

Fundamentos de Programación

Curso 2024-25 - Primer examen parcial, 15 de enero de 2025

Viviendas

Se dispone de determinados datos sobre las viviendas de una urbanización en un archivo CSV codificado en UTF-8. Cada línea del archivo incluye la siguiente información sobre una vivienda: el **propietario**, el **nombre de la calle**, el **número**, la **fecha de adquisición**, los **metros cuadrados construidos**, el **precio de compra** (en euros) y una **lista de mejoras** que se han realizado en las viviendas tras su adquisición.

Cada *mejora* contiene la siguiente información: la **denominación**, el **coste** (en euros) y la **fecha** en que se realizó. Hay viviendas en las que no se han realizado mejora en cuyo caso la lista está vacía (*vea la línea 9 del fichero*).

Los datos están almacenados en el archivo *urbanizacion.csv* cuyas cuatro primeras líneas se muestran a continuación:

```
APELLIDOS Y NOMBRE;CALLE;NÚMERO;FECHA DE ADQUISICIÓN;METROS;PRECIO;MEJORAS
AGUILAR CABALLERO, CARLOS;HALCÓN;30;10/10/2018;138,29;395488;BUHARDILLA-13379-30/11/2019*PLACAS
SOLARES-13502-04/04/2023*PISCINA-8945-29/10/2020*GARAJE-13177-20/01/2023
ALONSO GARCÍA, AGUSTÍN;PALOMA;5;01/03/2013;190,69;395379;PISCINA-10277-11/09/2014
ALONSO NAVARRO, BEATRIZ;PALOMA;11;17/09/2005;156,62;437807;GARAJE-11783-04/03/2007
```

La primera línea es una línea de cabecera. Se puede observar en la segunda línea del archivo CSV que **Aguilar Caballero, Carlos**, es el propietario de una vivienda en la calle **Halcón** número **30**, que adquirió el **10 de octubre de 2018**. La vivienda tiene **138,29** metros cuadrados construidos y el precio de compra fue de **395.488** euros. Adicionalmente se le han hecho cuatro mejoras: Una **buhardilla** con un coste de **13.379** euros el **30 de noviembre de 2019**, se le han colocado **placas solares** por un importe de **13.502** euros el **4 de abril de 2023**, también se le había hecho una **piscina** por **8.945** euros el **29 de octubre de 2020** y un garaje con un coste de **13.177** euros el **20 de enero de 2023**.

Utilice obligatoriamente las definiciones de `NamedTuple` que se indican a continuación:

```
from typing import NamedTuple
from datetime import datetime, date

Mejora = NamedTuple("Mejora",
    [("denominacion", str),
     ("coste", int),
     ("fecha ", date)])

Vivienda = NamedTuple("Vivienda",
    [("propietario", str),
     ("calle", str),
     ("fecha_adquisicion", date),
     ("numero", int),
     ("metros", float),
     ("precio", int),
     ("mejoras", list[Mejora])])
```

Observaciones

- Si una vivienda no tiene mejoras, se almacenará una **lista vacía** en el campo **mejora** de la tupla correspondiente.
- La lista de mejoras de una vivienda **no** está ordenada por fechas.
- El **valor del metro cuadrado** de una vivienda se calcula dividiendo el **precio** de compra por los **metros** cuadrados construidos.
- La cadena de formato para parsear las fechas es "%d/%m/%Y".
- El número de días entre dos fechas, fecha_inicio y fecha_fin, se calcula como (fecha_fin - fecha_inicio).days.
- Para calcular el valor absoluto use la función abs.

Estructura de las carpetas del proyecto

- **/src**: Contiene los diferentes módulos de Python que conforman el proyecto.
 - **urbanizacion.py**: Contiene funciones para explotar los datos de viviendas.
 - **urbanizacion_test.py**: Contiene funciones de test para probar las funciones del módulo urbanizacion.py. En este módulo está el main.
 - Puede añadir otros módulos para funciones auxiliares si así lo desea.
- **/data**: Contiene el dataset del proyecto.
 - **urbanizacion.csv**: Archivo con las viviendas de la urbanización.
- **/doc**: Contiene la documentación del proyecto.
 - **identificacion.md**: Archivo con los datos del alumno que realiza el examen.
 - **FP2425-Parcial1-Sesión 2-Enunciado.pdf**: Enunciado del examen.

