

ALGORITMO PARA LA PREVENCIÓN DEL ACOSO SEXUAL CALLEJERO Y LA DELICUENCIA EN MEDELLÍN

Juan Esteban Toro Caraballo
Universidad Eafit
Colombia
jetoroc@eafit.edu.co

Jose Manuel Sanchez
Villegas
Universidad Eafit
Colombia
jmsanchez@eafit.edu.co

Andrea Serna
Universidad Eafit
Colombia
asernac1@eafit.edu.co

Mauricio Toro
Universidad Eafit
Colombia
mtorobe@eafit.edu.co

RESUMEN

En la ciudad de Medellín se han presentado altos índices de delincuencia y acoso sexual callejero. Dadas estas circunstancias, en este proyecto se ha puesto el objetivo de indagar en algoritmos que ayuden a disminuir los índices mencionados. Se planean conseguir estos objetivos a través de diseñar un algoritmo que encuentre una ruta que reduzca los riesgos de sufrir acoso sexual en el espacio público o padecer un acto delictivo, sin dejar de tener en cuenta la longitud del camino. Se escogió el algoritmo de Dijkstra ya que presento muy buenos resultados en calidad de tiempo y ejecución. Para la realización del proyecto se utilizaron diferentes herramientas tecnológicas que permitieron analizar una base de datos, y con ayuda del algoritmo de Dijkstra, observar los caminos que redujeran el acoso sexual en la ciudad de Medellín.

Palabras clave

Camino más corto, acoso sexual callejero, identificación de rutas seguras, prevención del crimen.

1. INTRODUCCIÓN

Reducir el acoso sexual callejero es compromiso de todos: desde las autoridades responsables de la seguridad de las personas en espacios públicos hasta todos aquellos ciudadanos que son testigos de acoso sexual callejero y no hacen nada al respecto. En el caso particular de Medellín, el acoso sexual callejero se concentra en algunas zonas del territorio país; por lo tanto, las personas que deseen evitar estas zonas con el fin de esquivar las situaciones incómodas a las que son sometidas deben tener una solución. Encontrar un algoritmo que no solo permita eludir las áreas con tasas más altas de acoso, sino que también encuentre el camino más corto entre los posibles es una gran alternativa para que los individuos recuperen la seguridad a la hora de transitar espacios públicos.

1.1. Problema

El acoso sexual callejero es una forma de interacción que se da en espacios públicos y consiste en miradas, comentarios, persecuciones, exhibicionismo, silbidos, tocamientos, entre otros comportamientos de connotación sexual que no son consentidos.

Es una problemática que se trata de un tipo de violencia basada en género que sufre la mayoría de las mujeres que transitan las calles de Medellín, desde los diez y doce años

aproximadamente hasta la edad adulta; para ellas significa condicionamientos en su forma de vivir y consecuencias psicológicas negativas. Con ello no se niega que también los hombres sean objeto de otro tipo de violencias asociadas al género; por el contrario, se afirma que lo son de una manera distinta, en otros espacios y con otra afectación particular.

Queda claro que el acoso sexual callejero condiciona el comportamiento de las mujeres en el espacio público y que, por lo tanto, no hay un disfrute igualitario del mismo, sino diferenciado entre mujeres y hombres.

El problema consiste en encontrar un algoritmo que le dé la posibilidad a los peatones de escoger el camino más corto y con menos acoso sexual callejero en la ciudad de Medellín, Colombia.

1.2 Solución

Para ayudar a que las personas no se conviertan en víctimas del acoso sexual callejero en la ciudad de Medellín, nos planteamos como solución la utilización del algoritmo de Dijkstra o también conocido como algoritmo de caminos cortos. Es un algoritmo que nos permite determinar la ruta más corta entre los nodos de un grafo y se puede utilizar para solucionar grandes problemas debido a que tiene buenos resultados en complejidad de tiempo. De esta forma, las personas podrán tomar decisiones anticipadas ya que podrán ver el camino más corto posible y con menor probabilidad de recibir acoso sexual callejero.

1.3 Estructura del artículo

A continuación, en la Sección 2, presentamos trabajos relacionados con el problema. Posteriormente, en la Sección 3, presentamos los conjuntos de datos y los métodos utilizados en esta investigación. En la Sección 4, presentamos el diseño del algoritmo. Después, en la Sección 5, presentamos los resultados. Finalmente, en la Sección 6, discutimos los resultados y proponemos algunas direcciones de trabajo futuro.

