

## UNIDAD TEMÁTICA 4: Árboles Binarios I

### TRABAJO DE APLICACIÓN 9

#### Escenario: seguimos utilizando el mismo escenario del Trabajo de Aplicación #2 de UT3:

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 código, presentando además su stock.

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

Descargar el **proyecto** “código base TA9”, que contiene las interfases y clases básicas para trabajar, en base a Árboles binarios de búsqueda, en dos sub-equipos.

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

En sub-equipos, implementar las siguientes funcionalidades:

##### *Sub-equipo A:*

1. Implementar las funcionalidades para incorporar productos al almacén.
2. Compra de más unidades de un cierto producto o Incorporación de uno nuevo.
3. ¿Cuál es el valor económico agregado al stock?
4. Dado un archivo de entrada (“*altasPrueba.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).
5. El archivo “*altasPrueba.txt*” tiene la siguiente estructura (1 producto por línea, campos separados por comas)

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

6. Escribir y ejecutar los casos de prueba necesarios.

### *Sub-equipo B:*

1. Venta de un producto (buscar y reducir su stock de acuerdo a la venta).
2. ¿Cuál es el valor económico reducido del stock?
3. Dado un archivo de entrada ("*ventasPrueba.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 "*ventasPrueba.txt*" tiene la siguiente estructura (1 producto por línea, campos separados por comas)

**CODIGO PRODUCTO, CANTIDAD a VENDER.**

4. Eliminar un producto del almacén, indicando el valor de stock actualizado.
5. Dado un archivo de entrada ("*elimPrueba.txt*" – que contiene un código de producto por línea), eliminar los productos correspondientes del almacén e indicar el **monto total** en que se ha reducido el valor del stock.
6. Escribir y ejecutar los casos de prueba necesarios.

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

Integrar todo el código desarrollado por ambos sub-equipos y sincronizarlo con el repositorio GIT