

Alternativa a caminos para una movilización más segura en Medellín

Sebastián Escobar García
Universidad Eafit
Colombia
sescobarg6@eafit.edu.co

Juan Jose Castrillon Correa
Universidad Eafit
Colombia
jicastrilc@eafit.edu.co

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

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

Felipe Gomez Espinal
Universidad Eafit
Colombia
fgomeze1@eafit.edu.co

RESUMEN

En el presente informe se tratará el acoso sexual en la ciudad de Medellín enfocado en las mujeres, donde se presenta una alternativa que minimice esta problemática y sea más seguro movilizarse por la ciudad. Esta problemática se ha venido presentando durante siglos, y ya es momento de empezar a presentar soluciones para este tipo de situaciones y de esta forma brindar unas calles más seguras para las personas, en especial el género femenino. De igual forma se presentarán diferentes trabajos relacionados y algoritmos con posibles alternativas o soluciones no solo con temas de acoso sexual en la ciudad de Medellín, también se analizarán trabajos relacionados con las rutas más seguras respecto a la criminalidad y en diferentes lugares del mundo. Algunos ejemplos de problemas relacionados con estos en la poca seguridad para personas que son turistas en ciudades o países. También se puede relacionar la alta accidentalidad vial que se presenta tanto como en los nativos como en los turistas en puntos específicos.

Palabras clave

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

1. INTRODUCCIÓN

El acoso sexual en Medellín está siendo una problemática que cada vez se está incrementando y se buscan alternativas para minimizar al máximo el riesgo de que una mujer sea acosada mientras se moviliza por la ciudad de Medellín, donde se calcula una ruta con el menor porcentaje de riesgo sin exceder una distancia estimada y así lograr una mejor seguridad y protección mientras se movilizan. Incluya algunos antecedentes de este problema, y evitar situaciones de acoso tales como:

<https://www.eltiempo.com/colombia/medellin/metro-de-medellin-18-de-mujeres-revelaron-haber-sido-acosadas-647333>

3

<https://www.bluradio.com/blu360/antioquia/abren-investigacion-a-subdirector-del-inder-de-medellin-por-presunto-acoso-sexual-y-laboral>

<https://www.rcnradio.com/bogota/reconstruccion-de-episodios-de-violencia-sexual-denunciados-en-la-universidad-nacional>

1.1. Problema

El acoso sexual y la alta criminalidad es una problemática demasiado grave y requiere una solución inmediata, ya que las personas y en especial las mujeres cuando salen a las calles se sienten en constante amenaza y no se siente protegidas, es obvio que esto es un increíble problema. No puede suceder que a alguien le de miedo caminar a su hogar o estar en constante angustia porque en cualquier momento pueden suceder situaciones muy desagradables. El mundo es para todos los seres humanos, así que se debe hacer del mundo lo mejor posible para todos, un mundo que se puede considerar idealizado, pero si no se llega al punto de que todo sea perfecto, por lo menos poder brindar unas calles seguras para las mujeres y las personas en general que sufren de acoso.

1.2. Solución

Se escogió el algoritmo de dijkstra para la solución de la problemática del camino más corto considerando los índices de acoso que se presenta en la ciudad de Medellín, fue escogido este algoritmo debido a que fue considerado dentro diferentes opciones como el de una mejor complejidad y una rápida y efectiva respuesta a la hora de ser ejecutado.

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 Modelo de predicción para una ruta utilizando datos de criminalidad en New York

En este artículo investigativo tiene como objetivo lograr una predicción segura de una ruta para movilizarse dentro de la ciudad de New York utilizando como base de datos de criminalidad, y así creando porcentajes de probabilidad de posibles delitos que puedan ocurrir mientras se movilizan por dichos sectores.

Los algoritmos implementados en este proyecto fueron tres:

- Data preprocessing Mask Algorithm (Model Design Mask Algorithm)

Algorithm 1: Data preprocessing Mask Algorithm

1. Arrest data (crime data): AD1 \rightarrow nypd
2. Accident data: AD2 \rightarrow nypd
3. RM (law code, jurisdiction code) \rightarrow nypd
4. Na.rm \rightarrow nypd
5. RM (Outlier: OL) \rightarrow nypd
6. Column: rename \rightarrow nypd
7. 'Accident Score of a point' (A_s) \rightarrow
 - a. count of pedestrians injured - P_i
 - b. count of pedestrians killed - P_k
 - c. count of cyclist injured - C_i
 - d. count of cyclist killed - C_k
 - e. count of motorist injured - M_i
 - f. count of motorist killed - M_k
 - g. $A_s = P_i + P_k * 2 + C_i + C_k * 2 + M_i + M_k$
8. RM(Offences) \rightarrow nypd
9. 'Crime Score of a point' (C_s) \rightarrow 1 to 15
10. Dist_manhattan \rightarrow (AD1, AD2)

