



Se dispone de datos personales y deportivos sobre jugadores de golf. Dichos datos se facilitan en un archivo "**jugadores.csv**" con la siguiente información sobre cada uno de ellos:

- **apellidos y nombre:** apellidos y nombre (en un único campo), de tipo str.
- **licencia:** número (aunque también contiene letras) de licencia federativa, de tipo de tipo str.
- **fecha de nacimiento:** fecha en que nació el jugador, de tipo date.
- **federación:** nombre de la federación a la que pertenece el jugador, de tipo str.
- **handicap:** es un número que refleja la capacidad y potencial de cada jugador de golf. En el dataset están recogidos hándicaps entre -5,0 y +36,0. Cuanto menor hándicap, mejor jugador, de tipo float.
- **fecha y hora de última actualización:** la fecha y la hora en que se actualizaron los datos del jugador por última vez, de tipo datetime.
- **senior:** indica si el jugador es de categoría senior. En el dataset se identifica por S (*True*) o N (*False*). Es de tipo bool.
- **últimos resultados:** lista de número enteros que contiene los últimos resultados de cada jugador (el número de golpes realizados en los últimos torneos que ha jugado, cuantos menos golpes, mejor resultado). La lista contiene elementos de tipo int.

Por ejemplo, las siguientes líneas del fichero:

```
Rubio Duarte, Diana;AM3824387;06/03/1982;ANDALUZA;28.9;16/07/2020 17:19:33;S;94,97,91,92,96  
Bravo Rodríguez, Isabel A.;AK6247463;20/12/1983;MADRILEÑA;11.2;22/11/2020 12:12:55;N;99,82
```

indican que la jugadora Rubio Duarte, Diana, tiene número de licencia AM3824387, que nació el 6 de marzo de 1982, pertenece a la federación andaluza de golf, tiene hándicap 28,9, sus datos se actualizaron por última vez a las 17:19:33 horas del 16 de julio de 2020, es jugadora senior y se han registrado 5 últimos resultados con 94, 97, 91, 92 y 96 golpes, respectivamente.

La jugadora Bravo Rodríguez, Isabel A, tiene número de licencia AK6247463, nació el 20 de diciembre de 1983, pertenece a la federación madrileña de golf, tiene hándicap 11,2, sus datos se actualizaron por última vez a las 12:12:55 horas del 22 de noviembre de 2020, no es una jugadora senior y se han registrado 2 últimos resultados con 99 y 82 golpes, respectivamente.

Para almacenar los datos de un jugador se **usará obligatoriamente** la siguiente namedtuple:

```
Jugador = namedtuple('Jugador', 'ape_nom, licencia, fecha_ncto, federacion, handicap, fec_hor_act, senior, resultados')
```

Cree un módulo **jugadores.py** e implemente en él las funciones que se piden. También debe crear un módulo **jugadores_test.py** en el que defina una función de test para cada función solicitada **IMPORTANTE: Los test sólo puntúan si se ha implementado razonablemente el método que lo prueba**. Puede definir funciones auxiliares cuando lo considere necesario:

1. **lee_jugadores:** recibe una cadena de caracteres con el nombre de un fichero en formato CSV codificado en UTF-8, y devuelve una lista de tuplas de tipo Jugador conteniendo todos los datos almacenados en el fichero. Nota1: Utilice la función `datetime.strptime` con los siguientes formatos `"%d/%m/%Y"` o `"%d/%m/%Y %H:%M:%S"` según el campo de que se trate. Nota2: Para "parsear" la lista de resultados utilice el método `split(",")`, que trocea la cadena que invoca al método, devolviendo una lista de elementos str ya separados. (1 punto + 0,25 puntos el test).

