ALGORITMO PARA LA SELECCIÓN DE LA RUTA MÁS CORTA Y SEGURA

Vanessa Alexandra Velez Restrepo
Universidad Eafit
Colombia
vavelezr@eafit.edu.co

Luis Miguel Giraldo Gonzalez
Universidad Eafit
Colombia
lmgiraldo4@eafit.edu.co

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

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

Para cada versión de este informe: 1. Borre todo el texto en rojo. 2. Ajuste los espacios entre palabras y párrafos. 3. Cambiar el color de todos los textos a negro.

Texto en rojo = Comentarios

Texto en negro = Contribución de Andrea y Mauricio

Texto en verde = Para completar la primera entrega

Texto azul = A completar para el 2º entregable

Texto en color violeta = A completar para el tercer entregable

RESUMEN
El problema que se nos plantea trata sobre aplicar dos algoritmos para la ruta más corta, pero con una restricción, evitar pasar por lugares donde se considera que hay mucho acoso callejero. El problema es importante dado que este puede desencadenar a problemas mayores, entre ellos el feminicidio, el cual inicia usualmente con un simple acoso verbal y físico. ¿Cuál es el algoritmo que has propuesto para resolver el problema? ¿Qué resultados cuantitativos has obtenido? ¿Cuáles son las conclusiones de este trabajo? El resumen debe tener **como máximo 200 palabras**. (*En este semestre, debes resumir aquí los tiempos de ejecución, y los resultados del camino de menor riesgo y del camino más corto*).

Palabras clave

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

1. INTRODUCCIÓN

Los seres humanos estamos en constante apuros, sintiendo que nos falta el tiempo, y una medida para ahorrarlo, es dirigirnos por una ruta que nos parece más corta, pero esta trae consigo varios problemas y uno de ellos es la seguridad. Al momento de planear nuestra ruta, no nos fijamos en que sea segura porque estamos enfocados en querer ahorrar tiempo, muchas veces al momento de estar transitando por esta ruta, nos damos cuenta que es muy peligrosa, que puede tener altos índices de robos, actividades ilícitas y también acoso callejero. Pero por otro lado, cuando vemos que tenemos bastante tiempo queremos *evitar pasar por lugares inseguros, pero esto no siempre se puede hacer.*

1.1. Problema

En los últimos años se ha aumentado la tendencia de acoso callejero, pues aunque este inicia con un uso morboso de las palabras y comentarios innecesarios, desencadena que en los alrededores donde sucede esto se refleja una mala imagen del lugar, también que muchas personas se sienten inseguras al momento de salir a la calle a cualquier hora del día y da pie a que actividades ilícitas inicien en lugares como estos. Queremos resolverlo para evitar que el problema de acoso callejero pase a sucesos mayores.

1.2 Solución

Se busca encontrar el camino más corto con restricción de que no se pase un promedio ponderado de acoso. De esta forma lo primero que se pretende solucionar es que entre todos los caminos posibles, se encuentre el más corto y por consiguiente cuando ya se tenga este, agregar otra restricción que sería el promedio ponderado de acoso. Se eligió el algoritmo DFS (Búsqueda en profundidad) dado que es el más accesible para nosotros, además en el algoritmo del que poseemos más conocimiento para realizarlo; creemos que al realizar primero la búsqueda de todos los caminos posibles, al final es más sencillo devolver la ruta más corta en metros.

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.

3.1 Herramientas frente al acoso sexual callejero

Este proyecto de investigación pretende facilitar el uso de herramientas para prevenir el acoso sexual callejero, mediante el uso de la página web y aplicación móvil "Solas, ya no", la cual fue creada por el mismo autor; esta tecnología pone al servicio a cualquier mujer que este en peligro de sufrir acoso sexual callejero a un solo click de distancia.

"Solas, ya no" usa parámetros para poder determinar si es un caso de abuso sexual, entre ellos está la opción de presionar el botón de la aplicación por dos segundos, instantáneamente el sistema activara un sonido ensordecedor para apartar a los acosadores, y aunque eso pareciera insuficiente, seguidamente la aplicación le avisará a los cinco contactos más cercanos escogidos por la victima previamente acerca de su situación, además de su ubicación actual, la cual se actualiza cada treinta segundos. En cuanto a la víctima, la aplicación le mostrará los lugares más seguros cercanos y poseerá de llamadas instantáneas a un click como la policía y contactos seleccionados anteriormente. [1]

3.2 Incorporating a Safety Index into Pathfinding

Los accidentes de tránsito han sido uno de los problemas más preocupantes por los conductores, por lo cual este proyecto realizado en manos de Zhaoxiang He y Xiao Qin da solución a la seguridad, mostrando el camino más corto y fiable en un trayecto determinado, llevando a cabo un sistema que selecciona el camino más adecuado con base en los lugares con mayor y menor índice de accidentalidad y los trayectos posibles más cortos y más largos medidos en tiempo real con una red.