Model Design Mask Algorithm

- K-Mean

Algorithm 2: K-Mean

1. K-Mean (latitude, longitude) \rightarrow Nested (K-Means)
2. Elbow (Clusters) \rightarrow # of clusters
3. Nested_K Means \rightarrow store
4. Centroid (clusters) \rightarrow Nested (K-Means)
5. Index \rightarrow (AD1, AD2)
6. Cross_validation_train \rightarrow 70:30
7. 'Average crime score of a cluster' (C) \rightarrow Store
8. 'Average accident score of a cluster' (A) \rightarrow Store
9. $C = \sum_0^n C_s / n$
10. $A = \sum_0^n A_s / n$
11. Consisting of cluster number \rightarrow define
12. Store \rightarrow (C, A)

- K Nearest Neighbor

Algorithm 3: K Nearest Neighbor

1. Val(rmse) \rightarrow K
2. K-NN (latitude, longitude) \rightarrow Training
3. $R_2 \rightarrow$ Store
4. Start \rightarrow G_M (Google map)
5. Score \rightarrow min (total score)
6. Store \rightarrow (rmse, R_2)

Gracias a las bases de datos utilizadas, se generaron puntajes a cada zona de la ciudad y así clasificaron cuales serían los lugares menos probables que ocurran delitos en la ruta especificada y así evitar ingresar a zonas peligrosas, lo cual ayuda a todos los turistas y hasta a locales a moverse mucho más seguro en la ciudad de New York.[1]

2.2 Ruta más segura basada en la criminalidad, tiempo y accidentalidad vial.

En este trabajo se busca encontrar un camino seguro principalmente para las personas que no son conocedoras de la ciudad donde se encuentran, por lo general turistas. Se busca encontrar el mejor camino con tres variables principales, las cuales son la seguridad de la persona según los índices de delincuencia en puntos específicos, de igual manera usando los índices de accidentalidad en las vías y por último buscando el camino más rápido y eficiente que hay para la persona. Esto en diferentes calles y ciudades del mundo.

El algoritmo usado en este trabajo para dar solución al problema propuesto fue "Safety hazard index", el cual consta de muchas formulas usadas para resolver el problema anterior mencionado. Se usa la tasa de desaceleración mínima y máxima para calcular el riesgo de accidentalidad, también el tiempo ya la velocidad promedio de un punto A a un punto B, todo esto teniendo en cuentas variables como:

$S_{A,1-2}$ = distance A travels from Time Point 1 to Time Point 2,
 W_B = width of other Vehicle B,
 W_A = width of subject Vehicle A,
 L_B = length of other Vehicle B, and
 L_A = length of subject Vehicle A.

Los resultados obtenidos arrojan que este algoritmo y problema no pueden probar todas las posibles situaciones del mundo real, es por esto por lo que en muchos resultados los datos obtenidos no se asemejan a la vida real ya que variables como la edad de la persona, estado de la salud, etc., influye mucho en las situaciones del mundo real. Además, en el índice de accidentalidad vial solo se consideraron datos de choques traseros y de ángulo, por lo tanto, se dejan fuera muchos posibles casos del mundo real.[2]

2.3 Camino seguro para turistas motorizados

En este proyecto el problema a resolver es la creación del camino más seguro y al tiempo el más rápido para aquellas personas que se movilizan en vehículos motorizados, pero no son conocedores de los lugares peligrosos y de alto riesgo en temas de criminalidad. Esto en base de información geográfica voluntaria, datos gubernamentales y datos históricos de delitos de las estaciones de policía.

El algoritmo utilizado en este caso fue la utilización de mapas, se utilizaron los puntos donde se encuentran las estaciones de policía, las carreteras principales, los lugares con índices criminales más altos, espacio geográfico ocupado por bandas, entre otros. Luego comparando los diferentes datos obtenidos se pudo trazar la ruta más rápida, la más segura sin considerar posibles obstáculos y la más segura considerando los obstáculos, esto desde Long Beach airport a Hollywood Boulevard en Estados Unidos.