2. TRABAJOS RELACIONADOS

A continuación, explicamos cuatro trabajos relacionados con la búsqueda de caminos para prevenir el acoso sexual callejero y la delincuencia en general.

2.1 Planificación de Rutas Seguras

En la India los delitos y los casos de acoso sexual contra las mujeres aumentan a un ritmo exponencial en la última década. En este artículo se trata de encontrar una forma de resolver este creciente número de casos, mediante el diseño de una aplicación para rutas seguras que se puede utilizar para determinar la ruta más segura y rápida en lugar de solo la ruta más rápida. Los factores a tener en cuenta son: Las personas en el área, la iluminación, el número de casos. El algoritmo en que basaron su proyecto fue: El algoritmo de Dijkstra (o algoritmo CCP de Dijkstra's el Camino Mas Corto Primero. Con esto, las mujeres ahora tendrán la opción de otra ruta además de la ruta más rápida que se muestra comúnmente en las aplicaciones de mapeo de rutas.

2.2 Incorporación de un índice de seguridad en búsqueda de caminos

Los problemas de los que se habla en el artículo son: La inseguridad vial de algunas zonas de la ciudad, la distancia de recorrido. Para resolver el problema se tomaron en cuenta la desaceleración de los vehículos, modelos de predicción de accidentes, antecedentes, riesgos de colisión, tráfico pesado, entre otros, lograron algoritmos funcionales, aunque dependiendo del lugar, puede funcionar mejor uno u otro ya que en algunos lugares, los factores como tráfico pueden afectar positivamente, y en otros, de forma contraria.

2.3 Más allá de la ruta más corta: una encuesta sobre la navegación de rutas consciente de la calidad para peatones

En el artículo se buscan alternativas a los sistemas de navegación comunes que basan todos sus esfuerzos en encontrar la ruta más corta o rápida entre dos puntos. En el artículo se plantean la posibilidad de que existan ciertos deseos o expectativas de los usuarios con respecto a las características preferidas de la ruta que van más allá de la longitud de la ruta directa. La calidad de la ruta recomendada debe reflejar estas características deseadas. Los factores que tienen en cuenta son: Seguridad, Salud y Bienestar Físico, Esfuerzo, Exploración y Experiencia placentera. Utilizaron el algoritmo de Dijkstra. En el artículo se propone que los investigadores investiguen el desarrollo de algoritmos de

enrutamiento basados en grupos de múltiples calidades, un marco de evaluación transdominio y algoritmos de enrutamiento que puedan adaptarse a los cambios de las características de la calle y Preferencias del usuario en tiempo real.

2.4 Enrutamiento consciente de la seguridad para

La inseguridad de las calles, delincuencia y como afecta esto a la movilidad, para resolver este problema, se tuvo en cuenta datos de puntos de delincuencia, índices de delincuencia mayores, cercanía de estaciones de policía, iluminación de las calles y las carreteras, se creó un algoritmo capaz de evaluar la mejor ruta a partir de la mayor seguridad partiendo de la iluminación de las carreteras y su contacto con agentes de la ley.

3. MATERIALES Y MÉTODOS

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos de caminos que reducen tanto la distancia como el riesgo de acoso sexual callejero.

3.1 Recogida y tratamiento de datos

El mapa de Medellín se obtuvo de *Open Street Maps* (OSM)¹ y se descargó utilizando la API² OSMnx de Python. El mapa incluye (1) la longitud de cada segmento, en metros; (2) la indicación de si el segmento es de un solo sentido o no, y (3) las representaciones binarias conocidas de las geometrías obtenidas de los metadatos proporcionados por OSM.

