

## TRUSTED PATH. GENERADOR DE RUTAS ALTERNATIVAS PARA EVITAR EL ACOSO CALLEJERO

Santiago Rodríguez Mojica  
Universidad Eafit  
Colombia  
srodrigu24@eafit.edu.co

Miguel Angel Hoyos  
Universidad Eafit  
Colombia  
mahoyosv@eafit.edu.co

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

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

### RESUMEN

Una ciudad como Medellín, brinda una amplia gama de alternativas a la hora de movilizarse y la seguridad juega un papel fundamental. Actualmente el acoso y la inseguridad son fenómenos que nos golpean como sociedad en todos los aspectos de nuestra vida, incluyendo la movilidad, así que no basta con tomar la ruta más rápida, si al hacerlo nos exponemos a un gran riesgo mientras la transitamos. Es necesario integrar el factor seguridad y determinar también, cuáles son las rutas fiables para llegar a nuestros destinos tratando de equilibrar y minimizar los tiempos de desplazamiento y riesgos durante los trayectos.

Una problemática de esta índole acarrea repercusiones para todos y desemboca en más problemáticas que afectan la calidad de vida de todos, y aquí la importancia de encontrar alternativas. Independientemente de nuestro estrato social, rol en la sociedad, creencias religiosas, profesiones o lugar de trabajo, todos debemos desplazarnos en mayor o menor medida en la ciudad.

Se propone el algoritmo Dijkstra para calcular rutas alternativas y se obtienen resultados favorables en términos de reducir riesgo, distancia y tiempos de ejecución al implementarlo. Considerando su corto tiempo de ejecución, puede ser una herramienta muy útil para todos los habitantes de Medellín.

### Palabras clave

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



**Imagen 1:** Foto de Medellín tomado de <https://es.wikipedia.org/wiki/Medellín>.

### 1. INTRODUCCIÓN

El proyecto Trusted Path, nace del deseo de aportar desde la tecnología y la revolución 4.0 a la construcción de una ciudad con soluciones inteligentes. Como futuros ingenieros de sistemas, tenemos una gran responsabilidad en nuestras manos, mejorar la calidad de vida de las personas. Actualmente la tecnología permea todos los aspectos de nuestra vida, y de aquí nace la oportunidad de usarla para disminuir el acoso callejero hacia las mujeres, buscando brindar alternativas a las soluciones que ya se han planteado y no han sido efectivas. Medellín es un laberinto y presenta una gran cantidad de rutas alternas a la hora de movilizarse. Para una persona normal, tener en cuenta cada uno de los caminos posibles y que tan seguros son, es una tarea demorada, complicada, desgastante y no siempre con resultados precisos. Aquí es donde nace la integración con el software. Calcular el camino más rápido de un punto a, hasta un punto b, es un problema que hemos tenido desde siempre, y ya hay soluciones tecnológicas, como Google maps o Waze, el problema con estas es que ignoran la seguridad del trayecto y aquí es donde nace Trusted Path, que además de tener en cuenta una ruta eficaz, busca establecer un trayecto seguro.

#### 1.1. Problema

La violencia y acoso contra la mujer es un fenómeno supremamente complejo y data de muchísimo tiempo atrás. En el intervienen factores culturales, arraigados en las mismas bases de la sociedad, ineficiencia de organismos de control y seguridad, falta de educación y civismo, entre otros. Para fines de este proyecto, tomaremos como foco central la inseguridad y el acoso callejero. Esto gracias a que las cifras reflejan que es uno de los puntos críticos de este problema. Es importante dar soluciones y aplacar estos hechos desafortunados, ya que hace parte de nuestro desarrollo integral como sociedad. El progreso va más allá de la infraestructura, y el brindar seguridad y tranquilidad a todos debe ser una de las bases para empezar a construir ciudad, ya que frases normalizadas como “Me avisas cuando llegues” o “¿Llegaste bien?”, denotan que vivimos con la preocupación de que, al salir a la calle, algo pueda ocurrir, y esto definitivamente, no es calidad de vida.

