

ALGORITMO PARA EVITAR EL ACOSO SEXUAL CALLEJERO HACIA LA MUJER

Sofía Mendieta Marín
Universidad Eafit
Colombia
smendietam@eafit.edu.co

Juan José Villa Soria
Universidad Eafit
Colombia
jjvillas@eafit.edu.co

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

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

RESUMEN

El objetivo de este informe es analizar y hallar una solución para el problema del acoso sexual callejero hacia la mujer, esto calculando las rutas más cortas y con menos probabilidad de acoso. Lo que se busca es encontrar y utilizar un algoritmo que permita calcular el camino más corto sin superar un riesgo medio ponderado de acoso r y el camino con menor riesgo medio ponderado de acoso sin superar una distancia d .

La solución a este problema es de suma importancia debido a que estamos en una ciudad en la que lamentablemente muchas mujeres todavía viven el acoso callejero en su día a día, y como consecuencia se generan otros problemas tales como el miedo a salir solas a las calles, gastar más dinero en servicios de transporte, ya sea público o privado, cuando muchas veces no es necesario, entre otros. El algoritmo que se ha utilizado es una versión modificada de Dijkstra; decidimos emplearlo ya que es un algoritmo para la determinación del camino más corto dado un origen y un destino, de esta manera podemos implementar todos los elementos necesarios para solucionar la problemática.

Palabras clave

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

1. INTRODUCCIÓN

Hoy en día vivimos en una sociedad en la que en ciertos aspectos hay muchos avances, como en el tecnológico; sin embargo, todavía nos falta mejorar mucho otros aspectos, como el cultural, es por esto que nuestro progreso no se puede quedar solo en lo tecnológico, debemos tratar de mejorar cada día los valores que nos hacen humanos. Así que lo mejor que podemos hacer es utilizar aquello en lo que ya estamos avanzados para poder progresar en lo que no; como en este caso que deseamos utilizar la tecnología para ayudar a buscar la seguridad de la mujer. Hay muchísimos antecedentes del problema, son innumerables los casos de acoso que pueden llegar a sufrir las mujeres día a día. Por ejemplo, según un artículo de Ruta N Medellín, el acoso lo viven 34,6% de las adolescentes de Medellín que, según cifras de la Secretaría de las Mujeres de la Alcaldía, dijeron que son víctimas de acoso callejero varias veces al día; y el 60% de las mujeres que dijeron sentir que Medellín no es una ciudad segura para ellas.[1] Es por esto que necesitamos calcular el camino más corto y con menor riesgo.

1.1. Problema

El problema al cual nos enfrentamos se basa en crear un algoritmo que nos permita calcular el camino más corto sin superar un riesgo medio ponderado de acoso r y el camino

con menor riesgo medio ponderado de acoso sin superar una distancia d .

Resolver este problema sería muy útil debido a que ayudaríamos a cientos de mujeres que sufren el acoso en su vida cotidiana, mejorando así su calidad de vida y brindando por su seguridad.

1.2 Solución

La solución a lo anteriormente mencionado consiste en emplear algunos algoritmos que nos permiten encontrar caminos más cortos restringidos. En este caso hemos usado una versión modificada de Dijkstra; decidimos emplearlo ya que es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de vértices en un grafo con pesos en cada arista. De esta manera podemos implementar todos los elementos necesarios para solucionar el problema.

