



Street Fighter es un videojuego de lucha de uno contra uno desarrollado y publicado por Capcom. La compañía se dispone a lanzar Street Fighter 6, y necesita conocer los gustos de sus jugadores y extraer algunas conclusiones basadas en datos generados en su anterior entrega. Los datos que maneja Capcom tienen la siguiente estructura e información:

- **pj1** (str): nombre del personaje controlado por el jugador.
- **pj2** (str): nombre del personaje controlado por el oponente.
- **puntuacion** (int): puntuación obtenida en la partida.
- **tiempo** (float): tiempo total de duración de la partida en segundos.
- **fecha_hora** (datetime): fecha y hora en la que se jugó la partida.
- **golpes_pj1** (list): lista de los movimientos y ataques ejecutados por el personaje controlado por el jugador.
- **golpes_pj2** (list): lista de los movimientos y ataques ejecutados por el personaje controlado por el oponente.
- **movimiento_final** (str): movimiento especial final utilizado para terminar la partida. No tiene por qué estar incluido dentro de la lista de movimientos.
- **combo_finish** (bool): valor que indica si el combate ha terminado con un ultra combo finish (1 = Sí, 0 = No). Un ultra combo finish sucede cuando un combate acaba con un movimiento especial.
- **ganador** (str): nombre del personaje que ha ganado la partida.

La primera línea del fichero tiene el siguiente aspecto:

```
"Ryu", "Chun-Li", 3500, 120.5, "2023-06-05 14:30:00", ["Hadouken", "Shoryuken", "Tatsumaki Senpukyaku"], ["Spinning Bird Kick", "Lightning Legs"], "Shinku Hadoken", 1, "Ryu"
```

e indica que el 5 de junio de 2023 a las 14:30:00 hubo un combate entre Ryu y Chun-Li, que tuvo una puntuación de 3500 puntos y duró 120.5 segundos. Los movimientos de Ryu fueron "Hadouken", "Shoryuken" y "Tatsumaki Senpukyaku", mientras que los de Chun-Li fueron "Spinning Bird Kick" y "Lightning Legs". El combate lo ganó Ryu con un ultra combo finish con el movimiento especial "Shinku Hadoken".

Para la realización de los ejercicios se usará la siguiente definición de namedtuple, y su uso será **obligatorio**:

```
Partida = namedtuple("Partida", "pj1, pj2, puntuacion, tiempo, fecha_hora, golpes_pj1, golpes_pj2, movimiento_final, combo_finish, ganador")
```

Como ejemplo, para la partida anterior obtendremos la siguiente tupla. Fíjese con detalle en el tipo de cada uno de los campos:

```
'Ryu', pj2= 'Chun-Li', puntuacion= 3500, tiempo= 120.5, fecha_hora= datetime.datetime(2023, 6, 5, 14, 30), golpes_pj1= ['Hadouken', 'Shoryuken', 'Tatsumaki Senpukyaku'], golpes_pj2= ['Spinning Bird Kick', 'Lightning Legs'], movimiento_final= 'Shinku Hadoken', combo_finish= True, ganador= 'Ryu')
```

Implemente las siguientes funciones en un módulo partidas.py:

1. **lee_partidas**: recibe una cadena de texto con la ruta de un fichero csv, y devuelve una lista de tuplas Partida con la información contenida en el fichero. **(1 punto)**