El problema por solucionar y el reto que se nos presenta desde la ingeniería, es poder encontrar rutas alternativas, que

reduzcan tanto la distancia como el riesgo del acoso sexual callejero en las calles de Medellín. Al resolver esto, proporcionamos una herramienta a las mujeres, para que puedan sentirse más seguras. Y confiando en que podamos escalar de Medellín a todas las ciudades de Colombia, de Latinoamérica, y ¿Por qué no?, de todo el mundo.

## 1.2 Solución

Para dar solución al reto, en este caso desarrollar un algoritmo para peatones para reducir tanto la distancia como el riesgo de acoso, se ha elegido utilizar Dijkstra (3.2.1). El algoritmo de Dijkstra toma un nodo de partida y asigna en la primera iteración distancias infinitas al resto de nodos. Posteriormente va recorriendo todos los nodos adyacentes al nodo de partida y va seleccionando la ruta menor de cada uno de los nodos a los que va navegando, esto lo realiza iteración tras iteración, hasta que uno de los nodos fue visitado y se determina cual es el costo menor para ir desde ese nodo origen a todo el resto de los nodos 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 Walking Safe: Aplicación Móvil para el Trazado de Rutas Peatonales en las Ciudades

Por medio del uso de la inteligencia artificial, esta aplicación busca aumentar la seguridad vial de los peatones, guiándolos mediante rutas más seguras. Su principal diferencia (como lo definen, Ventaja) surge de la incorporación de un sistema que le permite a la aplicación, identificar la intención de cruce y giro de un peatón. No se especifica el algoritmo utilizado. [1]

### 2.2 RIDBIKE: Aplicación Móvil Para Crear Rutas Compartidas En Usuarios De Bicicletas

Aplicación para dispositivos móviles que permiten realizar diferentes búsquedas en tiempo real como crear rutas de un punto a otro calculando distancia, tiempo y seguridad, incorpora también la posibilidad de buscar rutas que hayan sido creadas previamente, valorar las diferentes rutas y unirse (o crear) rutas en las que otros usuarios también pueden comentar y valorar, esto permite identificar las rutas en las que se reduce el acoso callejero, por ejemplo. No

especifican el algoritmo utilizado, sin embargo, se basan en el API de Google Maps.[2]

### 2.3 Sistema Big Data Para La Identificación Y Predicción De Zonas Potencialmente Peligrosas Para Ciclistas

Basado en el desarrollo de un módulo de análisis de datos y de predicción, el proyecto se centra en la implementación de un sistema capaz de identificar zonas potencialmente peligrosas para ciclistas mediante el uso de técnicas basadas en Machine Learning, basándose en un historial de trayectos, accidentes y robos, adicionando información que es consultada de los usuarios en tiempo real. Este sistema implementa los algoritmos Dijkstra y Bellman-Ford. [3]

### 2.4 SAFEWAY: Un Sistema De Recomendación De Rutas Basado En El Contexto Para Fomentar La Seguridad Vial

Es un sistema de recomendación que intenta fomentar la seguridad vial mediante la recomendación de rutas seguras, considerando además del tiempo empleado, el riesgo que presentan las distintas alternativas de viajes entre dos puntos distintos dentro de un mismo territorio. Tiene como principal objetivo que el usuario sea quien comprenda el porqué de las recomendaciones que le proporciona el sistema. No especifica el algoritmo utilizado. [4]

## 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

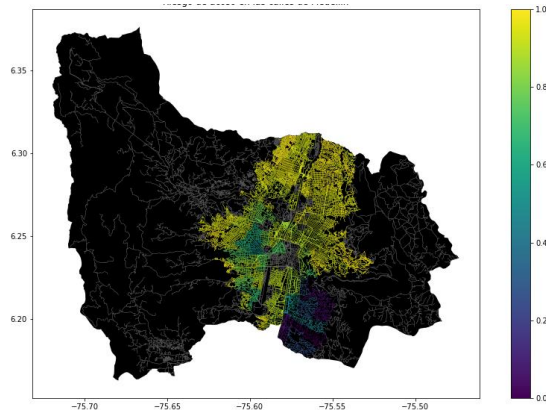
El mapa de Medellín se obtuvo de *Open Street Maps* (OSM)<sup>1</sup> y se descargó utilizando la API<sup>2</sup> OSMnx de Python. El mapa incluye (1) la longitud de cada segmento, en metros; (2) la indicación de si el segmento es de un solo sentido o no, y (3) las representaciones binarias conocidas 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

<sup>1</sup> <https://www.openstreetmap.org/>

<sup>2</sup> <https://osmnx.readthedocs.io/>

Figura 1 presenta el riesgo de acoso calculado. El mapa está disponible en GitHub<sup>3</sup>.



