



VNIVERSITAT ID VALÈNCIA

**Laboratorio de ESTRUCTURAS DE
DATOS Y ALGORITMOS**
Grado en Ciencia de Datos (1º)
Curso 2024-25**Práctica Nº 3: Definición de clases en Python**

Periodo de realización: Semana del 10 a 14/03/2025

Metodología de trabajo

Fases	Tipo	Dedicación
1. Resolver las tareas planteadas en este guion <u>antes</u> de iniciar la sesión presencial en el laboratorio.	No presencial	Máx. 4,5 h.
2. Resolver en el laboratorio un <u>nuevo ejercicio</u> basado en la resolución de las tareas previas.	Presencial	Máx. 3 h.

Introducción

En esta práctica se debe definir completamente una clase que pueda ser utilizada en diversos programas. Antes de realizar la sesión de laboratorio se deberá implementar la clase y el programa que se indican en los siguientes ejercicios, de acuerdo con los requisitos establecidos. Durante la sesión de laboratorio este código será reutilizado para resolver un problema concreto.

Ejercicios (Fase 1)**Tarea 1**

Se debe construir una clase para representar atletas que pueden participar en competiciones (carreras) internacionales. Se debe definir un nuevo tipo de datos `Atleta` (clase de Python) que cumpla los siguientes requisitos (utiliza como ejemplos las clases que ya se ha proporcionado en la práctica 2 o las utilizadas en las clases de teoría):

Representación: Los datos del tipo `Atleta` están caracterizados por la siguiente información:

- **Identificador:** Un identificador alfanumérico que identifica al atleta.
- **Nombre:** Nombre y apellidos del atleta (un único valor).
- **País:** País al que se adscribe el atleta.
- **Especialidad:** Especialidad en la que participa el atleta (solo una).

Operaciones: Los datos de este tipo podrán realizar las siguientes operaciones:

- **Constructor:** Deberá construir un nuevo objeto con valores por defecto para sus atributos. Todos los atributos alfanuméricos tendrán el valor "<Sin asignar>" y todos los atributos numéricos tendrán el valor 0 (cero).
- **SetValores:** Esta operación permite establecer los valores de todos los atributos del objeto.
- **SetId:** Permite modificar exclusivamente el identificador del atleta.

- **SetNombre:** Permite modificar exclusivamente el nombre del atleta.
- **SetPais:** Permite modificar exclusivamente el país del atleta.
- **SetEspecialidad:** Permite modificar exclusivamente la especialidad en la que compete el atleta.
- **GetId:** Permite conocer el identificador del atleta.
- **GetNombre:** Permite conocer el nombre del atleta.
- **GetPais:** Permite conocer el país del atleta.
- **GetEspecialidad:** Permite conocer la especialidad del atleta.
- **str:** Permite obtener una versión en formato string (cadena de caracteres) de los atributos de una persona. El formato debe ser el siguiente:

id;nombre;pais;especialidad

Las operaciones que modifican (mutan) el estado de un atleta (todas las que empiezan por Set...) deben validar que los valores a modificar sean correctos. Para ello, se deben cumplir los siguientes requisitos:

- El país de adscripción debe corresponder a alguno de los siguientes códigos normalizados (hay 20): "ALE", "ARG", "AUS", "BEL", "BOL", "BRA", "CHI", "DIN", "ESP", "FRA", "GBR", "GRE", "HOL", "HON", "ITA", "MEX", "POR", "SUE", "SUI", "URU". Los códigos tienen la siguiente equivalencia real¹:

Código	País
ALE	Alemania
ARG	Argentina
AUS	Austria
BEL	Bélgica
BOL	Bolivia
BRA	Brasil
CHI	China
DIN	Dinamarca
ESP	España
FRA	Francia
GBR	Gran Bretaña
GRE	Grecia
HOL	Holanda
HON	Honduras
ITA	Italia
MEX	México
POR	Portugal
SUE	Suecia
SUI	Suiza
URU	Uruguay

¹ Esta información es simplemente complementaria, pues el programa debe manejar exclusivamente los códigos.

- La especialidad en la que compete un atleta debe corresponder a alguno de los siguientes códigos normalizados (hay 14): "100m(H)", "100m(M)", "200m(H)", "200m(M)", "400m(H)", "400m(M)", "800m(H)", "800m(M)", "1500m(H)", "1500m(M)", "5000m(H)", "5000m(M)", "10000m(H)", "10000m(M)". Cada código indica la distancia de la carrera y si los competidores son hombres (H) o mujeres (M).

Muy importante: Las operaciones SetValores, SetPais y SetEspecialidad no deben modificar ningún dato del objeto si cualquiera de los valores de entrada es incorrecto. Todas las operaciones que modifican valores (Set...) devolverán un valor lógico (True/False) que indique si la modificación solicitada se ha podido realizar correctamente o si, por el contrario, ha sido rechazada.

Para facilitar los procesos de validación de datos se deben definir en la clase dos operaciones/métodos auxiliares y privados, que realicen cada una de las comprobaciones requeridas:

```
def __ValidarPais(p:str)->bool
def __ValidarEspecialidad(e:str)->bool
```

El argumento de cada método será el valor que se debe verificar y el resultado indicará si ese valor cumple el requisito establecido en cada caso (*True*) o no (*False*).

Esta clase se debe escribir en un archivo llamado "atleta.py" que pueda ser importado desde otros programas.

Tarea 2

Escribe un programa para probar la clase realizada. Este programa, llamado "pr3_v1.py" debe importar la clase Atleta creada previamente y leer los datos del archivo "atletas_small.csv", guardarlos en una lista solo aquellos atletas cuya información sea correcta, de acuerdo con los requisitos de validación indicados, y mostrar el número de atletas almacenados y la información de cada uno de ellos.

La salida esperada del programa para este archivo es:

```
* Número de atletas correctamente leídos = 10

* Datos de los atletas:
J20853;LOPETEGUI POCH, AATIKA;POR;400m(M)
KB032274;GARCIA PLIEGO ESTALL, ABDELKABIR;FRA;400m(H)
ZH650632;VARONA RISUENO, ABDELLATIF;GBR;100m(H)
Y68959;ORDONO BROCAL, ABDENAJI;ESP;1500m(H)
ZR232501;URGELLES OLIVER, ABDERRAHAMAN;BRA;400m(H)
HA541246;MECERREYES PRICE, ABDOU KARIM;CHI;100m(H)
US314164;ROLLE BESCOS, ABDOU LAHAT;ESP;10000m(H)
WN693675;GOENAGA BENZO, ABENCIO;ALE;800m(H)
LZ109053;POBES OULGHAZI, ABILIO;ESP;10000m(H)
XK081259;CABESTANY KHALID, ABRAHAM JESUS;ESP;100m(H)
```

Si pruebas el programa con el archivo "atletas_big.csv", el número de atletas correctamente leídos deberá ser 300.

Fase 2: Tarea final

Resolución de un nuevo ejercicio planteado durante la sesión presencial en el laboratorio.