ALGORITMO QUE TRAZA RUTAS SEGÚN DISTANCIA Y SEGURIDAD EN MEDELLÍN

Juan Felipe Vargas
Universidad Eafit
Colombia
jfvargasq@eafit.edu.co

Tomás Jaramillo
Universidad Eafit
Colombia
tjaramillm@eafit.edu.co

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

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

Mateo Alexander Zapata
Universidad Eafit
Colombia
mazapatag1@eafit.edu.co

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
Los problemas de acoso sexual callejero y los problemas de movilidad son evidentes en las ciudades lo que atraido descontente y preocupacion a la población. En la búsqueda de proponer opciones para afrontar esto se espera mejorar la calidad de vida de los habitantes en las ciudades y evitar problemas derivados como: robo, violacion, secuestro y homicidio.

Al analizar el problema se optó por el algoritmo de Dijkstra, por su análisis de datos y la forma en como los operaba era el indicado para el proyecto. A la hora de la implementación se vio como este se demoraba entre 4-5 minutos para correr todo el código, aunque fue un gran avance teniendo en cuenta que antes se podía tardar 20 minutos, así que se concluye que nuestro razonamiento para crear el algoritmo era válido, falto sintaxis y manejo de variables en el programa para que fuera mucho mas optimo, pero a pesar del tiempo empleado, el programa genera diferentes caminos tomando en cuenta ciertas operaciones y cumple con su trabajo asi que es valido

Palabras clave

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

1. INTRODUCCIÓN

Dentro de las ciudades se ha vuelto común el problema de la circulación, viéndose agravado con el tamaño poblacional de esta. Al mismo tiempo, se evidencia un aumento de los casos de acoso sexual en la calles, en gran medida por la normalizacion de acciones que violentan con la integridad fisica y mental de las personas. Lograr un cambio cultural que se refleje en la seguridad de las personas al transitar la ciudad no está próximo. Es así que tener en cuenta el camino que se transita se ha vuelto conveniente, tanto en materia de tiempo como en seguridad.

1.1. Problema

Los niveles de acoso en las calles van en aumento, junto con los problemas de movilidad en las ciudades. Esto ha generado incertidumbre, insatisfacción y aumento en los niveles de estrés de la población. Trabajando en estos problemas se busca traer seguridad y complacencia a los habitantes de las ciudades.

1.2 Solución

La solución que le hemos encontrado al problema es crear un programa que, usando un repositorio que contiene alrededor de 72 mil calles de Medellín, con sus respectivas distancias e índice de riesgo, calcule el camino más óptimo de un punto de la ciudad a otro. Para combinar la distancia y el riesgo, decidimos multiplicar ambos números, ya que entre menor sea cada uno, más seguro es el camino, y se podría decir que esta fórmula permite ver el riesgo total de cada camino. Ya cuando cada vía tiene su riesgo total independiente, la menor sumatoria total del recorrido nos da a conocer la vía más óptima. Ya para encontrar esta sumatoria mínima usamos el algoritmo de Dijkstra, el cual resulta útil asignar a cada nodo individual un único valor, lo cual ahorra mucho tiempo al no recorrer todos los casos posibles, que muchas veces pueden ser redundantes. Este algoritmo es de gran ayuda porque graba el nodo previo, lo cual permite graficar el recorrido más fácilmente.

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 Algoritmos para encontrar la ruta de mejor conciencia de calidad para peatones

Esta investigación habla acerca de cómo se usa el algoritmo de Dijkstra y A-estrella con la información geográfica extraída principalmente de crowdsourcing y redes sociales, para encontrar las rutas de mejor calidad para peatones y turistas. Estas rutas de calidad son una combinación de placer, seguridad y salud [1].

2.2 Algoritmos para rutas de transporte para niños usando buses escolares

En el artículo se habla acerca de la integración de un "Safe Map", el cual tiene parámetros de seguridad integrados y puede encontrar la ruta más segura y óptima para los estudiantes. Se usa una variación del algoritmo de dijkstra [2].

2.3 Aplicación móvil de algoritmos de rutas óptimas y su efecto en el desplazamiento de los conductores de vehículos en la ciudad de Trujillo

Este proyecto consistió en la generación de distintas rutas en la ciudad de Trujillo mediante varios algoritmos y su comparación. Se usan los algoritmos de Rutas Cortas, DFS, BFS, Dijkstra, Floyd-Warshall, Bellman-Ford y A-Estrella [3].

