

Guía Estructura de Datos:

Listas Enlazadas

Profesora Titular: Karina Rojas.
Profesor Laboratorios: Oscar Plaza de los Reyes.

Instrucciones:

Implementar dos programas uno para listas enlazadas simples y otro para dobles, que incorporen los métodos que se especifican a continuación.

Se pide:

Crear los siguientes métodos:

- **Constructores:** Lista() y ListaDoble()

- **public void insertarFinal(int x)**

Inserta al final de la lista el entero x.

Lista original: Inicio: 3-->1-->4-->2-->

insertarFinal(5): Inicio: 3-->1-->4-->2-->5-->

ListaDoble original: Inicio: 3<-->1<-->4<-->2<-->

insertarFinal(5): Inicio: 3<-->1<-->4<-->2<-->5<-->

- **public void insertarInicio(int x)**

Inserta al inicio de la lista el entero x.

Lista original: Inicio: 3-->1-->4-->2-->

insertarInicio(5): Inicio: 5-->3-->1-->4-->2-->

ListaDoble original: Inicio: 3<-->1<-->4<-->2<-->

insertarInicio(5): Inicio: 5<-->3<-->1<-->4<-->2<-->

- **public void insertarAntes(int x, int antesDe)**

Inserta antes del entero antesDe en la lista el entero x.

Lista original: Inicio: 3-->1-->4-->2-->

insertarAntes (5,1): Inicio: 3-->5-->1-->4-->2-->

ListaDoble original: Inicio: 3<-->1<-->4<-->2<-->

insertarFinal(5,1): Inicio: 3<-->5<-->1<-->4<-->2<-->

- **public void insertarDespues(int x, int despuesDe)**

Inserta despues del entero despuesDe en la lista el entero x.

Lista original: Inicio: 3-->1-->4-->2-->

insertarAntes (5,1): Inicio: 3-->1-->5-->4-->2-->

ListaDoble original: Inicio: 3<-->1<-->4<-->2<-->

insertarFinal(5,1): Inicio: 3<-->1<-->5<-->4<-->2<-->

- **public boolean encontrar(int x)**

Buscar dentro de la lista un valor x, si se encuentra se retorna true y si no esta false

Lista original: Inicio: 3-->1-->4-->2-->

encontrar (5): false

encontrar (1): true

ListaDoble original: Inicio: 3<-->1<-->4<-->2<-->

encontrar (5): false

encontrar (1): true

- **public boolean eliminar(int x)**

Elimina de la lista el entero x.

Lista original: Inicio: 3-->1-->4-->2-->

eliminar (1): Inicio: 3-->5-->4-->2-->

ListaDoble original: Inicio: 3<-->1<-->4<-->2<-->

eliminar (1): Inicio: 3<-->5<-->4<-->2<-->

- **public int contarElementos()**

Retorna la cantidad de elementos que contiene la lista.

Lista original: Inicio: 3-->1-->4-->2-->

contarElementos (): Inicio: 4

ListaDoble original: Inicio: 3<-->1<-->4<-->2<-->

contarElementos (): 4

- **public void intercambiar(int a, int b)**

Este método recibe un entero a y un entero b. Lo que hace es intercambiar en la lista todas las apariciones del elemento a por el elemento b.

Lista original: Inicio: 3-->5-->1-->4-->5-->2-->
 intercambiar(5,8): Inicio: 3-->8-->1-->4-->8-->2-->

ListaDoble original: Inicio: 3-->5-->1-->4-->5-->2-->
 intercambiar(3,8): Inicio: 8-->5-->1-->4-->5-->2-->

- **public void listaMayoresProm()**

Este método calcula el promedio de los elementos contenidos en la lista y solo deja los elementos que son mayores al promedio.

Lista original: Inicio 3-->8-->1-->4-->8-->2-->
 Promedio = 4,33
 listaMayoresProm(): Inicio 8-->8-->

ListaDoble original: Inicio 3<-->8<-->1<-->4<-->8<-->2<-->
 Promedio = 4,33
 listaMayoresProm(): Inicio 8<-->8<-->

- **public String toString()**

Retorna un string que representa a la lista.

Lista original: Inicio: 3-->1-->4-->2-->
 toString(): "3-->1-->4-->2-->"

ListaDoble original: Inicio: 3<-->1<-->4<-->2<-->
 toString(): "3<-->1<-->4<-->2<-->"