PUCE Virtual

DOCUMENTO DE PROFUNDIZACIÓN

Programación III

Laboratorio de Contenido 3: Árbol AVL – Auto-Balanceo

Laboratorio de Contenido 3: Árbol AVL – Auto-Balanceo

Explicación básica

Un árbol AVL es un BST auto-balanceado. Si la diferencia de altura supera 1, aplica rotaciones.

Objetivos

- Entender el factor de equilibrio.
- Implementar inserción con rotación.
- Aplicarlo a monitoreo en tiempo real.

Caso práctico

Insertar:
[10, 20, 30]
Esto genera desbalance → requiere rotación.

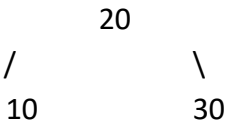
Problema

Balancear automáticamente tras insertar.

Aplicación paso a paso

1. Insertar 10 → raíz.
2. Insertar 20 → derecha.
3. Insertar 30 → desbalance (caso derecha-derecha).
4. Aplicar rotación izquierda.

Resultado:



Implementación en Python

```
class AVLNode:
  def __init__(self, valor):
    self.valor = valor
    self.izquierda = None
    self.derecha = None
    self.altura = 1

class AVL:
  def altura(self, node):
```

```
        return node.altura if node else 0

    def balance(self, node):
        return self.altura(node.izquierda) - self.altura(node.derecha)

    def rotar_izquierda(self, z):
        y = z.derecha
        T2 = y.izquierda
        y.izquierda = z
        z.derecha = T2
        z.altura = 1 + max(self.altura(z.izquierda), self.altura(z.derecha))
        y.altura = 1 + max(self.altura(y.izquierda), self.altura(y.derecha))
        return y

    def insertar(self, node, valor):
        if not node:
            return AVLNode(valor)
        if valor < node.valor:
            node.izquierda = self.insertar(node.izquierda, valor)
        else:
            node.derecha = self.insertar(node.derecha, valor)

        node.altura = 1 + max(self.altura(node.izquierda), self.altura(node.derecha))
        balance = self.balance(node)

        if balance < -1:
            return self.rotar_izquierda(node)

        return node

avl = AVL()
raiz = None
for v in [10, 20, 30]:
    raiz = avl.insertar(raiz, v)

print("AVL balanceado correctamente")
```

Conclusiones

El AVL garantiza rendimiento estable en O(log n).

En ciberseguridad es ideal para:

- Indexación dinámica de logs
- Gestión de IPs sospechosas
- Sistemas SIEM

- Monitoreo en tiempo real