Finalmente se pudo concluir que cuando no se incluyen los obstáculos la ruta más rápida y la más segura son muy similares, pero cuando se tiene en cuenta los obstáculos la ruta cambia muy considerablemente ya que se deben evitar muchos lugares que en los otros dos casos no se consideraron, pero también difiere en muchas ocasiones de la realidad ya que estos mapas no incluyen situaciones como si es de día o de noche, por lo que deberían agregar muchos más datos para que la ruta teórica más segura y rápida con obstáculos pueda ser realmente factible en el mundo real. [3]

2.4 Mas allá de una ruta corta una ruta de calidad

Se estudiaron y analizaron diferentes aplicaciones de movilidad urbana especializada en peatones donde se compararon los factores que utilizaban e interferirán en la valoración de las rutas sugeridas teniendo en cuenta factores como la seguridad, terrenos y adversidades que se puedan presentar durante el viaje diferencian con imagen de interiores durante la ruta para una mejor experiencia para todo tipo de usuarios hasta para usuarios con

movilidad reducida como personas en silla de ruedas utilizando diversos algoritmos. Se han utilizado varios algoritmos para recomendar una ruta adecuada a los usuarios en los documentos que identificamos. “Un algoritmo popular es el algoritmo de Dijkstra, Dado un nodo inicial s en un grafo ponderado G , el algoritmo de Dijkstra encuentra el camino más corto entre el nodo y todos los demás nodos del grafo. Más específicamente, el algoritmo de Dijkstra genera un árbol de ruta más corta con un nodo de inicio determinado como raíz. El algoritmo mantiene dos conjuntos de nodos, un conjunto de nodos en el árbol de la ruta más corta y un conjunto de otros nodos que aún no se han incluido en el árbol de la ruta más corta. En cada paso del algoritmo, se identifica un nodo que no está en el árbol de la ruta más corta y tiene una distancia mínima desde el nodo inicial y se mueve al conjunto de nodos en el árbol de la ruta más corta [77]. Este algoritmo o su derivado se ha utilizado en varios sistemas de navegación cubiertos en este documento, con características tales como seguridad, estética y la posible presencia de sustancias nocivas que representan el costo a lo largo de los bordes en el gráfico, por ejemplo, utilizó el algoritmo de Epstein, una variante del algoritmo de Dijkstra, que tiene como objetivo encontrar los caminos más cortos k entre los nodos s y d y considera los casos en los que se permiten vértices repetidos” y también se utilizó el algoritmo A^* (ver en el gráfico [80]), un algoritmo de “primera búsqueda de la mejor” que utiliza una función heurística además de la función de costo para buscar un camino apropiado. se concluyó que existen 5 categorías claves: seguridad, salud y bienestar, esfuerzo, exploración y placer, pero también se resaltó el tiempo, el costo y la limitación de que se necesitaría una constante supervisión manual de estas rutas. [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.

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

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.

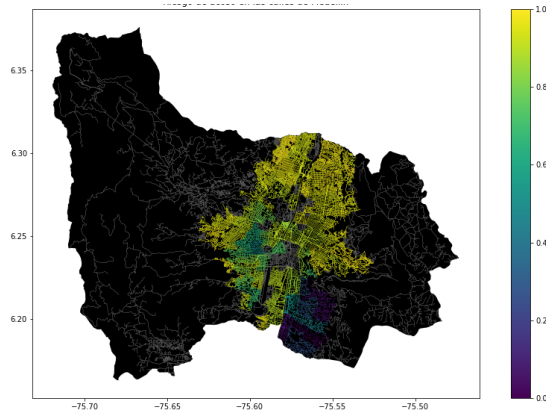


Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenida de la Encuesta de Calidad de Vida de Medellín, de 2017.

