**1. Importación de Módulos y Clases:**

from Arbol import AVL, Libro

import pandas as pd

import openpyxl

from openpyxl import load\_workbook

* **Explicación:** En esta sección, se importan los módulos y clases necesarios para el programa.
* **Elementos Teóricos:**
* **“import”:** Es una declaración en que permite importar módulos o clases desde otros archivos .
* **“from ... import ...”:** Permite importar módulos o clases específicas desde un archivo o módulo.
* **“AVL”** y **“Libro”:** Son clases que se utilizarán para crear y manipular el árbol AVL y los libros, respectivamente.
* **“pandas”:** Es una biblioteca de utilizada para manipular y analizar datos.
* **“openpyxl”:** Es una biblioteca de para leer y escribir archivos Excel (.xlsx).
* **“load\_workbook”:** Es una función específica de **“openpyxl”** que permite cargar un archivo Excel en memoria.

**2. Definición de Funciones:**

# Define una función para mostrar el menú de opciones

def mostrar\_menu():

* **Explicación:** En esta sección, se definen varias funciones que serán utilizadas para diversas operaciones en el programa, como mostrar el menú de opciones, buscar libros, insertar libros, entre otras.
* **Elementos Teóricos:**
* **“def”:** Es una declaración en utilizada para definir una función.
* **“def nombre\_funcion():”:** Es la sintaxis básica para definir una función sin parámetros.

**3. Implementación de la Lógica de Negocio:**

# Define una función para buscar libros disponibles en el árbol AVL

def buscar\_libros\_disponibles(arbol):

* + **Explicación:** Esta función busca y muestra los libros disponibles en el árbol AVL.
  + **Elementos Teóricos:**
  + **Recorrido de Árboles:** Se utiliza un recorrido inorden para visitar todos los nodos del árbol, verificando el estado de cada libro y mostrando los detalles de los libros disponibles.

4. Ejecución Principal:

# Crear árbol y cargar datos desde un archivo Excel

arbol = AVL()

arbol.cargar\_libros\_desde\_excel('datapijsonn.xlsx')

Menú principal

while True:

mostrar\_menu() # Muestra el menú de opciones

opcion = input("Seleccione una opción: ") # Solicita al usuario que seleccione una opción

* + **Explicación:** Esta sección inicializa el árbol AVL y carga los datos de los libros desde un archivo Excel. Luego, entra en un bucle principal que muestra un menú de opciones y solicita al usuario que seleccione una opción.
  + **Elementos Teóricos:**
  + **Bucles (“while”):** Permiten repetir un bloque de código mientras se cumpla una condición.
  + **Interacción con el Usuario:** Se utiliza la función “input()” para solicitar al usuario que ingrese datos desde la consola.
  + **Control de Flujo:** Se utiliza una estructura de control “if” para determinar qué acción tomar según la opción seleccionada por el usuario.

5. Interacción con el Usuario:

opcion = input("Seleccione una opción: ") # Solicita al usuario que seleccione una opción

- Explicación: Esta línea de código solicita al usuario que ingrese una opción del menú seleccionando un número.

- Elementos Teóricos:

- Entrada de Usuario: La función “input()” permite al usuario ingresar datos desde la consola.

6. Llamado a Métodos y Funciones:

buscar\_libro\_por\_id(arbol) # Llama a la función para buscar un libro por ID

- Explicación: Este código llama a la función “buscar\_libro\_por\_id()” para buscar un libro por su ID en el árbol AVL.

- Elementos Teóricos:

- Llamadas a Funciones: Se utilizan para ejecutar un bloque de código encapsulado en una función.

7. Gestión de Excepciones:

try:

id = int(input("Ingrese el ID del libro que desea buscar: "))

searchInBFS(arbol, id) # Llama a la función para buscar un libro por ID utilizando BFS

except ValueError:

print("Por favor, ingrese un ID válido.")

- Explicación: Este código intenta convertir la entrada del usuario en un entero. Si la conversión falla (debido a una excepción de tipo “ValueError”), se imprime un mensaje de error.

- Elementos Teóricos:

* + **Gestión de Excepciones (“try”, “except”):** Permite manejar errores y excepciones que pueden ocurrir durante la ejecución del programa.

**8. Actualización de Datos:**

# Devuelve el libro utilizando la función devolver\_libro del árbol AVL

arbol.devolver\_libro(arbol.raiz, id)

* + **Explicación:** Este código utiliza el método “devolver\_libro()” del árbol AVL para marcar un libro como disponible después de que ha sido devuelto por un usuario.
  + **Elementos Teóricos:** 
    - Manipulación de Datos: El programa actualiza el estado del libro en el árbol y en el archivo Excel para reflejar su disponibilidad.

**9. Finalización del Programa:**

print("¡Hasta luego!") # Imprime un mensaje de despedida

break # Sale del bucle while y termina el programa

* + **Explicación:** Estas líneas de código imprimen un mensaje de despedida y terminan la ejecución del programa.
  + **Elementos Teóricos:** 
    - Salida del Programa: Se utiliza la instrucción “break” para salir del bucle “while”, lo que termina la ejecución del programa.

**Explicación del método para la parte grafica del árbol**

**1. Creación de una Nueva Figura:**

plt.figure(figsize=(10, 6)) # Crea una nueva figura para el gráfico con tamaño 10x6 pulgadas

* + - **Explicación:** Esta línea de código crea una nueva figura para el gráfico con un tamaño de 10x6 pulgadas utilizando la biblioteca Matplotlib.
    - **Elementos Teóricos:**
    - **Gráficos:** En este caso, se utiliza la biblioteca Matplotlib para generar visualizaciones de datos, como el árbol AVL.

**2. Llamada al Método `plot\_avl()`:**

self.plot\_avl(self.raiz) # Llama al método plot\_avl para trazar el árbol AVL a partir de la raíz

* + - **Explicación:** Este código llama al método `plot\_avl()` para trazar el árbol AVL a partir de la raíz del árbol.
    - **Elementos Teóricos:**
    - **Llamadas a Métodos:** Los objetos en tienen métodos asociados que se pueden llamar utilizando la sintaxis `objeto.metodo()`. En este caso, se llama al método `plot\_avl()` del objeto actual (`self`) y se pasa la raíz del árbol como argumento.

**3. Mostrar el Gráfico:**

plt.show() # Muestra el gráfico

* + - **Explicación:** Esta línea de código muestra el gráfico generado utilizando la biblioteca Matplotlib.
    - **Elementos Teóricos:**
    - **Mostrar Gráficos:** Para que aparezca el gráfico en la interfaz, se utiliza la función `plt.show()`, que despliega la figura creada anteriormente con todos los elementos gráficos generados.