



Nombre: \_\_\_\_\_

DNI: \_\_\_\_\_ Aula: \_\_\_\_\_ Fila: \_\_\_\_\_ Columna \_\_\_\_\_

## Instrucciones

- El examen se recogerá en pendrive al terminar.
- NO está permitida la conexión a internet en el examen. Tampoco la utilización de tu propio portátil ni la consulta de apuntes o de otros códigos.
- Guarda el código de todos los ejercicios en un único fichero que se llame "ExamenFinal-NombreApellidos.py", por ejemplo: ExamenFinal-PabloGaraizar.py.

## Enunciado

**DeustoMarket** es un nuevo servicio de supermercado *online* para estudiantes de la Universidad de Deusto. Te han contratado para crear una aplicación que les ayude a gestionar sus productos y pedidos:

### Producto

Cada producto de DeustoMarket requiere almacenar la siguiente información:

- **id**: cadena de caracteres (str), por ejemplo "ABC123".
- **nombre**: cadena de caracteres (str), por ejemplo "Tomate frito".
- **precio**: en euros (float), mayor o igual a cero.
- **descuento**: en porcentaje (int), de 0 a 100.

### Pedido

Cada pedido de DeustoMarket requiere almacenar la siguiente información:

- **id**: código numérico (int), de 1 en adelante.
- **usuario**: cadena de caracteres (str), por ejemplo "garaizar".
- **productos**: lista de productos de este pedido.

Modela productos y pedidos como diccionarios o listas multidimensionales.

Crea un programa con una lista de productos vacía, una lista de pedidos vacía y programa las siguientes funciones:

**productoStr**: función que recibe un producto y devuelve una cadena de caracteres (str) con el siguiente formato: 'Tomate frito, 1.45 € (0% de descuento)' **(0.5 puntos)**

**totalPedido**: función que recibe un pedido y devuelve el precio total de todos sus productos **(1 punto)**

**pedidoStr**: función que recibe un pedido y devuelve una cadena de caracteres (str) con el siguiente formato: 'usuario: garaizar, 12 productos, total: 46.5 €' **(0.5 puntos)**



**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: **id;nombre;precio;descuento (1.5 puntos)**. Por ejemplo:

**ABC123;Tomate frito;1.45;0**

**Atención:** si no sabes hacer esta función, sustitúyela por la creación a mano de una lista de 10 productos. Esta solución alternativa valdrá 0.5 puntos.

**crearPedidos:** función que recibe la lista de productos llena y la lista de pedidos vacía y devuelve una lista de pedidos llena después de generar 100 pedidos de los usuarios 'usuario-1', 'usuario-2', 'usuario-3', 'usuario-4', y 'usuario-5' añadiendo aleatoriamente 5 productos de la lista de productos **(1.5 puntos)**

**guardarPedidos:** función que recibe la lista de pedidos y los almacena en el fichero 'pedidos.csv' que tiene el siguiente formato: **id;usuario;total (1.5 puntos)**. Por ejemplo:

**1;garaizar;65.45**

**totalDeustoMarket:** función que recibe la lista de pedidos y devuelve el sumatorio del precio total de todos los pedidos **(1 punto)**

**usuarioVip:** función que recibe la lista de pedidos y devuelve el nombre del usuario que más haya pagado por sus pedidos en DeustoMarket **(2 puntos)**

**stockNecesario:** función que recibe la lista de pedidos y devuelve un diccionario en el que se indicarán cuántas unidades de cada producto serán necesarias para completar todos los pedidos. Para ello, será necesario recorrer la lista de productos de cada pedido e ir contabilizando en el diccionario el total de unidades de cada uno de ellos. Las claves de este diccionario serán los códigos de producto y los valores serán el número de productos necesarios. Por ejemplo, si hubiera varios pedidos en los que se piden los productos 'ABC123', 'XYZ347' y 'DFG324', el resultado de esta función debería ser un diccionario indicando la cantidad total de cada producto después de recorrer todos los pedidos: **(2.5 puntos)**

**{ 'ABC123': 23, 'XYZ347': 12, 'DFG324': 27 }**

Como se ha comentado anteriormente, además de estas funciones, desarrolla un **programa principal** en el que haya una lista de **productos** vacía (que llenarás con la función **cargarProductos**) y una lista de **pedidos** vacía (que llenarás con la función **crearPedidos**) y se llame a las funciones **productoStr**, **pedidoStr**, **guardarPedidos**, **totalDeustoMarket**, **usuarioVip**, **productoTop** y **stockNecesario** **(0.5 puntos)**