2.4 Ruta más corta: soluciones algorítmicas para movilidad eficiente en la malla vial de Cundinamarca. Programación dinámica

La tesis presenta el uso de programas de programación dinámica, con los algoritmos Dijkstra, algoritmos A y colonia de hormigas, con el fin de encontrar la movilidad más corta y eficiente, usando la malla vial de Cundinamarca. [4]

3. MATERIALES Y MÉTODOS

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos de caminos que reducen tanto la distancia como el riesgo de acoso sexual callejero.

3.1 Recogida y tratamiento de datos

El mapa de Medellín se obtuvo de *Open Street Maps* (OSM)[1] y se descargó utilizando la API[2] 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 GitHub[3].

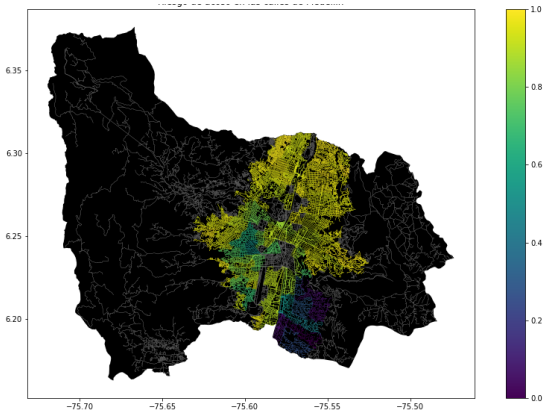


Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenidas de la Encuesta de Calidad de Vida de Medellín, de 2017.

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

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

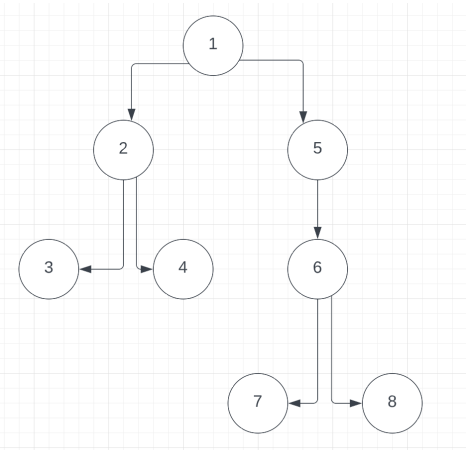
[3]https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets/

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.

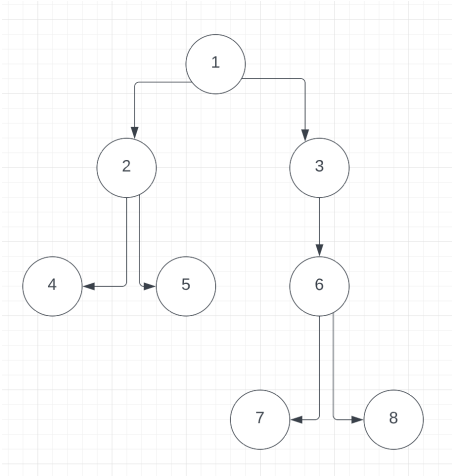
3.2.1 Depth first Search

Este algoritmo, usado para recorrer grafos, empieza por el nodo superior o el cual marca el inicio. Traza un camino por los nodos hasta llegar a uno cuyos adyacentes han sido visitados. Luego se devuelve por los vértices que se atravesó anteriormente para trazar desde un nodo dado una nueva ruta. Por la misma línea que se baja se buscan aristas sin recorrer para un nuevo trazado. Su complejidad viene dada por O(V+E), es decir, la suma de la cantidad de vértices V y de aristas E. [5]