2. **victoria_mas_rapida**: recibe una lista de tuplas de tipo Partida, y devuelve una tupla compuesta por los dos personajes y el tiempo de aquella partida que haya sido la más rápida en acabar. Implemente este ejercicio usando bucles y sin usar los métodos min o max de Python. **(1 punto)**
3. **top_ratio_media_personajes**: recibe una lista de tuplas de tipo Partida y un número entero n , y devuelve una lista con los n nombres de los personajes cuyas ratios de eficacia media sean las más bajas. La ratio de eficacia se calcula dividiendo la puntuación entre el tiempo de aquellas partidas que haya ganado el personaje. Es decir, si *Ryu* ha ganado 3 combates, su ratio media se calcula con los cocientes $puntuacion/tiempo$ de dichos combates. **(1.5 puntos)**
4. **enemigos_mas_debiles**: recibe una lista de tuplas de tipo Partida y una cadena de texto *personaje*. El objetivo de esta función es calcular los oponentes frente a los que el personaje ha ganado más veces. Para ello, esta función devuelve una tupla compuesta por una lista de nombres y el número de victorias, de aquellos contrincantes contra los cuales el número de victorias haya sido el mayor. Es decir, si introducimos como parámetro el valor *Ken* y este ha ganado 2 veces contra *Blanka*, 2 contra *Ryu* y 1 contra *Bison*, la función deberá devolver (`['Blanka', 'Ryu'], 2`). **(2 puntos)**
5. **movimientos_comunes**: recibe una lista de tuplas de tipo Partida y dos cadenas de texto *personaje1* y *personaje2*, y devuelve una lista con los nombres de aquellos movimientos que se repitan entre *personaje1* y *personaje2*. Tenga solo en cuenta los movimientos que aparecen listados en los campos *golpes_pj1* y *golpes_pj2*. **(2 puntos)**
6. **dia_mas_combo_finish**: recibe una lista de tuplas de tipo Partida, y devuelve el día de la semana en el que hayan acabado más partidas con un combo finish. El tipo datetime tiene una función `isoweekday()` que devuelve el día de la semana en formato numérico, siendo 1 el lunes y 7 el domingo. Para hacer la traducción del número al nombre, utilice una función auxiliar. **(1.5 puntos)**
7. Pruebe las funciones implementadas en un módulo `partidas_test.py`. Se recomienda que lo vaya haciendo a medida que vaya resolviendo los distintos apartados. **(1 punto)**

Anexo: Pruebas de las funciones

En esta sección, se muestra una posible ejecución de las pruebas de las funciones.

1. Test de lee_partidas:

Total de registros leídos: 49

Mostrando los tres primeros registros:

```
Partida(pj1='Ryu', pj2='Chun-Li', puntuacion=3500, tiempo=120.5,
fecha_hora=datetime.datetime(2023, 6, 5, 14, 30), golpes_pj1=['Hadouken', 'Shoryuken',
'Tatsumaki Senpukyaku'], golpes_pj2=['Spinning Bird Kick', 'Lightning Legs'],
movimiento_final='Shinku Hadoken', combo_finish=True, ganador='Ryu')
```

```
Partida(pj1='Guile', pj2='Ken', puntuacion=4200, tiempo=90.2,
fecha_hora=datetime.datetime(2023, 6, 5, 15, 45), golpes_pj1=['Sonic Boom', 'Flash
Kick'], golpes_pj2=['Shoryuken', 'Hadouken'], movimiento_final='Sonic Boom',
combo_finish=False, ganador='Guile')
```

```
Partida(pj1='Blanka', pj2='Dhalsim', puntuacion=2900, tiempo=150.7,
fecha_hora=datetime.datetime(2023, 6, 5, 16, 20), golpes_pj1=['Electric Thunder',
'Rolling Attack'], golpes_pj2=['Yoga Fire', 'Yoga Flame'], movimiento_final='Electric
Thunder', combo_finish=False, ganador='Blanka')
```

2. Test de victoria_mas_rapida

La partida más rápida fue una entre Bison y Guile que duró 70.3 segundos.

3. Test de top_ratio_media_personajes

El top 3 de ratios medias es: ['Zangief', 'Ryu', 'Ken']

4. Test de enemigos_mas_debiles

Los enemigos más débiles de Ken son (['Blanka', 'Ryu'], 2)

5. Test de movimientos_comunes

Los movimientos repetidos entre Ryu y Ken son: ['Shoryuken', 'Hadouken']

6. Test de dia_mas_combo_finish

El día que más Ultra Combo Finish ha habido es el Domingo