

SafeRoute- Prevención del acoso sexual a través de un

Algoritmo de búsqueda de caminos

Samuel serpa
Universidad Eafit
Colombia
sdserpaz@eafit.edu.co

Tomas Zuleta
Universidad Eafit
Colombia
Correo electrónico en
Eafit

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

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

Sebastián garzón
Universidad Eafit
Colombia
jsgarcia@eafit.edu.co

RESUMEN

El objetivo de este proyecto es brindar una pequeña ayuda a las mujeres de la ciudad de Medellín, con el tema del acoso sexual que ha venido en aumento en la ciudad, este acoso presentándose en modo de palabras, gestos, he incluso en roses o ataques físicos, por esta razón buscamos brindar nuestra ayuda con una app la cual trace rutas mas seguras para ellas y así poder evitar violaciones y feminicidios.

El algoritmo propuesto por nosotros es el Dijkstra, con el programa realizado podemos obtener un resultado muy satisfactorio para los usuarios, generamos rutas con eficacia, en ejecución el programa se demora entre 3 a 3:30 minutos en realizar el análisis del camino y mostrarlo al usuario, los 3 resultados obtenidos, eficacia, seguridad y promedio cumplieron satisfactoriamente con la meta que teniamos propuesta desde un principio.

Palabras clave

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

En algunas zonas hay alta cantidades acoso haciendo que estas zonas sean peligrosas para las mujeres, lugares donde se pueden vulnerar sus derechos o incluso vulnerar su integridad física.

1.2 Solución

Explica, brevemente, tu solución al problema (En este semestre, la solución es un algoritmo para peatones para reducir tanto la distancia como el riesgo de acoso. ¿Qué algoritmos has elegido? ¿Por qué?)

Hemos elegido el algoritmo de dijkstra, ya que nos parece el algoritmo más apto para este tipo de proyecto, no se trabaja con valores negativos en ningún momento, si ese fuera el caso podríamos hacer uso por ejemplo del algoritmo de bellman, pero el algoritmo seleccionado cuenta con la capacidad de dar con el camino más corto y apto para cada persona.

1. INTRODUCCIÓN

El acoso sexual es un problema común vivido principalmente por mujeres, expresado en palabras, sonidos, gestos, e incluso abuso físico, el acoso sexual es un problema muy serio, por eso por medio de este proyecto buscamos encontrar una manera de reducir estos índices de acoso en las calles de Medellín y así evitar tragedias como violaciones o feminicidios

1.1. Problema

en Medellín en comunas como Manrique, Villa Hermosa, La Candelaria y el corregimiento de AltaVista, el 61,5% de las mujeres dicen sentir disgusto por salir después de las 7, en estas comunas como en muchas otras se han vivido estos casos de acoso los cuales ya son habituales en la ciudad, empezando desde un piropo mal sonante, hasta un ataque físico.

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 sachetan: Aplicación de seguridad personal basada en crowdsourcing

En países en desarrollo como Bangladesh la falta de seguridad de los civiles debido a la alta tasa de delincuencia, causa miedo en mujeres y niños al salir a la calle

Algoritmo:

1	Match all known locations to the prefixes of each word in
2	<i>IF</i> a match is found:
3	Tag the article with the matched location.
4	<i>GOTO</i> line 14.
5	<i>ELSE</i> scan the body of the article for keywords
6	<i>FOR</i> each keyword match, <i>DO</i> :
7	Match all known locations with the prefixes of each wor
8	<i>IF</i> a match is found, add matched location to the tagged
9	<i>ELSE</i> scan up to N neighboring sentences forward and b
10	<i>FOR</i> each neighboring sentence s, <i>DO</i> :
11	<i>IF</i> s has been scanned before, <i>CONTINUE</i> .
12	<i>ELSE IF</i> location match within s is found:
13	Add the matched location to the tagged area list for
14	<i>RETURN</i>

Resultados:

“es posible que los usuarios puedan evitar mejor los lugares donde es más probable que ocurran accidentes o tomar las precauciones necesarias cuando viajen por uno de estos lugares”

2.2 Herd Routes: un sistema preventivo basado en IoT para Mejorar la seguridad de los peatones en las calles de la ciudad.