3.2 Alternativas de camino más corto con restricciones

A continuación, presentamos diferentes algoritmos utilizados para el camino más corto restringido. *(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 Encuentre el camino mas corto en un laberinto

Este algoritmo como se centre en encontrar el camino más corto para llegar a la salida de un laberinto. En este solo se permite un movimiento en 4 direcciones, se hace uso de llamados recursivos para encontrar que camino en el idóneo para resolver el problema, siempre se encuentra preguntándose si la casilla siguiente ya fue utilizada en caminos anteriores para no repetir los caminos, y de esta

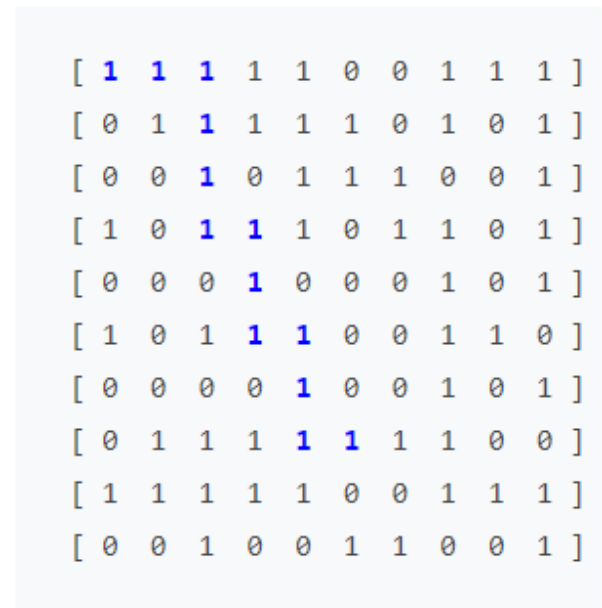
manera se evalúan todos los posibles caminos usando todas las casillas de movimiento. En este se suma uno o resta uno a las coordenadas x o y, y de esta forma se va avanzando en todas las direcciones. Y como se explicó anteriormente siempre se pregunta si este camino o casilla ya fue recorrida. [5]

Arriba (x,y) (x-1,y)

Izquierda (x,y) (x,y-1)

Abajo (x,y) (x+1,y)

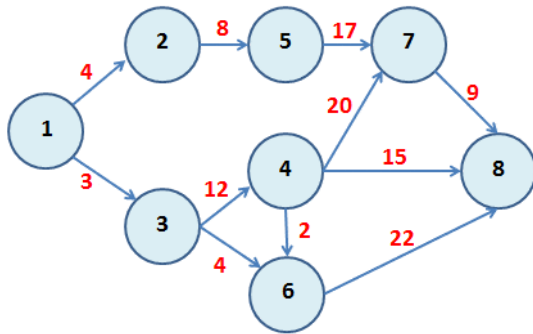
Derecha (x,y) (x,y+1)



3.2.2 Ruta más barata

Para la siguiente solución se implementan caminos que tienen un costo específico por recorrerlos, además cada nodo o casilla también contiene un valor específico, estos valores se pueden representar como el costo y puede ser en tiempo, dinero, etc. El objetivo es cruzar estos caminos hasta llegar al final con la suma de los diferentes valores, el camino que la suma de estos valores sea la menor será el camino que menos costo implicara para el sistema. [6]

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



3

Ruta 1-2-5-7-8: $4+8+17+9=38$ [km]

Ruta 1-3-4-7-8: $3+12+20+9=44$ [km]

Ruta 1-3-4-6-8: $3+12+2+22=39$ [km]

Ruta 1-3-4-8: $3+12+15=30$ [km]

Ruta 1-3-6-8: $3+4+22=29$ [km]

3.2.3 Algoritmo Floyd-Warshall

Este algoritmo combina el algoritmo de Floyd y el de Warshall, en este se usan grafos interconectados por flechas que contienen una cantidad determinada, por lo que se consulta la longitud mínima ya que estos valores pueden variar desde 1 hasta infinito, lo cual representa que no existe una flecha que conecte los grafos, es por esto que siempre se avanza en la dirección mas corta, y de igual forma se pregunta si existe un camino a seguir para llegar al final y no salirse nunca de la matriz. Este algoritmo evalúa en todos las aristas encontrando el que en total suma un peso menor de Arista, de esta forma de sabe el camino mas corto. Los llamados de cada camino se hacen de forma recursiva. Su llamado recursivo es: [7]

