



Desarrollo Entorno Servidor
Python – 1ª Evaluación



EJERCICIO 1 (4 puntos)

Leer una cadena y generar una lista de tuplas con las veces que cada vocal se repite ordenada de mayor a menor frecuencia.

1. Implementa una función `esVocal(letra) -> bool`
Recibe una letra y determina si es o no un vocal
2. Implementa una función `estaEnLista(letra,lista) -> int`
Recibe una letra y una lista de listas y retorna la posición en la que se encuentra la letra o bien -1 si no se encuentra

[['E', 3], ['A', 1]]

3. Implementa una función `transformar(lista) -> list[()]`
Recibe una lista como la del ejemplo del apartado anterior y la transforma en una lista de tuplas
[['E', 3], ['A', 1]] -----> [('E',3), ('A',1)]
4. Utiliza función lambda y programación funcional para ordenar la lista del apartado anterior de **mayor a menor frecuencia**

El proceso a implementar:

- a) Lee una palabra y construye una lista como la del apartado 2 con cada vocal de la palabra y las veces que se repite en la misma. Utiliza las funciones que se te han indicado.

Función `esVocal` **0,5 puntos**

Función `estaEnLista` **1 punto**

Construir lista **1 punto**

- b) Transforma la lista anterior en una lista de tuplas tal y como se te indica en el apartado 3. **(0,5 puntos)**
- c) Ordena la lista de tuplas tal y como se te indica en el apartado 4 **(1 punto)**

EJERCICIO 2 (3 puntos)

El directorio de los clientes de una empresa está organizado en una cadena de texto como la de más abajo, donde cada línea contiene la información del nombre, email, teléfono, nif, y el descuento que se le aplica. Las líneas se separan con el carácter de cambio de línea `\n` y la primera línea contiene los nombres de los campos con la información contenida en el directorio.

```
"nif;nombre;email;teléfono;descuento\n01234567;Luis González;luisgonzalez@mail.com;656343576;12.5\n71476342J;MacarenaRamírez;macarena@mail.com;692839321;8\n63823376M;Juan José Martínez;juanjo@mail.com;664888233;5.2\n98376547F;Carmen Sánchez;carmen@mail.com;667677855;15.7"
```

Escribir un programa que genere un diccionario con la información, donde cada elemento corresponde a un cliente y tiene por clave su nif y por valor otro diccionario con el resto de la información del cliente. Los diccionarios con la información de cada cliente tendrán como claves los nombres de los campos y como valores la información de cada cliente correspondientes a esos campos. Es decir, un diccionario como el siguiente **(2 puntos)**

```
{
'01234567L':
    {'nombre': 'Luis González', 'email': 'luisgonzalez@mail.com', 'teléfono': '656343576', 'descuento': 12.5},
'71476342J':
    {'nombre': 'Macarena Ramírez', 'email': 'macarena@mail.com', 'teléfono': '692839321', 'descuento': 8.0},
'63823376M':
    {'nombre': 'Juan José Martínez', 'email': 'juanjo@mail.com', 'teléfono': '664888233', 'descuento': 5.2},
'98376547F':
    {'nombre': 'Carmen Sánchez', 'email': 'carmen@mail.com', 'teléfono': '667677855', 'descuento': 15.7}
}
```

Debes imprimir el diccionario de la siguiente forma **(1 punto)**

```
-----
Cliente - 01234567X
{'nombre': 'Luis González', 'email': 'luisgonzalez@mail.com', 'teléfono': '656343576', 'descuento': '12.5'}
-----
Cliente - 71476342J
{'nombre': 'Macarena Ramírez', 'email': 'macarena@mail.com', 'teléfono': '692839321', 'descuento': '8'}
-----
Cliente - 63823376M
{'nombre': 'Juan José Martínez', 'email': 'juanjo@mail.com', 'teléfono': '664888233', 'descuento': '5.2'}
-----
Cliente - 98376547F
{'nombre': 'Carmen Sánchez', 'email': 'carmen@mail.com', 'teléfono': '667677855', 'descuento': '15.7'}
```

EJERCICIO 3 (3 puntos)

Una inmobiliaria de una ciudad maneja una lista de inmuebles como la siguiente:

```
[{'año': 2000, 'metros': 100, 'habitaciones': 3, 'garaje': True, 'zona': 'A'},  
{ 'año': 2012, 'metros': 60, 'habitaciones': 2, 'garaje': True, 'zona': 'B'},  
{ 'año': 1980, 'metros': 120, 'habitaciones': 4, 'garaje': False, 'zona': 'A'},  
{ 'año': 2005, 'metros': 75, 'habitaciones': 3, 'garaje': True, 'zona': 'B'},  
{ 'año': 2015, 'metros': 90, 'habitaciones': 2, 'garaje': False, 'zona': 'A'}]
```

1. Añadir a la lista anterior el precio de cada inmueble teniendo en cuenta (1 punto)

- Zona A: $\text{precio} = (\text{metros} * 1000 + \text{habitaciones} * 5000 + \text{garaje} * 15000) * \text{año}/100$
- Zona B: $\text{precio} = (\text{metros} * 1000 + \text{habitaciones} * 5000 + \text{garaje} * 15000) * \text{año}/100 * 1.5$

Implementa una función **anyadir_precio** que recibe un diccionario y añade el campo precio con el valor calculado según las fórmulas que se te han indicado

2. Utiliza **programación funcional** para hacer una búsqueda de inmuebles en función de un precio leído por teclado.
Debes retornar una lista con los inmuebles cuyo precio sea menor o igual que el dado (**una sola línea de código**). **(2 puntos)**
No se puntuará el apartado si usas programación tradicional en lugar de funcional