

## Parte práctica

Dada la siguiente estructura de información:

```
typedef struct producto
{
    int codigo;
    char descripcion[20];
    int stock ;
} Producto;
```

- 1) Desarrollar una función que permita la carga de productos en un arreglo dinámico. La función recibe la información de un producto y lo carga dentro del arreglo, retornando la cantidad total de elementos del arreglo. Si coincide el código de producto con un código existente, se deberá **actualizar** el stock disponible, manteniendo la descripción y el código sin modificación.

La función debe cumplir con el siguiente prototipo:

**int CargarProducto (Producto \*\*listado, int cantidadElementosArreglo, int codigo, char \* descripción, int stock);**

En caso de error, la función devuelve (-1) y debe garantizar la liberación de los recursos utilizados hasta ese momento. En caso de éxito, retorna la cantidad total de elementos del arreglo.

- 2) Desarrollar una función que a partir de un arreglo del tipo **Producto**, actualice un archivo, escribiendo al final del mismo, los datos correspondientes a un **código específico** utilizando el siguiente formato:

Codigo,Descripcion,Stock\n

Para realizar la función se pide cumplir con el siguiente prototipo:

**int EscribirInformacion(Producto \* listado, int cantidad, int codigoAFiltrar, char\* nombreArchivo);**

La función devuelve:

- 0 en caso de éxito
- 1 si no pudo crear / abrir el archivo
- 2 si el arreglo se encuentra vacío
- 3 si el código recibido no se encuentra en el arreglo

- 3) Desarrollar un programa que permita validar el funcionamiento de los módulos (1 y 2) previamente desarrollados.

El programa recibe por línea de comando la siguiente información:

**./programa archivoSalida codigoAFiltrar**

El proceso padre cargará los datos utilizando la función del punto (1) y finalizará la carga cuando el código sea 0. En caso de error en la función de carga, se notificará al usuario y finalizará el programa. El proceso de carga se realiza por parte del usuario mediante el teclado.

A continuación, el proceso padre será el encargado de pasarle a un proceso hijo (mediante IPC y no otro mecanismo) el nombre del archivo a generar. El proceso hijo invocará a la función del punto (2). Luego, el proceso hijo le notificará al padre el resultado de la operación y el padre informará al usuario si hubo error o se generó el archivo correctamente.

Procure realizar todas las validaciones y liberaciones de recursos.