**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.

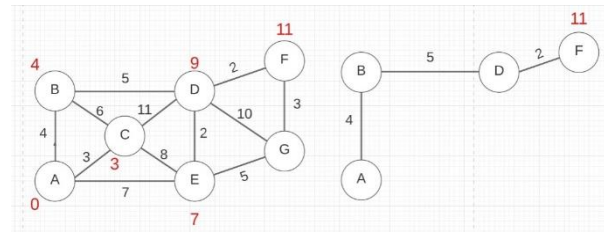
### 3.2 Alternativas de caminos que reducen el riesgo de acoso sexual callejero y distancia

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

#### 3.2.1 Dijkstra

Es un algoritmo desarrollado por Edsger Dijkstra en el año 1959. Su objetivo es encontrar el camino más corto entre dos nodos dados. Parte del nodo inicial y se detiene en el momento en que llega al nodo objetivo. Este asigna un valor de “infinito” a cada nodo que no ha visitado. A medida que va recorriendo el grafo, crea un registro y calcula distancias a cada nodo nuevo, partiendo de sumar la distancia anterior y reemplazando el valor de infinito anteriormente asignado. Si la distancia calculada es menor que la anterior, la reemplaza, y si no, la ignora y continua su ejecución. Es importante tener en cuenta lo anterior, ya que no se puede aplicar a grafos que tengan aristas negativas, ya que genera un error al ignorar caminos que podrían resultar más eficientes a largo plazo. Este algoritmo en particular cuenta con un orden de complejidad, dependiendo la manera en la que se desarrolle, tenemos entonces que en el caso de no utilizar cola de prioridad, el algoritmo tiene una complejidad

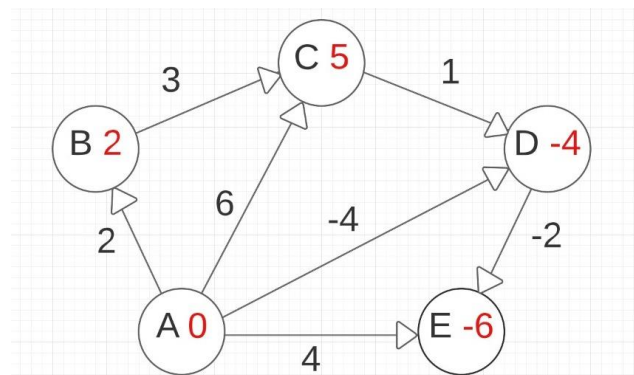
de  $O(|V|^2 + |A|) = O(|V|^2)$  y cuando se implementa con cola de prioridad, vemos que tiene una complejidad de  $O((|A| + |V|) \log |V|) = O(|A| \log |V|)$ . El algoritmo se ejemplifica en la Figura 2. [5]



**Figura 2:** Representación del algoritmo Dijkstra.

#### 3.2.2 Bellman-Ford

Es un algoritmo desarrollado por Richard Bellman, Samuel End y Lester Ford. Su objetivo es encontrar el camino más corto entre dos nodos dados. Es muy similar al algoritmo de Dijkstra, aunque se define de una manera más general. Tiene la particularidad, de que acepta ponderaciones negativas, pero a cambio de esto, es menos eficiente que Dijkstra. Su complejidad es de  $O(VA)$ . El algoritmo se ejemplifica en la Figura 3. [6]



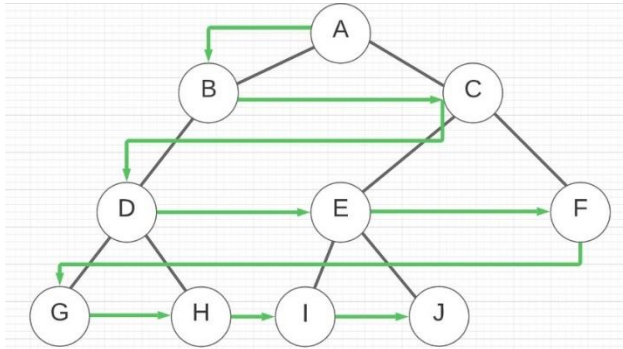
**Figura 3:** Representación del algoritmo Bellman-Ford.