Más de dos tercios de las mujeres de todas las edades en el Reino Unido han experimentado algún tipo de acoso sexual en un espacio público. Los trágicos incidentes recientes que involucran a peatones han resaltado algunos de los problemas de seguridad personal que las mujeres aún enfrentan en las ciudades hoy en día.

Algoritmo:

Algorithm 1 SUMO Simulation Main Algorithm

Data: Number N of agents; number T of simulation steps; fixed interval I for token-pricing updates.

Result: Specified number N of pedestrians are inserted to the simulation to validate the dynamic pricing, incentivisation of routes, and IOTA checkpoint elements of the Herd Routes system.

// Add agents into simulation with routes on permitted lanes.

Generate N simulated agents

Create connection to TraCI

for each lane in the road network that permits pedestrians
 append lane to pedestrianAllowedLanes list
end

for each simulated agent
 add pedestrians
 add route utilizing pedestrianAllowedLanes
end

// Determine which agents initially decide to take

// an incentivised route based on initial token pricing.

Set initial token price

for each simulated agent
 Let the agent decide
 if agents decides ‘yes’
 append agent ID to list of agents initially taking an
 incentivised route
 select incentivised route
 navigate agent onto selected
 incentivised route
 end

end

// Run the main part of the simulation.

while $i < T$:

if i is divisible by I :

 update the token price using **Algorithm 2**

for agents who decided to use an incentivised route:

 set the price using **Algorithm 3**

end

end

$i += 1$

Close connection to SUMO

(algoritmo principal de 4 en total)

Los resultados indican que este sistema tiene el potencial de tener un impacto real no solo al mejorar la seguridad de los peatones femeninos a corto plazo al generar calles más concurridas para caminar, sino también al tener un impacto a largo plazo al cambiar la forma en que todos los ciudadanos ven el cambio de comportamiento social. necesarias para minimizar la violencia de género en los espacios públicos.

2.3 Route-The Safe: un modelo robusto para la predicción de rutas más seguras utilizando datos de delitos y accidentes.

La seguridad y la protección se convirtieron en la máxima prioridad de las personas debido al aumento del número de delitos en las ciudades. Incluso el uso de mapas de Google mientras viaja, puede conducir a situaciones peligrosas y potencialmente mortales. Muchas mujeres toman rutas diferentes a las recomendadas por Google Maps y otras aplicaciones similares debido a problemas de seguridad.

La necesidad de una solución que sugiera una ruta segura se está volviendo mucho más importante que nunca. Con la ayuda de esta solución, las personas se sentirán más seguras que antes mientras viajan.

Algoritmo:

Algorithm 1: Data preprocessing Mask Algorithm

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

Model Design Mask Algorithm

Algorithm 2: K-Mean

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

Algorithm 3: K Nearest Neighbor

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

El resultado de la agrupación de significados K realizada en el distrito de Manhattan en función de la latitud y la longitud se muestra en la Figura 4. Nuevamente, la agrupación de significados K se realiza en las agrupaciones ya formadas en función de la latitud y la longitud, cuyo resultado se muestra en la Figura 5.

Los regresores KNN que predijeron la puntuación de accidentes y delitos se analizan utilizando el

puntuación R^2 . R-Squared, también llamado coeficiente de determinación, calcula estadísticamente qué tan cerca están los datos de la línea de regresión ajustada. Entonces, si el valor de R-Squared es alto, el modelo se ajustará mejor a sus datos

2.4 Prevención del acoso sexual a través de un algoritmo de búsqueda de rutas usando nearby search:

Construir un algoritmo de búsqueda de rutas en combinación con mapas de calor para

identificar 'puntos seguros' en relación con las coordenadas y direcciones de un usuario

Algoritmo:

El algoritmo de línea de Bresenham es un algoritmo de dibujo de línea que determina los puntos de un ráster de n dimensiones que deben seleccionarse para formar una aproximación cercana a una línea recta entre dos puntos. Los gráficos por computadora funcionan en unidades de píxeles, por lo que podemos ver cómo este algoritmo de búsqueda de ruta se originó a partir de la necesidad de sombrear las operaciones gráficas de píxeles.

