

1. Diseñar una aplicación Python que permita gestionar partidos asociados a diferentes deportes. Todos los partidos deben tener como atributos los siguientes: nombres de los dos contrincantes, fecha y competición. Además deben desarrollarse métodos para:
 - a. Retornar el ganador del partido.
 - b. Retornar el resultado.
 - c. Retornar una representación del partido con el formato "**XXX Vs. YYY - fecha - competición**".

Se debe tener en cuenta que los partidos pueden ser:

- De fútbol.
 - Deben guardarse los goles de los equipos participantes.
 - Puede darse el empate como resultado.
 - El resultado debe tener el formato: **Zafra 4 - 5 Badajoz**.
 - Lanzar una excepción y no crear el objeto en caso de insertar algún valor negativo en los goles. **Una excepción se lanza mediante `raise ValueError("Mensaje")`**. No es necesario capturar esta excepción, veremos las excepciones en la unidad 3.
- De baloncesto.
 - Deben almacenarse los puntos de cada equipo.
 - No puede darse el empate. En caso de empate o de puntuación negativa, lanzar una excepción y no crear el objeto.
 - El resultado debe tener el formato: **90 Cáceres - Badajoz 88**.
- De tenis.
 - Se necesita pasar dos listas, con el número de juegos ganados por cada jugador en cada set. El constructor debe tener este formato:

PartidoTenis('Rafa Nadal', 'Roger Federer', date(2020, 6, 2), 'Roland Garros', [6, 5, 7, 3, 7], [3, 7, 6, 6, 6]).

- El número de sets debe ser 3 o 5, y además debe ser el mismo para ambos jugadores. En cualquier otro caso generar una excepción al crear el objeto.
- Un resultado de un partido de tenis puede ser el siguiente:

Rafa Nadal	6	5	7	3	7
Roger Federer	3	7	6	6	6
- El ganador sería Nadal ya que ha ganado tres sets.
- **No podrán crearse instancias de partido y los métodos ganador, resultado y representación del partido serán de obligatoria implementación.**
- Pruebas:
 - Crear instancias de partidos de todas las clases y probar todos sus métodos.

2. En este ejercicio trabajaremos con la clase "Cadena" que tendrá un atributo de instancia que guardará una cadena de texto. Deberás programar en esta clase métodos para:
- a. **Sumar objetos Cadena.** Se obtendrá un objeto **str** cuyo contenido será el resultado de concatenar las cadenas de los objetos Cadena con los que se opera. Estudia el ejemplo:

```
# Sumar objetos Cadena
cadena1 = Cadena("Me gusta programar en ")
cadena2 = Cadena("Python")
print(cadena1 + cadena2)
```

✓ 0.0s

Me gusta programar en Python

- b. **Restar objetos Cadena.** Al restar dos objetos Cadena, las vocales de la segunda cadena se reemplazarán en la primera cadena por cadena vacía. **Por tanto, el primer objeto Cadena ve modificado el valor de su atributo.** Estudia el ejemplo:

```
# Restar objetos Cadena
cadena1 = Cadena("El temor de un hombre sabio")
cadena2 = Cadena("tormenta")
print(cadena1 - cadena2)
print(cadena1)
```

✓ 0.0s

El tmr d un hmbr sbi
El tmr d un hmbr sbi

- c. **Longitud de un objeto Cadena.** La longitud de una instancia de Cadena es el número de vocales que tiene, incluidas mayúsculas y minúsculas con o sin tilde.

```
# Longitud de una instancia de Cadena
cadena1 = Cadena("Leí 'La Rueda del Tiempo'")
print(len(cadena1))
```

✓ 0.0s

10

- d. **Compara objetos Cadena.** Dos objetos Cadena son iguales si tienen la misma longitud, es decir, el mismo número de vocales.

```
# Comparar objetos Cadena
cadena1 = Cadena("Python")
cadena2 = Cadena("Cobol")
cadena3 = Cadena("Java")
print(cadena1 == cadena2)
print(cadena2 == cadena3)
```

✓ 0.0s

False

True

- e. **Operador +=.** Al realizar la operación **cadena += valor_str**, se añadirá al atributo del objeto Cadena la cadena valor_str. El objeto original se verá modificado.

```
# Añadir valor str
cadena = Cadena("Sayonara ")
cadena += "baby"
print(cadena)
```

✓ 0.0s

Sayonara baby

- Pruebas:
 - Crea varios objetos Cadena y prueba los métodos creados.

Forma de entrega.

- Crea una carpeta por cada ejercicio, e incluye en ella los archivos para las clases y las pruebas.
- Crea un archivo comprimido que incluya los dos ejercicios y dale el nombre: **Apellido_nombre_POO_Tarea2.zip**.

Importante:

- En la resolución de los ejercicios de la tarea se deben utilizar las sentencias, estructuras y formas de trabajo vistas en clase.

Criterios de calificación.

- Ejercicio 1. 5 puntos.
 - Diseño de la clase, 3 puntos.
 - Pruebas, 2 puntos.
- Ejercicio 2. 5 puntos.
 - Diseño de la clase 4 puntos.
 - Pruebas, 1 punto.