

UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO  
Facultad de Ciencias



**Estructuras de Datos**

*Práctica 2: TDA Nodo*

Profesora:

Amparo López Gaona

Ayudante: Adrián Aguilera Moreno

Ayudante de Laboratorio: Kevin Jair Torres Valencia

## Objetivos

- Comprender el concepto de Tipo de Dato Abstracto (TDA) y su importancia en la programación.
- Aplicar los principios de la programación orientada a objetos para la definición y manipulación de estructuras de datos dinámicas.

## Introducción

Un Tipo de Dato Abstracto (TDA) es una estructura de datos que define un conjunto de operaciones y propiedades sin especificar su implementación interna.

Los tipos de datos abstractos (TDA) son una herramienta fundamental que permite a los desarrolladores definir estructuras de datos sin preocuparse por la implementación subyacente. Estos tipos de datos encapsulan la lógica de datos y las operaciones que se pueden realizar sobre ellos, proporcionando una interfaz clara y sencilla. Ejemplos comunes de TDAs incluyen pilas, colas, listas, entre otras.

## Desarrollo

Implementarán un TDA para representar un nodo de una lista doblemente ligada. A diferencia de la lista simplemente ligada, en esta estructura cada nodo contiene dos referencias: una al siguiente nodo y otra al nodo anterior. Esto permite recorrer la lista en ambas direcciones y facilita ciertas operaciones como la eliminación de nodos intermedios.

Para la implementación del nodo doblemente ligado, se deben seguir los siguientes pasos:

1. Definir una clase genérica `NodoDoble` que contenga:
  - Un atributo para almacenar el dato del nodo.
  - Una referencia al nodo siguiente.
  - Una referencia al nodo anterior.
2. Implementar un constructor que inicialice el nodo con un valor dado.
3. Crear métodos de acceso (get y set) para los atributos de la clase, permitiendo modificar y obtener los valores de los nodos adyacentes.

## Formato de Entrega

1. Las prácticas se entregarán en parejas.
2. Cada práctica (sus archivos y directorios) deberá estar contenida en un directorio llamado EquipoX\_pY, donde:
  - (a) X es el número de equipo correspondiente.
  - (b) Y es el número de la práctica.

Por ejemplo: Equipo09\_p01

3. NO incluir los archivos .class dentro de la carpeta.
4. Los archivos de código fuente deben estar documentados.
5. Se pueden discutir y resolver dudas entre los integrantes del grupo. Pero cualquier práctica plagiada total o parcialmente será penalizada con cero para los involucrados.
6. La práctica se debe subir al Github Classroom correspondiente.
7. La entrega en classroom debe contener el link HTTPS y SSH de su repositorio y es lo único que se debe entregar.
8. El horario y día de entrega se acordará en la clase de laboratorio y no deberá sobrepasar 2 clases de laboratorio.