

# Tecnológico de Monterrey

Campus Monterrey

Programación de estructuras de datos y algoritmos fundamentales

## Actividad Integral de Grafos

Profesor: Dr. Eduardo Arturo Rodríguez Tello

A01571226 Mariano Barberi Garza  
A00833623 Iván Alberto Romero Wells

22 de Noviembre de 2022

|   |          |
|---|----------|
| <b>Iván</b>                             | <b>3</b> |
| Introducción                            | 3        |
| Marco Teórico                           | 3        |
| Grafo                                   | 3        |
| Algoritmo de Dijkstra                   | 5        |
| Desarrollo                              | 5        |
| Conclusión                              | 6        |
| Referencias                             | 7        |
| <b>Mariano</b>                          | <b>8</b> |
| Importancia y Eficiencia del Uso Grafos |          |
| Métodos Implementados                   |          |
| <i>Grafos</i>                           |          |
| <i>Matriz de Adyacencia</i>             |          |
| <i>Listas de Adyacencia:</i>            |          |
| <i>Representación de Arco</i>           |          |
| <i>Algoritmo de Dijkstra</i>            |          |

## **Introducción**

Los ataques cibernéticos se presentan cada vez con más frecuencia en nuestro mundo digital, en donde, la información de todas las personas conectadas por medio del internet, con el uso de múltiples dispositivos, se pone en riesgo. Un ataque cibernético puede simbolizar un gran riesgo para todos sus afectados, ya que, pone en riesgo la información de los usuarios, abriendo múltiples vulnerabilidades de alto nivel, como acceso a información personal, información bancaria, o “backdoors” que afectan de manera directa a personas y compañías.

El rastreo de los ataques cibernéticos nos permiten, no solo darnos cuenta de que estos mismos suceden, sino que también, nos permiten encontrar un origen y comportamiento del ataque, y dar un seguimiento al mismo. De este modo, el determinar acciones por realizar nos permitirán acabar o disminuir los daños por los ataques.

## **Marco Teórico**

### *Grafo*

Un grafo es definido como una representación que nos permite establecer relaciones entre nodos. Esta representación nos permite analizar de múltiples maneras la información planteada al respecto. Dentro de un grafo, existen dos conceptos claves que nos permiten definir al mismo: los nodos y las conexiones (vértices, aristas o arcos). Los nodos, como fue mencionado anteriormente, son la manera de representar los datos almacenados de manera

individual. Por otra parte, los vértices nos permiten representar las conexiones entre los nodos. La forma en la que los vértices se comportan, nos permiten definir el tipo de grafo en cuestión. Esto, dado que un vértice puede contar con una dirección, en donde será denominado como un grafo dirigido; o puede contar con una ponderación, en donde el grafo será denominado como un grafo ponderado. Así mismo, existen posibles configuraciones entre este tipo de grafos, en donde se puede tener, por ejemplo un grafo ponderado no dirigido, entre otros.

Por otra parte, la forma en la que se representa un grafo también puede cambiar, dependiendo de la manera en que se decida computar. Existen tres principales tipos de representaciones de grafos:

**Matriz de Adyacencia:** en esta representación se utiliza una matriz que representa la adyacencia entre dos nodos, esta matriz tiene un tamaño de  $v \times v$  (siendo  $v$  la cantidad de nodos). Esta representación tiene la ventaja de tener un tiempo de consulta constante. Pero la desventaja de ocupar  $O(v \times v)$  de espacio en memoria, cosa que puede ser muy desventajosa en el caso de un grafo no dirigido, en donde básicamente se tendrá una repetición de datos. Así mismo, esta representación, no permite más de una relación entre nodos (dos, en caso de ser dirigido).

**Listas de Adyacencia:** en esta representación, se utilizan listas por cada nodo, en donde dentro de estas se contienen los vértices de adyacencia. Esta representación es muy utilizada, ya que facilita el manejo de la información; ocupa un espacio de  $O(e)$  en donde  $e$  representa la cantidad de vértices del grafo; así mismo, con una buena implementación, nos permite un tiempo de consulta constante.

**Representación de Arco:** esta representación es sumamente útil, ya que nos permite representar un grafo por los mismos nodos, es decir, que los vértices de adyacencia, están contenidos dentro de los mismos nodos, cosa que puede ser de mucha utilidad en sistemas complejos, donde los nodos y sus vértices contengan información compleja. Una representación por arco implica un espacio de memoria de  $O(v + e)$ , y cuyo tiempo de consulta es de  $O(e)$  (Cabe recalcar que dependiendo de la implementación estas complejidades pueden variar).

