

## UNIDAD TEMÁTICA 3: Listas, Pilas y Colas

### TRABAJO DE APLICACIÓN 2

#### Escenario: seguimos utilizando el mismo escenario del Trabajo de Aplicación #1:

La tienda “Grandeza y Elegancia ANte Todo” (por sus siglas *G.E.AN.T*), líder en el rubro de supermercados del país, necesita gestionar los productos de su supermercado, y nos ha encargado la construcción de un sistema software que permita hacerlo en forma eficiente.

La información que se tiene sobre un producto puede ser muy variable y extensa, pero como mínimo se tendrá:

- Nombre del producto,
- Código de identificación del producto
- Precio unitario.
- Cantidad existente del mismo en las góndolas y almacenes.

Nuestro sistema deberá implementar inicialmente las siguientes funcionalidades básicas:

1. Incorporar un nuevo producto al supermercado.
2. Agregar stock a un producto existente.
3. Simular la venta de un producto (reducir el stock de un producto existente)
4. Eliminar productos que ya no se venden (por no ser comercializados más).
5. Dado un código de producto, indicar las existencias del mismo en el almacén.
6. Listar todos los productos registrados, ordenados por nombre, presentando además su stock.

#### Ejercicio #1

##### PASO 1: Establecimiento de las clases en común del Equipo

1. Completar los métodos de Inserción y Búsqueda en el TDA Lista (de acuerdo a las interfaces publicadas)
2. Diseñar las estructuras de datos apropiadas para almacenar de manera eficiente la información necesaria para el producto de software requerido, utilizando el TDA Lista para ello.
3. Declarar las clases **Almacen** (que **contiene** una lista) y **Producto** (cada elemento de la lista) de acuerdo a las interfaces publicadas *iAlmacen* e *iProducto*.

## PASO 2: IMPLEMENTACION EN SUB-EQUIPOS – funciones básicas 1

En sub-equipos, implementar las siguientes funcionalidades:

### *Sub-equipo A:*

Compra de más unidades de un cierto producto o Incorporación de uno nuevo. ¿Cuál es el valor económico agregado al stock? Dado un archivo de entrada ("**altas.txt**") actualizar el almacén en forma correspondiente e indicar el **monto total** en que se ha incrementado el valor del stock (dinero gastado en comprar estos productos). El archivo "**altas.txt**" tiene la siguiente estructura (1 producto por línea, campos separados por comas)

**CODIGO PRODUCTO, DESCRIPCION DEL PRODUCTO, PRECIO UNITARIO, CANTIDAD**

### *Sub-equipo B:*

Venta de un producto (buscar y reducir su stock de acuerdo a la venta). ¿Cuál es el valor económico reducido del stock? Dado un archivo de entrada ("**ventas.txt**") actualizar el almacén en forma correspondiente e indicar el **monto total** en que se ha reducido el valor del stock (monto total vendido). El archivo "**ventas.txt**" tiene la siguiente estructura (1 producto por línea, campos separados por comas)

**CODIGO PRODUCTO, CANTIDAD a VENDER.**

**Nota:** si el stock disponible es menor que la venta deseada, se venderá lo que haya.

## PASO 3: INTEGRACION y VERIFICACIONES– funciones básicas 1

1. Integrar el código desarrollado en los sub-equipos para contener un programa único que permita probar las funcionalidades.

El programa recibirá como entrada los siguientes archivos (que serán liberados en la webasignatura antes de realizar las preguntas comunes):

- altas.txt
  - ventas.txt
2. Responder a las preguntas presentadas en pantalla.