

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY**  
**PROGRAMACIÓN DE ESTRUCTURAS DE DATOS Y ALGORITMOS**

**REFLEXIÓN**

TC1031.501: PROGRAMACIÓN DE ESTRUCTURAS DE DATOS Y ALGORITMOS (GPO 501)

Esteban Sánchez García  
Tecnológico de Monterrey, Campus SON NTE.  
[a01251440@itesm.mx](mailto:a01251440@itesm.mx)

Liga de repositorio: <https://github.com/MichelleArceo/TC1031.501/tree/main/Actividad4.3>

Entre las diferentes estructuras de datos vistos, la estructura de grafos es la más apropiada para el manejo de redes. Además que sirve para reconocer la manera en la que están interconectadas los diferente elementos en una red de información, también permite tener una representación visual fácil de entender de las mismas conexiones.

Entre las diferentes maneras de representar los grafos existen las listas de adyacencias y las matrices. Las matrices permiten ver la distancia entre los vértices y en ese sentido, es mejor para la creación de un mapa físico o diagrama de las conexiones. Mientras tanto, la lista de adyacencia despliega las conexiones de cada elemento en el grafo y los proyecta como una lista de listas; es decir, se crear un listado de los vértices y luego una lista de conexiones de cada vértice.

Para el ejercicio de la entrega Actividad 4.3, se usó la Lista de Adyacencia utilizando Vectores. El beneficio de este método viene con la creación del grafo mismo. Cuando son por medio de vectores, la creación es de  $O(1)$ .

Hay diferentes funciones que se pueden realizar con los Grafos. Durante el transcurso del módulo de Grafos, las funciones que se vieron son:

Operaciones	Eficiencia
Recorridos BFS y DFS	Tiene una complejidad de $O( Vértice  +  Arco )$ . Cuando se realiza el recorrido, se hace en los vértices y en los arcos al mismo tiempo. Por supuesto, esto tiene diferentes casos también. Es $O(1)$ en el mejor de los casos y $O(n^2)$ en el peor de los casos.
TopologicalSort()	Su complejidad es igual a la del recorrido BFS y DFS, ya que son usados para realizar el sorteo. Promedio de $O( Vértice  +  Arco )$ , mejor caso de $O(1)$ y peor de $O(n^2)$ .
isTree()	Promedio de $O( Vértice  +  Arco )$ , mejor caso de $O(1)$ y peor de $O(n^2)$ .
IsBapartite()	Promedio de $O( Vértice  +  Arco )$ , mejor caso de $O(1)$ y peor de $O(n^2)$ .

Como se mencionó anteriormente, el uso principal de los grafos en es aplicaciones de redes. Sean redes de computadoras, redes de Internet, e inclusive las redes sociales usan los grafos para almacenar la relación digital que existe entre un usuario o computadora y otro usuario o computadora. La ventaja de estos es que no se requiere conocer la cantidad de datos en el principio, se pueden ir agregando datos sin mucha dificultad. La mayor desventaja de esta estructura es el almacenaje de memoria.

Por supuesto que existen muchas ventajas obvias de los grafos y se espera que el programa realizado para la entrega “Actividad 4.3” sirva como un ejemplo del poder de los grafos.

## Referencias

- Allen, M. (2013). Data Structures & Algorithm Analysis in C++. Florida, United States of America. Prentice Hall.
- Drozdek, A. (2012). Data Structures and algorithms in C++. Boston, United States of America. Cengage Learning.