

## Trabajo Práctico N°2

## Informe Problema 2

En este problema del TP se nos pidió ayudar a Kevin a realizar sus consultas eficientemente implementando una base de datos en memoria principal "Temperaturas\_DB" que utilice internamente un árbol AVL; donde la base de datos debe permitir realizar determinadas operaciones. Por lo que decidimos basarnos, como se pidió, en la estructura de Árbol AVL, que mantiene el orden y el balance automáticamente, lo que hace que todas las operaciones básicas (insertar, eliminar, buscar) tengan una eficiencia bastante buena.

Para hacerlo más ordenado, separamos el código en tres partes principales (módulos):

- NodoArbol (representa cada nodo del árbol).
- AVLTree (maneja la lógica del árbol AVL).
- Temperaturas DB (clase usada como "interfaz" para trabajar con las temperaturas).

Como se podrá observar en el código, en la clase "Temperatura" se puede:

- Guardar una temperatura en una fecha.
- Obtener la temperatura de una fecha específica.
- Eliminar una medición.
- Consultar todas las temperaturas en un rango de fechas.
- Obtener la temperatura mínima, máxima o ambas entre dos fechas.
- Saber cuántas muestras hay cargadas.
- Imprimir el contenido completo del árbol ordenado por fecha.

Además, nos aseguramos de que las fechas queden siempre ordenadas justamente debido al AVL, lo que hace que las búsquedas y recorridos sean rápidos.

## Resolución de las actividades adicionales:

Presentamos la tabla con el análisis del orden de complejidad Big-O para cada uno de los métodos implementados para la clase "Temperaturas\_DB", explicando brevemente el análisis de los mismos.

MÉTODO	COMPLEJIDAD	ANÁLISIS BREVE
guardar_temperatura(temperatura, fecha)	O(log n)	Inserta un nodo en el AVL (fecha como clave).
devolver_temperatura(fecha)	O(log n)	Busca en árbol AVL por clave (fecha).
temperaturas_en_rango(fecha1, fecha2)	O(n)	Recorre todo el árbol en inorden y devuelve una lista ordenada de las fechas.



max_temp_rango(fecha1, fecha2)	O(n)	Recorre toda la lista de temperaturas_ en_rango y devuelve el máximo dentro del rango de fechas.
min_temp_rango(fecha1, fecha2)	O(n)	Recorre toda la lista de temperaturas_ en_rango y devuelve el mínimo dentro del rango de fechas.
temp_extremos_rango(fecha1, fecha2)	O(n)	Recorre toda la lista de temperaturas_ en_rango y devuelve el par (mínimo, máximo).
borrar_temperatura(fecha)	O(log n)	Elimina el nodo por fecha en el AVL y rebalancea si es necesario.
devolver_temperaturas(fecha1, fecha2)	O(n)	Devuelve todas las temperaturas entre dos fechas. Usa recorrido inorden
cantidad_muestras()	O(n)	Cuenta todos los nodos del árbol recorriéndolo completamente, devuelve una suma.
devolver_raiz	O(1)	Devuelve el primer nodo /nodo raíz.
str	O(n)	Construye una cadena recorriendo todo el árbol en orden.

<u>Nota/comentario</u>: aunque el árbol tiene eficiencia logarítmica en búsquedas individuales, todos los métodos que trabajan con rangos hacen un recorrido completo del árbol (inorden), lo que lleva tiempo O(n).

Implementación de una forma de probar el correcto funcionamiento de los métodos de Temperaturas\_DB:

Para probar el funcionamiento correctamente, creamos un bloque if \_\_name\_\_ == "\_\_main\_\_" donde usamos fechas y valores concretos, cubriendo inserciones, búsqueda por fecha, borrado, búsqueda de extremos en un rango, lista "formateada", etc.

Dicho bloque se encuentra al final en el módulo Temperaturas DB.

```
if _name_ == "_main_":
    temp_db = Temperatura()
    temp_db.guardar_temperatura(25, "01/01/2023")
    temp_db.guardar_temperatura(30, "02/01/2023")
    temp_db.guardar_temperatura(20, "03/01/2023")
    temp_db.guardar_temperatura(15, "04/01/2023")
    temp_db.guardar_temperatura(35, "05/01/2023")
    temp_db.guardar_temperatura(28, "06/01/2023")
    temp_db.guardar_temperatura(22, "07/01/2023")
    print("\nTemperaturas del 01/01/2023 al
03/01/2023:",temp_db.devolver_temperaturas("01/01/2023", "07/01/2023"))
```

## Morales, Moreira Yanz, Zampedri AyED 1C2025



```
print("\nTemperatura del
02/01/2023:", temp db.devolver temperatura("02/01/2023"))
        print("\nTemperatura minima:",temp_db.min_temp_rango("01/01/2023",
"07/01/2023"))
       print("\nTemperatura máxima:",temp db.max temp rango("01/01/2023",
"07/01/2023"))
       print("\nValores extremos de
temperatura:",temp_db.temp_extremos_rango("01/01/2023", "07/01/2023"))
        temp db.borrar temperatura("02/01/2023")
        print("\nCantidad de muestras tomadas:",temp db.cantidad muestras())
        temp_db.guardar_temperatura(30, "02/01/2023")
       print("\nRaiz del arbol:",temp db.devolver raiz())
        temp_db.borrar_temperatura("05/01/2023")
        print("\nÁrbol después de boorrar 05/01/2023:\n",temp_db)
        temp db.borrar temperatura("04/01/2023")
        print("\n\(\hat{A}\)rbol despu\(\hat{e}\)s de borrar 04/01/2023:\n", temp db)
        temp_db.guardar_temperatura(25, "05/01/2023")
        print("\n\()arbol despu\()es de volver a insertar 05/01/2023:\n", temp db)
        print("\nRaiz del arbol:",temp_db.devolver_raiz())
```