiante Jaimito:

```
jaimito = Estudiante("Jaimito", 19, 22334, 1005)
```

- Crear un objeto que corresponda a la asignatura Latín:

```
latin = Asignatura("Latín", 13)
```

- Luis imparte Física:

```
luis.añade_docencia(fisica)
```

- Jaimito cursa Filosofía:

```
filosofia.añade_estudiante(jaimito)
```

- Estudiantes matriculados en Filosofía:

```
filosofia.imprime_listado()
```

- Asignaturas y estudiantes matriculados de Pepe:

```
pepe.print_docencia()
```



## Ejercicio 2: Universidad (6/6)

### ◆ **Realice** un programa que:

- Cree todos los objetos necesarios.
- Asigne los valores adecuados a sus atributos.
- Imprima en pantalla las asignaturas que imparte cada profesor junto con la relación de estudiantes matriculados en ellas.

◆ **Opcional:** Realice una *función* adicional (ojo, no le pedimos un método) que reciba como parámetro el nombre de un estudiante (por ejemplo "Jaimito", pero no el objeto `jaimito`) e imprima las asignaturas en que éste está matriculado. Haga llamadas a esta función para comprobar su funcionamiento. *Ayuda:* Para simplificar el código puede serle de utilidad crear una lista con todas las asignaturas (lista de objetos).

# Aviso



Python 3 by C. Mañoso, A. P. de Madrid, M. Romero is licensed under a [Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Esta colección de transparencias se distribuye con fines meramente docentes.

Todas las marcas comerciales y nombres propios de sistemas operativos, programas, hardware, etc. que aparecen en el texto son marcas registradas propiedad de sus respectivas compañías u organizaciones.