

Algoritmo para la prevención del acoso sexual en las calles de Medellín mediante la búsqueda de rutas seguras.

Juan David Zapata Moncada
Universidad Eafit
Colombia
jdzapatam@eafit.edu.co

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

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

RESUMEN

El problema es la inseguridad en las calles, por el acoso, la delincuencia, los accidentes y todo aquello que perturba a los ciudadanos y que no se pueda transcurrir tranquilamente sin el temor de que suceda algo malo. Este problema es muy importante debido a que se ven perjudicados todas las personas y empresas que sitúan en los alrededores donde se frecuente dicho problema. Los problemas relacionados pueden ser aquellos robos a turistas, a tiendas, a personas que viven en esa misma zona y esto se produce por una gran tasa de desempleo.

¿Qué resultados cuantitativos has obtenido?

El algoritmo propuesto para la resolución del problema de acoso y distancia para peatones en Medellín es el Dijkstra.

Los tres caminos propuestos presentan ciertas diferencias el primero ($d \cdot r$) regula muy bien tanto la distancia como el acoso encontrando un punto medio; el segundo ($d \cdot (r \cdot 10)$) prioriza más el valor del riesgo de acoso por lo que propone caminos más largos con la finalidad de reducir notablemente el riesgo y el tercero ($30 \cdot d + 500 \cdot r$) propone un camino corto, aunque el riesgo sin importar el camino tiene un valor muy cercano a 0.7. Por otro lado, el algoritmo puede llegar a ser muy útil cuando hablamos de personas que transitan las calles a pie durante la noche, aunque los tiempos de ejecución deben optimizarse para que sea útil en la realidad pues por camino el algoritmo tarda alrededor de 2 minutos. Finalmente, para una aplicación recomendaría el primer camino pues es uno que deja en un punto medio los valores de distancia y riesgo de acoso.

Palabras clave

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

1. INTRODUCCIÓN

Se plantea diseñar un algoritmo que determine una ruta segura y rápida para el beneficio del usuario, se realiza debido a que la tasa de criminalidad y acoso sexual han aumentado a unas grandes cifras y se requiere una pronta intervención para disminuir este gran problema. Este trabajo se realiza para construir una sociedad más segura.

1.1. Problema

Actualmente el acoso es uno de los problemas que encabezan la lista de los más graves y comunes en el mundo y miles de personas sufren secuelas psicológicas y físicas que en ocasiones les generan trastornos todo gracias al abuso de otro. Esto impacta la sociedad negativamente ya que esto perjudica el desempeño en el trabajo por lo que indirectamente las empresas se ven afectadas y el deterioro de las relaciones sociales a causa de la desconfianza.

1.2 Solución

La solución la obtenemos mediante el algoritmo dijkstra el cual va comparando valores de nodos vecinos para encontrar el camino que establezca la distancia y el valor del riesgo de acoso, esto lo hacemos sumando la distancia y el riesgo multiplicado por la distancia ($d + (d \cdot h)$) con el valor resultante es con el que trabaja el algoritmo dijkstra. Lo elegimos porque nos pareció una solución práctica y poco compleja de implementar.

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.

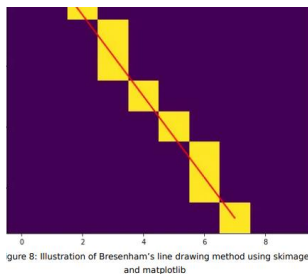
2. TRABAJOS RELACIONADOS

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

2.1 Prevención del acoso sexual a través de un algoritmo de búsqueda de rutas utilizando la búsqueda cercana

Este trabajo surge por la necesidad de encontrar puntos seguros y rutas con el menor riesgo de acoso posible en relación con una dirección y un desplazamiento de (x, y) punto inicial a (x, y) destino. Para llevar a cabo este objetivo buscaron la forma de crear un algoritmo de búsqueda de caminos para prevenir el acoso sexual y lo hicieron utilizando un mapa de calor, dividiendo el área en una cuadrícula de $\sim 1,74 \text{ km}^2$ cada cuadro a la cual se le asignaba una puntuación de riesgo que pueden ir de 0 hasta 4, siendo 0 el lugar más seguro y 4 siendo el más peligroso.

