

ALGORITMO PARA ENCONTRAR EL CAMINO MAS SEGURO PARA DESPLAZARCE EN MEDELLIN

Jhonnathan Stiven Ocampo
Diaz
Universidad Eafit
Colombia
jsocampod@eafit.edu.co

Jerónimo Cardona Osorio
Universidad Eafit
Colombia
jcardonao2@eafit.edu.co

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

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

RESUMEN

Podemos intentar buscar causas a este “fenómeno” y pese a que podemos intuir, incluso afirmar que se debe a tradiciones machistas de épocas pasadas y conservadoras, las cuales veían el piropo como una muestra de afecto y exaltación a la belleza, principalmente femenina, el hecho de hacerlo en situaciones, lugares y con personas totalmente desconocidas solo dejan como causante una gran falta de respeto, moralidad y educación, pues hay quienes hoy en día se enorgullecen de esta mala práctica y creen que entre más acosen a una persona, más valor y dominancia van a tener. [0]

Continuando, para la solución precisa de esta situación que aqueja a muchas personas alrededor de la ciudad, se decidió implementar un algoritmo llamado Dijkstra, también nombrado algoritmo de caminos mínimos, el cual se encarga de determinar el camino más corto, buscando desde un vértice de origen a los otros que se encuentran dentro de un grafo. Finalmente, al realizar la ejecución de los tres caminos se obtuvieron datos cuantitativos que nos indicaban el tiempo, el riesgo y la distancia necesarios dentro de los 3 trayectos. [3]

Palabras clave

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

1. INTRODUCCIÓN

A lo largo de la historia, la sociedad ha evolucionado creando normas con la finalidad de mantener el orden y la seguridad en la misma. Sin embargo, uno de los problemas a los que actualmente se enfrenta es la violencia de género, esto debido a los roles que se manejan todavía en una sociedad en este problema no solo la violencia física, psicológica sino también un problema mucho mayor que por nuestra cultura no tomamos en cuenta, y que forma parte de una violencia particular y es el “acoso callejero”, el cual en su mayoría lo sufren las mujeres.

1.1. Problema

El acoso sexual callejero se refiere a un tipo de violencia ejercida por una persona a otra, sin consentimiento y con el propósito de remarcar una posición superior de poder que se ha impuesto por la sociedad, las principales implicaciones frente a este problema son las emocionales y psicológicas ya

que hace que las personas afectadas se sientan invalidadas, vulnerables e inseguras y terminan generando traumas que hacen que los afectados no vuelvan a sentirse cómodos por el miedo a que esta situación vuelva a suceder. [1]

1.2 Solución

Para nuestra solución decidimos implementar un código que se encargue de ponderar tanto la distancia como el riesgo de acoso en las calles de Medellín, y le devuelva al usuario con base a sus coordenadas de origen y destino, la ruta más optima posible, siendo esta tanto rápida como segura. Para ello utilizamos el algoritmo de Dijkstra, el cual es un algoritmo muy popular alrededor del mundo gracias a que determina el camino más corto en un grafo, con base a un vértice de origen y los pesos de cada arista.

Dado nuestro caso convertimos las calles de Medellín en un grafo y definimos el peso de las aristas con la distancia y riesgo de cada calle respectivamente. Donde, implementado el Dijkstra, este devuelve la ruta más corta y segura a través del grafo.

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. CROWDSAFE: crowd sourcing of crime incidents and safe routing on mobile devices

CROWDSAFE, fusiona dos conceptos: crowdsourcing (que hace referencia a la colaboración de un colectivo para una tarea) de Internet y dispositivos inteligentes portátiles para permitir la búsqueda y el informe de incidentes delictivos basados en la ubicación y en tiempo real. Este proyecto está dirigido a usuarios interesados en información sobre delitos. El sistema aprovecha los datos de fuentes múltiples para proporcionar funciones novedosas, como un enrutador de

seguridad y análisis de delitos de valor agregado, permite al usuario buscar y reportar nueva información sobre delitos. La función de enrutador de seguridad ayuda al usuario a programar un plan de viaje seguro y conveniente. Este proyecto implementa dos algoritmos A* y Dijkstra dependiendo del desempeño. [2]

