



Trabajo Práctico N° 5

Temas: TDA Lista enlazada simple

Año 2025

Formato de presentación

- Subir los archivos realizados en cada ejercicio a un repositorio en github.
- Compartir el enlace en el aula virtual.
- Realizar al menos un commit por cada ejercicio.

1. De acuerdo a la definición de TDA lista simple, implemente el TDA y sus operaciones fundamentales considerando que la lista solo tiene un puntero con la dirección del primer nodo de la lista (HEAD). Luego, implemente la operación *mínimo* que permita extraer el menor valor almacenado en una lista **desordenada**.
2. Implemente una lista con puntero al inicio y al final además de un contador que registre la cantidad de elementos actuales que hay en la lista. Realice las modificaciones necesarias a los métodos agregar y eliminar para que actualice el contador, además de agregar una función que retorne la cantidad de elementos actuales de la lista.
3. Utilizando la implementación de lista del **punto 2**. Realice las siguientes funciones
 - a. Función **buscar_indice** que busque un valor dentro de la lista y retorne un índice numérico (int) correspondiente a su posición lógica dentro de la lista, si el valor no se encuentra retorna -1.
 - b. Función **buscar_por_indice** que busque un valor dentro de la lista según su posición lógica, y si existe retorne un puntero con la dirección del elemento buscado, caso contrario la función debe devolver NULL.
4. Utilizando listas simples implemente:
 - a. TDA Pila y sus operaciones fundamentales.
 - b. TDA Cola y sus operaciones fundamentales (incluya un contador de elementos para la cola).
5. Dada una colección de valores enteros, se pretende ordenar dicha colección aplicando el algoritmo de ordenación por selección. En virtud de ello, se solicita:
 - a. Implemente mediante listas simplemente enlazadas la colección de valores enteros.
 - b. Desarrolle los procedimientos/funciones necesarios para implementar el algoritmo de ordenación por selección para la estructura definida.

- c. Implemente dos métodos que permitan combinar dos listas, donde, el resultado final es una lista única con los elementos de ambas listas. Considerando que:
 - i. Para el primer método no importa el orden de los elementos de la lista.
 - ii. Para el segundo método sí importa el orden de los elementos de la lista.
6. Haciendo uso de listas genéricas que realice:
 - a. Que la lista se cargue con números enteros y realice una función que busque números primos dentro de la lista y los cuente. Muestre el resultado de la cuenta.
 - b. Que la lista se cargue con caracteres y realice la cuenta de mayúsculas y minúsculas dentro de la lista. Muestre los resultados por pantalla.
7. Analice los siguientes fragmentos de código, describa las acciones que realizan y determine sus propósitos.

```
void enigma(Nodo* nodo)
{
    if(nodo!=nullptr)
        if(nodo->getSiguiete() == nullptr)
            cout << nodo->getDato() << endl;
        else{
            enigma(nodo->getSiguiete());
            cout << nodo->getDato() << endl;
        }
}
```

```
int misterio (Nodo* nodo)
{
    if(nodo == nullptr)
        return 0;
    else
        return misterio(nodo->getSiguiete()) + 1;
}
```

```
Nodo* desconocido(Nodo* nodo)
{
    if(nodo == nullptr || nodo->getSiguiete() ==
    nullptr)
        return nodo;
    else
        return desconocido(nodo->getSiguiete());
}
```

Utilice la lista que se muestra a continuación para probar los algoritmos:

