

Trabajo Integrador – Programación I

Tema: Árboles binarios con listas

Alumnos: María Cecilia Fontana – cefontana16@gmail.com / Iván Domínguez – ivan_dominguez2016@hotmail.com

Materia: Programación I

Fecha de entrega: 09 de junio de 2025

Índice

1. Introducción
2. Marco Teórico
3. Caso Práctico
4. Metodología utilizada
5. Resultados obtenidos
6. Conclusiones
7. Bibliografía

1. Introducción

Este trabajo tiene como objetivo implementar y analizar el funcionamiento de un árbol binario de búsqueda utilizando listas y funciones en Python, sin el uso de clases. Se eligieron valores numéricos enteros para facilitar la representación del árbol y observar su estructura mediante distintos recorridos. En programación entender este tema es muy importante ya que ayuda a la optimización de procesos.

2. Marco Teórico

Un árbol binario de búsqueda es una estructura jerárquica donde cada nodo puede tener como máximo dos hijos. El nodo izquierdo contiene valores menores al nodo actual, y el derecho valores mayores o iguales. La representación elegida en este trabajo es mediante listas con la forma: [valor, subárbol izquierdo, subárbol derecho]. Se utilizan funciones recursivas para insertar y recorrer el árbol en tres formas: inorden, preorden y postorden.

3. Caso Práctico

La lista de valores utilizada es: 50, 30, 70, 20, 40, 60, 80. A medida que se insertan en el árbol, se respeta el orden definido por la estructura binaria de búsqueda. El árbol resultante fue recorrido utilizando funciones específicas que imprimen los valores según cada tipo de recorrido. Este procedimiento permite verificar visualmente que la estructura se formó correctamente.

4. Metodología utilizada

1. Definir la estructura del nodo como lista con valor y referencias a izquierda y derecha.
2. Crear funciones para insertar nodos en el árbol respetando el orden binario.
3. Desarrollar funciones recursivas para recorrer el árbol en inorden, preorden y postorden.
4. Ejecutar el programa con una lista de valores predefinidos.
5. Verificar los resultados de los recorridos para asegurar que el árbol fue construido correctamente.

5. Resultados obtenidos

El recorrido inorden mostró los valores en orden creciente, lo cual confirma que la inserción fue exitosa. El preorden y postorden permitieron observar el recorrido desde la raíz y desde las hojas respectivamente. El uso de listas en lugar de objetos no afectó la lógica del árbol, lo que demuestra la versatilidad del lenguaje Python para representar estructuras complejas con herramientas simples.

6. Conclusiones

La experiencia de trabajar con árboles binarios sin clases permitió fortalecer el concepto de recursividad y estructuras dinámicas. El código se mantuvo sencillo y accesible, ideal para un primer acercamiento a esta estructura. Además, se logró cumplir con el objetivo del trabajo integrador usando solamente funciones, listas y condicionales.

7. Bibliografía

- Apuntes de clase de Programación I
- Documentación oficial de Python
- Videos de la cátedra sobre árboles binarios