### *Algoritmo de Dijkstra*

El algoritmo de Dijkstra es uno de los más conocidos algoritmos, que dado un nodo origen en un grafo, nos permite conocer los caminos más cortos desde nuestro nodo origen hasta los demás nodos. Este algoritmo funciona de una manera simple, a base de una cola de prioridad, empezando desde nuestro nodo origen, por cada nodo que visitemos, consultamos nuestras adyacencias, en el caso en el que no se haya registrado una distancia previa, se toma la distancia desde nuestro nodo actual hasta nuestro nodo origen, más la ponderación entre nuestro nodo actual y el nodo adyacente, en el caso en el que haya existido una distancia, esta se compara con la posible nueva distancia (distancia hasta nodo origen más la ponderación entre nuestro nodo actual y el nodo adyacente), posteriormente, si se hizo un cambio (se asignó una nueva distancia al nodo adyacente) este se agrega a la cola de prioridad. Esto se repite hasta que nuestra cola de prioridad se encuentre vacía.

### **Desarrollo**

Para el desarrollo de esta actividad integradora, se exploró un escenario en donde podemos observar nuestro ataque cibernético de manera relacional, es decir, cada uno de nuestros registros determinan una dirección, de una dirección IP hacia otra, y la ponderación que dicha interacción costó; así como otros factores, como la fecha y la descripción de la incidencia. Con esta información, el camino que se tomó para la representación de los registros dados, fue de un grafo ponderado dirigido; este grafo, fue representado haciendo uso de la representación por listas de adyacencias, considerando que nos permite hacer mejores complejidades de consulta y espacio. Con esta representación, es sencillo obtener los grados respectivos de cada IP, en donde solamente se obtiene el tamaño de la lista de adyacencia para cada IP respectivamente. Una vez obtenida una lista de pares de IP y su grado respectivamente, nos es posible hacer uso de una cola de prioridad o Heap, para determinar las IPs con mayor grado. En este caso, tomamos esto en cuenta para obtener nuestro Boot Master, considerando que presuntamente este puede ser la IP de mayor grado.

Una vez que se haya obtenido el Boot Master, para analizar cuál será el camino más corto entre éste y cada uno de los otros nodos del grafo, se realizó una implementación del algoritmo de Dijkstra, para obtener los menores caminos posibles.

## **Conclusión**

La información en cada situación es dada de múltiples maneras, y el saber hacer uso de una implementación que nos permita utilizar esta información de una manera eficiente, nos permitirá sobrepasar problemas de alta dificultad con una sencillez más razonable y sobre todo, permitir que nuestro poder computacional sea mejor utilizado para la resolución de problemas. La implementación de grafos en este caso es sumamente útil comparado a otras implementaciones previamente realizadas, ya que nos permite determinar de una manera muy

eficiente relaciones entre computadores y de este modo analizar las relaciones entre cada una de una manera sencilla y eficiente.

### **Referencias**

- Menéndez Velázquez, A. (1998). Una breve introducción a la teoría de grafos. Suma.
- Alvarez Nuñez, M. F. (2013). Teoría de grafos.
- Muñoz Jugo, C. M. (2006). Introducción a la teoría de grafos.
- Torrubia, G., & Terrazas, V. (2012). Algoritmo de Dijkstra. Un tutorial interactivo. VII Jornadas de Enseñanza Universitaria de la Informática (JENUI 2001).
- Salas, A. (2008). Acerca del algoritmo de dijkstra. arXiv preprint arXiv:0810.0075.
- Olaya Oliveros, A. (2021). Ataques cibernéticos.

## Importancia y Eficiencia del Uso Grafos

Los ataques cibernéticos que estamos viendo cada día son más poderosos y cada vez es más fácil crear ataques y que estos dañen a los dispositivos de los usuarios, es por eso que es de suma importancia rastrear de dónde vienen estos ataques y saber la como dar seguimiento después de haber encontrado este mismo, pero para defendernos del ataque lo primordial es encontrarlo, ahí es donde entran los grafos. Lo que hace tan importante el uso de grafos para la ciberseguridad es que una base de datos en grafos le da la posibilidad de crear entidades y relaciones de cualquier tipo, esto termina creando muchas formas de las cuales se pueden hacer simulaciones para ver dónde en tu base de datos hay deficiencias.

## Métodos implementados

### *Grafo:*

Un grafo es una estructura matemática que permite modelar mediante una representación gráfica formada por nodos o vértices que muestra a los actores y las aristas que sirven para representar los lazos o relaciones entre los actores, las aristas pueden estar ponderadas, así haciendo que algunas conexiones tengan más importancia que otras.

### *Algoritmo de Dijkstra:*

El algoritmo de Dijkstra es un algoritmo de complejidad  $O(n^2)$  ( $n$  es el número de vértices). Se utiliza para encontrar la ruta más corta o menos



costosa desde cualquier nodo de origen a cualquier otro nodo en el gráfico. El algoritmo fue diseñado por Edsger Wybe Dijkstra en 1959.

### **Bibliografía:**

Orlando. (2021, February 11). *El Rol de los Grafos en la Ciberseguridad: Ciberseguridad*. GraphEverywhere. Retrieved November 22, 2022, from <https://www.grapheverywhere.com/el-rol-de-los-grafos-en-la-ciberseguridad/#:~:text=L a%20Ciberseguridad%20también%20es%20Big%20Data&text=Para%20cubrir%20de %20forma%20fiable,correcta%20con%20el%20contexto%20adecuado.>