2.2 Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data

Este proyecto realizado por la Escuela de Informática y TI, de la Universidad Manipal Jaipur, India. Proponen una solución para las altas tasas de crimen y la baja percepción de seguridad en la ciudad de New York, esta solución consta de 3 pasos, el primero que busca predecir una ruta segura utilizando datos de delincuencia y accidentes, así como considerando la distancia entre la origen y destino. El segundo que divide la ciudad de Nueva York en regiones de riesgo más pequeñas aplicando agrupamiento anidado en los datos y el último que hace un cálculo de la puntuación de riesgo de las rutas en función del riesgo puntuado de los clústeres cercanos. Usan 3 distintos algoritmos para desarrollar la solución propuesta: Data preprocessing, k-mean y K Nearest Neighbor. [3]

2.3 SAFETIPIN – Supporting Safer Cities

Safetipin es una organización social que trabaja con una amplia gama de actores urbanos, incluidos los gobiernos, para hacer que los espacios públicos sean más seguros e inclusivos para las mujeres. Recopilan datos utilizando sus 3 aplicaciones de teléfonos móviles (My Safetipin, que está disponible en la tienda de aplicaciones y Play Store; Safetipin Nite y Safetipin Site) y los presentan a las partes interesadas relevantes con recomendaciones. Las rutas se elaboran en la parte trasera de un automóvil por profesionales capacitados para cubrir todas las calles de la ciudad. El conductor ve la ruta que debe tomar. A medida que el automóvil se mueve, se toman fotografías a distancias predefinidas y se cargan en sus servidores. Luego se ejecutan algoritmos de aprendizaje automático (visión por computadora) en estas imágenes para extraer información sobre parámetros de seguridad. Por último, codificadores capacitados agregan algunos puntos de datos adicionales que usan esta información para auditar un punto. Esto constituye la principal fuente de datos para los parámetros de seguridad en una ciudad.

El sistema back-end que se utiliza para codificar los puntos de auditoría se puede adaptar para capturar información adicional basada en el proyecto. El sistema está disponible para todas las partes interesadas del proyecto y permite múltiples niveles de credenciales para permitir tanto la participación como la seguridad. [4]

2.4 SAFEWAY: An explainable context-aware recommender system for safe routes

SAFEWAY es un sistema de recomendación basado en casos para proponer y explicar la ruta más segura al usuario teniendo en cuenta las restricciones de su contexto. El sistema SAFEWAY funciona con una memoria de casos obtenida del dataset de Seguridad Vial que incluye accidentes de tráfico en GB desde 1979. Luego utiliza una metáfora visual para dar al usuario los detalles de la ruta en cuanto a su seguridad. En esta metáfora los marcadores numéricos muestran la similitud entre el contexto del incidente pasado y el contexto del usuario actual. Además, el color del icono refleja la gravedad del incidente. La explicación de la gravedad de todo el recorrido se complementa con el uso de la agrupación de incidencias en una localización similar. Esto permite al usuario obtener una vista global de la ruta, que luego se amplía cuando el usuario hace zoom para ver los detalles. [5]

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.

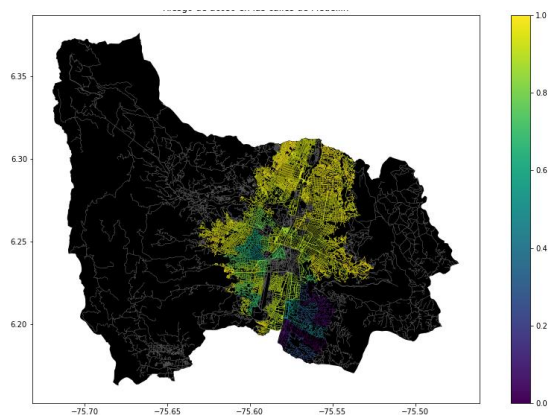
Para este proyecto, se calculó una combinación lineal (CL) que captura la máxima varianza entre (i) la fracción de hogares que se sienten inseguros y (ii) la fracción de hogares con ingresos inferiores a un salario mínimo. Estos datos se obtuvieron de la encuesta de calidad de vida de Medellín, de 2017. 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 Algoritmo de Dijkstra