Para determinar la mejor ruta lo que se hizo fue que calcular el promedio de las puntuaciones de riesgo de los cuadros por donde pasan las ruta lo cual, para determinar la mejor implica encontrar la que tiene menor distancia euclidiana mediante su ecuación. Por otro lado, para que se pudiera saber si una ruta tocaba una cuadrícula realizaron al método del dibujo lineal de Bresenham, el cual, es un algoritmo que según la distancia de línea a punto determina si sombreara el punto de dicha intersección. Esto se hace con la intención de tomar los valores de riesgo de los cuadros donde están los puntos y que quede así:



Finalmente se puede ver que algunos cuadros no fueron incluidos en el cálculo por lo que optaron por incluir otro algoritmo que se encargue de eso y que su función es encontrar la “supercubierta” del segmento de línea. El resultado fue un algoritmo que recibía un inicio y un destino y te daba una ruta que prioriza la seguridad antes que la distancia más reducida.

2.2 Un sistema de integración y análisis de datos para la planificación de rutas seguras

Este trabajo surge por un gran índice de casos de acosos sexual contra las mujeres y además por el aumento de criminalidad causada por el desempleo y la pérdida de recursos por el periodo del COVID-19. Se da la idea de diseñar una aplicación que puede determinar las rutas más seguras y a su vez más rápidas. Unas de las maneras más eficaces para resolver el problema es desarrollar un modelo que nos brinde con exactitud qué tan seguro es un lugar. Este utilizaría varios insumos, que podrían influir en la tasa relativa de acoso en un área. A pesar de lo complicado que es determinar una ruta 100% segura se puede dar de que la ruta no sea la rápida posible y es por eso que debe haber un equilibrio.

Para darle solución se realizaron encuestas a varias mujeres preguntándoles que condiciones las haría sentir seguras y cuales las harían sentir inseguras.

El algoritmo se ensaya con suponiendo que hay 5 rutas:

y_1, y_2, y_3, y_4, y_5 , cada valor será de 0 a 100 siendo la máxima seguridad 500 y en cada ruta hay una diferente situación y se trata de determinar cuál es aquella que cumple con más de las especificaciones en las encuestas realizadas a varias mujeres.

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

Este trabajo surge de la funcionalidad de grandes sistemas como por ejemplo Google Maps, queriendo profundizar su función dándole un mejor beneficio al usuario implementando algoritmos que den las rutas con menos probabilidad de riesgo. Se propone que la información que se necesita para crear dicho algoritmo sea brindada por la propia sociedad en distintas redes sociales.

Se define una ruta de navegación consciente de la calidad sistema para peatones basado en la siguiente característica:

El sistema brinda al peatón una ruta que le permite moverse de un punto A hasta un punto B y cuando el usuario ingresa más puntos para visitar en la ruta, la tarea se convierte en encontrar una serie de rutas entre ubicaciones intermedias consecutivas X_i a X_{i+1} dentro de una ruta.

Con base en la literatura identificada, se proporciona una descripción general de los sistemas de navegación de ruta conscientes de la calidad que han sido propuesto en base a los atributos que se utilizan para representar las cualidades de una ruta, el tipo de datos que se han utilizado para implementar tales sistemas, los algoritmos utilizados para implementarlos y los enfoques de evaluación utilizados para implementar la ruta consciente de la calidad de los sistemas de navegación.



2.4 ruta segura: un modelo sólido para la predicción de rutas más seguras utilizando datos sobre delitos y accidentes

Este trabajo surge por la inseguridad en las ciudades y esto es causado por todos los casos de delincuencia. Incluso los mapas de Google pueden dar una ruta rápida pero no segura. Está claro que las personas locales a la zona conocen mejor las calles y sus peligros, pero los turistas no tienen ni idea y están confiando en una aplicación que no está diseñada para garantizar seguridad, por ende, el gobierno reúne datos sobre crímenes, accidentes, rutas, etc. para diseñar un sistema que pueda brindar una mayor función que la anterior mencionada.

Se toma como ejemplo para desarrollar el sistema la ciudad de New York. Se propone como solución en un primer caso predecir una ruta segura utilizando datos de delincuencia y accidentes, considerando un punto de origen y un punto destino, en segundo caso se divide la ciudad en varias regiones de riesgos para dar mejores predicciones ya que se tienen en cuenta áreas de delincuencia más pequeñas y como tercer caso se calcula la puntuación de riesgo más baja para sugerir la ruta más segura de acuerdo a los datos obtenidos de los clústeres más cercanos.

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

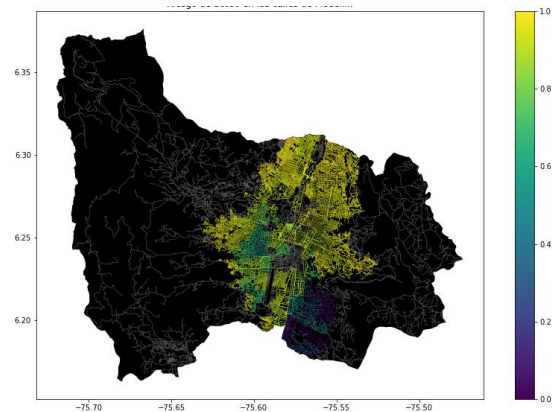


Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenidas de la Encuesta de Calidad de Vida de Medellín, de 2017.

