Análisis de Eficiencia (Complejidad Temporal)

1. Agregar Persona (`add_person`)

Este método agrega un nodo al grafo.

- **Complejidad Big O (O)**: O(1)
- **Complejidad Theta (Θ)**: Θ(1)
- **Complejidad Omega (Ω)**: Ω(1)

Como agregar un nodo no depende de la cantidad de nodos existentes en el grafo, esta operación es **constante**.

2. Agregar Relación (Padre-Hijo) (`add_relationship`)

Este método agrega una arista entre dos nodos, representando una relación de padre-hijo.

- **Complejidad Big O (O)**: O(1)
- **Complejidad Theta (Θ)**: Θ(1)
- **Complejidad Omega (Ω)**: Ω(1)

Al igual que agregar un nodo, agregar una arista es una operación constante, ya que `networkx` maneja las conexiones directamente y no se requieren cálculos adicionales.

3. Eliminar Persona (`remove_person`)

Elimina un nodo y todas sus aristas asociadas.

- **Complejidad Big O (O)**: O(E)
- **Complejidad Theta (Θ)**: Θ(E)
- **Complejidad Omega (Ω)**: Ω(1)

En el peor de los casos, la persona tiene conexiones con todos los demás nodos (es el nodo más conectado), lo que lleva a una complejidad de O(E). En el mejor caso, el nodo no tiene conexiones, por lo que es Ω(1).

4. Encontrar Relación (`find_relationship`)

Este método busca el camino más corto entre dos personas en el árbol genealógico para determinar su grado de parentesco.

- **Complejidad Big O (O)**: O(N + E)
- **Complejidad Theta (Θ)**: Θ(N + E)
- **Complejidad Omega (Ω)**: Ω(1)

Para encontrar el camino más corto en un grafo, se emplea una búsqueda como BFS (Búsqueda en Anchura). En un grafo dirigido, BFS tiene una complejidad de O(N + E). En el caso ideal donde `person1` y `person2` están directamente conectados, solo se requiere una operación para verificar la relación, resultando en Ω(1).

5. Detectar Endogamia (`detect_inbreeding`)

Este método busca múltiples caminos entre dos nodos para detectar relaciones redundantes o posibles casos de endogamia.

- **Complejidad Big O (O)**: O(N^3)
- **Complejidad Theta (Θ)**: Θ(N^3)
- **Complejidad Omega (Ω)**: Ω(1)

Para cada nodo en el grafo, el método evalúa todos los posibles caminos desde cada otro nodo, lo cual resulta en una complejidad aproximadamente cúbica O(N^3) en el caso más denso. En el mejor de los casos, si no hay conexiones adicionales o ciclos, el método no encuentra rutas múltiples, logrando Ω(1).

6. Guardar y Cargar Árbol (`save_tree` y `load_tree`)

Estas funciones permiten guardar y cargar la estructura del árbol en un archivo JSON.

- **Guardar (save_tree)**:
  - **Complejidad Big O (O)**: O(N + E)
  - **Complejidad Theta (Θ)**: Θ(N + E)
  - **Complejidad Omega (Ω)**: Ω(1)

Para guardar el árbol, el código debe convertir todos los nodos y aristas a formato JSON. En el peor caso, debe procesar todos los nodos y aristas, resultando en O(N + E). En el mejor caso (árbol vacío), solo requiere Ω(1).

- **Cargar (load_tree)**:
  - **Complejidad Big O (O)**: O(N + E)
  - **Complejidad Theta (Θ)**: Θ(N + E)
  - **Complejidad Omega (Ω)**: Ω(1)

Cargar los datos también implica procesar nodos y aristas, lo que resulta en O(N + E). En el mejor caso, el archivo no tiene nodos ni aristas, y la complejidad es Ω(1).

7. Visualización del Árbol (`visualize_tree`)

Genera una representación gráfica del árbol genealógico.

- **Complejidad Big O (O)**: O(N + E)
- **Complejidad Theta (Θ)**: Θ(N + E)
- **Complejidad Omega (Ω)**: Ω(1)

Dibujar el grafo requiere recorrer los nodos y aristas para su representación. En el caso ideal (árbol vacío), no hay nodos ni aristas para dibujar, por lo que es Ω(1).

Resumen de Complejidades

Método	Big O (O)	Theta (Θ)	Omega (Ω)
Agregar Persona	O(1)	Θ(1)	Ω(1)
Agregar Relación	O(1)	Θ(1)	Ω(1)
Eliminar Persona	O(E)	Θ(E)	Ω(1)
Encontrar Relación	O(N + E)	Θ(N + E)	Ω(1)
Detectar Endogamia	O(N^3)	Θ(N^3)	Ω(1)
Guardar Árbol	O(N + E)	Θ(N + E)	Ω(1)
Cargar Árbol	O(N + E)	Θ(N + E)	Ω(1)
Visualizar Árbol	O(N + E)	Θ(N + E)	Ω(1)

Este análisis muestra cómo el rendimiento depende de la cantidad de personas y relaciones en el árbol genealógico, destacando que las funciones de búsqueda y detección de endogamia tienen mayor complejidad, especialmente en árboles densos.