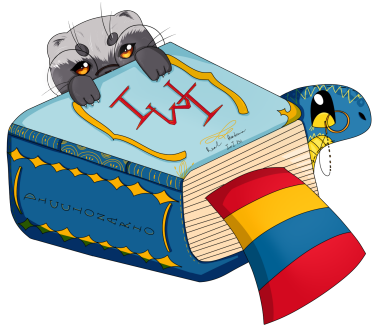
Tarea UVA 8: Diccionarios



La inmobiliaria Pyhouses almacena la oferta de casas y departamentos en arriendo y venta en una lista de listas como la siguiente:

```
oferta = [
  ['B6396', 'C', 'V', 'Providencia', 4, 3, 1, 63, True, False, 140513611],
  ['B2003', 'C', 'A', 'Ñuñoa', 4, 1, 2, 72, False, False, 122655402],
  ['B4510', 'D', 'V', 'Valparaíso', 3, 3, 1, 53, True, False, 137661254],
  ['A6803', 'D', 'A', 'Providencia', 4, 1, 0, 80, False, False, 151368062],
  # ...
]
```

La estructura de cada lista interior es:

```
[id, tipo, operación, comuna, dormitorios, baños, estacionamientos, m2, piscina, gimnasio, precio]
```

Aquí, **`id`** es un identificador único, **`tipo`** puede ser `'C'` para casa o `'D'` para departamento, **`operación`** puede ser `'A'` para arriendo o `'V'` para venta, y **`comuna`** indica dónde se encuentra la propiedad. Por otra parte, **`dormitorios`**, **`baños`** y **`estacionamientos`** representan la cantidad de cada uno de estos elementos que tiene la propiedad, mientras que **`m2`** corresponde al área. Los campos **`piscina`** y **`gimnasio`** son booleanos, e indican si se cuenta con esos espacios. Finalmente, **`precio`** es un valor entero que corresponde al precio de arriendo o venta de la propiedad.

1. Implemente la función `comunas_con_mas_oferta(lista, operacion)`, que recibe la lista con la oferta de propiedades y una operación (`'A'` ó `'V'`). La función debe retornar una lista de *strings* con las 3 comunas que tienen más oferta para la operación indicada, ordenada de mayor a menor de acuerdo a la cantidad de propiedades. Puede ver ejemplos de la ejecución esperada de la función al final del documento. Suponga que siempre existirán al menos 3 comunas que cumplan los criterios.

2. Implemente la función `filtrar(lista, tipo, operacion, min_dormitorios, min_m2)`, que recibe la lista con la oferta de propiedades, así como un indicador de tipo y operación, y una cantidad mínima de dormitorios y metros cuadrados que se está buscando. La función debe retornar un diccionario, cuyas llaves son las distintas comunas y los valores son listas con las propiedades que cumplen con los criterios de búsqueda, con la estructura: `[precio, id, dormitorios, baños, estacionamientos, m2, piscina, gimnasio]`.
Cada lista asociada a una comuna debe estar ordenada de menor a mayor de acuerdo al precio de la propiedad ofertada. Puede ver ejemplos de la ejecución esperada de la función al final del documento.

3. Considere una lista de solicitudes que los clientes han hecho a la empresa inmobiliaria, con la siguiente estructura:

```
solicitudes = [
   [nombre_cliente, [tipo, operacion, min_dormitorios, min_m2]],
   # ...
]
```

Implemente la función `buscar(oferta, solicitudes)`, que recibe la lista con la oferta de propiedades y la lista de solicitudes de los clientes. La función debe retornar una lista con las propiedades que cumplen con los criterios de búsqueda de cada cliente, pero **limitándose a la propiedad de menor valor en cada comuna**. Esta función **debe** utilizar la función `filtrar` de la pregunta anterior. La lista resultante debe tener la siguiente estructura:

```
[
 [nombre_cliente1, [
   [comuna1, precio, id, dormitorios, baños, estacionamientos, m2, piscina, gimnasio],
   [comuna2, precio, id, dormitorios, baños, estacionamientos, m2, piscina, gimnasio],
   # ...
   ]
 ]
 [nombre_cliente2, [
   [comuna1, precio, id, dormitorios, baños, estacionamientos, m2, piscina, gimnasio],
   [comuna2, precio, id, dormitorios, baños, estacionamientos, m2, piscina, gimnasio],
   # ...
   ]
 ]
 # ...
]
```

