Lista Doblemente Enlazada

Realizado Por:

Diego Cardenas

Zayra Gutiérrez

Felipe Calvache

Presentado A:

Sebastián Camilo Martínez Reyes

Escuela Colombiana De Ingeniería Julio Garavito

Algoritmos Y Estructuras De Datos

Programa De Ingeniería De Sistemas

Bogotá D.C.

1. **Contexto**

Realizar la implementación de la lista doblemente enlazada usando como referencia el código adjunto ( DoubleLinkedList ) extendiendo la definición de Nodo a tener un apuntador a aun nodo previo

1. **Diseño**

Para implementar la lista doblemente enlazada, agregamos un nuevo atributo al constructor el cual es el nodo previo, y se agregaron dos nuevos métodos los cuales tienen como objetivo obtener el valor previo (get), y asignar el valor previo (set).

* 1. Casos de Prueba

Creamos una lista doblemente enlazada mediante código.

1. **Análisis**
   1. Temporal

* Los métodos get y set de nuestra implementación tiene complejidad O(1).
* El método delete tiene complejidad O(n)
* El método para eliminar duplicados tiene complejidad O(n).
* El método para combinar dos DoubleLinkedList, tiene complejidad O(1).
* El método length tiene complejidad O(n).

Pila y Cola

Realizado Por:

Diego Cardenas

Zayra Gutiérrez

Felipe Calvache

Presentado A:

Sebastián Camilo Martínez Reyes

Escuela Colombiana De Ingeniería Julio Garavito

Algoritmos Y Estructuras De Datos

Programa De Ingeniería De Sistemas

Bogotá D.C.

1. **Contexto**

Realizar la implementación de Pila y Cola (Queue, Stack) en donde en su definición estructural se reuse la definición realizada en el punto 1. En las operaciones CRUD recuerde mantener la política de cada estructura (LIFO - FIFO)

1. **Diseño**
2. Para implementar Stack, esta clase de pila hereda todas las propiedades y métodos de DoubleLinkedList, los cambios son la forma de agregar y eliminar valores porque Stack usa LIFO como estrategia para agregar nuevos elementos y eliminar LIFO.
3. Para implementar Queue, esta clase Queue hereda todas las propiedades y métodos de DoubleLinkedList, lo que cambia la forma en que se agregan y eliminan los valores, ya que las Colas tienen una estrategia de último en entrar, primero en salir para agregar nuevos elementos y FIFO para eliminar a ellos.
   1. Casos de Prueba

Como caso de prueba se utilizó la simulación de una pila de libros donde ingresan 6 libros, se prestan 3 de estos y devuelven uno.

Texto

Descripción generada automáticamenteUna captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

Como caso de prueba se utilizó la simulación la cola de un banco en donde hay 3 personas en fila y llega otra a ser atendida.

Texto

Descripción generada automáticamentePantalla azul con letras blancas

Descripción generada automáticamente con confianza media

1. **Análisis**

* El método pop tiene complejidad O(n).
* El método push tiene complejidad O(n).

3.2

* El método dequeue tiene complejidad O(1).
* El método enqueue tiene complejidad O(n).