#### 3.2.3 BFS

Es un algoritmo que viene de su abreviación en inglés “breadth-first search” o en español, búsqueda de anchura. Su objetivo puede ser, tanto el buscar un nodo especificado, como el de recorrer el grafo por completo. Inicialmente, explora todos los nodos que se conectan directamente con el nodo de origen especificado, y a partir de estos, los adyacentes a estos, hasta llegar al nodo objetivo o terminar de recorrer todo el grafo. Es importante aclarar que determina el camino más corto en función de los vértices

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

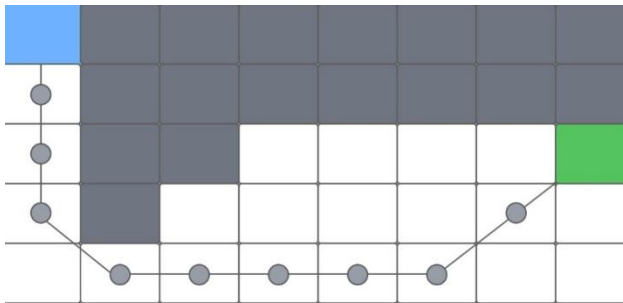
que recorre, y no del costo variable que puede tener ir de un nodo a otro, por lo que se usa normalmente en grafos no ponderados. Su coste o tiempo de ejecución (complejidad) es de  $O(|V|+|A|)$ . El algoritmo se ejemplifica en la Figura 4. [7]



**Figura 4:** Representación del algoritmo BFS.

### 3.2.4 A Star

Es un algoritmo de búsqueda. Fue desarrollado en 1968 por Peter E. Hart, Nils J. Nilsson y Bertram Raphael y es conocido como “A\*”, “A estrella”, etc. Su objetivo es encontrar el camino más corto, especificándole un nodo inicial y uno objetivo. Usa la heurística para calcular la distancia más optima al nodo objetivo, sin sobreestimar el resultado. Cuenta con una complejidad exponencial en el peor de los casos de  $O(2^n)$ . El algoritmo se ejemplifica en la Figura 5. [8] [9]



**Figura 5:** Representación del algoritmo A Star.

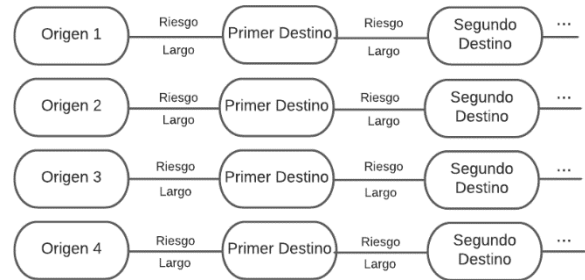
## 4. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

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 GitHub<sup>4</sup>.

### 4.1 Estructuras de datos

La estructura de datos que utilizamos para representar el grafo de las calles de la ciudad fue la lista de adyacencia. La implementamos por medio de diccionarios. En la lista de

adyacencia, las claves del diccionario maestro son todos los vértices de orígenes únicos. Los diferentes valores de este diccionario son todos los vértices a los que está conectado cada origen respectivamente. Adicional, utilizamos un segundo diccionario, que toma como clave los destinos, y como valor, tuplas del riesgo y largo de ir del nodo origen al nodo destino. La estructura de los datos se presenta en la Figura 6.