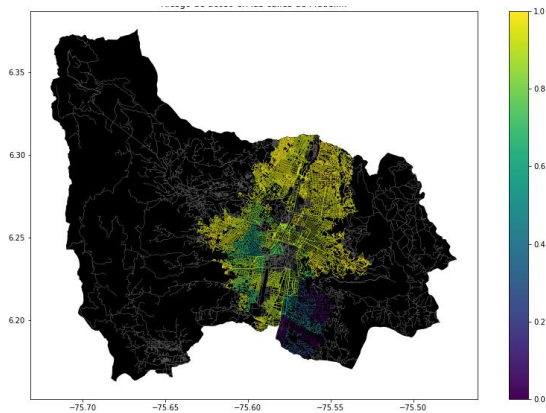
Para este proyecto, se calculó una combinación lineal (CL) que captura la máxima varianza entre (i) la fracción de hogares que se sienten inseguros y (ii) la fracción de hogares con ingresos inferiores a un salario mínimo. Estos datos se obtuvieron de la encuesta de calidad de vida de Medellín, de 2017. La CL se normalizó, utilizando el máximo y el mínimo, para obtener valores entre 0 y 1. La CL se obtuvo mediante el análisis de componentes principales. El riesgo de acoso se define como uno menos la CL normalizada. La Figura 1 presenta el riesgo de acoso calculado. El mapa está disponible en GitHub³.

Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

³ <https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets/>



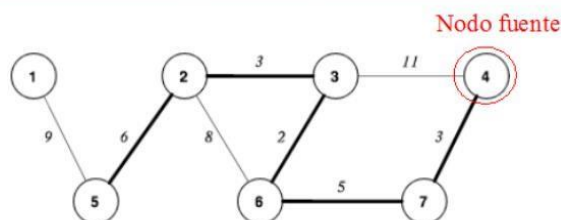
un salario mínimo, obtenidas de la Encuesta de Calidad de Vida de Medellín, de 2017.

3.2 Alternativas de caminos que reducen el riesgo de acoso sexual callejero y distancia

A continuación, presentamos diferentes algoritmos utilizados para un camino que reduce tanto el acoso sexual callejero como la distancia.

3.2.1 El algoritmo de Dijkstra

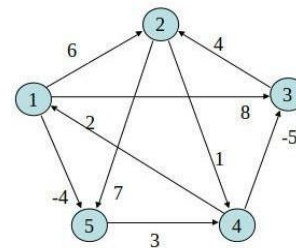
Es un algoritmo eficiente (de complejidad $O(n^2)$, donde “n” es el número de vértices) que sirve para encontrar el camino de coste mínimo desde un nodo origen a todos los demás nodos del grafo. Fue diseñado por el holandés Edsger Wybe Dijkstra en 1959. Este algoritmo es un típico ejemplo de algoritmo ávido, que resuelve los problemas en sucesivos pasos, seleccionando en cada paso la solución óptima con el objeto de resolver el problema



3.2.2 El algoritmo de Floyd-Warshall

Es la opción utilizada cuando se desea determinar el camino mínimo entre todos los pares de vértices de un grafo, comparando todos los posibles caminos logra mejorar

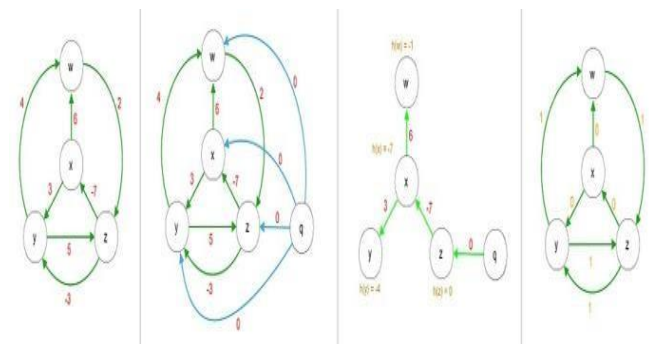
paulatinamente la estimación hasta llegar a la óptima. Complejidad: $O(n^3)$



	1	2	3	4	5
1	0	6	8	∞	-4
2	∞	0	∞	1	7
3	∞	4	0	∞	∞
4	2	∞	-5	0	∞
5	∞	∞	∞	3	0

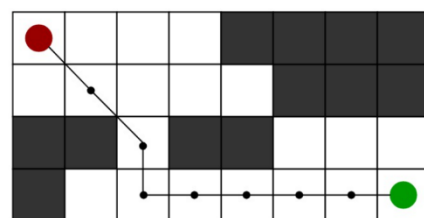
3.2.3 El algoritmo de Johnson

Es una forma de encontrar el camino más corto entre todos los pares de vértices de un grafo dirigido disperso. Permite que las aristas tengan pesos negativos, si bien no permite ciclos de peso negativo. Funciona utilizando el algoritmo de Bellman-Ford para hacer una transformación en el grafo inicial que elimina todas las aristas de peso negativo, permitiendo por tanto usar el algoritmo de Dijkstra en el grafo transformado. Su nombre viene de Donald B. Johnson, quien fuera el primero en publicar la técnica en 1977.