El algoritmo de Dijkstra nos permite encontrar el camino más corto entre dos vértices cualesquiera de un grafo. Es un algoritmo eficiente que sirve para encontrar el camino de coste mínimo desde un nodo origen a todos los demás nodos del grafo. Este algoritmo se fundamenta en la manera que dado un grafo a cuyos arcos se han asociado una serie de pesos, se define el camino de coste mínimo de un vértice “u” a otro “v”, esto porque el camino donde la suma de los pesos de los arcos que lo forman es la más baja entre las de todos los caminos posibles de “u” a “v”. [6]

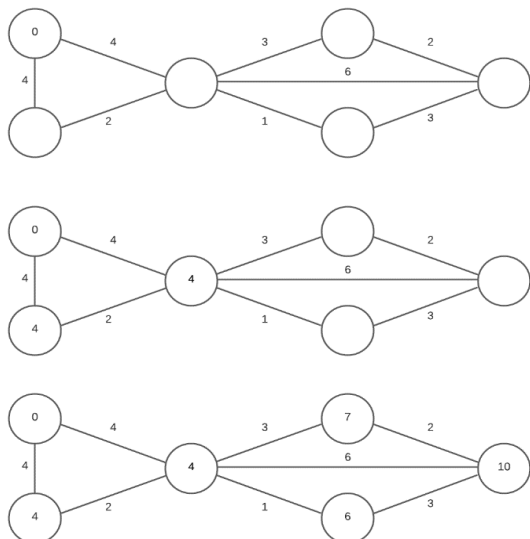


Imagen 1. Algoritmo de Dijkstra

3.2.2 Breadth First Search (BFS)

BFS tiene varias aplicaciones como encontrar el camino más corto entre 2 nodos, medido por el número de nodos conectados o para sistemas de navegación GPS, para encontrar localizaciones vecinas. Es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo, comenzando en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo), para luego explorar todos los vecinos de este nodo. A continuación, para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el grafo. [7]

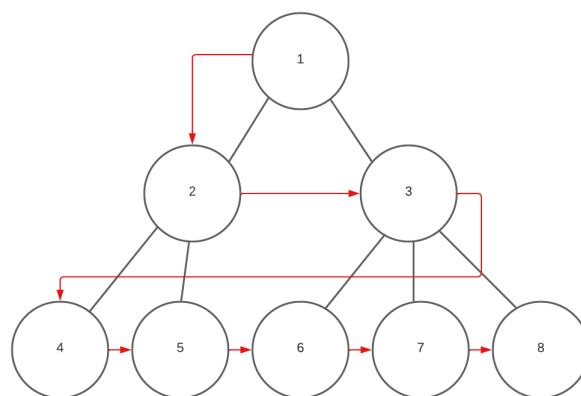


Imagen 2. Algoritmo BFS

3.2.3 Depth First Search (DFS)

Este es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo. Su funcionamiento consiste en ir expandiendo cada uno de los nodos que va localizando, de forma recurrente (desde el nodo padre hacia el nodo hijo). Cuando ya no quedan más nodos que visitar en dicho camino, regresa al nodo predecesor, de modo que repite el mismo proceso con cada uno de los vecinos del nodo. [7]

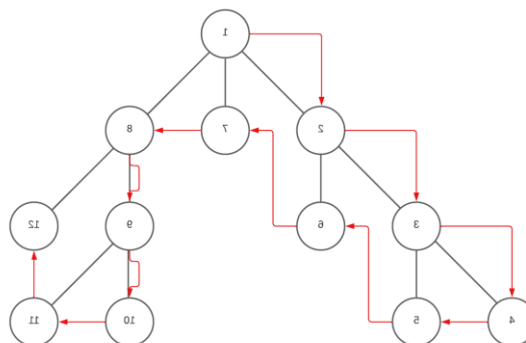


Imagen 3. *Algoritmo DFS*

3.2.4 Algoritmo de búsqueda A* El algoritmo de búsqueda A* es una de las mejores y más populares técnicas utilizadas en la búsqueda de rutas y recorridos de gráficos. Lo que hace el algoritmo de búsqueda A* es que, en cada paso, selecciona el nodo de acuerdo con un valor: 'f', que es un parámetro igual a la suma de otros dos parámetros: 'g' y 'h'. En cada paso, elige el nodo/celda que tiene la 'f' más baja y procesa ese nodo/celda. [8]

