

Ejercicio *Compras* (de un examen del curso 2020/21)

Se quieren analizar los datos de las compras en los supermercados más relevantes de cada provincia de Andalucía. Para ello se dispone de un archivo en formato CSV codificado en UTF-8. En cada línea del archivo se recoge la siguiente información: el dni del cliente (de tipo string); el nombre de la marca del supermercado (de tipo string); el nombre de la provincia (de tipo string); la fecha y hora en la que llega el cliente (de tipo datetime); la fecha y hora en la que sale el cliente (de tipo datetime); y el importe de la compra (de tipo float). Las primeras líneas son las que se muestran a continuación:

```
dni,supermercado,provincia,fecha_llegada,fecha_salida,total_compra
62745110Y,Carrefour,Almeria,01/01/2019 04:43,01/01/2019 06:19,175.71
94320158X,Aldi,Malaga,01/01/2019 14:53,01/01/2019 15:55,70.04
55993608Q,Lidl,Sevilla,01/01/2019 20:09,01/01/2019 20:19,43.09
...
```

El objetivo del ejercicio es leer estos datos, realizar distintas operaciones con ellos e implementar los test que permitan probarlos. Cada operación se implementará en una función distinta. Se pide implementar las siguientes funciones y sus test correspondientes, teniendo en cuenta que se pueden definir funciones auxiliares cuando se considere necesario:

1. **lee_compras**: recibe el nombre de un fichero y devuelve una lista de tuplas de tipo `Compra` conteniendo todos los datos almacenados en el fichero. (1 punto)
2. **compra_maxima_minima_provincia**: recibe una lista de tuplas de tipo `Compra` y una provincia. Devuelve una tupla que contiene el importe máximo y el mínimo de las compras que se han realizado en la provincia dada como parámetro. Si la provincia toma el valor `None`, se devuelve una tupla con el importe máximo y el mínimo calculados a partir de todas las compras. (1 punto)
3. **hora_menos_afluencia**: recibe una lista de tuplas de tipo `Compra` y devuelve una tupla con la hora en la que llegan menos clientes y el número de clientes que llegan a dicha hora. (1,5 puntos)
4. **supermercados_mas_facturacion**: recibe una lista de tuplas de tipo `Compra` y un número entero `n`, con valor por defecto 3. Devuelve un ranking, es decir, una lista de tuplas (`posición_ranking`, (`supermercado`, `facturación`)) con las `n` marcas de supermercados que más facturan, en orden decreciente de facturación. El ranking debe empezar por la posición 1. (1,5 puntos)
5. **clientes_itinerantes**: recibe una lista de tuplas de tipo `Compra` y un entero `n`, y devuelve una lista de tuplas con el dni del cliente y la lista de provincias donde el cliente ha realizado sus compras, ordenadas alfabéticamente. Solo se devolverán aquellos clientes que hayan comprado en un número de provincias mayor que el parámetro `n`. (2 puntos)

6. **dias_estrella**: recibe una lista de tuplas de tipo `Compra`, un supermercado y una provincia, y devuelve una lista ordenada cronológicamente con las "fechas estrella" de ese supermercado en esa provincia. Se consideran "fechas estrella" aquellos días en los que el supermercado factura más que el día anterior y más que el día siguiente. (2 puntos)
7. Complete el código de las funciones de test¹ del módulo `compras_test.py` para probar las funciones anteriores, teniendo en cuenta que la salida por consola de la ejecución de los test debe ser la siguiente. (1 punto)

EJERCICIO 1

Número de registros leídos: 2414

Tres primeros registros: [Compra(dni='62745110Y', supermercado='Carrefour', provincia='Almeria', fecha_llegada=datetime.datetime(2019, 1, 1, 4, 43), fecha_salida=datetime.datetime(2019, 1, 1, 6, 19), total_compra=175.71), Compra(dni='94320158X', supermercado='Aldi', provincia='Malaga', fecha_llegada=datetime.datetime(2019, 1, 1, 14, 53), fecha_salida=datetime.datetime(2019, 1, 1, 15, 55), total_compra=70.04), Compra(dni='55993608Q', supermercado='Lidl', provincia='Sevilla', fecha_llegada=datetime.datetime(2019, 1, 1, 20, 9), fecha_salida=datetime.datetime(2019, 1, 1, 20, 19), total_compra=43.09)]

Tres últimos registros: [Compra(dni='89803474C', supermercado='Dia', provincia='Malaga', fecha_llegada=datetime.datetime(2019, 12, 30, 21, 8), fecha_salida=datetime.datetime(2019, 12, 30, 21, 26), total_compra=41.58), Compra(dni='06286245T', supermercado='Aldi', provincia='Sevilla', fecha_llegada=datetime.datetime(2019, 12, 30, 21, 50), fecha_salida=datetime.datetime(2019, 12, 30, 22, 30), total_compra=180.92)]

EJERCICIO 2

Importe máximo de la provincia de Huelva: 199.92. Importe mínimo: 1.28

EJERCICIO 3

La hora con menos afluencia es: 21 h. con 83 llegadas de clientes

EJERCICIO 4

Los 2 supermercados con más facturación son: [(1, ('Dia', 37343.600000000001)), (2, ('Mas', 36564.8999999999965))]

EJERCICIO 5

Los clientes itinerantes que han comprado al menos en 7 provincias son: [('90225569L', ['Almeria', 'Cadiz', 'Cordoba', 'Granada', 'Huelva', 'Jaen', 'Malaga', 'Sevilla'])]

EJERCICIO 6

Los días estrella del supermercado Aldi de la provincia de Huelva son: ['04/01/2019', '18/02/2019', '10/03/2019', '20/03/2019', '07/04/2019', '27/05/2019', '31/05/2019', '16/07/2019', '04/08/2019', '07/10/2019', '13/11/2019', '02/12/2019']

¹ Cuando vayas programando debes hacer la función de testeo a la vez que la función que estás programando. Es la única forma de programar bien pues, en caso contrario, es muy sencillo que te equivoques en algún detalle. De hecho, en mi caso, siempre escribo la función de testeo antes incluso que el cuerpo – el contenido – de la función que voy a programar.