3.2.4 Algoritmo A*

Es un algoritmo de búsqueda inteligente o informada que busca el camino más corto desde un estado inicial al estado meta a través de un espacio de problema, usando una heurística óptima. Como ignora los pasos más cortos (más "chatos") en algunos casos rinde una solución



subóptima.

4. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

A continuación, explicamos las estructuras de datos y los algoritmos utilizados en este trabajo. Las implementaciones de las estructuras de datos y los algoritmos están disponibles en Github⁴.

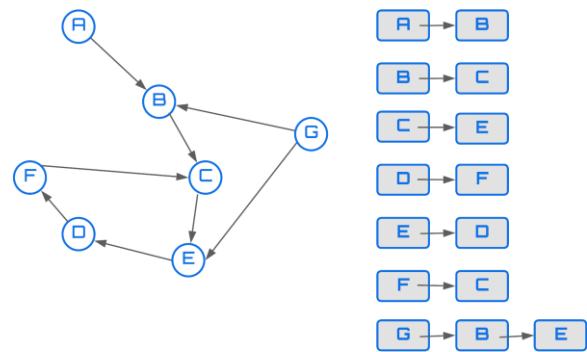
4.1 Estructuras de datos

Para representar el mapa de Medellín escogimos la lista de adyacencia utilizando diccionarios. En primera instancia, hicimos uso de la librería Pandas, que nos da la posibilidad de analizar, explorar y manipular datos. Las coordenadas de la ciudad de Medellín venían contenidas en un archivo csv, pero gracias a la utilización de Pandas, las podemos trabajar en un DataFrame. Posteriormente, creamos un diccionario que se define como una colección de datos que se almacenan como pares claves, de esta forma: Clave → Valor. Por último, representamos el grafo con una lista de adyacencia que es una estructura de datos con la idea de asociar cada vértice i del grafo con una lista que contenga todos aquellos vértices j que sean adyacentes a él. De esta forma sólo reservará memoria para los arcos adyacentes a i y no para todos los posibles arcos que pudieran tener como origen i y así las búsquedas son más rápidas al costo de almacenamiento extra. La escogimos debido a que podemos llegar a la lista de adyacencia de cada vértice en un tiempo constante, porque solo tenemos que indexar en un arreglo.

También se utilizaron las colas de prioridad para obtener el camino con menor distancia. Una cola de prioridad es una poderosa herramienta que puede resolver problemas tan variados como escribir un programador de correo electrónico, encontrar el camino más corto en un mapa o fusionar archivos de registro. La programación está llena de problemas de optimización en los que el objetivo es encontrar el mejor elemento. Las colas de prioridad y las funciones del módulo `heapq` de Python pueden ayudar a menudo a ello.

La estructura de los datos se presenta en la Figura 2.

Figura 2:



4.2 Algoritmos

En este trabajo, proponemos un algoritmo para un camino que minimiza tanto la distancia como el riesgo de acoso sexual callejero.

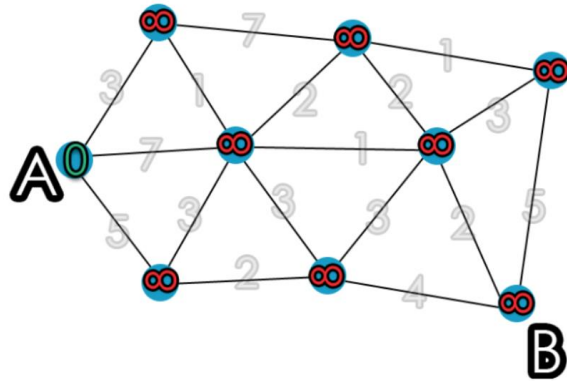
4.2.1 Algoritmo para un camino que reduce tanto la distancia como el riesgo de acoso sexual callejero

Para solucionar el problema escogimos el algoritmo Dijkstra ya que es un algoritmo óptimo para trabajar con problemas grandes y presenta buenos resultados en complejidad de tiempo. El algoritmo de Dijkstra ha visto cambios a lo largo de los años y existen varias versiones y variaciones. Originalmente se usó para calcular el camino más corto entre dos nodos. Debido a la forma en que funciona, se adaptó para calcular la ruta más corta entre un nodo inicial y cualquier otro nodo en el grafo. El fundamento sobre el que se basa este algoritmo es el principio de optimizar: si el camino más corto entre los vértices “ u ” y “ v ” pasa por el vértice “ w ”, entonces la parte del camino que va de “ w ” a “ v ” debe ser el camino más corto entre todos los caminos que van de “ w ” a “ v ”. De esta manera, se van construyendo sucesivamente los caminos de coste mínimo desde un vértice inicial hasta cada uno de los vértices del grafo, y se utilizan los caminos conseguidos como parte de los nuevos caminos.