Estos dos investigadores basaron su algoritmo principalmente mediante el índice MADR, el cual tiene en cuenta muchas variables, entre ellas el peso del vehículo, las condiciones de la vía, y el tráfico de la ruta; por consiguiente, si el programa arroja un índice bajo indica que es un camino seguro, pero si por el contrario arroja un índice alto indica que la ruta indicada no es la más recomendable.

Este sistema ha sido experimentado exitosamente en la Universidad de Wisconsin, arrojando cinco rutas con el trayecto más corto y con su respectivo índice de MADR, permitiendo al usuario escoger la ruta de su preferencia. [2]

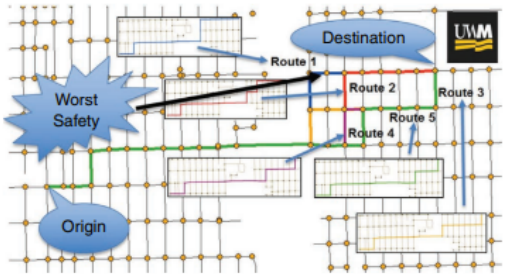


Figura 5. Posibles rutas dentro de la Universidad de Wisconsin arrojadas por el programa de acuerdo al índice MADR.

3.3 Beyond the Shortest Route: A Survey on Quality-Aware Route Navigation for Pedestrians

En este artículo se evidencian los criterios que se toman al recomendar una ruta, una categorización de esos sistemas basados en el algoritmo SWEEP (En inglés: Safety-Whealth-Effort-Exploration-Pleasure) y también en sus conclusiones muestran los problemas potenciales en sus estudios anteriores, además de la futura dirección que le darán a su estudio.

El artículo habla de las diferentes calidades, atributos, fuentes de datos y algoritmos que se han utilizado para implementar este tipo de sistemas pero para peatones. Antes se solían basar los sistemas de recomendación de rutas para minimizar distancias, recursos y tiempo. Pero ahora, ya se basan en más factores, como por ejemplo en las experiencias positivas de otros usuarios que han pasado por ahí, también basándose en la seguridad y en evitar lugares donde el usuario pueda enfrentar algún peligro. [3]

3.4 Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search

Este artículo de lo que trata es que buscan lugares seguros y de esos lugares seguros buscan hacer que una ruta pase por ahí. Se puede evidenciar como utilizan un mapa donde los puntos en rojo es en donde hay más acoso sexual y los puntos verdes es en donde menos hay.

Usaron la API de los mapas de google para saber las direcciones y con esto construyeron una ruta de un punto A a un punto B que pasaba por lugares seguros para caminar. El primer algoritmo que utilizaron fue la fórmula de Euclides para calcular esta distancia. Pero ¿Cómo determinaron que una ruta era segura? Primero clasificaron los puntos de riesgo el 0 siendo el más seguro y 4 siendo el punto más peligroso.

El problema lo llevaron de cosas más simples a más complejas:

- Determinaron el riesgo asociado con el destino.
- Calculan el promedio del riesgo de la ruta en una línea recta desde el origen hasta el destino.
- Calculan el promedio del riesgo de la ruta en una línea recta de cada paso hasta el destino. [4]

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 API[2] 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 normaliza, 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 GitHub[3].

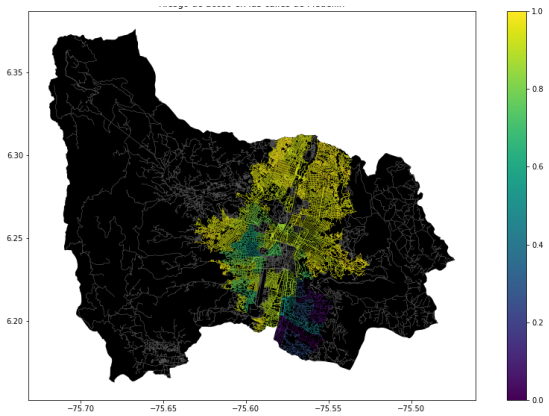


Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenida de la Encuesta de Calidad de Vida de Medellín, de 2017.

