



PRÁCTICA Nº 8: Listas

Fecha de realización: Del 20 al 24 de Mayo

Duración: 2.5 horas presenciales

Fecha de entrega de ejercicios previos: Antes del Domingo 19 de Mayo a las 23:55

Fecha de entrega de la práctica: Hasta 6 días después de la sesión de prácticas de cada grupo.

Objetivos de la práctica

- Desarrollar una aplicación en C++ desde cero.
- Implementar una estructura de tipo Lista basada en punteros.

Material a entregar

- Solución de los ejercicios previos (1, 2 y 3): pr8_previo.doc. Se debe subir al Aula Virtual antes del Domingo 19 de Mayo a las 23:55.
- Solución de la práctica: archivo zip o rar con los ficheros de la solución final de la práctica. Se debe subir al Aula Virtual hasta 6 días después de la sesión de prácticas de cada grupo.

Introducción

Se va a desarrollar una aplicación para la gestión de la lista de la compra de una familia. La lista de la compra está formada por un conjunto de productos (para cada producto indicaremos qué cantidad deseamos comprar).

Para la representación de la lista vamos a utilizar la solución vista en clase de implementar una lista mediante punteros (lista enlazada). De esta manera cada nodo de la lista tiene **un solo puntero** que apunta al siguiente elemento.



La clase `Producto` va a representar una entrada de la lista de la compra.

TAD `Producto`

Dominio: Representa a los productos a adquirir en la compra especificando los atributos:

- *nombre*: nombre del producto.
- *tienda*: nombre del establecimiento donde comprar el producto.
- *seccion*: sección donde se encuentra el producto.
- *observaciones*: cualquier observación para ayudar al proceso de compra
- *cantidad*: número de unidades que se quiere comprar

Operaciones:

InicializarProducto(Producto, cadena, cadena, cadena, cadena, entero) -> Ninguno

AsignaNombre(Producto, cadena)-> Ninguno

AsignaTienda(Producto, cadena)-> Ninguno

AsignaSeccion(Producto, cadena)-> Ninguno

AsignaObservaciones(Producto, cadena)-> Ninguno

AsignaCantidad(Producto, entero)-> Ninguno

DevuelveNombre(Producto) -> cadena

DevuelveTienda(Producto) -> cadena

DevuelveSeccion(Producto) -> cadena

DevuelveObservaciones(Producto) -> cadena

DevuelveCantidad(Producto) -> entero

Axiomas:

Sea $p \in \text{Producto}$, $x \in \text{Entero}$, $n, t, s, o \in \text{Cadena}$ entonces

InicializarProducto(p, n, t, s, o, x) Asignar los datos de un producto (nombre, tienda, sección, observaciones y cantidad)

AsignaNombre(p, n) Asigna el nombre n al producto p .

AsignaTienda(p, t) Asigna la tienda t al producto p .

AsignaSeccion(p, s) Asigna la sección s al producto p .

AsignaObservaciones(p, o) Asigna el nombre n al producto p .

AsignaCantidad(p, q) Asigna la cantidad x al producto p .

DevuelveNombre(p) Devuelve el nombre del producto p .

DevuelveTienda(p) Devuelve la tienda del producto p .

DevuelveSeccion(p) Devuelve la sección del producto p .

DevuelveObservaciones(p) Devuelve las observaciones del producto p .

DevuelveCantidad(x) Devuelve la cantidad del producto p .



Ejercicios

Ejercicio 1: Implementar el TAD Producto (ficheros *Producto.cpp*, *Producto.h*). Sobrecargar el operador << con una función amiga para escribir por pantalla los datos de un producto.

Ejercicio 2: Implementar el TAD Lista (ficheros *Lista.cpp*, *Lista.h*) basada en punteros de similar al visto en clase (lista enlazada). Sobrecargar el operador << con una función amiga para escribir por pantalla los datos de la lista. Sobrecargar también el operador >> con una función amiga para escribir por pantalla los datos de la lista en el orden inverso (desde fin hasta inicio).

Ejercicio 3: Crear un proyecto Dev C++, añadir los archivos creados en los ejercicios anteriores y un archivo p8.cpp que contenga el programa principal.

Ejercicio 4: Implementar el programa principal. Éste debe de tener un menú que se ejecute hasta que se elija la opción salir. El menú permitirá elegir entre las opciones:

1. Ingresar producto
2. Borrar producto
3. Ver lista en orden directo
4. Ver lista en orden inverso
5. Salir

La opción 1 insertará un producto de manera que los productos de la misma sección estén agrupados.

La opción *Borrar producto* borrará un producto de la lista en función de su nombre.

La opciones 3 y 4 mostrará la lista en el orden que corresponda sin modificar su punto de interés.