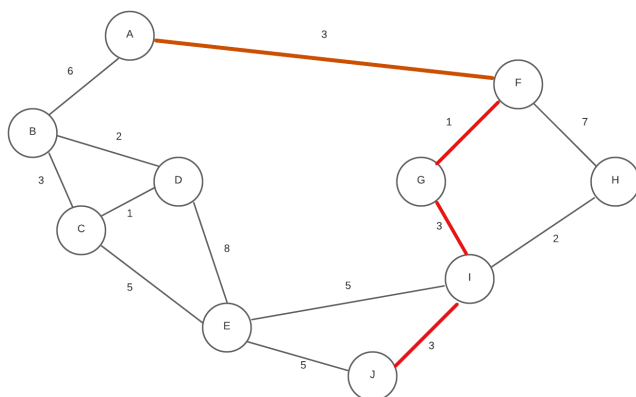


Imagen 4. *Algoritmo A**

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

Los datos fueron entregados en un archivo CSV, así que se decidió representar estos datos en un DataFrame de pandas, a partir de esto se implementó una lista de adyacencia de diccionarios para hacer el grafo que representa la ciudad de Medellín, en donde las "Keys" son todos los orígenes y los "values" son otro diccionario que posee como "keys" todos los destinos del origen al que están asociados y los valores de este diccionario son el dato que representa la ponderación del riesgo y la distancia del camino.

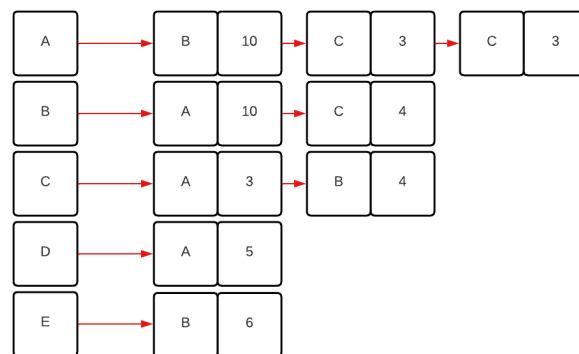


Imagen 5. *Grafo como lista de adyacencia donde la columna de la izquierda representa la "Key" y los datos de la derecha son sus "Values"*

Figura 2: Un ejemplo de mapa de calles se presenta en (a) y su representación como lista de adyacencia en (b). (Por favor, siéntase libre de cambiar esta gráfica si utiliza una estructura de datos diferente).

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

El algoritmo que se emplea para la solución propuesta es Dijkstra, este necesita de un grafo (Un grafo, es una estructura de datos que permite modelar mediante, una representación gráfica formada por nodos que muestra a los actores y aristas que sirven para representar las relaciones entre los actores. [9]) y de los nodos de origen y destino. Comienza en un nodo de su elección (el nodo de origen) y analiza el grafo para encontrar el camino más corto entre ese nodo y el nodo elegido como destino (como el objetivo es encontrar la ruta más segura se tiene como distancia no solo el "peso" de la longitud del camino sino el valor del riesgo de acoso).

El algoritmo realiza una validación de la distancia más corta actualmente conocida de cada nodo al nodo de origen y actualiza esos valores cuando encuentra una ruta más corta.

Si el algoritmo encuentra la ruta más corta entre el nodo de origen y otro nodo, ese nodo se marca como "visitado" y se agrega a la ruta (esta ruta es una lista y es el valor de retorno de la función).

El proceso continúa hasta que todos los nodos del grafo se agregan a la ruta. Por lo tanto, tenemos una ruta que conecta el nodo de origen con todos los demás nodos, siguiendo la

⁴ <https://github.com/jhothinnan/ST0245-Eafit>

ruta más corta posible del nodo origen al que se marcó como destino.