3.2 Alternativas de camino más corto con restricciones

A continuación, presentamos diferentes algoritmos utilizados para el camino más corto restringido. (En este semestre, ejemplos de dichos algoritmos son DFS, BFS, una versión modificada de Dijkstra, una versión modificada de A*, entre otros).

3.2.1 On an exact method for the constrained shortest path problem

Este algoritmo semi-complejo se encarga de encontrar el camino más corto a partir del nodo inicial ($V_{0)}$ y del nodo final (V_k) (ambos datos ingresados por el usuario), posteriormente el algoritmo enumera todas las rutas posibles de V_0 y V_k, esto lo hace posibles gracias a que el sistema tiene un registro vector binario de los nodos ya visitados; seleccionando el camino más factible a partir de la ruta que tenga el mínimo consumo para cada nodo del trayecto. [5]

Figura 6: Selección del camino más corto con distancias y nodos.

3.2.2 Algoritmos para la ruta más corta en un Grafo.

Es un algoritmo que calcula los costos mínimos desde un punto de origen a un punto de destino. Este algoritmo lo que busca es optimizar, en cada paso selecciona un vértice donde la distancia de este es desconocida y lo que hace es mirar las conexión entre este vértice y otro. Es un algoritmo que está categorizado como algoritmo de grafos. Después de hacer todas las conexiones entre los vértices y de hallar las distancias entre cada vértice, lo marca como ya es conocido.

En este artículo nos hablan de como un ejemplo podría ser encontrar la ruta entre 2 ciudades, los vértices serían las ciudades y las aristas la duración del vuelo entre cada ciudad. Después de ir por todas las rutas posibles, el

[1] https://www.openstreetmap.org/

[2] https://osmnx.readthedocs.io/

[3]https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets/

algoritmo lo que hace es almacenar el total de distancia y nos devuelve la distancia más corta y por ende la ruta que tiene esta distancia. [6]

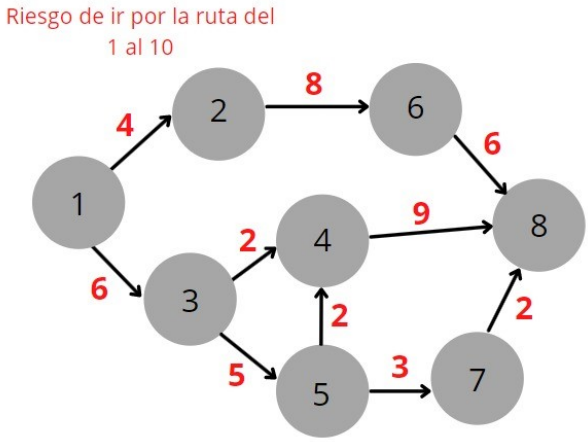


Figura 7. Algoritmo de Dijkstra modificado en riesgo de rutas.

3.2.3 Shortest path in a maze – Lee Algorithm

El algoritmo se basa primordialmente en matrices binarias, lo cual solo se puede moverse hacia cuatro direcciones comunes (, ,↑,↓). Básicamente lo que hace el código es en un ciclo toparse con cada posición, mira las 4 posiciones adyacentes a la actual y a aquellas que tienen un valor de uno, se mueve hacia esa posición y añade un +1 al contador, el algoritmo hace este proceso repetitivamente hasta que se encuentra con la posición final, ahí retorna el contador(distancia), repite el mismo proceso para mirar las otras posibles rutas y devuelve finalmente la ruta más corta con la condición de que haya recorrido todas las posiciones de la matriz, de lo contrario arrojaría "false". [7]

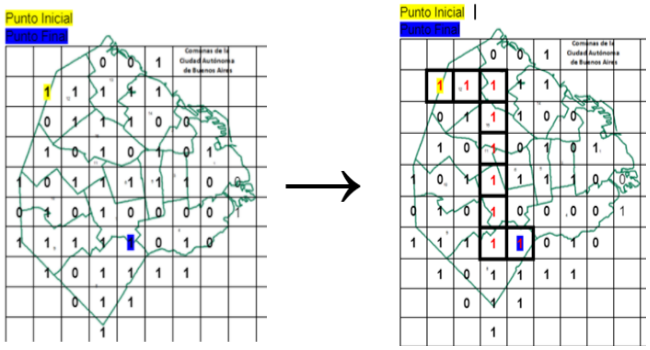


Figura 8. Algoritmo en matriz binaria.

