

Video24 (1)

Conceptos Principales

1. a
2. b
3. c

Notas

Árboles de Expresión

Un **árbol de expresión** es un tipo especial de **árbol binario**, pero **no es de búsqueda** (no sigue la regla izquierda < raíz < derecha).

En cambio, se utiliza para **representar expresiones matemáticas o de código**.

Definición

- Cada **nodo interno** representa un **operador** (ej: `+`, `,`, `=`).
- Cada **hoja** representa un **operando** (ej: variables `a`, `b`, `c` o valores numéricos).
- Se utilizan mucho en:
 - **Compiladores** (como parte del Árbol de Sintaxis Abstracta – AST).
 - **Calculadoras** o programas que interpretan expresiones.

Ejemplo

Expresión:

```
a = (a + b) * c
```

Árbol de expresión:

```
      =
     /   \
    a   *
       / \
      (+)  c
         / \
```

a b

Recorridos para evaluación

La evaluación de la expresión depende del tipo de recorrido del árbol:

- **Preorden** → (= a (* (+ a b) c))
- **Inorden** → (a = ((a + b) * c))
- **Postorden** → (a b + c * a =)

En compiladores y calculadoras, normalmente se transforma la expresión **in-fija (normal)** a **post-fija (notación polaca inversa)** para construir el árbol y facilitar su evaluación.

Construcción

1. Se toma la expresión como **texto** (**string**).
 2. Se convierte de **in-fija** → **post-fija**.
 3. Con esa notación postfija se **carga en el árbol**.
 4. Luego se puede recorrer el árbol para **evaluar** o **traducir**.
-

Resumen

- Un árbol de expresión **no es de búsqueda** → se organiza según **operadores y operandos**.
- **Hojas** → operandos.
- **Nodos internos** → operadores.
- Se usa en:
 - Compiladores (AST).
 - Calculadoras y parsers.
- Para evaluarlo, se usan **recorridos** (preorden, inorder, postorden).