La idea general detrás de este algoritmo es: dado un punto final inicial de un segmento de línea, el siguiente punto de la cuadrícula que atraviesa para llegar al otro punto final es Construir un algoritmo de búsqueda de ruta para prevenir el acoso sexual...

determinado evaluando dónde se cruza el segmento de línea en relación con el punto medio (arriba o abajo) de las dos posibles opciones de puntos de cuadrícula

```
from skimage.draw import line
import matplotlib.pyplot as plt
# create array of zeros
example_arr = np.zeros((10, 10))
# set endpoints for line segment
x0, y0 = 1, 1
x1, y1 = 7, 9
# draw the line
rr, cc = line(y0, x0, y1, x1)
example_arr[rr, cc] = 1
# plot it out
plt.figure(figsize=(8, 8))
plt.imshow(example_arr, interpolation='nearest')
plt.plot([x0, x1], [y0, y1], linewidth=2, c
```

Los resultados de este estudio de caso dependen del trabajo previo sobre mapas de calor, que

predecir lugares con alto riesgo de incidentes de acoso sexual.

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)¹ y se descargó utilizando la API² OSMnx de Python. El mapa incluye (1) la longitud de cada segmento, en metros; (2) la indicación de si el segmento es de un solo sentido o no, y (3) las representaciones binarias conocidas de las geometrías obtenidas de los metadatos proporcionados por OSM.

Para este proyecto, se calculó una combinación lineal (CL) que captura la máxima varianza entre (i) la fracción de hogares que se sienten inseguros y (ii) la fracción de hogares con ingresos inferiores a un salario mínimo. Estos datos se obtuvieron de la encuesta de calidad de vida de Medellín, de 2017. La CL se normalizó, utilizando el máximo y el mínimo, para obtener valores entre 0 y 1. La CL se obtuvo mediante el análisis de componentes principales. El riesgo de acoso se define como uno menos la CL normalizada. La Figura 1 presenta el riesgo de acoso calculado. El mapa está disponible en GitHub³.

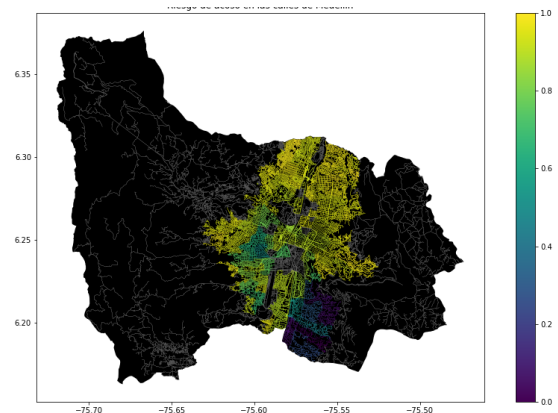


Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenidas de la Encuesta de Calidad de Vida de Medellín, de 2017.