3.2.4 Shortest path in a maze using backtracking

Primero en este algoritmo se dio un problema, el cual es una matriz rectangular donde se simula un laberinto y se pide encontrar el camino más corto donde solo se pase por casillas que tengan valor de 1 y solo se puede dar 1 paso en cualquier dirección. También cabe aclarar, que este algoritmo sólo permite movimientos así:

Arriba: (x, y-1)

Abajo: (x, y+1)

Derecha: (x+1, y)

Izquierda: (x-1, y)

El algoritmo vuelta atrás (En inglés backtracking) nos dice que podemos agotar todos los caminos posibles en el laberinto desde la posición inicial hasta la posición de meta hasta que se agoten todas las posibilidades, esto de forma recursiva mirando hacia todas las direcciones.

Para saber cuál fue el camino más corto, se tiene una longitud de la ruta y siempre que se alcance la celda de destino, esta se actualizará con el número total de celdas por las que se pasó. Y antes de explorar cualquier celda, se ignorará la celda si ya está cubierta en la ruta actual. [8]

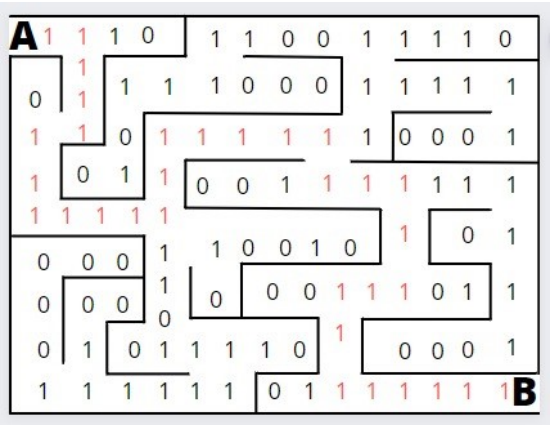


Figura 9. Resolución de problema de backtracking en un laberinto..

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[4].

4.1 Estructuras de datos

Se utilizará la estructura de datos matriz de adyacencia. La estructura de los datos se presenta en la Figura 2.