Ejercicios

Se pide implementar las siguientes funciones (en el módulo urbanizacion.py) y sus funciones de test correspondientes con los parámetros adecuados (en el módulo urbanizacion_test.py). Las puntuaciones indicadas para cada ejercicio incluyen la realización de dichos tests. Tenga en cuenta que se pueden definir funciones auxiliares cuando se considere necesario.

Para cada función solicitada, se le proporciona el prototipo de la función (donde puede ver los parámetros, sus tipos y el tipo de salida), así como una posible salida del test correspondiente.

Ejercicio 1 (1,5 puntos)

```
def lee_viviendas(ruta: str) -> list[Vivienda]
```

Recibe la ruta de un fichero CSV y devuelve una lista de tuplas de tipo Vivienda conteniendo todos los datos almacenados en el fichero.

Antes de implementar la función anterior, implemente una función auxiliar que dada una cadena de texto con las mejoras separadas por *, devuelva una lista de tipo Mejora. Tenga en cuenta que si la cadena de texto está vacía la función debe devolver una lista vacía (compruébelo en el penúltimo registro de los resultados esperados).

```
def parsea_mejoras(mejoras_str:str) -> list[Mejora]
```

```
test_lee_viviendas
```

Número de registros leídos:106

Las dos primeras: [Vivienda(propietario='AGUILAR CABALLERO, CARLOS', calle='HALCÓN', numero=30, fecha_adquisicion=datetime.date(2018, 10, 10), metros=138.29, precio=395488, mejoras=[Mejora(denominacion='BUHARDILLA', coste=13379, fecha=datetime.date(2019, 11, 30)), Mejora(denominacion='PLACAS SOLARES', coste=13502, fecha=datetime.date(2023, 4, 4)), Mejora(denominacion='PISCINA', coste=8945, fecha=datetime.date(2020, 10, 29)), Mejora(denominacion='GARAJE', coste=13177, fecha=datetime.date(2023, 1, 20))]), Vivienda(propietario='ALONSO GARCÍA, AGUSTÍN', calle='PALOMA', numero=5, fecha_adquisicion=datetime.date(2013, 3, 1), metros=190.69, precio=395379, mejoras=[Mejora(denominacion='PISCINA', coste=10277, fecha=datetime.date(2014, 9, 11))])]

Las dos últimas: [Vivienda(propietario='VICENTE ROJAS, ELOY', calle='ÁGUILA', numero=6, fecha_adquisicion=datetime.date(2005, 6, 17), metros=197.48, precio=446610, mejoras=[]), Vivienda(propietario='VIDAL CAMPOS, DANIEL', calle='HALCÓN', numero=23, fecha_adquisicion=datetime.date(2009, 6, 10), metros=106.36, precio=302212, mejoras=[Mejora(denominacion='PISCINA', coste=12233, fecha=datetime.date(2011, 12, 31)), Mejora(denominacion='BUHARDILLA', coste=13474, fecha=datetime.date(2012, 5, 3)), Mejora(denominacion='GARAJE', coste=13096, fecha=datetime.date(2011, 6, 30)), Mejora(denominacion='PLACAS SOLARES', coste=13168, fecha=datetime.date(2015, 4, 1))])]

Ejercicio 2 (1,25 puntos)

```
def total_mejoras_por_calle (viviendas: list[Vivienda], par_impar: str) -> dict[str,int]
```

Devuelve un diccionario que hace corresponder cada calle con el número total de mejoras de las viviendas de número par o impar de esa calle, según sea el valor del parámetro par_impar. El parámetro par_impar puede tomar los valores "par" o "impar".

```
test_total_mejoras_por_calle
```

Número de mejoras en viviendas: par

HALCÓN --> 33

PALOMA --> 18

ÁGUILA --> 16

GOLONDRINA --> 12

LECHUZA --> 11

CODORNÍZ --> 11

PERDÍZ --> 8

ZORZAL --> 14

Número de mejoras en viviendas: impar

PALOMA --> 16

HALCÓN --> 41

LECHUZA --> 9

ÁGUILA --> 26

ZORZAL --> 10

GOLONDRINA --> 17

PERDÍZ --> 7

CODORNÍZ --> 12

Ejercicio 3 (1,25 puntos)

```
def vivienda_con_mejora_mas_rapida(viviendas: list[Vivienda]) -> tuple[str, str, int, int, str]:
```