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 algoritmo lee

El algoritmo lo que busca es encontrar el camino más corto desde un punto a otro, el laberinto es construido como una matriz binaria y son las celdas que tengan el número 1 por donde se puede avanzar. Lo primero que el algoritmo hace es crear una cola vacía y poner en cola la celda de origen que tenga una distancia 0 y luego la marca como visitada y cada vez que se mueva de celda se aumentará la distancia en 1,

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

además, si este es el nodo de destino la función retorna la distancia y si no se llega al destino se retorna false. También hay que tener en cuenta que las celdas donde halla un 1 se recorren primero

3.2.2 Encuentra el camino más corto en un laberinto

En este caso desde un principio se buscan todos los caminos posibles de un laberinto de posición inicial a destino. Esto se realiza explorando los 4 caminos posibles (arriba, abajo, derecha, Izquierda) y verifica recursivamente si conducen al destino o no, además la “longitud mínima” se compara y actualiza cada vez que se llega por un camino al destino y si se encuentra en un callejón sin salida hace un retroceso para explorar otros caminos, además no tiene en cuenta si una celda ya fue visitada por otra ruta.

3.2.3 Encuentra el camino más corto en un laberinto usando retroceso

Este es un algoritmo de retroceso en el cual tenemos una matriz binaria y nuevamente solo podemos avanzar si la celda cercana a la actual contiene un 1, el movimiento puede ser vertical u horizontal y va de 1 en 1. Este algoritmo lo que hace es que avanza al 1 más cercano y luego analiza si puede ir arriba y si no puede entonces agota todas las posibilidades y si ninguna sirve entonces se devuelve una celda para intentar otro camino. Al llegar al destino explora otros caminos para finalmente determinar cuál es el más corto

3.2.4 algoritmo dijkstra

dist, una matriz de distancias desde el nodo de origen ss a cada nodo en el gráfico, inicializado de la siguiente manera: $\text{dist}(\text{ss}) = 0$; y para todos los demás nodos vv, $\text{dist}(\text{vv}) = \infty$. Esto se hace al principio porque a medida que avanza el algoritmo, la distdist desde la fuente hasta cada nodo vv en el gráfico se recalculará y finalizará cuando se encuentre la distancia más corta a vv. QQ, una cola de todos los nodos en el gráfico. Al final del progreso del algoritmo, QQ estará vacío. SS, un conjunto vacío, para indicar qué nodos ha visitado el algoritmo. Al final de la ejecución del algoritmo, SS contendrá todos los nodos del gráfico.

El algoritmo procede como sigue:

Si bien QQ no está vacío, extraiga el nodo vv, que aún no está en SS, desde QQ con la distdist más pequeña (vv). En la primera ejecución, se elegirá el nodo de origen ss porque

distdist(ss) se inicializó en 0. En la siguiente ejecución, se elige el siguiente nodo con el valor de distdist más pequeño. Agregue el nodo vv a SS, para indicar que se ha visitado vv Actualice los valores distdist de los nodos adyacentes del nodo actual vv de la siguiente manera: para cada nuevo nodo adyacente uu, si $\text{dist}(\text{vv}) + \text{peso}(\text{u,v}) < \text{dist}(\text{uu})$, se encontró una nueva distancia mínima para uu, así que actualice distdist (uu) al nuevo valor de distancia mínima; de lo contrario, no se realizan actualizaciones en distdist (uu).

4. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

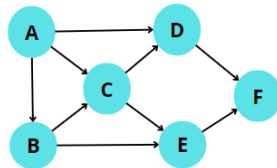
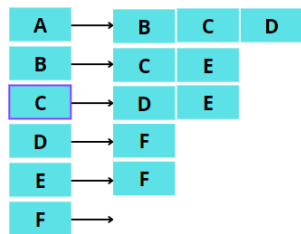
A continuación, explicamos las estructuras de datos y los algoritmos utilizados en este trabajo. Las implementaciones de las estructuras de datos y los algoritmos están disponibles en GitHub⁴.

