

Cristian Rodriguez Rodriguez  
Lunes 24 de Noviembre de 2025.

## Estructuras no Lineales: Árboles y Grafos en Python.

Bienvenidos a nuestra exploración de estructuras de datos no lineales en python. Nos enfocaremos en árboles y grafos, herramientas fundamentales para resolver problemas complejos en informática. Estas estructuras ofrecen flexibilidad y eficiencia en el manejo de datos jerárquicos y relacionales.

### Introducción a las estructuras no lineales.

- Estructuras Lineales: Datos Organizados Secuencialmente, Ejemplos: array, listas enlazadas
- Estructuras no lineales: Datos con relaciones jerárquicas o en red, Ejemplo: árboles, grafos.
- Características: Mayor flexibilidad, eficiencia en búsqueda compleja, representación de relaciones multidimensionales.

### Árboles: Estructura jerárquica.

- 1.- Nodo: Elemento básico que contiene datos y referencias a otros nodos.
- 2.- Arista: Conexión entre nodos que establece relaciones padre-hijo.
- 3.- Raíz: Nodo Superior del árbol, Sin padre.
- 4.- Hoja: Nodo Sin hijo, ubicado en los extremos del árbol.

### Clasificación de árboles

| Tipo                      | Características                        | Uso Común            |
|---------------------------|--|----------------------|
| Árbol binario             | Máximo 2 hijos por nodo                | Expresión algebraica |
| Árbol binario de búsqueda | Ordenado, búsqueda eficiente           | Diccionario, índices |
| Árbol Balanceado          | Altura mínima, operaciones optimizadas | Bases de dato.       |

Cristian Rodriguez Rodriguez

Lunes 27 de Noviembre de 2023.

## Operaciones básicas sobre árboles binarios

1.- Inserción: Agregar nuevo nodo manteniendo la estructura.

Complejidad:  $O(\log n)$  en árboles balanceados

2.- Eliminación: Remover nodo y reorganizar conexiones. Complejidad  $O(\log n)$  en árboles balanceados.

3.- Recorrido: Visitar nodos en Orden específico: inorden, preorden, postorden, Complejidad:  $O(n)$

## Aplicación de árboles

Busqueda eficiente: Árboles binarios de búsqueda para encontrar elementos rápidamente en Colecciones Ordenadas.

Ordenamiento: Heapsort utiliza árboles binarios para ordenar elementos con complejidad  $O(n \log n)$

Estructura de Archivos: Sistemas de archivos utilizan árboles para organizar directorios y Archivos jerárquicamente.

Toma de decisiones: Árboles de decisión en machine learning para Clasificación y predicción.

## Grafos: Nodos y Aristas

• Grafo no dirigido: Conexiones bidireccionales entre nodos. Ejemplo: red Social de amistades

• Grafo dirigido: Conexiones unidireccionales. Ejemplo: Flujo de trabajo en un proyecto

• Características: flexibles para modelar relaciones complejas. Utilizado en redes, mapas y sistemas distribuidos

Cristian Rodriguez Rodriguez  
Lunes 24 de Noviembre de 2025

## Representación y Operaciones en Grafos.

- Representación: Matrices de adyacentes para grafos densos. Listas de adyacencia para grafos dispersos.
- Busqueda en profundidad (DFS): Explora ramas Completas antes de retroceder. Util para detectar ciclos
- Busqueda en amplitud (BFS) Explora todos los vecinos antes de profundizar. Ideal para encontrar Caminos más Cortos