

Introducción a la Programación

2ª Prueba de Evaluación continua

2022-2023

DNI:

NOMBRE Y APELLIDOS:

Se pide crear una aplicación que ayude a la gestión de los productos y las ventas de una farmacia.

De cada producto se guarda la siguiente información:

- **id:** identificador del producto
- **nombre:** nombre del producto
- **precio_coste:** el precio por el que la farmacia ha adquirido el producto
- **precio_venta:** el precio por el que la farmacia vende el producto
- **num_unidades:** el número de unidades disponibles del producto

De cada venta se almacena la siguiente información:

- **código:** el código de la venta
- **día:** día en el que se realizó la venta
- **mes:** mes en el que se realizó la venta
- **productos vendidos:** diccionario en el que, por cada producto vendido en esa venta, se guarda el identificador del producto (clave) junto con el número de unidades vendidas de ese producto (valor).

Los datos de ventas serán los siguientes:

```
ventas = [['C201', 1, 2, {'1': 2, '2': 1}], ['C302', 2, 3, {'4': 2, '3': 3}],  
['C203', 1, 2, {'1': 1, '4': 2}], ['C205', 1, 2, {'2': 2, '3': 1}]]
```

Partiendo de esta información se pide que implementéis las siguientes funciones en un único fichero denominado **examen.py**:

1. **cargarProductos:** función que recibe la lista de productos vacía y añade a la lista los productos almacenados en el fichero "productos.csv" que tiene el siguiente formato **(0,5 puntos)**:

 productos.csv: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
id;nombre;precio_coste;precio_venta;num_unidades  
1;Frenadol;1.5;2.8;20  
2;Ibudol;2;3.5;30  
3;Strepsils;3.5;4.5;20  
4;Paracetamol;1;2;15
```

```
cargarProductos(productos)

print(productos)#Devuelve [['1', 'Frenadol', '1.5', '2.8', '20'], ['2', 'Ibudol', '2', '3.5', '30'],
['3', 'Strepsils', '3.5', '4.5', '20'], ['4', 'Paracetamol', '1', '2', '15']]
```

2. **obtenerProducto**: función que recibe la lista de productos y el identificador del producto a buscar y devuelve la información del producto en caso de encontrarlo en la lista. Si el producto no se encuentra en la lista de productos, se devolverá 'Producto no existe'. **(0,5 puntos)**

Ejemplo de ejecución 1:

```
print(obtenerProducto(productos, '2')) #Devuelve ['2', 'Ibudol', '2', '3.5', '30']
```

Ejemplo de ejecución 2:

```
print(obtenerProducto(productos, '5')) #Devuelve Producto no existe
```

3. **obtenerProductosVendidosMes**: función que recibe la lista de ventas y el mes para el que se quiere obtener sus ventas y devuelve una lista con el identificador de los productos vendidos ese mes. **(0,6 puntos)**

Ejemplo de ejecución 1:

```
print(obtenerProductosVendidosMes(ventas, 3))#Devuelve ['4', '3']
```

Ejemplo de ejecución 2:

```
print(obtenerProductosVendidosMes(ventas, 2))#Devuelve ['1', '2', '4', '3']
```

4. **calcularBeneficiosTotal**: función que recibe la lista de ventas y la lista de productos y devuelve el beneficio total obtenido con las ventas. **(0,7 puntos)**

```
print(f'{calcularBeneficiosTotal(ventas, productos):.2f}')
#Devuelve 16.40
```

5. **ventasPorProducto**: función que recibe la lista de ventas y devuelve un diccionario en el que se indicarán cuántas unidades de cada producto se han vendido. Para ello, será necesario recorrer la lista de ventas e ir contabilizando en el diccionario el total de unidades de cada uno de ellos. Las claves de este diccionario serán los identificadores de producto y los valores serán el número de productos. **(0,7 puntos)**

```
print(ventasPorProducto(ventas))
#Devuelve {'1': 3, '2': 3, '4': 4, '3': 4}
```