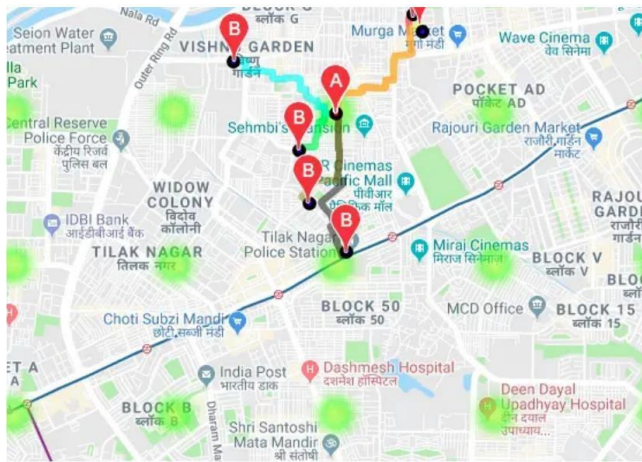
2. TRABAJOS RELACIONADOS

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

3.1 Prevención del acoso sexual

El problema que resolvieron fue encontrar las rutas más seguras para prevenir el acoso sexual, teniendo en cuenta la distancia y los puntajes de riesgo (los cuales ilustran con colores cálidos). Todo esto mediante la creación de un algoritmo de búsqueda de rutas combinado con mapas de calor para identificar puntos seguros en relación con las coordenadas y direcciones del usuario. Como resultado obtuvieron un algoritmo el cual, primeramente, prioriza la seguridad de la ruta, antes de tener en cuenta la distancia (es decir, se sugiere primero la ruta más segura, y si dos rutas están empatadas en seguridad, luego se sugiere el destino más cercano), todo esto utilizando el análisis de mapas de calor para prevenir los casos de acoso sexual.

Por ejemplo:



Como se puede ver, hay varias ubicaciones de hospitales en un radio de 800 metros; sin embargo, es bastante intuitivo evitar las ubicaciones superiores, ya que las puntuaciones de riesgo son más altas (ilustradas por colores más cálidos). Mirando esto, se puede simplemente tomar una decisión sobre por dónde pasar: Priorizando la seguridad general de la ruta y eligiendo el lugar más cercano.[6]

3.2 La ruta segura en NY

El problema que resolvieron fue predecir las rutas más seguras utilizando datos sobre delitos y accidentes. En este trabajo se propone una solución que sugiere a las personas el camino más seguro para viajar desde el origen hasta el destino. La tecnología de la API de Google combinada con modelos de aprendizaje automático, concretamente el algoritmo de agrupación K Means y el algoritmo KNN Regressor, se utilizan en esta solución propuesta. Se tienen en cuenta conjuntos de datos de arrestos y accidentes de la ciudad de Nueva York para predecir la ruta más segura.

En contraste con la sugerencia de rutas Google maps, que se basa puramente en la distancia más corta; esta solución es modificada y recomienda rutas que son seguras para viajar. La ruta segura en esta solución significa la ruta con la puntuación de riesgo más baja y se calcula en base a los accidentes y delitos ocurridos en esa ruta o en las regiones cercanas.

Esta solución es muy útil especialmente para las personas que son nuevas en la ciudad o son turistas.[7]

3.3 Búsqueda de rutas para viajeros

Este estudio ha desarrollado un método para incorporar la seguridad en el proceso de búsqueda de rutas, mediante el desarrollo y la aplicación de un nuevo índice de seguridad, este estudio ha desarrollado una metodología para incorporar aspectos de seguridad en el proceso de búsqueda de rutas de los viajeros.

El proceso de búsqueda de rutas seguras consta de dos partes principales: un índice de riesgo de seguridad específico de la ruta y un algoritmo de búsqueda de rutas que tiene en cuenta

tanto el tiempo de viaje como la seguridad. El índice de riesgo de seguridad se formuló utilizando el mecanismo de colisión a lo largo del segmento de carretera y en la intersección, la información específica de los automovilistas (por ejemplo, tipo de vehículo, edad, estado del pavimento), y en el modelo de índice de seguridad para poder tener en cuenta las necesidades individuales de los viajeros.

El algoritmo de búsqueda de rutas, que combina movilidad y seguridad, tiene tres objetivos: Reducir el tiempo de viaje, reducir el índice de riesgo de seguridad de la ruta y evitar los lugares con mayor índice de riesgo para la seguridad a lo largo de la ruta. Ya que se debe asignar una puntuación de seguridad a instalaciones de tráfico individuales (por ejemplo, segmento de carretera, intersección o a toda la ruta), utilizaron Dijkstra, el cual calculó las puntuaciones de cada ruta utilizando criterios propuestos, como la minimización de las transiciones entre categorías de carreteras, la minimización del número de giros a la izquierda y la densidad de las intersecciones lo más baja posible.[8]

3.4 Seguridad para los turistas motorizados

En este trabajo se construyó una aplicación de enrutamiento para que los turistas motorizados eviten conducir en zonas relativamente peligrosas.

Se presenta un enfoque de enrutamiento que evita las zonas relativamente peligrosas de una ciudad. La información sobre la peligrosidad de algunas zonas urbanas se obtiene utilizando información geográfica voluntaria (VGI), datos abiertos gubernamentales para detectar propiedades y funcionalidades de la infraestructura urbana y datos históricos de delincuencia de los departamentos de policía para detectar puntos calientes de delincuencia. Por lo tanto, se presenta los fundamentos de la cartografía y el análisis de la delincuencia con SIG, el uso práctico de la VGI para el enrutamiento y se describe la contribución dentro del campo de las soluciones de enrutamiento.