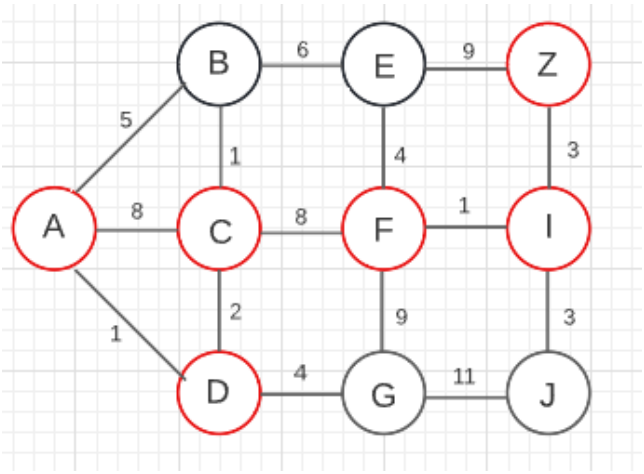
3.2.2 Breadth first search

Este algoritmo, cuyo uso radica en recorrer grafos, funciona recorriendo desde un nodo todos los vértices adyacentes a este. De este modo, los nodos ya visitados se rotulan como marcados, evitando la repetición y bucles infinitos. El algoritmo termina su ejecución cuando todos los nodos han sido marcados como visitados. Su complejidad está dada por O(V+E), es decir, la suma de la cantidad de vértices V y de aristas E. [6]



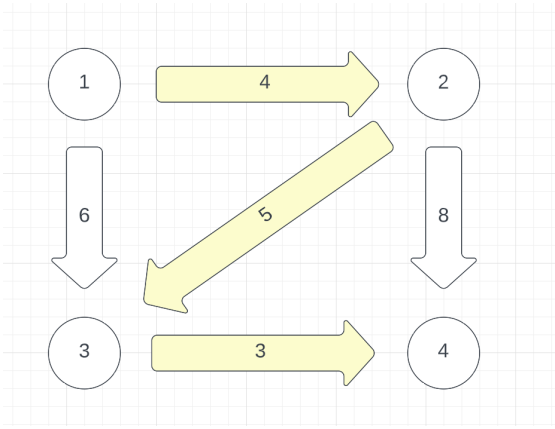
3.2.3 A Star

Este algoritmo busca el camino con menor costo de un nodo de inicio hasta un destino. Este se dice que es informado porque requiere de conocer cuál es el nodo final. Este se basa en la función f(n)=g(n)+h(n), donde g(n) mide el coste de llegar desde el nodo de inicio hasta n y h(n) desde el nodo hasta el final, buscando qué nodo minimiza f(n). Esto se evalúa respecto a los vértices adyacentes del nodo en que se encuentran. La complejidad puede ser tanto exponencial como lineal. [7]



3.2.4 Algoritmo de Dijkstra

Este algoritmo tiene como función recorrer completamente un grafo de la forma más corta posible. Este algoritmo trabaja por etapas, sin tener a consideración consecuencias en el futuro. En su forma original, tenemos que la complejidad está dada por O((V+E)log V), donde la suma de la cantidad de vértices V y de aristas E. [8]



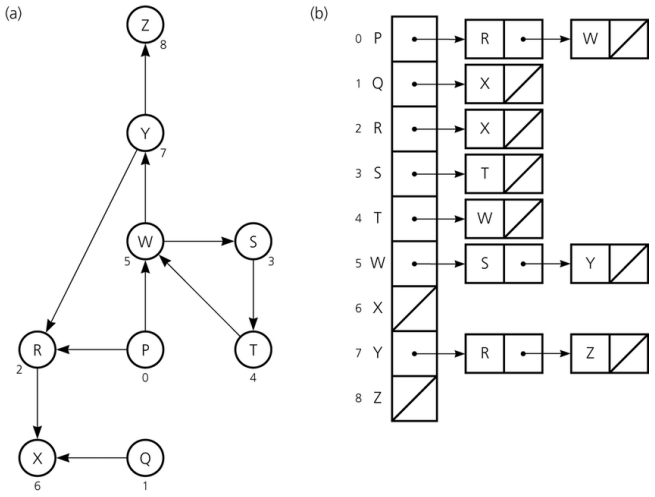
4. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

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 Github[4].

4.1 Estructuras de datos

La estructura de datos usada en el código fue una lista de adyacencia, utilizando diccionarios. La estructura de esta correspondía a un diccionario donde las claves eran las coordenadas, y el valor asociado a cada clave es otro diccionario, que contiene todas las coordenadas a las que se puede acceder directamente con el nodo de la clave (ambos nodos conforman una sola calle). El valor de las aristas asociada a la longitud por el acoso, se asigna mediante el llamado con 2 claves en el diccionario, donde la primera representa la coordenada inicial de la calle y la segunda representa la coordenada final de la calle

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



[4]https://github.com/tjaramillm/EDA-GPS

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

Para que el algoritmo de dijkstra funcione de la forma apropiada, se requiere que a la función se les ingrese 3 parámetros iniciales: nodo inicial, nodo final y el grafo con todos los nodos.