4.1 Estructuras de datos

La estructura de datos utilizada para almacenar los datos de la ciudad de Medellín fue una lista de adyacencia utilizando diccionarios a partir de la información del dataframe. Lo primero que se hizo fue llenar los espacios de riesgo del dataframe que no tenían valor con la media de los valores de “riesgos de acoso”, luego se determinaron cuáles eran los orígenes únicos del dataframe para posteriormente crear las llaves del diccionario con estos orígenes únicos. Finalmente, dentro de las llaves de origen se creó otra llave con los destinos la cual almacena un arreglo con dos valores: la distancia y el riesgo de acoso. Además, había que tener en cuenta si la calle era doble vía; si este valor en el dataframe era “true” entonces ese destino debía ser tratado también como un origen así que se agregó a las llaves y dentro de este va otra llave que es la del origen la cual contiene los valores de distancia y riesgo de acoso. En las siguientes imágenes se puede ver:

1. La representación de una lista de adyacencia
2. La representación como un grafo de la lista de adyacencia
3. Como se implementa una lista de adyacencia con diccionarios en Python.

Lo único que no incluimos en las imágenes fue la información que pueden opcionalmente contener los enlaces y nodos.



```

grafo={
  "a"={"b","c","d"}
  "b"={"c","e"}
  "c"={"d","e"}
  "d"={"f"}
  "e"={"f"}
  "f"={}
}

```

4.2 Algoritmos

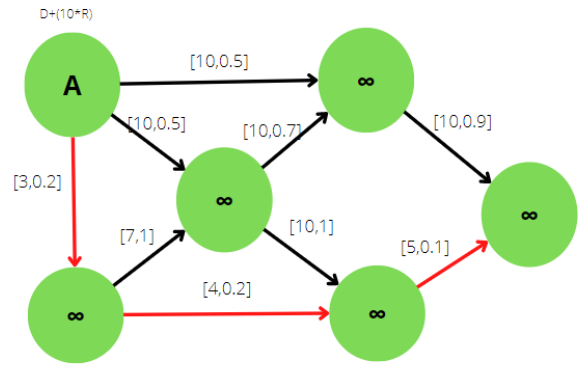
En este trabajo, proponemos un algoritmo para un camino que minimiza tanto la distancia como el riesgo de acoso sexual callejero.

4.1 algoritmo dijkstra modificado para encontrar el mejor camino tomando en cuenta el riesgo de acoso sexual callejero.

Lo primero que ocurre es que el algoritmo recibe un origen, un destino y un grafo. En principio se definen 4 diccionarios uno para ir guardando las distancias otro para los riesgos de acoso uno que es con los valores del algoritmo y otro que es para hacer un rastreo del camino, también se define un número muy grande y finalmente un arreglo vacío que al final va a contener el camino más óptimo.

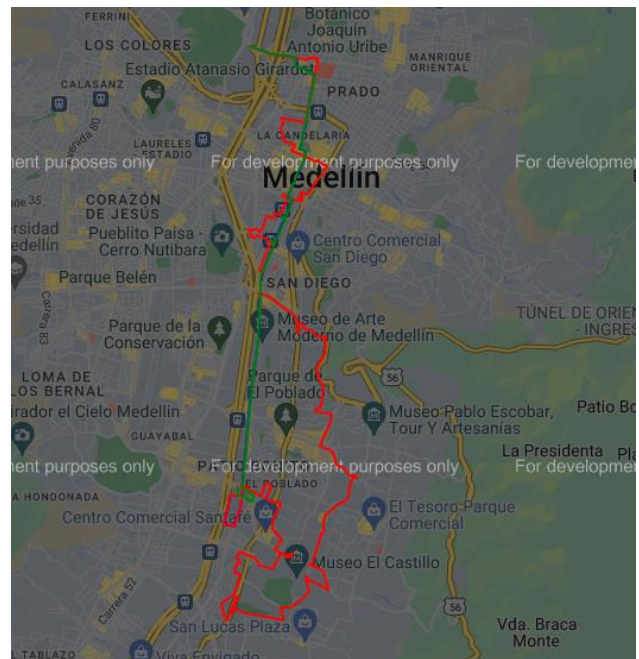
Lo primero que se hizo fue darles un valor de infinito a todos los nodos exceptuando al origen, posterior a esto se realiza otro ciclo para encontrar el nodo que contenga menor valor. Luego de haberlo encontrado identificamos a los nodos “vecinos” para irles cambiando el valor que se inicializo en un número muy grande por el menor posible, además es aquí donde se van guardando los antecesores, las distancias y los riesgos. Finalmente, luego de que termine esta iteración se elimina el nodo de menor distancia que se había escogido al inicio del algoritmo y se vuelve a repetir este proceso hasta que se hayan sacado todos los nodos del grafo.