Para el uso práctico, se simplifica toda la información de la infraestructura urbana y se propone un índice de seguridad, que representa la seguridad relativa en el área de investigación. Además, se detectan los puntos conflictivos históricos y se utilizan como obstáculos de enrutamiento. Los arcos de la red de carreteras se ponderan en función del índice de seguridad y los puntos conflictivos históricos se introducen como polígonos de obstáculos.

En este trabajo se utiliza un algoritmo de dijktra modificado para resolver la problemática.[9]

3. MATERIALES Y MÉTODOS

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos del camino más corto restringido para abordar el 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. La (i) 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 se obtuvieron de los metadatos proporcionados por OSM.

Para este proyecto, se calculó la 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.

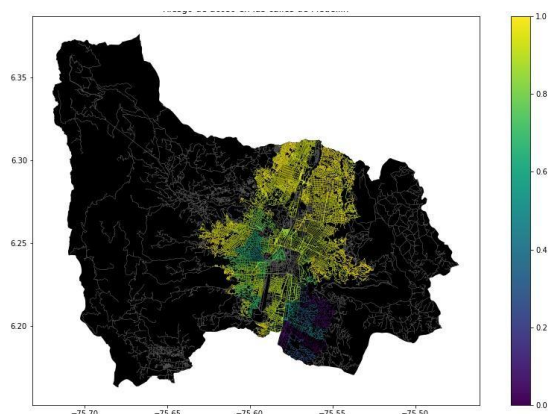


Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenida de la Encuesta de Calidad de Vida de Medellín, de 2017.

3.2 Alternativas de camino más corto con restricciones

A continuación, presentamos diferentes algoritmos utilizados para el camino más corto restringido.

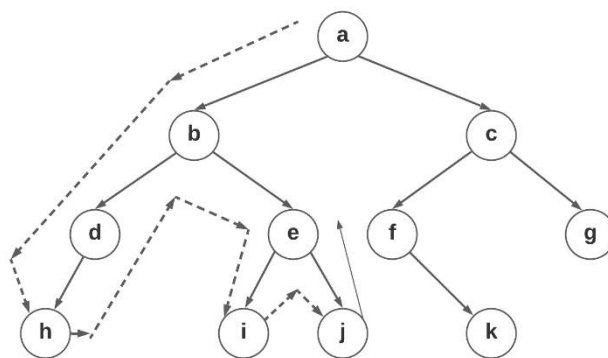
3.2.1 DFS

Una búsqueda en profundidad (DFS) 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. Cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

La búsqueda en profundidad se usa cuando se quiere probar si una solución entre varias posibles cumple con ciertos requisitos. [2]

Complejidad: $O(V + E)$

Ejemplo de su figura vectorial:



3.2.2 BFS

Una búsqueda en anchura (BFS) 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. Cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

La búsqueda por anchura se usa para aquellos algoritmos en donde resulta crítico elegir el mejor camino posible en cada momento del recorrido. [3]

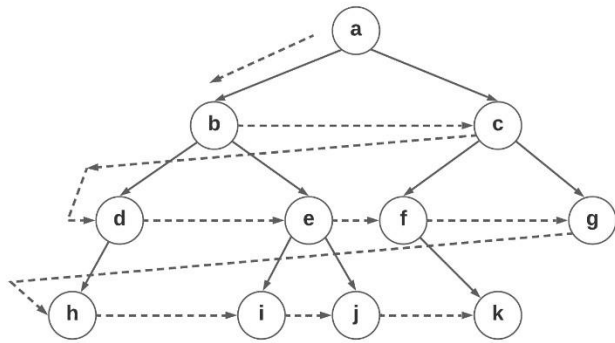
Complejidad: $O(V + E)$

Ejemplo de su figura vectorial:

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

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

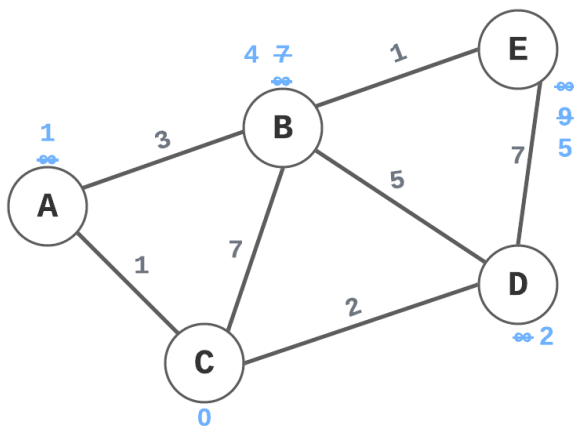


3.2.3 Dijkstra

El algoritmo de Dijkstra. También llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de vértices en un grafo con pesos en cada arista. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959. [4]