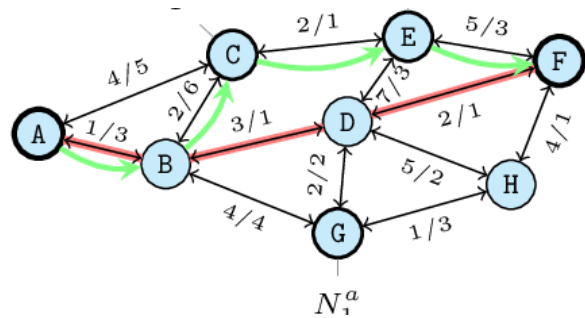


Figura 3: Cálculo de un camino que reduce tanto la distancia como el riesgo de acoso.

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

Los otras dos rutas funcionan como una solución óptima que se ejecuta con las mismas condiciones del camino principal, la diferencia presente es el foco del algoritmo, en cada uno, respectivamente, una es la distancia más corta a calcular y la otra, es la ruta más segura (con menos riesgo) por obtener, es decir, ahora la prioridad no es compartida y cada ruta se centra en un aspecto en específico, cosa que se evidencia al ver los datos resultantes entre ambas, con una distancia mayor pero con menos riesgo y viceversa.

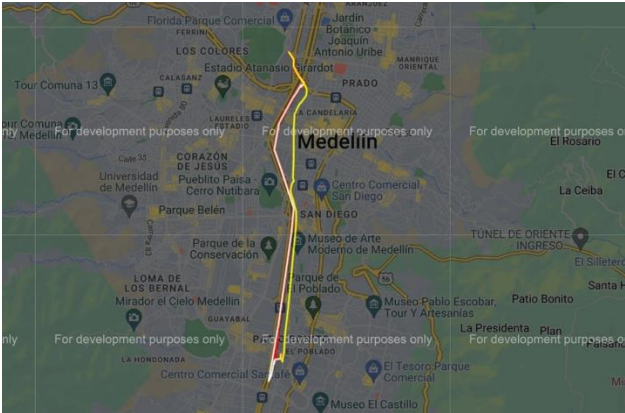


Figura 4: Mapa de la ciudad de Medellín donde se presentan tres caminos para peatones que reducen tanto el riesgo de acoso sexual como la distancia en metros entre la Universidad EAFIT y la Universidad Nacional.

4.3 Análisis de la complejidad del algoritmo

En nuestro código, el algoritmo de Dijkstra visita cada nodo una vez (es por esto que $O(V)$) e intenta alcanzar todos los nodos adyacentes a través de las aristas ($O(V)$), además el tiempo de acceso a una cola de prioridad (en nuestro caso heap) tiene un tiempo de $O(\log V)$ es por esto que la complejidad es $O(E+V \log |V|)$

Algoritmo	Complejidad temporal
Dijkstra	$O(E+V \log V)$
Dijkstra (Sin cola de prioridad)	$O(V^2)$

Tabla 1: Complejidad temporal de Dijkstra. En esta complejidad la V representa los vértices y la E representa las aristas.

Estructura de datos	Complejidad de la memoria
Listas de adyacencia	$O(V^2)$

Tabla 2: Complejidad de memoria de la estructura de datos de listas de adyacencia es $O(V^2)$, donde V representa los vértices.

4.4 Criterios de diseño del algoritmo

En nuestro proyecto se decidió escoger el algoritmo Dijkstra, ya que su funcionamiento de búsqueda tipo BFS, nos resultaba muy útil para el problema planteado, claramente no podíamos simplemente usar el código en su forma primitiva, por lo que se debió implementar con ciertas características especiales, como un nodo específico de inicio (el nodo que representaba la calle de origen en este caso), y un peso con criterio de evaluación distinto para cada una de las tres implementaciones (distancia, riesgo, ponderación).

Y, por último, un detalle muy importante, fue el uso de una cola de prioridad (priority queue), la cual decidimos implementar como una forma de “apoyo” al Dijkstra que ya teníamos, debido a que el mismo, por sí solo, tardaba demasiado en recorrer el grafo, por lo cual, como una forma de optimización, se decidió por usar este apoyo al grafo, donde dicha influencia se ve reflejada después, al utilizar un tiempo muchísimo menor que antes.