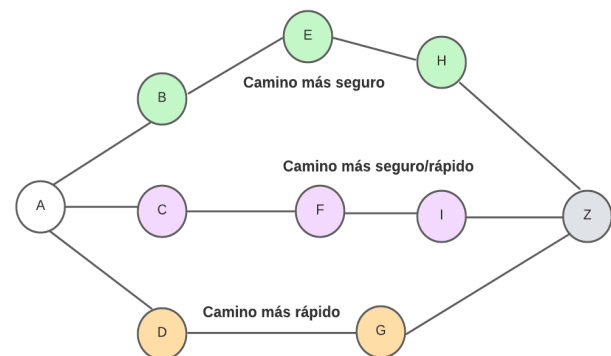
**Figura 6:** Representación de mapa de calles como lista de adyacencia

## 4.2 Algoritmos

En este trabajo, proponemos un algoritmo para encontrar tres caminos que nos permitan identificar de qué manera podemos minimizar 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 se implementó como se especificó en la sección 1.2. Se implementa con una prioridad mixta entre seguridad y rapidez. Se toma el valor de la distancia y se divide entre el riesgo del camino. No es lo mismo recorrer un riesgo x 10m que 100m, por ende, al aplicar esta fórmula, mientras más distancia haya, mayor será la ponderación y de esta forma el algoritmo evitará estos caminos. Terminada la ponderación se le aplica Dijkstra, y sus resultados son generados en forma de un arreglo que indica las coordenadas. El algoritmo se ejemplifica en la Figura 7 y 8 con el color morado.



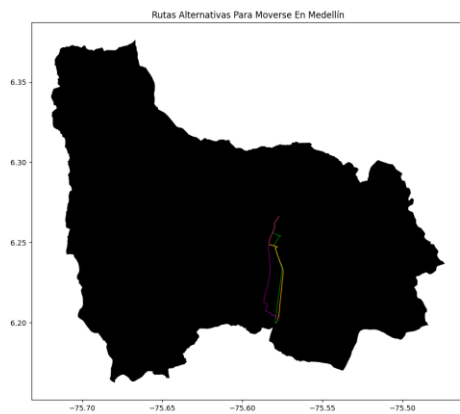
<sup>4</sup> <https://github.com/Sadriica/ST0245-5001>

**Figura 7:** Representación de los tres caminos a encontrar por el algoritmo propuesto.

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

Para el cálculo de los otros dos caminos, lo único que cambia en cada caso es la ponderación del grafo. Para la ponderación que prioriza la distancia, se calcula el camino seleccionando el largo, dividiéndolo entre 10 y sumándole el 25% del riesgo, esto con el objetivo de hacer las distancias más significativas. Terminada la ponderación se le aplica Dijkstra, y sus resultados son generados en forma de un arreglo que indica las coordenadas. El algoritmo se ejemplifica en la Figura 7 y 8 con el color naranja.

La segunda ponderación, selecciona un camino más seguro. Se toma el valor completo del riesgo y sumamos el 25% de la distancia. De esta manera se le da mayor relevancia al riesgo. Terminada la ponderación se le aplica Dijkstra, y sus resultados son generados en forma de un arreglo que indica las coordenadas. El algoritmo se ejemplifica en la Figura 7 y 8 con el color verde.



**Figura 8:** 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

Para este algoritmo en el peor de los casos la complejidad es de  $O(|A| \log |V|)$ , esto se calculó debido a que, en el peor de los casos, el algoritmo va a tener que recorrer el grafo en su totalidad, sin embargo, esto no lo va a ser una sola vez. Suponiendo que se quedó con algunas rutas en la cola de prioridad, este va a seguir recorriendo el logaritmo de los vértices.

Algoritmo	Complejidad temporal
Dijkstra	$O(( A + V ) \log  V )$

**Tabla 1:** Complejidad temporal del Dijkstra, donde A es el total de aristas y V son los vértices.

Estructura de datos	Complejidad de la memoria
Lista de adyacencia	$O(A^2)$

**Tabla 2:** Complejidad de memoria usando listas de adyacencia, donde A es la cantidad de aristas.

#### 4.4 Criterios de diseño del algoritmo

Escogimos Dijkstra por ser más óptimo en tiempo y en uso de memoria. Analizando la complejidad en tiempo, BFS era el algoritmo ganador. Sin embargo, para trabajar con grafos ponderados positivamente, se usa BFS con una modificación que conocemos como Dijkstra. Su complejidad en tiempo es la menor entre las opciones restantes ( $A^*$   $O(2^n)$  y Bellman-Ford  $O(VA)$ ). Adicionalmente las estructuras de datos en las que se apoya son de las más eficientes para realizar operaciones (insertar, remover, buscar, etc.) y a su vez ocupan poco espacio en memoria (Cola de prioridad -  $O(n)$  y lista de adyacencia -  $O(n^2)$ ).

