



Práctica 3 - Pilas y Colas

1. Implemente pilas utilizando los arreglos dinámicos de números enteros de la práctica 1.
 - a) Escriba un archivo cabecera de nombre `arregloenteros.h` con las declaraciones a exportar para manipular de manera externa los arreglos dinámicos.
 - b) Considere la siguiente definición de pilas que guarda un puntero a un arreglo dinámico y un entero con la última posición ocupada del arreglo:

```
#include "arregloenteros.h"
struct _Pila {
    ArregloEnteros *arr;
    int ultimo;
};
typedef struct _Pila *Pila;
```

- c) Defina las siguientes operaciones básicas de pilas. Procure manipular el arreglo dinámico únicamente a través de las funciones declaradas en su interfaz.
 - i. Pila `pila_crear(int)`: crea e inicializa una nueva pila vacía con la capacidad dada.
 - ii. `void pila_destruir(Pila)`: libera la memoria requerida para la pila.
 - iii. `int pila_es_vacia(Pila)`: retorna 1 si la pila está vacía, y 0 en caso contrario.
 - iv. `int pila_tope(Pila)`: retorna el elemento que se encuentre en el tope de la pila, pero sin eliminarlo.
 - v. `void pila_apilar(Pila, int)`: inserta un elemento en el tope de la pila, en caso que la pila se encuentre llena, deberá aumentar al doble la capacidad del arreglo.
Ayuda: puede llamar a la función `arreglo_enteros_ajustar`.
 - vi. `void pila_desapilar(Pila)`: elimina el elemento que se encuentra en el tope de la pila.
 - vii. `void pila_imprimir(Pila)`: imprime en orden los elementos de la pila.

2. Implemente pilas generales utilizando las listas simplemente enlazadas generales de la práctica 2. Utilice la siguiente declaración:

```
typedef GList Pila;
```

Defina las mismas operaciones que en el ejercicio anterior, ajustando los prototipos según lo requiera. Procure manipular la lista únicamente a través de las funciones declaradas en su interfaz.

3. Escriba una función que utilice una pila para revertir el orden de una lista simplemente enlazada.
4. Implemente colas generales utilizando listas enlazadas generales.
 - a) Piense qué tipo de lista será necesaria (simple, doble, circular, con un puntero al último nodo, u otra) para implementar colas de manera eficiente. En caso de no tener implementada la lista elegida de manera general, provea una implementación de la misma, junto con su respectivo archivo cabecera.

- b) Defina las siguientes operaciones básicas para colas. Nuevamente, procure manipular la lista únicamente a través de las funciones declaradas en su interfaz.
- i. `cola_crear`.
 - ii. `cola_destruir`.
 - iii. `cola_es_vacia`.
 - iv. `cola_inicio` (que retorna el elemento que se encuentre en el inicio de la cola, sin eliminarlo).
 - v. `cola_encolar`.
 - vi. `cola_desencolar`.
 - vii. `cola_imprimir`.