Tarea #6
Estructuras de datos
Carolina Elizabeth Gomez

Introduccion:

La implementación de la clase Lista Enlazada presenta una estructura de datos fundamental en el desarrollo de software, proporcionando una forma dinámica de almacenar y manipular elementos. Basada en nodos enlazados, esta clase ofrece flexibilidad para insertar, eliminar y acceder a elementos de la lista, lo que la hace adecuada para situaciones donde la cantidad de elementos puede variar dinámicamente. El código muestra cómo se puede implementar una lista enlazada en C++, aprovechando las capacidades de plantillas para hacerla reutilizable con diferentes tipos de datos.

Descripción:

La clase nodo representa un nodo en la lista enlazada, que contiene un dato de tipo genérico objeto y un puntero al siguiente nodo. Se implementan constructores para inicializar un nodo con un dato y un puntero al siguiente nodo. La clase Lista Enlazada tiene un nodo de ancla para facilitar las operaciones de inserción y eliminación. El constructor de la clase inicializa este nodo de ancla. El destructor libera la memoria de todos los nodos de la lista.

Métodos principales:

- pushBack(const object &data): Inserta un elemento al final de la lista.
- pushFront(const object &data): Inserta un elemento al principio de la lista.
- push(const object &data, int position): Inserta un elemento en una posición específica en la lista.
- popFront(): Elimina el primer elemento de la lista.
- popBack(): Elimina el último elemento de la lista.
- pop(int position): Elimina el elemento en una posición específica en la lista.
- peek(int position) const: Retorna el elemento en una posición específica sin modificar la lista.
- replace(const object &data, int position): Cambia el valor de un elemento en una posición específica de la lista.
- find(const object &data) const: Busca un elemento en la lista y devuelve su posición (si se encuentra).
- display() const: Muestra todos los elementos de la lista. size() const: Retorna el tamaño de la lista.
- empty() const: Retorna true si la lista está vacía, false en caso contrario.

Conclusiones:

En esta actividad aprendimos como funcionan las listas enlazadas y nos ayudó a reforzar nuestros conocimientos con POO, la lista enlazada tiene una complejidad de búsqueda de O(n) y de inserción O(1).