Resultados esperados:

Registros leídos: 85

Los dos primeros:

```
Jugador(ape_nom='Rubio Duarte, Diana', licencia='AM3824387', fecha_ncto=datetime.date(1982, 3, 6), federacion='ANDALUZA', handicap=28.9, fec_hor_act=datetime.datetime(2020, 7, 16, 17, 19, 33), senior=True, resultados=[94, 97, 91, 92, 96])
```

```
Jugador(ape_nom='Bravo Rodríguez, Isabel A.', licencia='AK6247463', fecha_ncto=datetime.date(1983, 12, 20), federacion='MADRILEÑA', handicap=11.2, fec_hor_act=datetime.datetime(2020, 11, 22, 12, 12, 55), senior=False, resultados=[99, 82])
```

Los dos últimos:

```
Jugador(ape_nom='Cruz Sevillano, Javier Jesús', licencia='AN8289428', fecha_ncto=datetime.date(1964, 8, 20), federacion='CATALANA', handicap=28.9, fec_hor_act=datetime.datetime(2020, 8, 20, 9, 14, 57), senior=True, resultados=[96, 76, 67])
```

```
Jugador(ape_nom='Álvarez Mayo, David', licencia='AM5705532', fecha_ncto=datetime.date(1979, 7, 15), federacion='ANDALUZA', handicap=11.8, fec_hor_act=datetime.datetime(2022, 2, 15, 8, 37, 44), senior=False, resultados=[82, 93, 76])
```

2. **mejores_jugadores**: recibe una lista de tuplas de tipo Jugador, un año y un número entero n, y devuelve una lista de tuplas con los números de licencia, apellidos y nombre y handicap de los n jugadores más buenos que nacieron en dicho año. (1 punto + 0,25 puntos el test).

Resultados esperados para 1969 y n=4:

Los 4 mejores jugadores nacidos en el 1969 son: [('BB7714568', 'Díaz Chamorro, Amador', 1.4), ('AK3167484', 'Sánchez Valenzuela, Alfonso', 3.4), ('AM8942631', 'Ortega Ortega, Antonio', 21.1), ('AN5777232', 'Alonso Jiménez, Cristina', 24.1)]

3. **jugadores_por_golpes**: recibe una lista de tuplas de tipo Jugador y devuelve una lista de tuplas con el número de golpes y asociado a dicho número un conjunto con los números de licencia de los jugadores que tienen ese número de golpes entre sus últimos resultados. La lista debe estar ordenada de mayor a menor número de golpes. (1,25 puntos + 0,25 por el test)

Resultados esperados para los dos primeros y los dos últimos elementos de la lista:

Jugadores por golpes
(100, {'CA4656346', 'BB8295828', 'AK8406564', 'AM6906798', 'AK7077045'})
(99, {'BA7219579', 'BB7217998', 'AK6247463', 'BB7714568', 'AK7148346', 'BA9695506', 'BJ4128343', 'BA2863610', 'AN3248377', 'CA2259148'})
...
(66, {'BA8656908', 'AK3167484', 'AN2885686', 'BB7714568', 'BA2003146', 'AM8942631', 'AN6285953', 'BA9984796'})
(65, {'BJ3221814', 'AK3167484', 'AK8040755', 'CA4617255', 'BA9984796', 'AN9956600'})

4. **promedio_ultimos_resultados**: recibe una lista de tuplas de tipo Jugador y dos valores de tipo date f1 y f2, que por defecto pueden tomar el valor None, y devuelve una lista de tuplas con el número de licencia y el promedio de golpes de los últimos resultados de los jugadores de categoría senior (True), cuya fecha de actualización esté entre f1 y f2 (ambas inclusive). Si f1 es None, se tendrán en cuenta los jugadores con fecha de actualización anteriores a f2 (inclusive). Si f2 es None, se tendrán en cuenta los posteriores a f1 (inclusive). Si ambas son None, se tendrán en cuenta los registros de todos los jugadores. (1,5 puntos + 0,25 puntos el test)

Resultados esperados para las fechas f1=1 de marzo de 2020 y f2=31 de mayo de 2020:

El promedio de cada jugador senior con fecha de actualización entre 2020-03-01 y 2020-05-31 son:
(('AN7802072', 87.0), ('BJ5650943', 80.66666666666667), ('CA8729905', 79.33333333333333))

5. **jugador_menor_handicap_por_federacion**: recibe una lista de tuplas de tipo Jugador, y devuelve un diccionario que a cada federación le haga corresponder una tupla con el nombre y apellidos y el hándicap del mejor jugador de dicha federación. (1,5 puntos + 0.25 puntos el test).

Resultados esperados:

Los mejores jugadores de cada federación son:
(('ANDALUZA', ('Ortega Gutiérrez, Antonia', -1.1))
(('MADRILEÑA', ('Ruiz Rodríguez, Beatriz', 1.8))
(('CATALANA', ('Ayala Chamorro, Ana Belén', 3.6))
(('CANARIA', ('Parejo López, Jorge', -2.2))
(('GALLEGA', ('Riquelme Bonilla, Julián Alberto', 8.7))
(('MURCIANA', ('Roldán Aleta, Pablo', -4.3))
(('VALENCIANA', ('Escalona Díaz, Juan Manuel', -4.3))

6. **comparativa_de_mejores_resultados_segun_handicap**: recibe una lista de tuplas de tipo Jugador y devuelve una lista de tuplas (hándicaps, diferencia), donde hándicaps es una cadena que representa los dos hándicaps que se comparan y diferencia es un real, que se calcula como la diferencia de los promedios de los mejores resultados de todos los jugadores que tienen esos hándicaps. Para resolver este ejercicio descomponga el problema en dos problemas, definiendo una función auxiliar para cada uno de ellos:

- a) Defina una función auxiliar que construya un diccionario donde las claves sean los hándicaps y los valores del diccionario, el promedio de los mejores resultados de cada uno de los jugadores que tienen ese hándicap. Por ejemplo, si solo hay dos jugadores con hándicap 28.9

```
Jugador(ape_nom='Rubio Duarte, Diana', licencia='AM3824387', fecha_ncto=datetime.date(1982, 3, 6), federacion='ANDALUZA', handicap=28.9, fec_hor_act=datetime.datetime(2020, 7, 16, 17, 19, 33), senior=True, resultados=[94, 97, 91, 92, 96])
```

```
Jugador(ape_nom='Cruz Sevillano, Javier Jesús', licencia='AN8289428', fecha_ncto=datetime.date(1964, 8, 20), federacion='CATALANA', handicap=28.9, fec_hor_act=datetime.datetime(2020, 8, 20, 9, 14, 57), senior=True, resultados=[96, 76, 67])
```

Entonces en el diccionario la clave 28.9 tendría asociado el valor 79 (media de 91 y 67). Es decir, 79 es el promedio de los mejores resultados de los jugadores de ese hándicap.

- b) Defina otra función auxiliar que a partir del diccionario anterior devuelva una lista con las diferencias entre cada valor promedio y el anterior. Por ejemplo, si para el hándicap 29.0 el valor promedio es 75, en la lista de diferencias habría un valor ("28.9 vs 29.0", -4)

Nota: cada hándicap debe ser comparado con el inmediatamente siguiente en su orden natural. (2,25 puntos + 0,25 puntos el test).

Los primeros y últimos resultados esperados:

```
[('4.3 vs 3.1', -3.5), ('3.1 vs 2.2', 11.0), ('2.2 vs 1.5', -13.0), ('1.5 vs 1.1', 10.0), ('1.1 vs 0.5', 2.0), ('0.5 vs 1.3', -19.0), ('1.3 vs 1.4', 24.0), ('1.4 vs 1.7', -5.0), ('1.7 vs 1.8', -1.0), ('1.8 vs 2.5', -4.0), ('2.5 vs 3.4', 11.0), ('3.4 vs 3.6', 0.0), ('3.6 vs 3.9', -19.0), ('3.9 vs 5.4', 16.0), ('5.4 vs 5.7', 2.0), ('5.7 vs 6.0', -8.0), ('6.0 vs 6.4', 4.0), ('6.4 vs 6.6', 1.0), ('6.6 vs 7.9', -10.0), ('7.9 vs 8.7', 9.0), ('8.7 vs 9.1', 3.0), ('9.1 vs 9.6', -9.0),
.....
('28.7 vs 28.9', -2.66666666666666714), ('28.9 vs 29.0', -10.333333333333329), ('29.0 vs 29.9', 19.0), ('29.9 vs 30.3', 3.0), ('30.3 vs 30.9', -17.0), ('30.9 vs 31.1', 15.0), ('31.1 vs 31.3', -7.0), ('31.3 vs 31.5', -6.0), ('31.5 vs 33.3', 11.0), ('33.3 vs 33.9', -1.0), ('33.9 vs 34.1', 0.0), ('34.1 vs 34.3', 1.0), ('34.3 vs 34.8', 1.0), ('34.8 vs 35.1', 2.0), ('35.1 vs 35.6', -3.0)]
```