A la hora de haber creado el grafo con todos los nodos, se implementó un diccionario de diccionarios, lo cual permite que las claves del diccionario inicial sean todas las coordenadas del archivo, las claves del segundo diccionario corresponden a los nodos a los que se puede acceder desde el nodo que aparece en la primera clave y el valor asociado será la distancia entre ambas coordenadas.

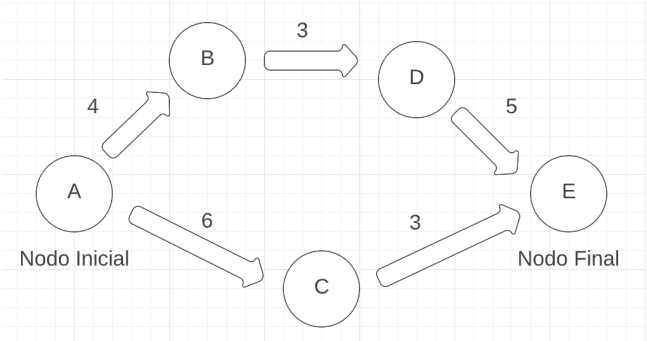
Al iniciar el algoritmo, todas las distancias del nodo inicial a los demás nodos se computa como infinito, menos la distancia del nodo inicial a sí mismo, la cual se computa como 0, lo cual permite inicial el algoritmo desde el nodo inicial.

De ahí se repite en ciclos estos 2 pasos hasta marcar todos los nodos como visitados:

- El nodo con menor distancia acumulada y sin visitar, se elige como nodo corriente y se visita todos sus nodos vecinos usando el diccionario previamente mencionado

- Cuando se visiten todos los vecinos, también se actualiza su distancia de infinito a la distancia que de el diccionario y el nodo corriente se marca como visitado

Desde la computación de los diccionarios, ya el valor de la arista se implementó como la multiplicación de la distancia del camino por el riesgo del camino, por lo que este paso no será requerido a la hora de correr el algoritmo.



Vertice/Paso	Paso 1	Paso 2	Paso 3	Paso 4	Paso 5	Paso Final
A	[0,-]	VISITADO	VISITADO	VISITADO	VISITADO	VISITADO
B	∞	[4,A]	VISITADO	VISITADO	VISITADO	VISITADO
C	∞	[6,A]	[6,A]	VISITADO	VISITADO	VISITADO
D	∞	∞	[7,B]	[7,B]	VISITADO	VISITADO
E	∞	∞	∞	[9,C]	[9,C] o [12,D]	[6,A]+[9,C]

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

Los otros dos caminos se hicieron de igual manera por medio de la multiplicación de dos términos, pero difiere en que en uno se prioriza la menor distancia elevando al cuadrado el riesgo ya que son números entre (0,1), y asi se baja su importancia, dándole más importancia al valor distancia.

El otro camino se le da mayor importancia al valor “acoso sexual”, estandarizando el valor distancia en una escala logaritmica, haciendo que sus valores sean menores y el valor acoso sea de mayor impacto en el producto.

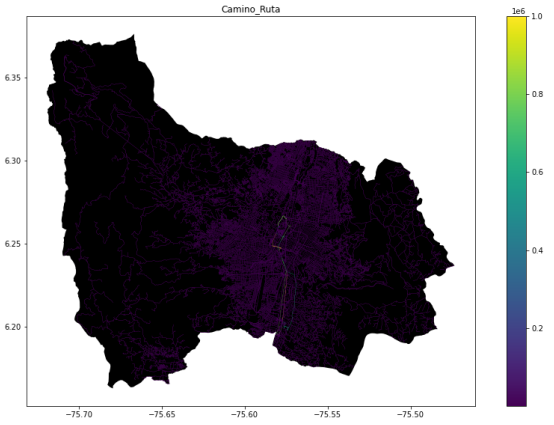


Figura 4: Mapa de la ciudad de Medellín donde se presentan tres caminos para peatones que reducen tanto el riesgo de acoso sexual como la distancia en metros entre la Universidad EAFIT y la Universidad Nacional.

