

## Fundamentos de la programación – Grupos A y B

Curso 2020–2021

### Primera convocatoria

Tiempo disponible: 2:30 horas

Desarrolla un programa en C++ para gestionar la venta de bebidas por parte de una máquina expendedora.

Funcionamiento del sistema. El sistema ofrece una lista de bebidas disponibles. El usuario puede elegir las que quiera. En todo momento se muestra en un “carrito” el estado actual del pedido. Cuando finaliza el pedido el sistema muestra en un fichero de texto una factura y finaliza la ejecución.

Funcionamiento detallado. La lista de productos disponibles se obtiene de un fichero de texto llamado `catalogo.txt` con el formato: *código del pedido, precio por unidad del producto, número de unidades disponibles y nombre del producto*. La señal de fin de archivo (centinela) es el valor -1. Ejemplo:

```
0 0.55 20 coca-cola
1 0.61 50 fanta
2 0.45 30 cerveza
3 0.15 10 vino de mesa
4 0.99 20 gaseosa
5 0.89 20 agua mineral
6 0.37 10 cafe helado
-1
```

Observa que los códigos de producto son números enteros consecutivos y comienzan siempre con el valor 0.

Tras cargar el contenido inicial se muestran los productos por pantalla y un menú con tres opciones: 1º) *Añadir producto al carrito*, 2º) *eliminar producto del carrito* y 3º) *efectuar pedido y terminar*.

Para la primera opción el usuario elige qué producto (a través del código de producto) y cuántas unidades comprar. Cuando se agotan las existencias de un producto desaparece de la lista visualizada. En la segunda opción el usuario elige qué producto (a través del código de producto) desea eliminar del carrito. Con esta opción se eliminan todas las unidades de ese producto y se restauran los valores del catálogo. Después de cada una de estas opciones se

muestra el contenido del carrito: *código de producto*, el *nombre* y el *precio de los productos* comprados de cada tipo. Además, se muestra el *precio total* del contenido del carrito. Finalmente, cuando el usuario elige la opción de *efectuar pedido y terminar* se guardan en el fichero `catalogo.txt` la lista de los productos con el valor de las existencias actualizado y se genera un fichero de texto llamado `factura.txt` con la lista de los nombres y costes de los productos comprados, el coste total y el coste con el IVA del 21%.

Utiliza `enum`, `structs`, `arrays` o combinaciones de ambos para implementar al menos los siguientes tipos:

- `tProducto` para representar los distintos productos,
- `tProductosComprados` para representar cada una de las bebidas seleccionadas (que están en el carrito),
- `tProductos` para representar la lista de todos los productos (estén agotados o no). Tiene como máximo 100 productos,
- `tCarrito` para representar el contenido del carrito. Debe incluir el coste total del pedido.

## Programa Principal (0.5 puntos)

El programa comenzará cargando la lista de productos de `catalogo.txt`. Si esta carga tiene éxito, el programa muestra un menú con las tres opciones mencionadas.

## Carga de Datos (1.5 puntos)

Implementa, al menos:

- ✓ `bool cargar(tProductos &productos)`: carga en `productos`, el contenido del fichero `catalogo.txt` y devuelve el valor `true` si todo ha ido bien.
- ✓ `void guardar(const tProductos &productos)`: guarda la lista de productos con sus cantidades disponibles actualizadas en `catalogo.txt`.
- ✓ `void generarFactura(tCarrito carrito)`: a partir de la información del carrito se genera el contenido de `factura.txt`

## Validación de datos (2 puntos)

Implementa, al menos:

- ✓ `bool codigoCorrecto(int codigo, const tProductos &productos)`: determina si es posible pedir un producto dado el valor de su código.
- ✓ `bool cantidadCorrecta(int codigo, int cantidad, const tProductos &productos)`: determina si es posible pedir cierta cantidad de un producto dado el valor de su código y la cantidad.

## Otros procedimientos y funciones (4 puntos)

---

Implementa, al menos:

- ✓ `int elegirItem(tProductos &productos, tCarrito &carrito):` muestra los productos disponibles, el estado del carrito y permite elegir un producto (a través de su código) y una cantidad. Además, comprueba que, tanto el código como la cantidad son correctas. Finalmente, llama a comprar.
- ✓ `void comprar(int codigo, int cantidad, tProductos &productos, tCarrito &carrito):` dado del código de un producto disponible añade la cantidad especificada al carrito y actualiza el estado de productos.
- ✓ `void eliminarItem(tProductos &productos, tCarrito &carrito):` se pide por pantalla qué código de producto se quiere eliminar del carrito y se actualizan productos y carrito.
- ✓ `void mostrar(const tProductos &productos, const tCarrito &carrito):` se muestra por pantalla los productos disponibles y el contenido del carrito.

## Ejemplo de ejecución

Productos disponibles:

Codigo Precio Cantidad Nombre

=====

0	0.55	10	coca-cola
1	0.61	40	fanta
2	0.45	30	cerveza
3	0.15	5	vino de mesa
4	0.99	60	gaseosa
5	0.89	20	agua mineral
6	0.37	10	cafe helado

Estado del carrito:

Carrito vacio

1.Aniadir al carrito

2.Eliminar del carrito

3.Efectuar pedido y terminar

Elija una opcion:1

Que producto desea comprar? 1



Cuántas unidades quiere comprar?: 2

Productos disponibles:

Código	Precio	Cantidad	Nombre
--------	--------	----------	--------

=====

0	0.55	10	coca-cola
1	0.61	38	fanta
2	0.45	30	cerveza
3	0.15	5	vino de mesa
4	0.99	60	gaseosa
5	0.89	20	agua mineral
6	0.37	10	cafe helado

Estado del carrito:

Código	Nombre	Precio
--------	--------	--------

=====

1	fanta	1.2
---	-------	-----

Total del pedido: 1.2

1.Añadir al carrito

2.Eliminar del carrito

3.Efectuar pedido y terminar

Elija una opción:1

Que producto desea comprar? 0

Cuántas unidades quiere comprar?: 10

Productos disponibles:

Código	Precio	Cantidad	Nombre
--------	--------	----------	--------

=====

1	0.61	38	fanta
2	0.45	30	cerveza
3	0.15	5	vino de mesa
4	0.99	60	gaseosa
5	0.89	20	agua mineral
6	0.37	10	cafe helado



Estado del carrito:

Codigo	Nombre	Precio
--------	--------	--------

=====

0	coca-cola	5.5
---	-----------	-----

1	fanta	1.2
---	-------	-----

Total del pedido: 6.7

1.Aniadir al carrito

2.Eliminar del carrito

3.Efectuar pedido y terminar

Elija una opcion:

## Corrección

**Recuerda** que no se permite el uso de variables globales ni de instrucciones de salto, salvo un return como última instrucción de las funciones. **Se valorará** (2 puntos) la corrección del diseño, la ausencia de código repetido, utilización de esquemas, la eficiencia del código, etc.

## Instrucciones de Entrega

- Añade al inicio de tu archivo .cpp un comentario con tus datos:  
/\*  
Nombre completo y DNI  
Grupo  
Laboratorio y puesto  
\*/
- Sube tu examen (solo el archivo .cpp) al enlace disponible en el campus.
- Pasa por el ordenador del profesor, muestra el DNI y firma.