Complejidad: $O(|V|^2 + |E|)$

Ejemplo de su figura vectorial:



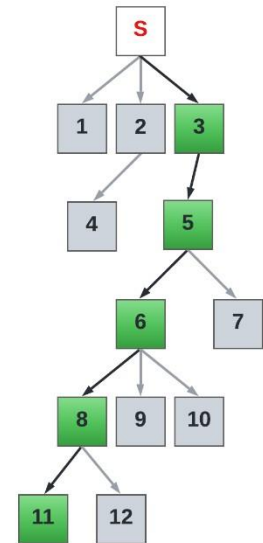
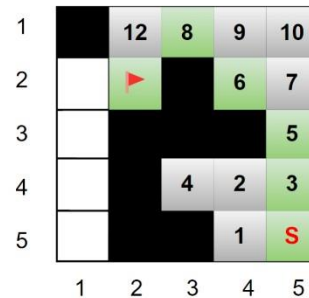
3.2.4 A*

El algoritmo A* es un algoritmo de búsqueda que puede ser empleado para el cálculo de caminos mínimos en una red. Se va a tratar de un algoritmo heurístico, ya que una de sus principales características es que hará uso de una función de evaluación heurística, mediante la cual etiquetará los diferentes nodos de la red y que servirá para determinar la

probabilidad de dichos nodos de pertenecer al camino óptimo. [5]

Complejidad: $O(n^3)$

Ejemplo de su figura vectorial:



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 implementar el algoritmo del camino más corto restringido se utilizó una estructura de datos llamada lista de adyacencia utilizando un diccionario, la cual es una representación de todas las aristas o arcos de un grafo mediante una lista; sin embargo, en este caso en nuestra implementación de la clase Vertice usaremos un diccionario en lugar de una lista. La estructura de los datos se presenta en la Figura 2.

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

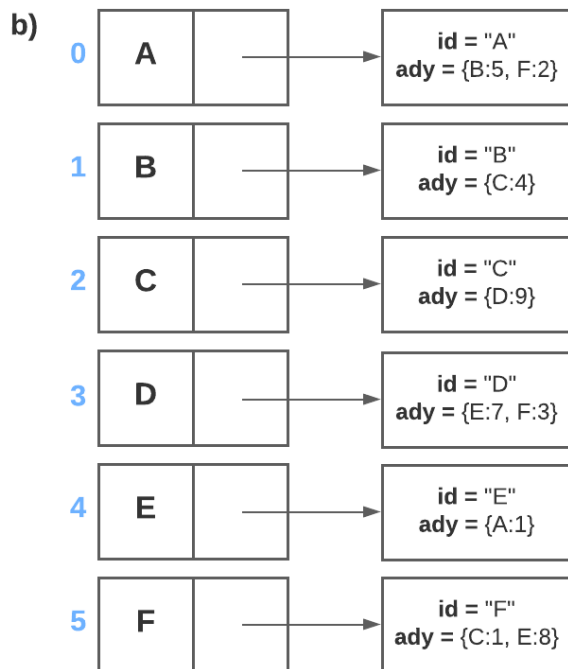
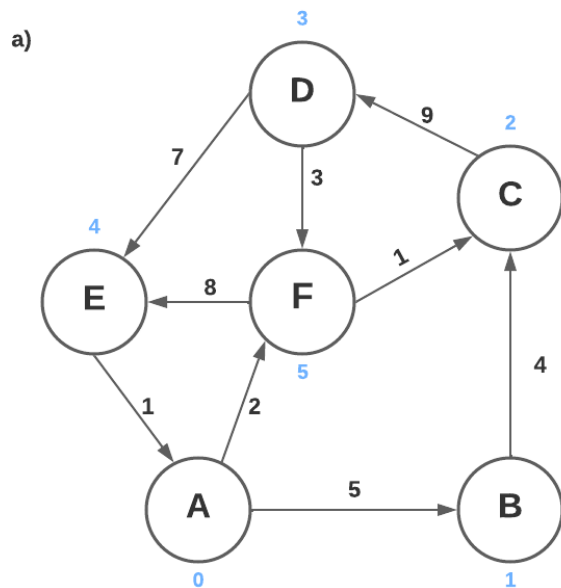


Figura 2: Un ejemplo de mapa de calles se presenta en (a) y su representación como lista de adyacencia utilizando un diccionario en (b).