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

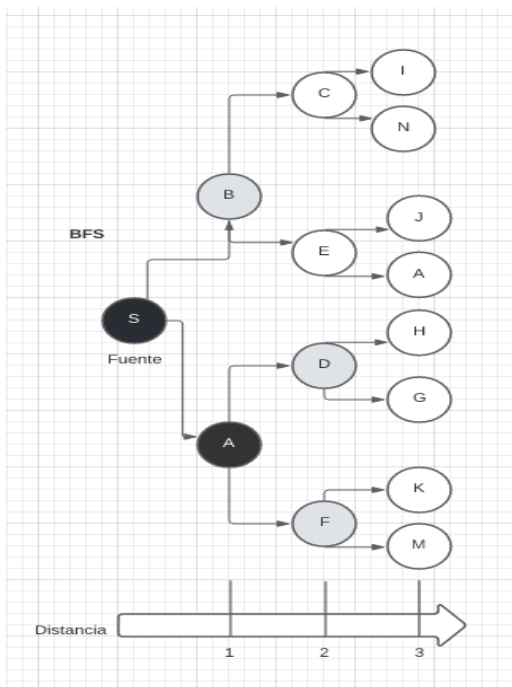
³ <https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets/>

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.

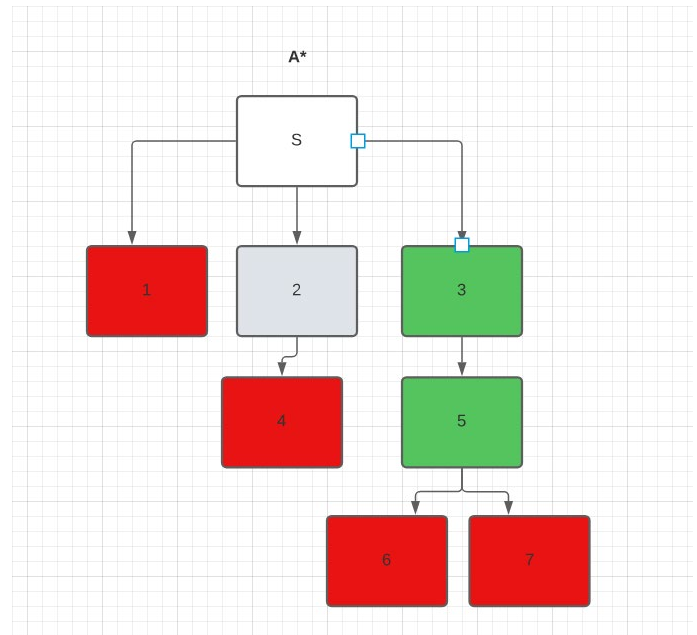
3.2.1 BFS

El algoritmo BFS (Breadth first search - Búsqueda en anchura), se utiliza en sistemas de navegación tipo GPS, cambiando su funcionalidad de encontrar lugares cercanos de interés a lugares seguros, mayormente concurridos con apoyo en caso de peligro, con todo esto se puede planear una ruta más cómoda y segura.



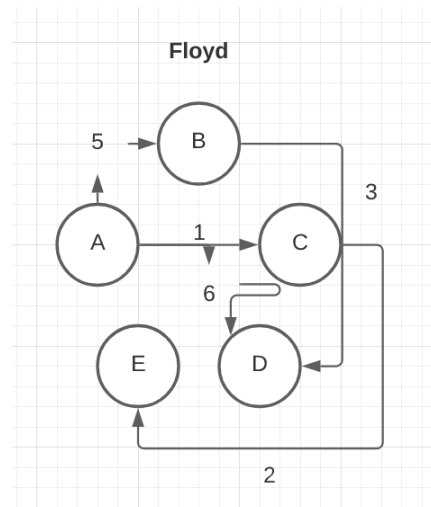
3.2.2 A*

El algoritmo A* es una combinación entre la búsqueda en anchura y profundidad, evalúa el valor heurístico del nodo actual hasta el final, para definir el costo real del camino que recorre hasta el nodo deseado, este mantiene dos estructuras de datos auxiliares que son abiertos, implementando así una cola de prioridad que este ordenando por el valor de cada nodo, los cerrados se encargan de guardar información de los nodos que ya se visitaron.