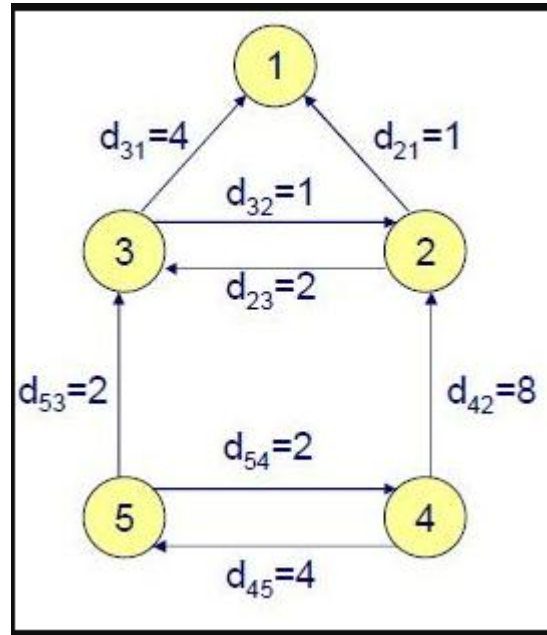
```
caminoMinimo(i, j, k) = min(caminoMinimo(i, j, k-1), caminoMinimo(i, k, k-1) + caminoMinimo(k, j, k-1));
caminoMinimo(i, j, 0) = pesoArista(i, j);
```

3.2.4 Algoritmo de Bellman-Ford

se relaja las aristas, y lo hace $|V|-1$ veces, siendo $|V|$ el número de vértices en el grafo, las repeticiones nos arrojan las distancias mínimas recorrer el árbol, ya que si no hay ciclos negativos, el camino más corto solo visita una vértices a la vez [8]

Existen dos versiones:

- Versión no optimizada para grafos con ciclos negativos, cuyo coste de tiempo es O .
- Versión optimizada para grafos con aristas de peso negativo, pero en el grafo no existen ciclos de coste negativo, cuyo coste de tiempo, es también O [8]



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.

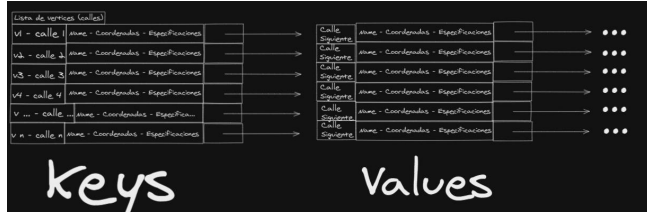
4.1 Estructuras de datos

Se utilizó una lista de adyacencia donde se representan todos los vértices mediante una lista, y nos apoyamos en un diccionario para obtener los valores a la hora de desarrollar e implementar el algoritmo, empleando datos como calle, nombre, condenadas y especificaciones. cuando el peso de decisión del algoritmo se basa en distancia y probabilidad de acoso. La estructura de los datos se presenta en la Figura 2.

Figura 2: Gráficamente se puede observar los "keys" que serían las listas de vértices de la lista de adyacencia y los "values" que serían los vértices a los que apuntan los vértices anteriormente mencionados ("keys") Donde los

⁴ <https://github.com/juanjcc10/ST0245-002>

"keys" son las coordenadas de las calles



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

Inicializamos con un vértice inicial (Calle) donde él lo que va a buscar es el siguiente vértice que menos probabilidad de acoso tenga recorriendo antes todos sus posibles caminos guardando y comparando los resultados, se decide por el vector que menos probabilidad de acoso tenga y así sucesivamente con el siguiente vértice. El algoritmo se ejemplifica en la Figura 3.

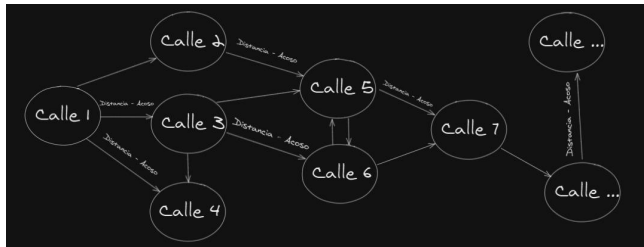


Figura 3: Revisa cada distancia de la calle actual a la siguiente teniendo como variable de decisión la probabilidad de acoso sin superar el ponderado, y así llegar a su destino final.

4.4 Análisis de la complejidad de los algoritmos

Es una suma aproximada del número de aristas y una complejidad logarítmica con base en el número de vértices. La complejidad se infiere desde el análisis del código en su peor caso en este caso se ve desde el algoritmo Dijkstra.

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

Tabla 1: Complejidad temporal del Dijkstra donde V es el número de vértices y E es el número de aristas.

Estructura de datos	Complejidad de la memoria
Dijkstra	$O(E + V)$

Tabla 2: Complejidad temporal del Dijkstra donde V es el número de vértices y E es el número de aristas.

