

## Trabajo práctico N°1

### Creación de nuestras primeras clases

#### EJERCICIO 1

Necesitamos gestionar los jugadores de básquet en una liga y los equipos que la integran.

##### Clase **JUGADOR**

De los jugadores necesitamos saber el dni, nombre completo, fecha de nacimiento, altura, equipo y puntos totales.

##### Clase **EQUIPO**

De los equipos debemos saber el

- nombre
- ciudad
- partidos jugados
- partidos ganados.

Definir un constructor que inicialice los atributos

Analizar los datos indicados en los archivos adjuntos

- 2025-TR-jugadores.csv
- 2025-TR-equipos.csv

y crear una función global para cada uno de estos casos:

- **jugadoresMasAltos()**: Obtener una lista con el nombre del jugador o jugadores más altos (en caso de haber empate).
- **equiposMasGanadores()**: Obtener una lista con el nombre del equipo o equipos con más partidos ganados.
- **jugadoresConMasPuntos()**: Obtener una lista con al menos 10 de los jugadores con más puntos.
- **alturaPromedio()**: Obtener una lista con la altura promedio de

jugadores por de cada equipo.

- (Complicado) `jugadoresMasPuntosPeorEquipo()`: Obtener una lista del jugador o jugadores con más puntos en el o los equipos con más partidos perdidos.

## EJERCICIO 2

### PasswordAnalyzer

Crear una clase llamada PasswordAnalyzer que siga las siguientes condiciones:  
Que tenga los atributos

- `números`
- `mayúsculas`
- `longitudMinima`.

Por defecto, `números` será 2, `mayúsculas` será 2 y `longitudMinima` será 8.

Los métodos que implementados serán:

- `esClaveFuerte( clave )`: que analizará la clave dada y devolverá un booleano indicando si la clave es fuerte o no. Una clave es fuerte si sigue un criterio que utilice los atributos internos de la clase.
- `generarClave()`: genera una clave aleatoria que satisfaga el criterio de la función previa.

Para inicializar por defecto los valores en un objeto podemos definirlos en el constructor

```
1 class Estudiante:
2     def __init__(self,
3                 nombre:str = 'nombre por defecto',
4                 notas:list = [0,0,0]):
5         self.nombre= nombre
6         self.notas = notas
7
```

Obtener la lista de claves débiles del archivo adjunto `claves.txt` utilizando

un analizador configurado en números 3 mayúsculas 1 y longitud mínima 6.  
Obtener una segunda lista con un analizador configurado en números 4  
mayúsculas 3 y longitud mínima 12.  
Se debe crear una función llamada `analizarClaves()`.