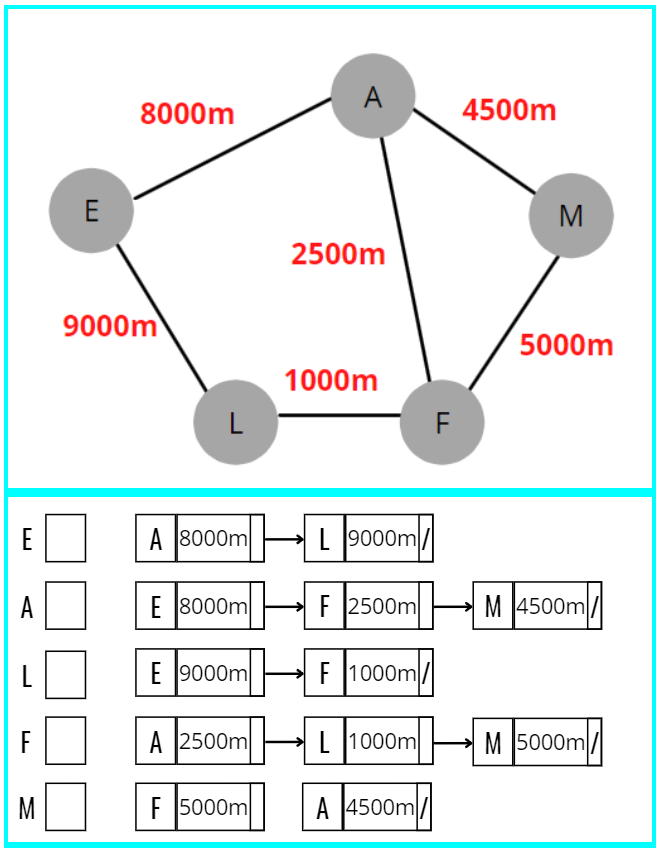


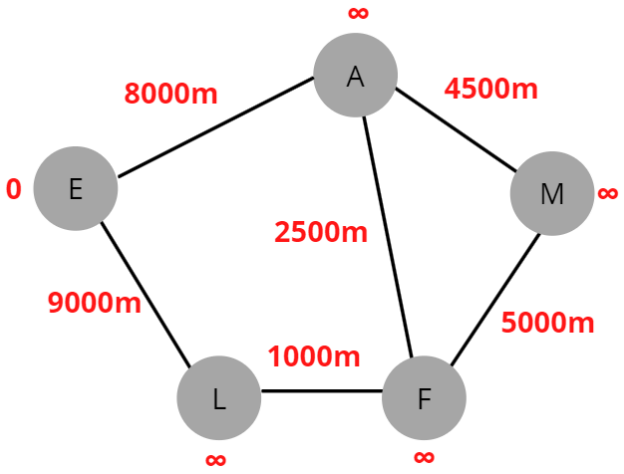
Figura 2: Mapa de calles representado en un grafo ponderado no dirigido, y su representación como matriz de adyacencia para un grafo ponderado no dirigido

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

El algoritmo que se utilizará será el algoritmo de Dijkstra. La primera parte de nuestro algoritmo será utilizado para calcular la ruta más corta. Posteriormente encontraremos la ruta con menos riesgo de acoso.



El 0 es nuestro vértice inicial y en nuestro algoritmo miramos si el 0+8000 < ∞, en este caso si es menor entonces cambiamos ese infinito por 8000 y así con los siguientes vértices.

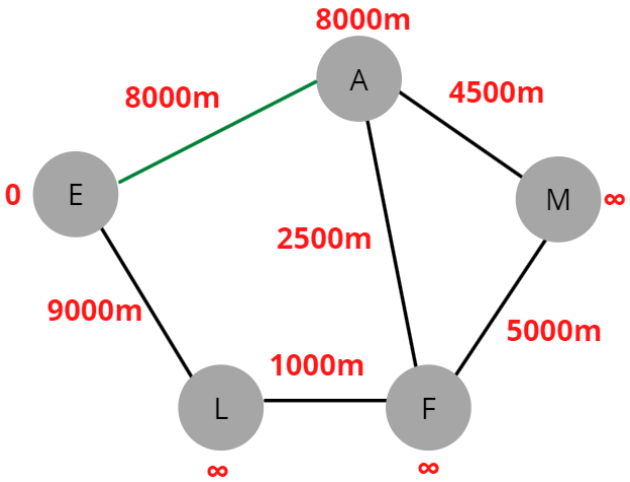


Figura 3: Resolución del problema del camino más corto restringido con Dijkstra.

4.2.2 Segundo algoritmo

[4] https://github.com/gotaluism/ST0245-002

El algoritmo Dijkstra busca resolver los caminos que hay entre diferentes vértices de un grafo, teniendo en cuenta los pesos, es muy importante saber que no funciona si alguno de estos pesos son negativos. Encuentra la ruta más corta desde el vértice inicial hasta dónde se quiere llegar y nos devuelve la ruta que según la suma de sus pesos es la más corta.

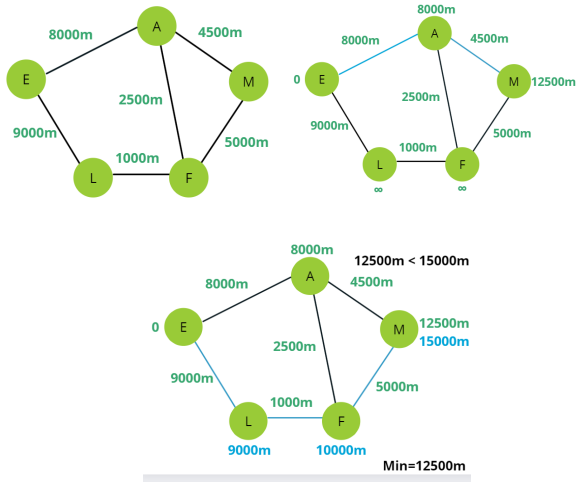


Figura 4: Resolución del problema del camino más corto restringido con Dijkstra

4.4 Análisis de la complejidad de los algoritmos

Es el análisis de la complejidad de los algoritmos y estructuras de datos en el peor de los casos.

Algoritmo	Complejidad temporal
Dijkstra	O(E*log(V))

Tabla 1: Complejidad temporal del algoritmo Dijkstra;, donde V es el vértice de 1 a n, y E es el número total de aristas.

Estructura de datos	Complejidad de la memoria
Lista de Adyacencia	$O(V^2)$

Tabla 2: Complejidad en memoria de la lista de adyacencia, donde V es el vértice de 1 a n, y E es el número total de aristas.

