TÍTULO DEL PROYECTO DE INVESTIGACIÓN[[1]](#footnote-1)

## Alejandra Díaz Navarro. Estudiante – Ingeniería de Sistemas.

Si buscas resultados distintos, no hagas siempre lo mismo.

Albert Einstein

# Resumen

En el presente informe se exponen los errores en el código desarrollado por la docente junto a las adiciones de las funciones al mismo que se encuentran indicadas en la definición del ejercicio del laboratorio.

# INTRODUCCIÓN

En el presente informe se documenta el desarrollo y corrección de una implementación de un Árbol AVL, una estructura de datos auto-balanceada basada en árboles binarios de búsqueda.

El objetivo principal del trabajo es garantizar que las operaciones de inserción mantengan el equilibrio del árbol, lo cual permite asegurar una eficiencia óptima en las búsquedas, inserciones y eliminaciones. Durante el análisis del código proporcionado inicialmente, se detectaron errores relacionados con la lógica de rotaciones, los cuales impedían el correcto balanceo del árbol. Respecto a esto, se realiza una corrección del código, que es ejecutada por el estudiante.

Ahora bien, el código desarrollado por el estudiante cumple con los requisitos establecidos para la implementación de un árbol AVL. Por lo cual, este incluye la clase AVLTree con una clase auxiliar Node, funciones auxiliares para altura y balanceo, rotaciones simples, inserción con mantenimiento del balance AVL, recorrido in-order para visualizar los elementos ordenados y funciones necesarias para asegurar la integridad estructural del árbol.

Además, se ha agrega la funcionalidad de eliminación de nodos con rebalanceo posterior, así como una visualización en consola opcional que permite verificar la estructura del árbol mostrando los nodos, sus alturas y balance; lo cual termina facilitando la validación de que el árbol permanece correctamente balanceado. Estas modificaciones se presentan en el archivo denominado , presente en el link de repositorio de GitHub indicado en los anexos.

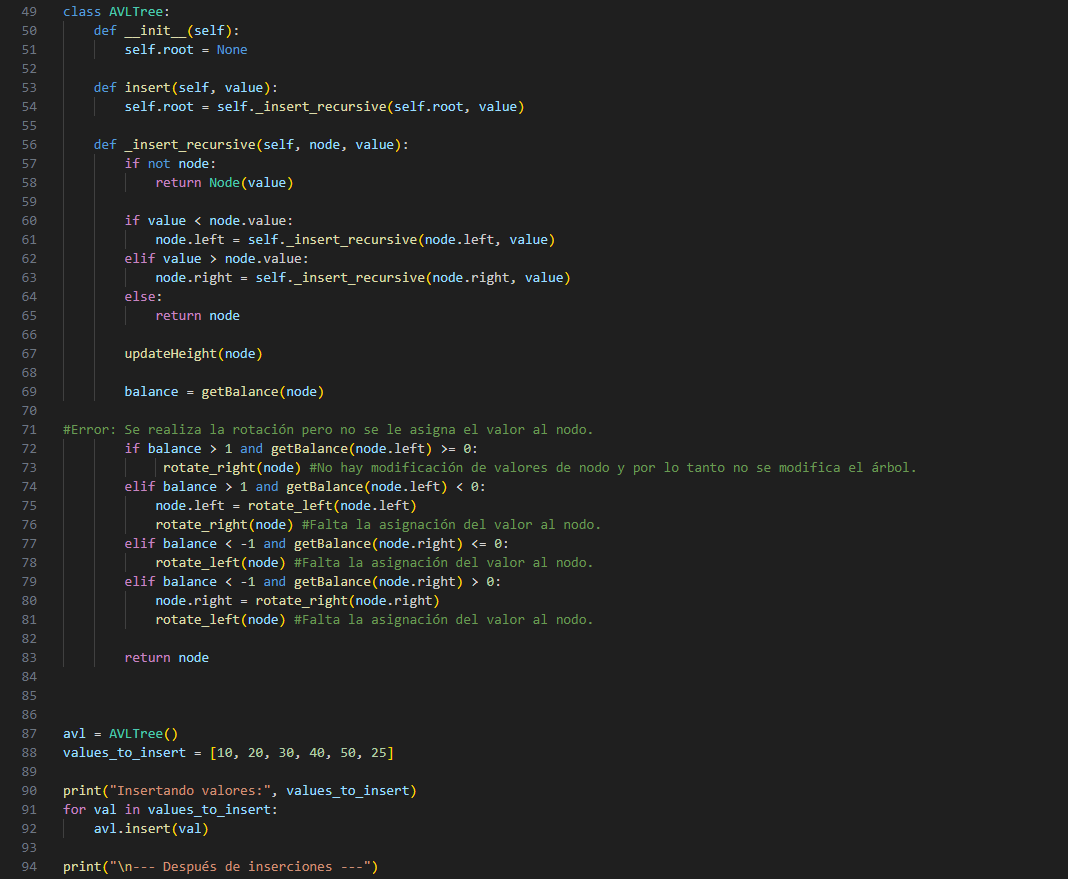
# METODOLOGÍA

El proceso a seguir corresponde a analizar línea por línea el archivo de código, para poder manejar las definiciones de las acciones a ejecutar el árbol AVL. En este, caso se presentan las funciones: *getHeight(node)* obtiene la altura de un nodo, mientras que *getBalance(node)* calcula el factor de balanceo para detectar desequilibrios. La función *updateHeight(node)* actualiza la altura de un nodo después de insertar o rotar. Las funciones *rotate\_left(node)* y *rotate\_right(node)* realizan rotaciones necesarias para reequilibrar el árbol cuando se detecta un desbalance. Finalmente, dentro de la clase AVLTree, la función *insert(value)* y su auxiliar *\_insert\_recursive(node, value)* se encargan de insertar nuevos valores de forma recursiva, ajustando el árbol y aplicando las rotaciones necesarias para mantenerlo balanceado.

# ANÁLISIS DE RESULTADOS.

Mediante el análisis de las funciones presentadas en el código se obtienen las siguientes incongruencias en el mismo.

**Figura 1 Errores presentes en el código.**



**Fuente**: Código LabError en Visual Studio Code.

# CONCLUSIONES

Aquí se describen las conclusiones de la investigación. Las conclusiones deben tener mínimo 200 palabras y estar en coherencia con los objetivos de la investigación y el análisis de resultados. En esta sección es necesario que el estudiante realice un balance sobre la comprensión lograda acerca de los aspectos teóricos y experimentales.

# ANEXOS

En esta sección deben ir las fotografías o imágenes escaneadas de las tablas de datos que registró en hoja de trabajo que usó durante la sesión práctica en el laboratorio, que incluye la fecha y firma de su profesor de Laboratorio.

1. [↑](#footnote-ref-1)