3.2.3 Floyd

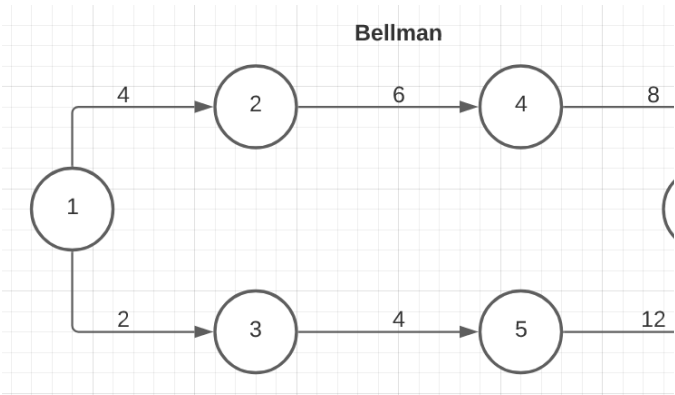
El algoritmo de Floyd compara todos los posibles caminos entre cada par de nodos hasta encontrar la mejor, la iteración de este se da sobre los nodos intermedios, hasta no llegar a la última matriz no se encuentran las distancias mínimas, basa su funcionamiento en comparaciones.



3.2.4 Bellman

El algoritmo de Bellman-Ford da una solución al problema de la ruta más corta, desde un nodo origen de modo más general que el algoritmo de dijkstra, porque este permite valores negativos en el arco, también se le puede encontrar solución a la ruta más larga la cual puede ser transformada

en el de la ruta más corta cambiando los procesos de iniciación y relajación.



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

la estructura de datos usada es un grafo, compuesto de un diccionario de diccionarios, que guarda orígenes, destinos y el riesgo.

La estructura de los datos se presenta en la Figura 2.

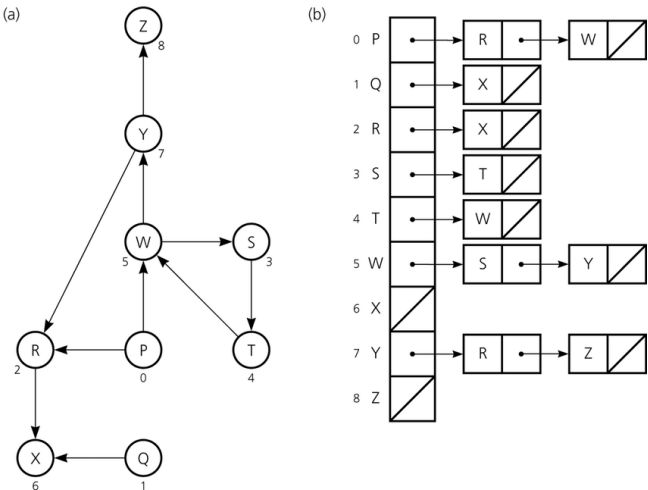


Figura 2: Un ejemplo de mapa de calles se presenta en (a) y su representación como lista de adyacencia en (b).

4.2 Algoritmos

En este trabajo, proponemos un algoritmo para un camino que minimiza tanto la distancia como el riesgo de acoso sexual callejero.

4.2.1 Algoritmo para un camino que reduce tanto la distancia como el riesgo de acoso sexual callejero

El algoritmo se ejemplifica en la Figura 3.

El algoritmo que utilizamos en nuestro trabajo fue dijkstra un algoritmo que permite encontrar el camino menos costoso entre unos nodos, en este caso vamos a ver cuál es el camino más efectivo y con menos riesgo de acoso entre 2 puntos(nodos), teniendo esto en cuenta, procedo con la explicación:

Explicación general del proceso que se ejecuta

Se abre el CSV (**Valores separados por coma**), se reemplazan los valores que no están por el promedio del riesgo, aplicamos un grafo, el cual se hizo con un diccionario dupla, se implementa el dijkstra y finalmente se grafica la ruta con gmpot.