Ejemplos

```
oferta = [['B6396', 'C', 'V', 'Providencia', 4, 3, 1, 63, True, False, 140513611],
['B2003', 'C', 'A', 'Ñuñoa', 4, 1, 2, 72, False, False, 122655402], ['B4510', 'D',
'V', 'Valparaíso', 3, 3, 1, 53, True, False, 137661254], ['A6803', 'D', 'A',
'Providencia', 4, 1, 0, 80, False, False, 151368062], ['A9442', 'D', 'A', 'Ñuñoa',
3, 1, 0, 114, True, False, 148007915], ['C4297', 'D', 'V', 'Ñuñoa', 2, 3, 2, 74,
True, True, 201075569], ['B1017', 'C', 'V', 'Valparaíso', 3, 3, 2, 60, True, False,
224580934], ['C6171', 'D', 'A', 'Valparaíso', 1, 3, 2, 63, True, False, 192273786],
['A1721', 'D', 'V', 'Ñuñoa', 2, 3, 2, 112, False, False, 118911622], ['B5197', 'C',
'A', 'Ñuñoa', 2, 2, 1, 82, True, False, 195287817], ['B4913', 'D', 'V',
'Concepción', 4, 2, 2, 73, False, True, 186961907], ['A5610', 'D', 'A', 'Viña del
Mar', 1, 2, 2, 115, False, True, 210702365], ['C7718', 'D', 'A', 'Concepción', 1, 1,
2, 112, False, True, 189009541], ['B9228', 'D', 'A', 'San Joaquín', 2, 3, 2, 110,
True, False, 248914620], ['A8676', 'C', 'V', 'Ñuñoa', 2, 1, 1, 62, False, False,
165369148], ['A2990', 'D', 'A', 'Viña del Mar', 3, 2, 1, 97, True, False,
248708274], ['A3150', 'C', 'V', 'San Joaquín', 3, 1, 1, 79, False, False,
202722466], ['B4236', 'C', 'A', 'San Joaquín', 1, 1, 1, 104, False, False,
153363382], ['B5625', 'C', 'V', 'Providencia', 3, 1, 1, 109, False, False,
86181180], ['C8119', 'D', 'A', 'Ñuñoa', 1, 3, 0, 60, False, True, 235617495],
['A7013', 'C', 'V', 'Viña del Mar', 2, 3, 0, 120, True, False, 146430526], ['A2983',
'D', 'V', 'Ñuñoa', 2, 2, 1, 80, True, False, 195192915], ['B2691', 'D', 'V', 'Viña
del Mar', 2, 2, 2, 80, False, False, 109713495], ['C2187', 'D', 'V', 'Ñuñoa', 1, 2,
2, 77, False, True, 195733294], ['C4132', 'C', 'A', 'Ñuñoa', 1, 2, 1, 119, False,
False, 67104298]]

solicitudes = [
   ['Andrea', ['D', 'V', 3, 70]],
   ['Fede', ['C', 'V', 2, 60]],
   ['Pedro', ['D', 'A', 1, 60]]
]

>>> print(comunas_con_mas_oferta(oferta, 'V'))
['Ñuñoa', 'Viña del Mar', 'Valparaíso']
>>> print(comunas_con_mas_oferta(oferta, 'A'))
['Ñuñoa', 'Viña del Mar', 'San Joaquín']

>>> print(filtrar(oferta, 'D', 'A', 1, 60))
{
  'Providencia': [[151368062, 'A6803', 4, 1, 0, 80, False, False]],
  'Ñuñoa': [[148007915, 'A9442', 3, 1, 0, 114, True, False],
            [235617495, 'C8119', 1, 3, 0, 60, False, True]
           ],
  'Valparaíso': [[192273786, 'C6171', 1, 3, 2, 63, True, False]],
  'Viña del Mar': [[210702365, 'A5610', 1, 2, 2, 115, False, True],
                   [248708274, 'A2990', 3, 2, 1, 97, True, False]
                  ],
  'Concepción': [[189009541, 'C7718', 1, 1, 2, 112, False, True]],
  'San Joaquín': [[248914620, 'B9228', 2, 3, 2, 110, True, False]]
}
```

```
>>> print(buscar(oferta, solicitudes))
[
  ['Andrea', [ ['Concepción', 186961907, 'B4913', 4, 2, 2, 73, False, True] ]],
  ['Fede', [ ['Providencia', 86181180, 'B5625', 3, 1, 1, 109, False, False],
             ['Valparaíso', 224580934, 'B1017', 3, 3, 2, 60, True, False],
             ['Ñuñoa', 165369148, 'A8676', 2, 1, 1, 62, False, False],
             ['San Joaquín', 202722466, 'A3150', 3, 1, 1, 79, False, False],
             ['Viña del Mar', 146430526, 'A7013', 2, 3, 0, 120, True, False]
           ] ],
  ['Pedro', [ ['Providencia', 151368062, 'A6803', 4, 1, 0, 80, False, False],
              ['Ñuñoa', 148007915, 'A9442', 3, 1, 0, 114, True, False],
              ['Valparaíso', 192273786, 'C6171', 1, 3, 2, 63, True, False],
              ['Viña del Mar', 210702365, 'A5610', 1, 2, 2, 115, False, True],
              ['Concepción', 189009541, 'C7718', 1, 1, 2, 112, False, True],
              ['San Joaquín', 248914620, 'B9228', 2, 3, 2, 110, True, False]]
            ] ]
```

Requisitos del programa

Las funciones deben cumplir con los siguientes requisitos:

1. Puede utilizar **sólo la materia estudiada hasta la UVA 8** de IWI-131 Programación.
2. La implementación de la función `buscar` debe llamar a la función `filtrar`.
3. La tarea debe realizarse de manera **estrictamente individual**.
4. Las funciones solicitadas deben **cumplir fielmente** la especificación dada, incluyendo los parámetros y valores de retorno. Asimismo, debe respetarse completamente los formatos de las listas y diccionarios descritos.
5. Tome en cuenta que **no sabemos de antemano cuántas propiedades o clientes hay**.

Importante: Puede suponer que los datos serán siempre correctos y apegados a los formatos descritos.