Programación II – ALGORTIRMOS Y ESTRUCTURAS DE DATOS II

**Docente:** Esp. Lic. PEREZ, Nicolás Ignacio

[nicoperez@uade.edu.ar](mailto:nicoperez@uade.edu.ar)

Repositorio de la materia: <https://github.com/NicolasPerezUNLaSMN/PROG_II_UADE_JAVA>

Facultad de Ingeniería y Ciencias Exactas

Universidad Argentina de la Empresa

Logotipo, nombre de la empresa

El contenido generado por IA puede ser incorrecto.

Programación de Lista Genérica en Java bajo el Paradigma TDA

# 📚 Teoría

## 🔹 Paradigma TDA (Tipo de Datos Abstracto)

- Un TDA describe qué operaciones se pueden hacer sobre un tipo de datos, sin especificar cómo están implementadas.  
- Separamos interfaces (contratos) de implementaciones (código concreto).  
  
Ejemplo de TDA en nuestro trabajo:  
- ILista<T> describe una lista.  
- Lista<T> implementa esa lista.  
- INodo<T> describe un nodo.  
- Nodo<T> implementa el nodo.

## 🔹 Qué es <T> (Tipos Genéricos)

**- <T> representa un parámetro de tipo.  
- Permite crear clases, interfaces y métodos que funcionan para cualquier tipo de dato.  
- 'T' es solo una letra por convención (Type), pero podría llamarse X, Dato, etc.**  
Ejemplo:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.  
  
Ventajas:  
- Reutilización: No duplicamos código para Integer, String, etc.  
- Seguridad: El compilador chequea tipos en tiempo de compilación.

# 📘 Código Implementado

## 1. INodo<T>

Texto

El contenido generado por IA puede ser incorrecto.

## 2. Nodo<T>

Texto

El contenido generado por IA puede ser incorrecto.

## 3. ILista<T>

Texto

El contenido generado por IA puede ser incorrecto.

## 4. Lista<T>

Texto

El contenido generado por IA puede ser incorrecto.

## 5. Main de Pruebas

Texto

El contenido generado por IA puede ser incorrecto.  
}

# 📈 Conclusiones Profesionales

- Siempre se programó usando interfaces para mantener la abstracción.  
- Se aplicó programación genérica con <T> para manejar cualquier tipo de dato.  
- Los métodos permiten insertar, eliminar, buscar, imprimir y verificar si la lista está vacía o su tamaño.  
- Se demostró que la lista funciona con Integer, String y Double.

# Para práctica:

* Descargar el código subido al github en la Clase VII, hacer todas las primitivas de Lista que faltan (están mencionadas en el archivo Lista).
* Crear una clase cualquiera, o reutilizar alguna vieja (Como Persona y vehículo) y testear una lista con esa clase.
* Investigar sobre el concepto de Lista<Objets>, evaluar los pro y contras de este tipo de implementación:   
    
  Texto

  El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Si venís bien con los temas te desafío a que hagas este tipo de lista y me preguntes lo que necesites. (Puede ser una linda pregunta para el 10 en el final ; ) )