El algoritmo se ejemplifica en la Figura 3.

⁴ <http://www.github.com/?????????/.../proyecto/>

Figura



3:

Se implemento el algoritmo Dijkstra con lista de adyacencia y colas de prioridad en los vértices. El caso más desfavorable, sería que el algoritmo tenga que recorrer todas las intersecciones(vértices) que se añadan a la cola de prioridad y tenga que analizar también por todas las intersecciones siguientes para determinar cual tiene menor distancia. De esta forma se concluye que la complejidad del algoritmo es de $O(E+V) + O(\log(V))$

Algoritmo	Complejidad temporal
Algoritmo de Dijkstra	$O((E+V)*\log(V))$

Tabla 1: Complejidad temporal del algoritmo de Dijkstra

4.2.2 Cálculo de otros dos caminos para reducir tanto la distancia como el riesgo de acoso sexual callejero

En algoritmo de Dijkstra combinamos las variables de distancia y acoso de diferentes maneras para obtener buenos resultados. Se pudo observar como la relación de las tres variables priorizo reducir la distancia, sin embargo, también tuvo en cuenta el acoso en las calles de la ciudad de Medellín. El camino de color verde fue diseñado elevando al cuadrado ambas variables (distancia y acoso) y sumándolas entre sí. Se puede observar que el camino de color azul fue pensado para encontrar las menores distancias posibles y se obtiene a partir de la suma entre la distancia y el acoso. Y el camino rojo se obtuvo a partir de multiplicar los cuadrados de las variables analizadas-El algoritmo se ejemplifica en la Figura 4.

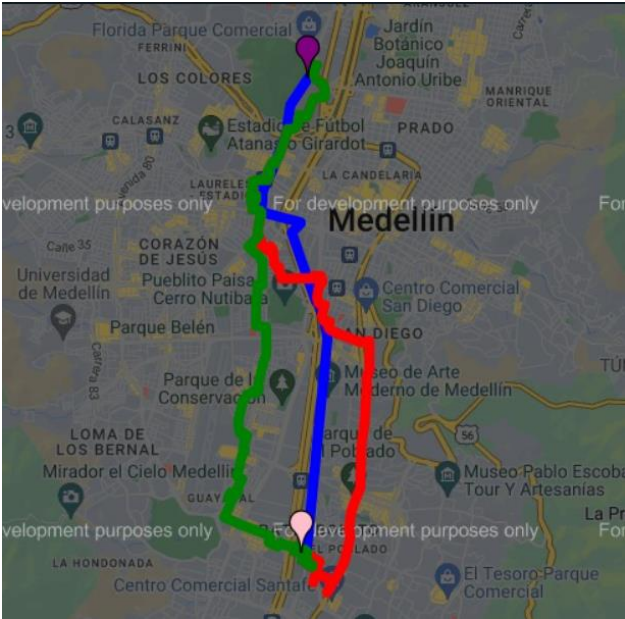


Figura 4: Mapa de la ciudad de Medellín donde se presentan tres caminos para peatones que reducen tanto el riesgo de acoso sexual como la distancia en metros entre la Universidad EAFIT y la Universidad Nacional.

4.3 Análisis de la complejidad del algoritmo

Estructura de datos	Complejidad de la memoria
Lista de adyacencia	$O(V)$

Tabla 2: Complejidad de memoria de la lista de adyacencia

4.4 Criterios de diseño del algoritmo

Desde un principio, se busco que el algoritmo fuera rápido y lo más efectivo posible. El algoritmo de Dijkstra cumplía con las características que buscábamos, y lo implementamos con las colas de prioridad, ya que son muy útiles cuando se quiere calcular distancias cortas. También buscamos que el algoritmo relacionar lo visto en las clases con nuestras investigaciones para completar el conocimiento y comprender de mejor manera lo que se estaba realizando. Para graficar los caminos que resultaban del algoritmo se decidió utilizar gmpplot ya que facilitaba en gran medida la visualización de estos. También, gmpplot nos permite jugar con los colores y de esta forma obtener un resultado útil y estético.

