



Facultad de  
**Ingeniería**

**UNER**

**FACULTAD DE INGENIERÍA**

Algoritmos y Estructuras de Datos

TP N°2: Aplicaciones de estructuras jerárquicas y grafos.

Integrantes:

- Frederich, Rocio
- Merlo, María Fernanda
- Sivila, Matías

Fecha de entrega: 06-06-2025

## **PROBLEMA 2: “Temperaturas DB”:**

En este problema se solicitó crear un registro de temperaturas que utilice internamente un Árbol AVL. Para resolver el problema creamos la clase “AVL” dentro del módulo Arbol\_AVL y la clase “Temperaturas\_DB” siendo la clase AVL la forma de organizar las temperaturas para su manejo, y en la clase Temperaturas\_DB los métodos que fueron pedidos por la cátedra para que opere con la base de datos. Para la clase AVL también se usó internamente la clase “NodoArbol” la cual la inicializamos con los datos de si tiene padre, en caso afirmativo cual es, si tiene hijo izquierdo o hijo derecho, al inicializar estos son None. Y su factor de equilibrio inicializado en 0, así como cada uno tiene la clave y su cargarUtil. La clase AVL se trata de un árbol binario de búsqueda de clase AVL que en todo momento se mantiene equilibrado gracias al factor de equilibrio. La clase “Temperaturas\_DB” se inicializa como un arbol AVL, a este le agregamos le agregamos las temperaturas con los datos de fecha,temperatura(en grados centígrados).

Método	Complejidad Big-O
guardar_temperatura	$O(\log n)$
devolver_temperatura	$O(\log n)$
max_temp_entre_fechas	$O(k + \log n)$
min_temp_entre_fechas	$O(k + \log n)$
temp_extremos_rango	$O(k + \log n)$
devolver_temperaturas	$O(k + \log n)$
cantidad_muestras	$O(1)$
rango_nodos	$O(k + \log n)$
__recorrer_rango	$O(k + \log n)$
borrar_temperatura	$O(\log n)$

A continuación un análisis de estos órdenes de complejidad:

-guardar\_temperatura: Inserta una nueva temperatura en el árbol AVL, y la inserción en AVL toma  $O(\log n)$  debido al balanceo del árbol.

-devolver\_temperatura: Busca una temperatura en el árbol AVL usando la fecha, y la búsqueda en AVL tiene un tiempo de  $O(\log n)$ .

-max\_temp\_rango: Recorre el árbol para encontrar nodos en un rango. El recorrido en AVL tiene un costo  $O(k)$ , donde  $k$  es el número de nodos en el rango, más  $O(\log n)$  para encontrar los límites del rango.

-min\_temp\_rango:Igual que el método anterior, ya que realiza un recorrido en rango para buscar la mínima temperatura

-temp\_extremos\_rango:Hace un recorrido en rango para obtener el mínimo y máximo valor, por lo que tiene la misma complejidad que los métodos anteriores

-borrar\_temperatura: Elimina una temperatura del árbol AVL, y la eliminación en AVL tiene una complejidad de  $O(\log n)$ .

-devolver\_temperaturas: Recorre el árbol en el rango dado para devolver las temperaturas, similar a los métodos anteriores de rango, con  $O(k)$  por el número de nodos en el rango.

-cantidad\_muestras: Este método simplemente devuelve el tamaño del árbol, que se almacena como una variable. El acceso a esta variable toma  $O(1)$ .

-temperaturas\_en\_rango:Este método llama a nodos\_en\_rango, y la complejidad está dominada por el recorrido en el rango que toma  $O(k)$  más el tiempo para encontrar los límites  $O(\log n)$ .

-nodos\_en\_rango: Realiza un recorrido en el árbol AVL para devolver los nodos en un rango, tomando  $O(k)$  más el tiempo para encontrar los nodos iniciales, que es  $O(\log n)$ .

-recorrer\_rango: Este método realiza un recorrido en profundidad sobre los nodos que están en el rango, por lo que tiene un costo proporcional a  $k$ , que es el número de nodos en el rango