

## **Análisis de Eficiencia - Complejidad Big O, Theta y Omega**

Este análisis explora la complejidad computacional de las operaciones implementadas en la aplicación de árbol genealógico. A continuación, se detalla el comportamiento asintótico para cada operación relevante.

### **1. Agregar Miembro (padre, madre, hijo o hermano)**

- **Operación:** La adición de un miembro al árbol implica la creación de un nuevo nodo y posiblemente una o más aristas que conectan al nuevo miembro con sus familiares.
- **Complejidad en el peor caso (Big O):**  $O(1)$ , ya que la operación de agregar un nodo y una arista es constante en un grafo.
- **Complejidad en el caso promedio (Theta):**  $\Theta(1)$ .
- **Complejidad en el mejor caso (Omega):**  $\Omega(1)$ .

### **2. Eliminar Miembro**

- **Operación:** Eliminar un miembro implica remover un nodo y todas las aristas relacionadas. La operación también requiere reorganizar los descendientes si el nodo tiene hijos.
- **Complejidad en el peor caso (Big O):**  $O(n)$ , ya que, en el peor de los casos, la eliminación de un nodo implica la reasignación de sus descendientes. Donde  $n$  es el número de descendientes que deben ser reubicados.
- **Complejidad en el caso promedio (Theta):**  $\Theta(d)$ , siendo  $d$  el número promedio de descendientes directos por nodo.
- **Complejidad en el mejor caso (Omega):**  $\Omega(1)$ , en caso de que el nodo a eliminar no tenga descendientes.

### **3. Buscar Relación**

- **Operación:** La búsqueda de la relación entre dos miembros requiere hallar el camino más corto en un grafo no dirigido.
- **Complejidad en el peor caso (Big O):**  $O(V + E)$ , donde  $V$  es el número de nodos (miembros) y  $E$  es el número de aristas (relaciones). Este es el caso cuando el grafo es grande y contiene muchas conexiones.
- **Complejidad en el caso promedio (Theta):**  $\Theta(V)$ , ya que se exploran nodos hasta hallar la relación deseada.
- **Complejidad en el mejor caso (Omega):**  $\Omega(1)$ , cuando los miembros están directamente conectados, como en el caso de hermanos o padre-hijo.

#### 4. Visualizar Árbol Genealógico

- **Operación:** Dibuja el grafo en una ventana gráfica, lo que implica una complejidad en función del número de nodos y aristas que se deben renderizar.
- **Complejidad en el peor caso (Big O):**  $O(V + E)$ , ya que networkx y matplotlib deben procesar todos los nodos y sus relaciones.
- **Complejidad en el caso promedio (Theta):**  $\Theta(V)$ , ya que se espera que la visualización incremente linealmente con el número de miembros.
- **Complejidad en el mejor caso (Omega):**  $\Omega(1)$ , para un árbol muy pequeño.