Imagen que contiene Texto

Descripción generada automáticamente

**FACULTAD:**

ING. SISTEMAS COMUTACIONALES

**CURSO DE:**

ESTRUCTURA DE DATOS II

**TEMA:**

TAREA #1

**ESTUDIANTE:**

JOSE G. DONDIS 4-812-1989

**PROFESOR:**

LEONARDO ESQUEDA

**FECHA DE ENTREGA:**

sábado, 11 de junio de 2022, 00:00

1. Examinar el código de programación en Python ([Binary Tree](https://moodle.uip.edu.pa/mod/url/view.php?id=1529055)) y realizar una función para evaluar el factor de equilibrio del árbol que se cree

* Para poder completar esta tarea lo primero que tuve que hacer era crear la clase Height (altura) para definir ciertos parámetros

class Height:

            def \_\_init\_\_(self):

                self.height = 0

* Luego de definir la clase Height debíamos indicar al detalle cada parámetro para que este funcionara correctamente.

def sibalanceado(root, height):

            left\_height = Height()

            right\_height = Height()

            if root is None:

                return True

            l = sibalanceado(root.left, left\_height)

            r = sibalanceado(root.right, right\_height)

            height.height = max(left\_height.height, right\_height.height) + 1

            if abs(left\_height.height - right\_height.height) <= 1:

                return l and r

            return False

        height = Height()

        root = Node(1)

        root.left = Node(2)

        root.right = Node(3)

        root.left.left = Node(4)

        root.left.right = Node(5)

* Una vez termine con definir la función/clase coloque el mensaje que debía aparecer si se cumplió o no con la función. Ósea, si estaba o no estaba balanceado

        if sibalanceado(root, height):

            print('El arbol esta balanceado')

        else:

            print('El arbol no esta balanceado')

nota: Todo esto lo agregue al Binary Tree.py de la plataforma del Moodle. Adjunto a este mini trabajo escrito le estaré enviando el código con la función ya añadida.