Explicación de la aplicación

Definimos un punto de partida el cual tiene un valor de 0, no cuesta nada para nuestro camino, luego se definen todos los posibles caminos y se observa cuanto tiempo nos tomaría recorrer cada una, el costo que cada uno conllevaría y el riesgo que este tendría , una vez realizada la prueba con cada nodo y llegar hasta el final del recorrido, se define cual es la ruta más rápida y segura para la persona

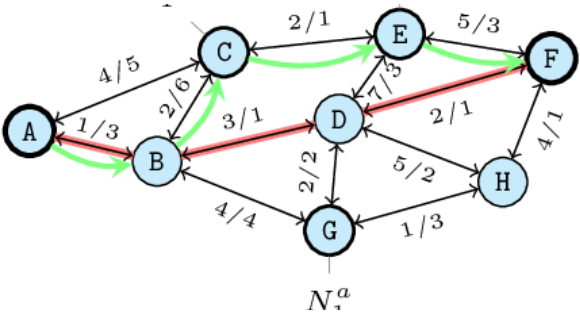


Figura 3: Cálculo de un camino que reduce tanto la distancia como el riesgo de acoso.

⁴ [http://www.github.com/ ???????? /.../proyecto/](http://www.github.com/?????????/.../proyecto/)

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

Los otros 2 caminos, tienen razones distintas por lo que se ejecutan por separado, una de ellas tiene como finalidad, la rapidez, eficiencia en cuanto a camino, ya que nos manda la ruta más corta, mientras que la otra nos envía la ruta más segura, para esto hicimos uso de grafos, los cuales representan nodos (vértices) y arcos (aristas).

El algoritmo se ejemplifica en la Figura 4.

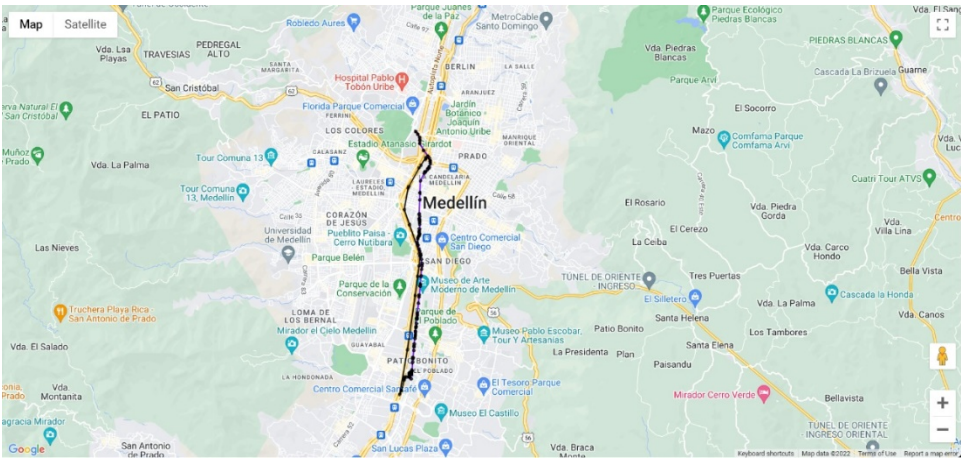


Figura 4: Mapa de la ciudad de Medellín donde se presentan tres caminos para peatones que reducen tanto el riesgo de acoso sexual como la distancia en metros entre la Universidad EAFIT y la Universidad Nacional.

4.3 Análisis de la complejidad del algoritmo

Calcular la complejidad lógica de nuestro programa fue hecho mediante el algoritmo anteriormente mencionado dijkstra.

Algoritmo	Complejidad temporal
Dijkstra	$O(V ^2 + E) = O(V ^2)$

Tabla 1: Complejidad temporal de dijkstra, en este la E representa las aristas y la V los vertices.

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

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

4.4 Criterios de diseño del algoritmo

Decidimos en este proyecto hacer uso de dijkstra, ya que, para nosotros, este era el algoritmo que más se acomodaba para nuestro programa, gracias a su eficiencia para recorrer y hacer el análisis de un nodo inicial, hasta un nodo final, todo esto con la finalidad de brindar un camino de calidad para los usuarios que van a utilizar a futuro nuestro programa.