Devuelve una tupla con información de la vivienda que menos días tardó en hacer una mejora desde que fue adquirida. La tupla contendrá el propietario, la calle, el número, el número de días transcurridos entre la adquisición de la vivienda y la primera mejora, y la denominación de esa mejora. Tenga en cuenta que las mejoras no están ordenadas por fecha.

```
test_vivienda_con_mejora_mas_rapida
```

```
La vivienda que hizo una mejora en menos tiempo es ('VELASCO PARRA, ALICIA', 'ÁGUILA', 12, 173, 'PISCINA')
```

Ejercicio 4 (2 puntos)

```
def calle_mayor_diferencia_precios(viviendas: list[Vivienda]) -> str:
```

Devuelve la calle en la que hay una mayor diferencia entre la suma de los precios de las viviendas con número impar y la suma de los precios de las viviendas con número par. Utilice el valor absoluto (función abs) de la diferencia de ambas sumas de precios.

```
test_calle_mayor_diferencia_precios
```

```
La calle es: HALCÓN
```

Ejercicio 5 (2 puntos)

```
def n_viviendas_top_valoradas_por_calle(viviendas: list[Vivienda],
                                         fecha: date | None = None,
                                         n: int = 3) -> dict[str, list[tuple[str, int, int]]]
```

Devuelve un diccionario que hace corresponder cada calle con una lista de tuplas formadas por nombre del propietario, el número de la vivienda y el **valor** de cada vivienda, de las n viviendas más valoradas de la calle de que se trate, para las viviendas adquiridas en la fecha dada o con posterioridad. Si fecha es None, se tendrán en cuenta todas las viviendas. El **valor** de una vivienda se calcula como el precio de compra más el coste de las mejoras que se le hayan realizado.

```
test_n_viviendas_top_valoradas_por_calle
```

```
Para n=4 y fecha=2020-01-01 son:
```

```
HALCÓN --> [('FLORES PRIETO, ALFONSO', 14, 399036), ('MEDINA SANZ, ELENA', 10, 379355), ('CAMPOS REYES, JOAQUÍN', 24, 370440), ('GALLEGO CALVO, ALICIA', 21, 337682)]
```

```
ÁGUILA --> [('PARRA SÁEZ, ALEJANDRO JESUS', 13, 433622), ('SOLER VELASCO, ÁNGEL LUIS', 19, 413162), ('FERRER CARMONA, FRANCISCO DE BORJA', 4, 391585), ('SOTO CRESPO, DANIEL', 8, 376474)]
```

```
LECHUZA --> [('CRUZ MÉNDEZ, SERGIO', 8, 362371)]
```

```
PERDÍZ --> [('GARCÍA RODRÍGUEZ, SALVADOR', 1, 400599)]
```

```
GOLONDRINA --> [('GIMÉNEZ HERRERO, SERGIO', 4, 444569), ('SANTANA VARGAS, MARCO', 1, 386332)]
```

```
PALOMA --> [('GUTIÉRREZ ALONSO, ALEJANDRO', 4, 381495)]
```

Con parámetros por defecto son:

```
HALCÓN --> [('NUÑEZ MEDINA, FELIPE PATRICIO', 9, 494211), ('DÍEZ AGUILAR, JUAN', 29, 462402), ('AGUILAR CABALLERO, CARLOS', 30, 444491)]
PALOMA --> [('ROMERO GUTIÉRREZ, ALEJANDRO JESUS', 3, 509324), ('GUTIÉRREZ ALONSO, MARIANO', 10, 478869), ('ROMERO GUTIÉRREZ, JOSÉ MARÍA', 9, 476457)]
ÁGUILA --> [('ARIAS MORA, JOAQUÍN', 2, 465273), ('PASTOR VELASCO, ÁNGEL LUIS', 11, 451214), ('VICENTE ROJAS, ELOY', 6, 446610)]
GOLONDRINA --> [('DURAN IBÁÑEZ, LUIS', 11, 493751), ('LORENZO SANTIAGO, RAFAEL', 8, 486973), ('VARGAS PASCUAL, RAFAEL', 2, 451940)]
LECHUZA --> [('SANTOS GUERRERO, MARÍA', 4, 463438), ('CANO CRUZ, SALVADOR', 7, 460523), ('LOZANO CANO, RAFAEL', 6, 457160)]
CODORNÍZ --> [('HERNÁNDEZ RUIZ, ELENA', 6, 497687), ('RUIZ DÍAZ, DANIEL', 7, 473237), ('MORENO MUÑOZ, ANTONIO', 9, 434395)]
ZORZAL --> [('GIL SERRANO, MANUELA', 5, 530466), ('RAMÍREZ GIL, YOLANDA', 4, 497259), ('MORALES MOLINA, MARÍA DEL CARMEN', 7, 466794)]
PERDÍZ --> [('FERNÁNDEZ LÓPEZ, MARCO', 4, 462676), ('MARTÍNEZ SÁNCHEZ, JUAN', 6, 424170), ('GARCÍA RODRÍGUEZ, SALVADOR', 1, 400599)]
```

