

**EJERCICIO No. 1** El depósito de drogas **LA SALUD** de la ciudad lleva actualmente un registro manual de la información de los medicamentos que compra a distintos laboratorios proveedores y que vende a distintas droguerías del Departamento del Cesar. Se pide desarrollar e implementar una aplicación de software que permita automatizar el procesamiento de la información que se maneja en el depósito de drogas **LA SALUD**, para lo cual hemos hecho un "levantamiento de información" que nos permitió identificar lo siguiente:

**INFORMACIÓN DE INVENTARIO DE MEDICAMENTOS:** registra la siguiente información por cada medicamento en depósito:

- Refe\_Medicamento:

Referencia del medicamento (Ej: 1005, 2001,3006, etc)
- Nom\_Medicamento:

Nombre del medicamento (Ej: Amoxicilina, Trimetrazol, Etc).
- Presentación

: Como se presenta (Ej: Caja x 50, Solución x 500 ml, etc)
- Cant\_existente

: Cuantas unidades hay actualmente en bodega de este medicamto.
- Máximo

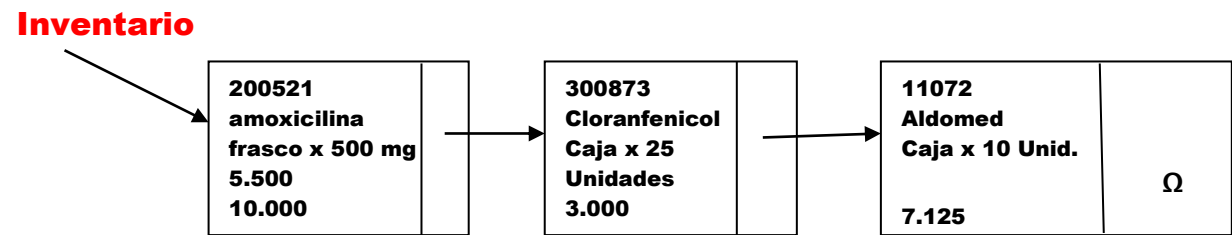
: Máxima cantidad a almacenar por limitaciones de ambiente, de Deposito en bodega, temperatura, etc. Según el medicamento.
- Minimo

: Mínima cantidad a almacenar para garantizar el negocio.

MISIÓN:

Se pide a los alumnos del curso de estructuras de datos desarrollar una aplicación (previo algoritmo en pseudocódigo), que permita automatizar la gestión de información relacionada con este proceso descrito.

Para ello sugiero que utilice UNA LISTA ENLAZADA SIMPLE, así:



MENU PRINCIPAL

- 1.-- Registrar Información de Inventario
- 2.- Gestión de Inventario
- 3.- Fin de la Aplicación

SUBMENU GESTION DE INVENTARIO

- Actualizar Info. De Inventario
- Borrar Registro de Inventario
- Consulta Información de Inventario
- Informe de Inventarios
- RETORNAR AL MENU PRINCIPAL

**EJERCICIO No. 2** Escriba una solución al problema consistente en **RETIRAR** un valor de una Lista Enlazada Simple que contiene números enteros en cada nodo.

**EJERCICIO No. 3** Escriba una nueva versión del algoritmo anterior teniendo en cuenta que ahora deben retirarse **TODOS** los valores que estén **REPETIDOS** en la lista. Tenga en cuenta que un valor se considera *repetido* si aparece 2 o más veces en la estructura. (apóyese en el ejemplo resuelto en clase, pero tenga en cuenta que ahora **NO** se pide "pasar" a otra lista los **REPETIDOS** sino "**BORRAR**" esos nodos de la lista.

**EJERCICIO No. 4** Escriba una solución al problema consistente en **AGREGAR** nombres a una lista de estudiantes de tal manera que solo se admitan estudiantes que **NO** estén ya matriculados, es decir, incluidos en dicha lista.

**EJERCICIO No. 5** Escriba una solución al problema consistente en ACTUALIZAR un campo de información de una Lista Enlazada Simple basado en los datos de Otra lista igualmente enlazada simple. (*A esta actividad se le denomina CRUZAR las listas*).

**EJERCICIO No. 6** Escriba una solución al problema consistente en **RETIRAR** de una Lista Enlazada Simple cuya cabeza es **cab1**, todos aquellos elementos que se encuentren en **OTRA** lista cuya cabeza es **cab2**. Considere que ambas listas contienen únicamente datos numéricos enteros en cada nodo.

**EJERCICIO No. 7** Se tienen **2** listas enlazadas simples cuyas cabezas son: **cab1** y **cab2** respectivamente. Escriba una solución al problema consistente en **AGREGAR** o **RETIRAR** de la lista **cab1**, todos aquellos nodos que en la lista **cab2** tengan en el campo **Accion** un carácter 'A' o 'R', respectivamente.

Considere que las listas 1 y 2 tienen la siguiente **estructura de información** en cada nodo:

```
Struct L1
{
    int   Codigo;
    L1    *SIG;
};
```

```
=====
Struct L2
{
    int   Codigo;
    char   Accion;
    L2    *SIG;
};
```

**EJERCICIO No. 8** Se tiene una Lista enlazada SIMPLE que contiene la siguiente información de Estudiantes:

**CODIGO, NOMBRE, GRUPO, SEMESTRE**

**MISIÓN:** Permitirle al usuario de la aplicación "**CERRAR**" grupos de estudiantes, para lo cual debe pedir el número del **GRUPO** que va a "**CERRAR**" y proceder a "**BORRAR**" de la lista enlazada original **TODOS LOS NODOS** que contengan ese número en el campo **GRUPO**.

Además, todo estudiante que sea "**BORRADO**" de la lista enlazada original debe "**AGREGARSE**" a una segunda Lista Enlazada SIMPLE, cuyo ÚNICO campo de información es el **NOMBRE** del estudiante.

**Nota:** LA SOLUCIÓN A ESTE ULTIMO PUNTO ESTÁ EN DOS (2) ARCHIVOS ANEXOS A ESTE DOCUMENTO