4.5 Criterios de diseño del algoritmo

Usamos el algoritmo Dijkstra ya que es muy útil para encontrar el camino más corto teniendo en cuenta que se tienen muchos nodos por la cantidad de calles que se presentan. También, decidimos usar este algoritmo ya que la complejidad es muy conveniente con la gran cantidad de datos que manejamos en este problema y porque es muy útil por la agilidad a la hora de correr el código y por su corto tiempo de espera. El espacio en memoria también es algo positivo a la hora de usar este algoritmo ya que es la suma de los vértices y los aristas, con una gran cantidad de datos suena como si fuera a gastar mucha memoria pero en realidad es mucho mejor que por ejemplo una complejidad de $O(n^2)$.

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	6115.456	0.84
Universidad de Antioquia	Universidad Nacional	815.437	0.83
Universidad Nacional	Luis Amigó	1469.177	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	0.24860675382964015	5,000 excedió en 6,000
Universidad de Antioquia	Universidad Nacional	0.0	7,000
Universidad Nacional	Universidad Luis Amigó	0.0827358980424558	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	56.53 s
De la Universidad de Antioquia a la Universidad Nacional	56.98 s
De la Universidad Nacional a la Universidad Luis Amigó	55.875 s

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

6. CONCLUSIONES

Los caminos cortos y los de menor riesgo de abusos no son tan diferentes cuando la distancia entre los puntos no es muy larga. Empiezan a cambiar cuando la distancia se vuelve más extensa y hay más rutas alternas para ir de un lugar a otro, por ejemplo cambia significativamente en la ruta desde la Universidad EAFIT hasta la Universidad de Medellín. Los tiempos de ejecución en teoría no son de mucha espera pero plataformas como uber, waze y otras plataformas los hacen casi instantáneos. En una situación de riesgo casi que se necesitan estos tiempos instantáneos para poder actuar con agilidad. Más que todo esto tiene utilidad en el área para bajar el riesgo de peligro en la ciudad y mantener a los ciudadanos a salvo.

6.1 Trabajos futuros

Generalmente pensamos que hemos hecho un buen trabajo al desarrollar este proyecto. En el futuro queremos intentar desarrollar el proyecto en un mejor computador y con mejores datos para poder realizar pruebas con variables en el momento preciso como lo es el tráfico. Además podremos seguir trabajando para optimizar los tiempos de espera y trabajar con plataformas internacionales como waze. Es interesante poder crear un juego o un tipo de aplicación de simulación de rutas más cortas así como un tipo de Mario Kart o algo de este tipo.

AGRADECIMIENTOS

Este proyecto ha sido apoyado y posible por la Universidad EAFIT en específico la clase Estructuras de Datos y Algoritmos.

Agradecemos a los estudiantes Juan Jose Escobar y Simon Montoya por la ayuda durante el desarrollo del proyecto y por sus criticas constructivas para mejorar este proyecto.

Agradecemos a nuestras familias por la matrícula y a la Universidad EAFIT por la beca 30% por ser los que nos facilitan estar en este espacio.

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] Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data (2019) <https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/Trabajos-relacionados/Route-The-Safe.pdf>
- [2] Incorporating a Safety Index into Pathfinding by Zhaoxiang He and Xiao Qin <https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/Trabajos-relacionados/Incorporating-a-Safety-Index-into-Pathfinding.pdf>
- [3] Safety-aware routing for motorised tourists based on open data and VGI (2016) https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/Trabajos-relacionados/Safety-aware_routing_for_motorised_tourists_based_.pdf
- [4] Beyond the Shortest Route: A Survey on Quality-Aware Route Navigation for Pedestrians(2016) https://github.com/mauriciotoro/ST0245-Eafit/blob/master/proyecto/Trabajos-relacionados/Beyond_the_Shortest_Route_A_Survey_on_Quality-Awar.pdf

[5] Shortest path in a maze – Lee Algorithm

<https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/>

[6] Ejemplo del Problema del Camino Más Corto en Programación Entera

<https://www.gestiondeoperaciones.net/programacion-entera/ejemplo-del-problema-del-camino-mas-corto-en-programacion-entera/>

[7] Algoritmo de Floyd-Warshall

https://es.wikipedia.org/wiki/Algoritmo_de_Floyd-Warshall#Algoritmo

[8] Algoritmo de Bellman-Ford

https://es.wikipedia.org/wiki/Algoritmo_de_Bellman-Ford