Ejercicio 6 (2 puntos)

```
def valor_metro_cuadrado_por_calle_y_año(viviendas: list[Vivienda]) -> list[tuple[int, list[tuple[str, float]]]]:
```

Se quiere obtener el **precio medio del metro cuadrado** de las viviendas de cada calle por año. Para ello, la función debe devolver una lista de tuplas, ordenada de menor a mayor año. Cada tupla estará formada por dos elementos:

1. El año.
2. Una lista de tuplas con el nombre de la calle y el precio medio del metro cuadrado de las viviendas de la calle en ese año. Esta lista estará ordenada de mayor a menor precio medio.

```
test_valor_metro_cuadrado_por_calle_y_año
```

```
(2005, [('PALOMA', 2795.345422040608), ('LECHUZA', 2721.4185529907263), ('HALCÓN', 2497.6665778915103), ('ÁGUILA', 2463.1517284325346)])
(2006, [('HALCÓN', 2323.146286084982), ('ZORZAL', 2134.7223126505633), ('CODORNÍZ', 2010.0065675562719)])
(2007, [('PALOMA', 2346.298443698505), ('HALCÓN', 1969.3380587630045)])
(2008, [('LECHUZA', 2456.813744213916), ('CODORNÍZ', 2384.542262663104), ('HALCÓN', 2266.3322566664974)])
(2009, [('CODORNÍZ', 2884.980037143334), ('HALCÓN', 2851.387822038723), ('GOLONDRINA', 2537.121338912134), ('PALOMA', 2479.3693911664186)])
(2010, [('HALCÓN', 2498.0901981776246), ('PALOMA', 2387.6323823979365)])
(2011, [('CODORNÍZ', 2405.5942198374332), ('GOLONDRINA', 2385.4682454251883), ('ÁGUILA', 2379.5311273668895)])
(2012, [('PERDÍZ', 2289.035496222705), ('HALCÓN', 2270.9011643949007), ('ÁGUILA', 2248.4111669985145), ('ZORZAL', 2232.9243786356424), ('PALOMA', 2223.0526571220576)])
(2013, [('GOLONDRINA', 3016.9066098700378), ('PERDÍZ', 2509.138527995544), ('HALCÓN', 2415.6233734687935), ('LECHUZA', 2369.515312777169), ('PALOMA', 2073.412344643138)])
(2014, [('GOLONDRINA', 2646.0371606909566), ('HALCÓN', 2635.359681803511), ('LECHUZA', 2388.0957951999007), ('ÁGUILA', 2270.1369029510192), ('PERDÍZ', 2202.268760907504)])
(2015, [('CODORNÍZ', 3421.2869967114034), ('PALOMA', 3116.0843898190265), ('ÁGUILA',
```

2468.1085112333367), ('ZORZAL', 2359.1300286877045), ('HALCÓN', 2194.264430478129))
(2016, [('ÁGUIA', 2783.3219877467664), ('PALOMA', 2750.5733856692063), ('ZORZAL',
2453.875418484326))
(2017, [('ÁGUIA', 3374.1882330134727), ('GOLONDRINA', 3186.1931364724664), ('ZORZAL',
2570.567240091742), ('CODORNÍZ', 2458.5933253614526))
(2018, [('ZORZAL', 3393.0243998425817), ('HALCÓN', 2678.348873364085), ('GOLONDRINA',
2357.44825069086), ('PALOMA', 2314.314251169637), ('CODORNÍZ', 2291.0841193952706))
(2019, [('PERDÍZ', 3010.135489183322), ('PALOMA', 2683.602132841814), ('GOLONDRINA',
2560.8499219795767), ('ÁGUIA', 2311.777598511685), ('HALCÓN', 2224.224658658977))
(2020, [('ÁGUIA', 3305.6402912323315), ('LECHUZA', 3098.28330206379), ('PALOMA',
2999.1745283018868), ('HALCÓN', 2863.8752052545155))
(2021, [('ÁGUIA', 2711.7594306237434), ('PERDÍZ', 2708.8564712389384), ('GOLONDRINA',
2690.4363071776615), ('HALCÓN', 2513.9513992839597))