4.2 Algoritmos

En este trabajo, proponemos algoritmos para el problema del camino más corto restringido. El primer algoritmo calcula el camino más corto sin superar un riesgo medio ponderado de acoso r . El segundo algoritmo calcula el camino con el menor

riesgo medio ponderado de acoso sin superar una distancia d .

4.2.1 Primer algoritmo

Este algoritmo está diseñado para calcular la menor distancia posible sin superar cierto riesgo de acoso como se representa en la gráfica mostrando las flechas rojas como la zona de riesgo, y las flechas verdes como el camino más corto y seguro del mapa. El algoritmo se ejemplifica en la Figura 3.

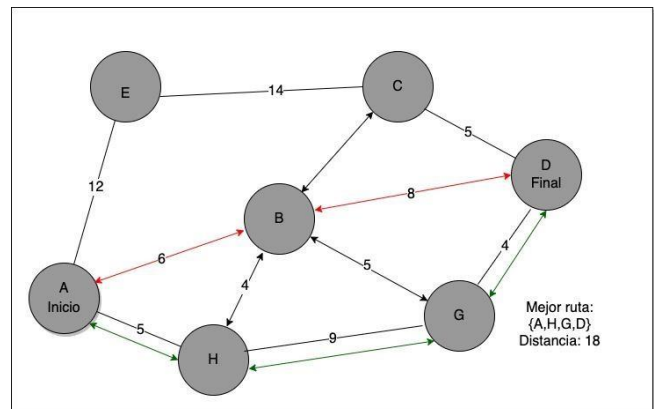


Figura 3: Resolución del problema del camino más corto restringido con una una versión modificada de Dijkstra.

4.2.2 Segundo algoritmo

Como Segundo algoritmo, decidimos usar nuevamente una una versión modificada de Dijkstra para calcular el camino con el menor riesgo medio ponderado de acoso sin superar una distancia. El algoritmo se ejemplifica en la Figura 4.

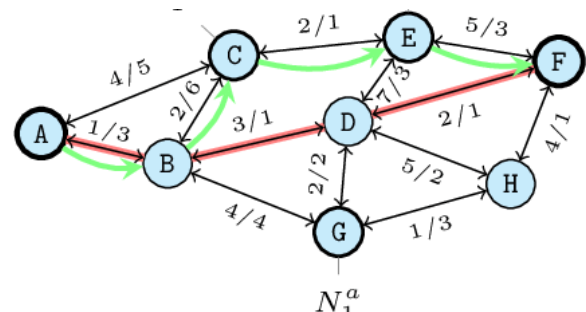


Figura 4: Resolución del problema del camino más corto restringido con una una versión modificada de Dijkstra.

4.4 Análisis de la complejidad de los algoritmos

El análisis para el peor caso utilizando la notación O para algoritmo de Dijkstra será de $O(V^2)$. Esto se sabe ya que el peor caso es que el dato, es decir, la coordenada que estoy buscando se encuentre al final del diccionario.

Algoritmo	Complejidad temporal
Dijkstra	$O(V^2)$

Tabla 1: Complejidad temporal del algoritmo de Dijkstra, donde V es la cantidad de nodos que se recorrerán.

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

Tabla 2: Complejidad de la memoria de la lista de adyacencia utilizando un diccionario, donde V es la cantidad de coordenadas almacenadas.

4.5 Criterios de diseño del algoritmo

El algoritmo fue diseñado de esta manera ya que es el que más se acoplaba tanto a los conocimientos que teníamos a la mano, como a las herramientas que teníamos a nuestra disposición; a pesar de que la complejidad no es la mejor, sigue siendo un algoritmo funcional en el cual se podrá seguir trabajando a lo largo del tiempo para seguir obteniendo mejores resultados.

5. RESULTADOS

En esta sección, presentamos algunos resultados cuantitativos sobre el camino más corto y el camino con menor riesgo.

5.1.1 Resultados del camino más corto

A continuación, presentamos los resultados obtenidos para el camino más corto, sin superar un riesgo medio ponderado de acoso r , en la Tabla 3.