5. RESULTADOS

En esta sección, presentamos algunos resultados cuantitativos sobre los tres caminos que reducen tanto la distancia como el riesgo de acoso sexual callejero.

5.1 Resultados del camino que reduce tanto la distancia como el riesgo de acoso sexual callejero

A continuación, presentamos los resultados obtenidos de tres caminos que reducen tanto la distancia como el acoso, en la Tabla 3.

Origen	Destino	Distancia	Riesgo
Eafit	Unal	8458	0.67
Eafit	Unal	11888m	0.63
Eafit	Unal	11433m	0.80

Tabla 3. Distancia en metros y riesgo de acoso sexual callejero (entre 0 y 1) para ir desde la Universidad EAFIT hasta la Universidad Nacional caminando.

5.2 Tiempos de ejecución del algoritmo

En la Tabla 4, explicamos la relación de los tiempos medios de ejecución de las consultas presentadas en la Tabla 3.

Cálculo de v	Tiempos medios de ejecución (s)
$v = \text{distancia} + \text{acoso}$	1.00465 s
$v = (\text{distancia} * \text{acoso}) * 2$	1.04324 s
$v = \text{distancia} ** 2 + \text{acoso} ** 2$	1.00345s

6. CONCLUSIONES

Se obtuvieron buenos resultados en calidad de tiempo y ejecución. Los tres caminos, a pesar de ser diferentes, comparten en común un enfoque a reducir la distancia entre un punto y otro. Con ese programa, los ciudadanos de Medellín pueden transportarse de un lugar otro de forma optima y segura. El algoritmo Dijkstra es funcional y veloz para determinar el camino más corto lo que fue beneficioso para la realización de este proyecto. Debido a los buenos resultados concluimos que los tres caminos son eficientes y se complementan entre sí. Estamos satisfechos con los resultados obtenidos, especialmente, por tener la posibilidad de analizar los caminos de forma gráfica.

6.1 Trabajos futuros

Nos gustaría optimizar el método que trata y manipula la base datos suministrada para la realización de este proyecto. Con esto en mente, el tiempo de ejecución se va a reducir de forma sustancial. Nos gustaría seguir realizando proyectos relacionados a la utilización de bases de datos y estructuras de datos ya que si se les da un buen manejo permiten modelar soluciones de problemas de la vida cotidiana.

AGRADECIMIENTOS

Los autores agradecen al profesor Juan Carlos Duque, de la universidad Eafit, por facilitar los datos de la Encuesta de Calidad de Vida de Medellín, de 2017, procesados en un archivo Shapefile.

Los autores agradecen al exprofesor Mauricio Toro, de la universidad Eafit por motivar y enseñar siempre de la mejor manera a los alumnados interesados en superarse.

REFERENCIAS

1. Gloria Sánchez Torrubia. Algoritmo de Dijkstra. Un Tutorial Interactivo. <https://bioinfo.uib.es/~joemiro/aenui/procJenui/ProcWeb/actas2001/saalg223.pdf>
2. Cristian Bastidas. Algoritmo de Dijkstra en Python. <https://cristianbastidas.com/my-blog/algorithms/python/tutorial/wiki/2021/04/20/python-dijkstra.html>
3. Bryan Salazar Lopez. Algoritmo de Dijkstra. <https://www.ingenieriaindustrialonline.com/investigacion-de-operaciones/algoritmo-de-dijkstra/>
4. Juan Carlos Villa. Representar grafos. <https://es.khanacademy.org/computing/computer-science/algorithms/graph-representation/a/representing-graphs>
5. Bala Priya. Diccionarios en Python. <https://www.freecodecamp.org/news/dictionary-comprehension-in-python-explained-with-examples/>
6. Programación IV. Guía No. 10. Algoritmos para la ruta más corta en un Grafo. https://www.udb.edu.sv/udb_files/recursos_guias/informatica-ingenieria/programacion-iv/2019/ii/guia-10.pdf
7. Aryan Gupta. Bhavye Khetan. “Safe-Route-Planning”. International Journal of Science and Research (IJSR). 10/2020.