### 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	8.9693 km	0.3176
EAFIT	UNAL	8.4147 km	0.4178
EAFIT	UNAL	8.9752 km	0.4138

**Tabla 3.** Distancia en kilómetros 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)
Camino que prioriza el riesgo	0.0340 s
Camino que prioriza la distancia	0.0369 s
Camino balanceado	0.0349 s



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

## 6. CONCLUSIONES

Basándose en los resultados que se obtuvieron, se puede encontrar que, en todos los casos, los resultados cambian de la manera esperada. Para el camino que prioriza el riesgo, se tiene que, respecto a los otros dos, es el más seguro (0,1 menos riesgo en una escala de 0 a 1). Para el camino que prioriza la distancia, se tiene que, respecto a los otros dos, es el más corto (0,5 km menos). En el caso del camino que balancea ambas variables, se puede ver una disminución en el riesgo comparado con el camino más rápido, y un ligero aumento en la distancia recorrida, comparándolo con el camino más seguro. Estos resultados, considerando su corto tiempo de ejecución, pueden ser una herramienta muy útil para todos los habitantes de Medellín que busquen una alternativa para planificar sus rutas. Se considera que, según el contexto de cada usuario, si se desea migrar esta solución a una aplicación web, deberían ser mostrados los 3 caminos, ya que cada persona puede priorizar ciertos factores por encima de otros.

### 6.1 Trabajos futuros

En un futuro se pueden realizar las siguientes mejoras al algoritmo y a la aplicación. En primer lugar, se buscaría tener un output más amigable para el usuario, permitiendo visualizar de una mejor manera las rutas generadas por el algoritmo. En segundo lugar, reemplazar el dataframe CSV con una base de datos. Por último, se trabajaría en implementar un aprendizaje retroalimentado por búsquedas de usuarios y sus experiencias con las rutas, permitiendo que el algoritmo cuente con un banco de respuestas y así que sea más eficiente.

## 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*.

## REFERENCIAS

- [1] Lozano Domínguez, J. And De J. Mateo Sanguino, T., 2021. Walking Safe: Aplicación Móvil Para El Trazado De Rutas Peatonales En Las Ciudades., P.5.
- [2] Cuervo Sanchez, J. And Gonzalez Martinez, L., 2019. Aplicación Móvil Para Crear Rutas Compartidas En Usuarios De Bicicletas De Bogotá. Universidad Distrital Francisco José De Caldas, P.81.
- [3] Gómez De Merodio, P., 2019. Sistema Big Data Para La Identificación Y Predicción De Zonas Potencialmente Peligrosas Para Ciclistas. , P.94.
- [4] Del Valle Corral, V., 2018. SAFEWAY: Un Sistema De Recomendación De Rutas Basado En El

Contexto Para Fomentar La Seguridad Vial. Universidad Complutense De Madrid, P.79.

- [5] Peñaranda Ortega, M., Osca Lluch, J., López Ferrer, M., Civera Mollá, C. And Tortosa Gil, F., 2019. Funcionamiento, Representación Y Comportamiento De Diferentes Algoritmos Frente Al Cálculo De Un Small World En Ciencia. International Society For Knowledge Organization, P.15.
- [6] Geeksforgeeks, 2022. Bellman–Ford Algorithm | DP-23 - Geeksforgeeks. Geeksforgeeks.
- [7] López Mamani, M. And Murillo, J., 2020. Difference Between Breadth Search (BFS) And Deep Search (DFS). Encora.
- [8] Anon, 2005. ALGORTIMO A\*. Instituto Tecnológico De Nuevo Laredo, P.12.
- [9] Geeksforgeeks, 2022. A\* Search Algorithm - Geeksforgeeks. Geeksforgeeks.