GTI0141	Elsy Joselyn Godinez Juarez
25 | 11 | 25	

"Implementacion de una lista Doblemente Enlazada con Nodos en Java"

2. Estructura de Clases en Java

Implementaremos el uso de Tipos Genericos (<T>) y Encapsulamiento para buenas practicas de Poo

Clase Nodo<T>

```
Public class Nodo<T>{
Private Nodo<T> siguente;
Private Nodo<T> anterior;

  // Constructor
Public Nodo(T dato){
   this.dato = dato;
   this.Siguiente = null;
   this.anterior = null;
}
  // Getters y Setters
Public T getDato(){
   this.dato = dato;
}
Public Nodo<T> getDato(){
 return dato;
}
Public Nodo<T> getSiguente(){
 return siguente;
}
Public void setSiguiente(Nodo<T> siguiente){
   this.Siguiente = siguiente;
}
Public Nodo<T> getAnterior){
 this.anterior = anterior;
}
```

Elsy Joselyn Godinez Juarez

C-TIDI41

25 | 11 | 25

3: Operaciones clave con Representacion Visual y clase ListaDoble<T> Completa.

```
Public class ListaDoble<T>{

Private Nodo<T> cabeza;
Private Nodo<T> cola;

Public ListaDoble(){
  this.cabeza = null;
  this.cola = null;
}
//3.1 Agregar al Inicio
Public void agregaAlInicio(T dato){
Nodo<T> nuevoNodo = new Nodo<>(dato);

if (cabeza == null){
   cabeza = nuevoNodo;
   cola = nuevoNodo;
}else{
   // cabeza.actual.anterior = NN
cabeza.setAnterior(nuevoNodo);

  // NN.Siguiente = Cabeza
nuevoNodo.setSiguiente(cabeza);

   // Cabeza = NN
  cabeza = nuevoNodo;
}
//3.2 Eliminar al final
Public void eliminarAlFinal(){
  if(cola != null){
```

```java
if (cabeza == Cola) {
    Cabeza = null;
    Cola = null;
} else {
    // cola = Cola.anterior
Cola = cola.getAnterior();

 // nueva cola.siguiente = null
  Cola.getSiguiente(null);
}
} else {
    System.out.println("La lista esta vacia");
}
}
   // Recorrer hacia adelante
Public void imprimirlista(){
 Nodo<T> actual = cabeza;
System.out.Print("Lista (Adelante): ");

While (actual != null){
 System.out.Print(actual.getDato() + " -> ");
 actual = actual.getSiguiente();
 }
 System.out.Print("null");
}
     // 3.3 recorrer hacia atras
Public Void imprimirHaciaAtras(){
  Nodo<T> actual = cabeza;
System.out.Print("Lista (Atras): ");

While (actual != null){
System.out.Print(actual.getDato() + " -> ");
actual = actual.getAnterior();
}
```

OK
Falta representación Visual
Revisado 25 Nov

Elsy Joselyn Godinez Juarez

C++ ID141
25 A 11 A 25

```
System.out.println("null");
}
```

4: Caso Pratico: Gestion de una cola de Mensajes.

```
Public class Main{
  Public static void Main(String [] args ){

ListaDoble<String> mensaje = new ListaDoble<>();

 // Agregar mensajes al inicio
mensajes. agregarAlinicio ("Error Critico");
mensajes. agregarAlinicio ("Aviso de Pago");
mensajes. agregarAlinicio ("Resumen Diario");

 // Resultado Esperado
// Resumen Diario -> Aviso de Pago -> Error Critico -> null

 // imprimir Hacia adelante
mensaje. imprimirLista();

  // imprimir Hacia atras
mensaje. ImprimirHaciaAtras();

  // Procesar mensaje mas antiguo (eliminar al final)
System. out. println("\n Procesando mensaje mas antiguo...");
mensajes. eliminarAlfinal();

  // Imprimir Lista actualizada
mensajes. imprimirLista();
}
}
```

Empecé ejercicios Prácticos de Pilas y Colas Ordenamiento y Búsqueda