Por último, se ejecuta otro ciclo con el objetivo de extraer del diccionario del rastreo de camino la mejor opción y pasarla a un arreglo para luego ser mostrada por pantalla. Además, se hizo un try catch por si era imposible ir del origen al destino que se recibieron.



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

Para encontrar los otros 2 caminos se utilizó de igual manera el algoritmo Dijkstra explicado anteriormente, lo único que varía es la ecuación utilizado para calcular la variable con la que funciona dicho algoritmo. Las 3 ecuaciones son las siguientes: 1. $(d*r)$, 2. $(d*(10*r))$, 3. $(30*d+500*r)$. a continuación, una imagen de cómo se ven los 3 caminos partiendo de Eafit y teniendo como destino la universidad Nacional.



4.3 Análisis de la complejidad del algoritmo

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

Tabla 1: Complejidad temporal del Dijkstra, donde V es la cantidad de aristas y E son los vértices

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

Tabla 2: Complejidad de memoria de la lista de adyacencia con diccionario donde V es la cantidad de llaves

4.4 Criterios de diseño del algoritmo

Escogí el algoritmo Dijkstra porque luego de buscar gran variedad de formas de solucionar el problema, llegué a la conclusión de que Dijkstra era muy eficiente y relativamente sencillo de implementar por su forma de recorrer el grafo cubriendo todos los caminos posibles, además de que es rápido en tiempos de ejecución.

Por lo anterior es que como estructura de datos escogí una lista de adyacencia con diccionarios pues de esta forma podía conocer los vecinos del nodo actual. Además de que los diccionarios son muy eficientes en términos de tiempo y memoria.

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	10127	0.6881
Eafit	Unal	24706	0.5602
Eafit	Unal	9336	0.7383

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 (Dijkstra) D=distancia R=riesgo	Tiempos medios de ejecución (s)
$v = d*r$	120.1 s
$v = d**(10*r))$	124.3 s
$v = 30*d+500*r$	135.6 s

Tabla 4: algoritmo Dijkstra para cada uno de los tres caminos calculadores entre EAFIT y Universidad Nacional.

6. CONCLUSIONES

Los tres caminos presentan ciertas diferencias el primero ($d*r$) regula muy bien tanto la distancia como el acoso encontrando un punto medio; el segundo ($d**(r*10)$) prioriza más el valor del riesgo de acoso por lo que propone caminos más largos con la finalidad de reducir notablemente el riesgo y el tercero ($30*d+500*r$) propone un camino corto, aunque el riesgo sin importar el camino tiene un valor muy cercano a 0.7. Por otro lado, el algoritmo puede llegar a ser muy útil cuando hablamos de personas que transitan las calles a pie durante la noche, aunque los tiempos de ejecución deben optimizarse para que sea útil en la realidad pues por camino el algoritmo tarda alrededor de 2 minutos. Finalmente, para una aplicación recomendaría el primer camino pues es uno que deja en un punto medio los valores de distancia y riesgo de acoso.

6.1 Trabajos futuros

En un futuro me gustaría encontrar una forma en la que el algoritmo en 2 minutos arrojará los 3 caminos, en otras palabras, que solo fuera necesario 1 grafo y que en una sola ejecución del algoritmo Dijkstra se fueran almacenando los 3 caminos distintos.

AGRADECIMIENTOS

Los autores agradecen enormemente a las personas que permiten en primera instancia el aprendizaje por medio de su apoyo económico y emocional, Luz Dary Moncada y John Jairo Zapata

Agradecemos también la ayuda de los monitores de la materia quienes contribuyeron con sus observaciones para mejorar el trabajo.

Finalmente, e igual de importante 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*.

Referencias

1. Sandeep, C., Gauri Shankar, V., Soni, S. 2019. Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data. Recuperado agosto 13, 2022 de https://www.researchgate.net/publication/338096313_Route-The_Safe_A_Robust_Model_for_Safest_Route_Prediction_Using_Crime_and_Accidental_Data
2. Jigang, W., Jin, S., Ji, H. Srikanthan, T. 2011. Algorithm for Time-dependent Shortest Safe Path on Transportation Networks. Recuperado agosto 13, 2022 de <https://www.sciencedirect.com/science/article/pii/S1877050911001591>
3. Sarraf, R. and P., M., 2018. A Data Integration and Analysis System for Safe Route Planning. Proquest.com.
4. Ma, D. 2022. Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search. Omdena | Building AI Solutions for Real-World Problems.
5. Anon, Shortest Path in Maze using Backtracking – Pencil Programmer. Pencil Programmer
6. Dijkstra's Algorithm. 2022. Programiz.com. <https://www.programiz.com/dsa/dijkstra-algorithm>