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 (m)	Riesgo
Eafit	Unal	7.744	0,69
Eafit	Unal	9.179	0,54
Eafit	Unal	8.080	0,57

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)
$v = d$	0,068 s
$v = r$	0,089 s
$v = d + 80*r$	0,078 s

Tabla 4: Tiempos de ejecución del Dijkstra para cada uno de los tres caminos calculados entre EAFIT y Universidad Nacional.

6. CONCLUSIONES

Comparando los resultados obtenidos, pese a la similitud grafica entre las tres rutas, es curioso notar los puntos de diferencia, en ciertas partes muy marcada, que hay entre estas, y como el foco central de cada algoritmo influye en el resultado final de cada ruta. Además de resultar muy gratificante el corto o casi nulo tiempo de ejecución de los algoritmos, lo cual representa una buena optimización y eficiencia para un posible uso futuro, uso que sería un gran apoyo para la misma ciudad, la cual puede implementar los algoritmos con muchas más direcciones de origen y destino, para reconocer posibles focos de riesgo e implementar nuevas medidas de seguridad y así intentar mitigar esta gran problemática (el acoso).

Pese a la similitud en las 3 rutas, recomendamos profesionalmente utilizar la ruta ponderada, ya que por un lado era el objetivo central del proyecto, “Obtener la ruta más corta y segura”, y por el otro, analizando los datos resultantes de cada ruta, la ponderada, aunque un poco obvio, es la que mejor maneja la distancia junto con el riesgo, ya que las otras dos poseen una diferencia más significativa entre distancia y riesgo, respectivamente.

6.1 Trabajos futuros

Con la vista en el futuro nos gustaría implementar la mayor parte de conocimientos que adquiramos en el transcurso de nuestra carrera (Ing. Sistemas), donde podamos implementar nuestro algoritmo optimizado con algún tipo de interfaz interactiva, en un ámbito tanto web como móvil, además claro está, de ampliar su alcance a más ciudades, departamentos y posteriormente, porque no, a un nivel internacional.

AGRADECIMIENTOS.

Esta investigación agradece a aquellos que han contribuido en pequeña y gran medida, a quienes nos orientaron y quienes tuvieron la disposición de dirigir nuestros conocimientos y dudas a la realización concreta del proyecto.

Además, se agradece a padres, amigos y mascotas por haber sido un apoyo tanto técnico como Psicológico durante nuestra travesía en este proyecto.

Los dos primeros autores fueron apoyados por la beca Generación E, financiada por el gobierno nacional, donantes externos y la propia institución educativa. Todos los autores agradecen a la Vicerrectoría de Descubrimiento y Creación, de la Universidad EAFIT, su apoyo en esta investigación.

Todos 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

0. <http://remca.umet.edu.ec/index.php/REMCA/article/download/239/281>.
1. Tuyub, J., Valle, V. y Alpuche, S., 2021. Repercusiones psicológicas del acoso sexual callejero en mujeres meridianas. *Alternativas.me*.
2. Shah, S., Bao, F., Lu, C. y Chen, I., 2011. CROWDSAFE: crowd sourcing of crime incidents and safe routing on mobile devices. *ACM*.
3. Soni, S., Gauri, V. y Chaurasia, S., 2019. Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data. *Research gate*.
4. Safetipin | Safetipin, Creating Safe Public Spaces for Women: 2022. <https://safetipin.com/>.
5. https://web.archive.org/web/20220420000129id_/http://eur-ws.org/Vol-2567/paper11.pdf.
6. Escuela de computación, UDB, 2022. Algoritmos para la ruta más corta en un Grafo., Bogotá: udb.
7. López, M. y Murillo, J., 2020. Difference between Breadth Search (BFS) and Deep Search (DFS). Encora.
8. GeeksforGeeks, 2022. A* Search Algorithm - GeeksforGeeks. GeeksforGeeks.
9. Rochina, P., 2022. Teoría de Grafos: Análisis relacional de las Redes Sociales. *Canal Informática y TICS*.