

11. Trabajo Práctico 11 - Estructuras de Datos

Utilizando la siguiente estructura:

```
typedef struct Nodo {  
    char usuario[30];  
    int clave;  
} Nodo;
```

11.1. Ejercicio 1

Se desea implementar un módulo de manejo de arreglos dinámicos (vectores). Para ello, dicho módulo debe poseer las siguientes funciones:

- **int insertar(Nodo** vector, int* largo, Nodo nodo, int posicion);** - Inserta un nodo en la posición indicada.
- **int eliminar(Nodo** vector, int* largo, Nodo* nodo, int posicion);** - Elimina el nodo presente en la posición indicada.
- **int obtener(Nodo* vector, int largo, Nodo* nodo, int posicion);** - Devuelve el valor del nodo presente en la posición indicada.
- **int buscar(Nodo* vector, int largo, Nodo nodo, int* posicion);** - Devuelve la posición del nodo con un valor coincidente al indicado.
- **int mostrar(Nodo* vector, int largo);** - Imprime el contenido del vector.

11.2. Ejercicio 2

Se desea implementar un módulo de manejo de pilas. Para ello, dicho módulo debe poseer las siguientes funciones:

- **int push(Nodo** pila, int* largo, Nodo nodo);** - Inserta un nodo en la pila.
- **int pop(Nodo** pila, int* largo, Nodo* nodo);** - Quita un nodo de la pila.
- **int espiar(Nodo* pila, int largo, Nodo* nodo);** - Devuelve el valor del próximo nodo a quitar de la pila.
- **int mostrar(Nodo* pila, int largo);** - Imprime el contenido de la pila.

11.3. Ejercicio 3

Se desea implementar un módulo de manejo de colas. Para ello, dicho módulo debe poseer las siguientes funciones:

- **int encolar(Nodo** cola, int* largo, Nodo nodo);** - Inserta un nodo en la cola.
- **int desencolar(Nodo** cola, int* largo, Nodo* nodo);** - Quita un nodo de la cola.
- **int espiar(Nodo* cola, int largo, Nodo* nodo);** - Devuelve el valor del próximo nodo a quitar de la cola.
- **int mostrar(Nodo* cola, int largo);** - Imprime el contenido de la cola.

Importante: Todas las funciones deben devolver EXITO o ERROR según corresponda (tipo enumerativo).