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	8.876m	0.80
Eafit	Unal	11.562m	0.70
Eafit	Unal	13.234m	0.56

Tabla 3. Distancia en metros y riesgo de acoso sexual callejero (entre 0 y 1) para ir desde la Universidad EAFIT hasta la Universidad Nacional caminando.

5.2 Tiempos de ejecución del algoritmo

En la Tabla 4, explicamos la relación de los tiempos medios de ejecución de las consultas presentadas en la Tabla 3.

Cálculo de v	Tiempos medios de ejecución (s)
$v = d$	79.15s
$v = r$	80.00s
$v = d + 100r$	79.00s

Tabla 4: Tiempos de ejecución del nombre del algoritmo dijkstra

6. CONCLUSIONES

Al analizar los resultados que obtuvimos gracias al algoritmo desarrollado, podemos observar que los caminos generados son diferentes, nuestro programa cumpliría con una función parecida a la que tiene por ejemplo waze, el cual te plantea distintos caminos con tiempos estimados, nosotros por

nuestra parte, brindaríamos caminos con mayor seguridad y rapidez, el tiempo de ejecución podría ser mejorado considerablemente más adelante cuando tengamos más conocimiento y capacidad de aplicar sistemas más eficaces, recomendaríamos abiertamente el camino seguro para aplicación móvil o web por la situación actual de Colombia

6.1 Trabajos futuros

En un futuro cercano, nos gustaría mejorar la velocidad y rendimiento de nuestro programa, hasta el punto de llegar a compararlo con algunas aplicaciones famosas para viajes, tales como google maps o waze, sin embargo, con el plus que en nuestro algoritmo buscamos brindar a nuestros usuarios y es la posibilidad de escoger una ruta ligera o una ruta de mayor seguridad, evitando así el acoso que se vive diariamente en Medellín, también implementando algo como un lente de contacto con la suficiente tecnología como para dar indicaciones en tiempo real y con la opción de pedir apoyo en caso de algún accidente.

AGRADECIMIENTOS

Este proyecto fue posible gracias a nuestros padres, a las becas generación E, nuestras capacidades y habilidades para la resolución de problemas planteados, entre muchas cosas que nos ayudaron a lograr completar este proyecto de manera exitosa, queremos agradecer por la ayuda brindada por los monitores, compañeros y profesor del curso, cada parte de este equipo fue fundamental para la realización correcta de este proyecto.

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. Mamani, M., 2022. Difference between Breadth Search (BFS) and Deep Search (DFS). [online] Encora. Available at: <<https://www.encora.com/es/blog/dfs-vs-bfs>> [Accessed 25 May 2020].
2. Goodrich and Tamassia, 2022. Breath-first search. [online] Ics.uci.edu. Available at: <<https://www.ics.uci.edu/~goodrich/teach/cs260P/notes/BFS.pdf>> [Accessed 17 August 2015].
3. Ecured.cu. 2022. Algoritmo de Búsqueda Heurística A* - EcuRed. [online] Available at: <https://www.ecured.cu/Algoritmo_de_B%C3%BAsqveda_Heur%C3%ADstica_A%2A> [Accessed 17 August 2022].
4. Meiassests.blob.core.windows.net. 2019. amsp - Floyd's Algorithm. [online] Available at: <<https://meiassests.blob.core.windows.net/amsponloads/uploads/files/93b078d4e8cb0c5bedef258c7c8194ac.pdf>> [Accessed 25 February 2019].
5. Klappenecker, A., 2022. The Bellman-Ford Algorithm. [online] People.engr.tamu.edu. Available at:

<<https://people.engr.tamu.edu/andreas-klappenecker/csce411-f17/csce411-graphs6.pdf>> [Accessed 17 August 2022].