Origen	Destino	Distancia más corta	Sin exceder r
Universidad EAFIT	Universidad de Medellín	6 km	1.02
Universidad de Antioquia	Universidad Nacional	4 km	1.52
Universidad Nacional	Universidad Luis Amigó	1,4 km	0.84

Tabla 3. Distancias más cortas sin superar un riesgo de acoso medio ponderado r .

5.1.2 Resultados de menor riesgo de acoso

A continuación, presentamos los resultados obtenidos para el trayecto con menor riesgo de acoso medio ponderado, sin superar una distancia d , en la Tabla 4.

Origen	Destino	Acoso más bajo	Sin exceder d
Universidad EAFIT	Universidad de Medellín	0.87	5,000
Universidad de Antioquia	Universidad Nacional	1.82	7,000
Universidad Nacional	Universidad Luis Amigó	0.86	6,500

Tabla 3. Menor riesgo de acoso ponderado sin superar una distancia d (en metros).

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.

	Tiempos medios de ejecución (s)
Universidad EAFIT a Universidad de Medellín	28.1s
De la Universidad de Antioquia a la Universidad Nacional	32.5s
De la Universidad Nacional a la Universidad Luis Amigó	34.25s

Tabla 4: Tiempos de ejecución del algoritmo Dijkstra para las consultas presentadas en la Tabla 3.

6. CONCLUSIONES

Con base al proceso realizado con el código y con las librerías extraídas, hemos llegado a la conclusión de que los caminos si se recortaron de manera efectiva, también teniendo en cuenta los caminos de menor riesgo de acoso, la utilidad que le podemos dar a este proyecto es de gran interés ya que se puede mejorar y así generar interés al observador, con base a la utilidad de la ciudad es fabuloso, ya que ayudaría a muchas personas a tomar una mejor decisión a la hora de acortar el camino y prevenir la zona de mayor riesgo. Ya para finalizar la conclusión creemos que debemos mejorar el tiempo de ejecución para que así su implementación en la vida real sea más optima.

6.1 Trabajos futuros

Nos gustaría mejorar en el futuro más la interfaz grafada para que sea más amigable con el observador. Con respecto a mejorar el algoritmo de la aplicación, nos gustaría reducir el código para que así también se optimice la complejidad a la hora de ejecutar el programa. Por otro lado, la metodología de mostrar este programa seria en una plataforma web o aplicativo para que así se accesible para todo el mundo.

AGRADECIMIENTOS

Esta investigación ha sido apoyada/parcialmente apoyada por la beca Sapiencia, financiada por el municipio de Medellín. Agradecemos la ayuda de nuestros compañeros que nos ayudaron a guiarnos en momentos de dificultad y por los comentarios que mejoraron en gran medida este manuscrito. 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 *Shapefile*.

REFERENCIAS

1. Ruta N. Reducir el acoso callejero: el reto de la secretaría de mujeres. Recuperado el 22 de febrero de 2022 de:
<https://www.rutanmedellin.org/es/programas-vigentes/2-uncategorised/592-reto-de-mujeres>

2. López, M. DFS vs BFS. Recuperado el 22 de febrero de 2022 de:
<https://www.encora.com/es/blog/dfs-vs-bfs>
3. López, M. DFS vs BFS. Recuperado el 22 de febrero de 2022 de:
<https://www.encora.com/es/blog/dfs-vs-bfs>
4. Algoritmo de Dijkstra. Recuperado el 22 de febrero de 2022 de:
https://www.ecured.cu/Algoritmo_de_Dijkstra
5. Algoritmo A*. Recuperado el 22 de febrero de 2022 de:
<http://idelab.uva.es/algoritmo>
6. Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search. Recuperado el 22 de Febrero de 2022 de:
<https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/Trabajos->

relacionados/Preventing-Sexual-Harassment-Through-a-Path-Finding-Algorithm-Using-Nearby-Search.pdf

7. Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data, Venkatesh Gauri Shankar and Chaurasia Sandeep. Recuperado el 22 de febrero de 2022 de:
<https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/Trabajos-relacionados/Route-The-Safe.pdf>
8. Incorporating a Safety Index into Pathfinding. Recuperado el 22 de febrero de 2022 de:
<https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/Trabajos-relacionados/Incorporating-a-Safety-Index-into-Pathfinding.pdf>
9. Safety-aware routing for motorised tourists based on open data and VGI. Recuperado el 22 de febrero de 2022 de:
https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/Trabajos-relacionados/Safety-aware_routing_for_motorised_tourists_based_.pdf