4.5 Criterios de diseño del algoritmo

Usamos la librería de networkx para el código, puesto que esta librería la vimos como una forma de optimizar código ya que muchos de sus métodos nos permiten usar de manera más profesional el algoritmo de dijkstra, por otra parte, usamos pydeck como medio para mostrar los resultados del algoritmo dado que es una manera más amena de mostrar los resultados de la ruta más corta sin superar un índice de acoso; además, ayuda al usuario con el mapa a entender de una forma mucho más clara la ruta a tomar.

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	6252	0.84
Universidad de Antioquia	Universidad Nacional	3226	0.83
Universidad Nacional	Universidad Luis Amigó	359	0.85

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	0.79	5,000
Universidad de Antioquia	Universidad Nacional	0.83	7,000
Universidad Nacional	Universidad Luis Amigó	0.85	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.

Calcule el tiempo de ejecución de las consultas presentadas en la Tabla 3. Indique los tiempos de ejecución medios.

	Tiempos medios de ejecución (s)
Universidad EAFIT a Universidad de Medellín	10 s
De la Universidad de Antioquia a la Universidad Nacional	8 s
De la Universidad Nacional a la Universidad Luis Amigó	2 s

Tabla 4: Tiempos de ejecución de *Dijkstra.*

6. CONCLUSIONES

A raíz de estos resultados obtenidos, concluimos que es este proyecto realizado reduce significativamente el riesgo de peatones al querer usar una ruta, ya que podría encontrar una ruta no muy larga y con poco riesgo de que sea acosado. Normalmente si se quería buscar una ruta con poco acoso, se miraba que la ruta era demasiado larga y esto no es lo más conveniente siempre. También pudimos observar que la ruta más corta, es una de las que más riesgo de acoso tiene porque se pasan por lugares donde hay mucha delincuencia y así la ruta con menos acoso es la más larga, por lo tanto logramos encontrar una ruta media entre estos dos factores.

6.1 Trabajos futuros

Nos gustaría mejorar su presentación, para que el usuario pueda elegir el nombre de donde quieres llegar y no unas coordenadas porque esto no es nada fácil de descifrar, también en temas de interfaz, para que el mapa sea más bonito y nos gustaría agregar un mapa de calor en donde se pueda observar las zonas con más acoso. También se podría implementar para otro tipo de factores, por ejemplo las zonas en donde no solo sea el riesgo de acoso sexual, sino en temas de robos, etc.

AGRADECIMIENTOS

Agradecemos la ayuda con la implementación de la estructura de datos a Kevin Sossa, monitor de la materia Estructura de datos y algoritmos 1 que fue de gran ayuda al momento de realizar el proyecto.

Agradecemos a algunos tutoriales en Youtube por enseñarnos a como era el funcionamiento del algoritmo Dijkstra.

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].Sanchéz Velasco, H .2019. *HERRAMIENTAS FRENTE AL ACOSO SEXUAL CALLEJERO*, Valencia,ESP.

[2]. He, Z .Qin, X. 2019. *Incorporating a Safety Index into Pathfinding*.

[3]. Siriaraya, P, Y. Wang, Y. Zhang, S. Wakamiya, P. Jeszenszky, Y. Kawai y A. Jatowt. 2020. *Beyond the Shortest Route: A Survey on Quality-Aware Route Navigation for Pedestrians.* IEEE Access.

[4]. Ma, D. 2020. Preventing Sexual Harassment Through a Path Finding Algorithm UsingNearby Search. Omdena.

[5].Lozano, L. Medaglia,A. 2012. *On an exact method for the constrained shortest path problem*, Bogotá,COL.

[6].N.A. 2021. *Shortest Path in Maze using Backtracking*. Pencilprogrammer. De: https://pencilprogrammer.com/algorithms/shortest-path-in-maze-using-backtracking/#:~:text=We%20can%20easily%20find%20the,(x%2C%20y%2D1)

[7]. FAANG. 2018. *Shortest path in a maze – Lee Algorithm*. De: https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze

[8]. N.A. Tema: Algoritmos para la ruta más corta en un Grafo. Udb. De: https://www.udb.edu.sv/udb_files/recursos_guias/informatica-ingenieria/programacion-iv/2019/ii/guia-10.pdf

[9]. Networkx. 2022. *Dijkstra algorithm* De:Dijkstra's algorithm | NetworkX Guide