4.3 Análisis de la complejidad del algoritmo

Orden de complejidad del algoritmo:

O(|V|²+|A|) = O(|V|²), sin utilizar cola de prioridad, :O((|A|+|V|) log |V|) = O(|A| log |V|) utilizando cola de prioridad (por ejemplo, un montículo binario o un árbol binario balanceado). Por otro lado, si se utiliza un montículo de Fibonacci, sería O(|V| log |V|+|A|).

La complejidad computacional del algoritmo de Dijkstra se puede calcular contando las operaciones realizadas:

El algoritmo consiste en n-1 iteraciones, como máximo. En cada iteración, se añade un vértice al conjunto distinguido.

En cada iteración, se identifica el vértice con la menor etiqueta entre los que no están en Sk. El número de estas operaciones está acotado por n-1.

Además, se realiza una suma y una comparación para actualizar la etiqueta de cada uno de los vértices que no están en Sk.

Luego, en cada iteración se realizan a lo sumo 2(n-1) operaciones.

Entonces:

Teorema: El algoritmo de Dijkstra realiza O(n²) operaciones (sumas y comparaciones) para determinar la longitud del camino más corto entre dos vértices de un grafo ponderado simple, conexo y no dirigido con n vértices.

El While se ejecuta tantas veces como el tamaño del arreglo de vértices sin repetir, es decir V. Así mismo, tenemos dentro un for que se ejecuta desde V hasta 1 vez, es decir V/2 en promedio, aunque este valor es constante. Pero la cantidad de aristas es máximo 2V. Entonces obtenemos que lo que está dentro del while es asintóticamente V y la complejidad sería V^2.

Algoritmo	Complejidad temporal
Algoritmo de Dijkstra	$O(V^2)$
No se probó	

Tabla 1: Complejidad temporal del algoritmo de Dijkstra:

Donde V son la cantidad de vértices y E son la cantidad de aristas.

Estructura de datos	Complejidad de la memoria
Lista de adyacencia	O(V+E)
No se uso	

Tabla 2: Complejidad de memoria de la lista de adyacencia donde V toma el valor de número de vértices y E toma el valor de conexiones entre vértices.

4.4 Criterios de diseño del algoritmo

En este caso se usó el algoritmo de Dijkstra por los siguientes motivos:

1. Resulta ser de los más eficientes frente al tiempo de ejecución y memoria. (Tomando en cuenta el algoritmo base)
2. Es un algoritmo que toma en cuenta nodos y sus aristas, los cuales usamos para generar nuestro recorrido en el grafo.

3. Los retornos que se podían hacer luego de ejecutar nuestro algoritmo permite hallar la distancia con una complejidad de O(1), siempre y cuando esta fuera la variable asignadas a las aristas, de forma eficiente se podían encontrar el camino para graficarlo en el mapa.
4. Porque se puede implementar con estructura de datos de baja complejidad, en nuestro caso la lista de adyacencia.

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	7474.982 m	0.79130
Eafit	Unal	10231.246m	0.7818
Eafit	Unal	9793.24m	0.79999

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*A^2	387s
v = Ln(D)*A	255s
v = A*D	367s

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

6. CONCLUSIONES

En la tabla 4, se ve cómo varían los tiempos de ejecución para cada camino, el primer dato es el camino donde se prioriza la menor distancia, el dato 2 es el camino en el que se prioriza el menor acoso y el tercero es la multiplicación de ambos términos, el camino 2 tiene un menor tiempo de recorrido ya que empieza a trabajar con valores a menor escala. Los caminos no varían mucho excepto donde se prioriza la menor distancia, ya que ahorra 2 km aproximadamente de recorrido, por lo que toma una ruta mucho más recta hacia la universidad nacional, en vez de andar por ciertos tramos donde toca dar ciertas vueltas para llegar al lugar. Los tiempos de ejecución son válidos si uno quiere tener un mapa estático y planearlo con una base de datos estática. Pero si es de caso inmediato no es muy útil ya que el tráfico es fluctuante y cambia a cada segundo, entonces esperar tanto es tener muchos datos del pasado que no me sirven para hacer un recorrido inmediato. Para comparar nuestro diseño con uno que ya está al mercado como es Waze, usaría el modelo donde se prioriza la menor distancia, ya que uno usando estos GPS lo que busca es llegar al punto lo más rápido posible, aunque eso implique entrar en lugares con mayor riesgo de acoso.

6.1 Trabajos futuros

El proyecto tiene todo un universo por mejorar y tiene un gran potencial para futuros trabajos de base de datos y programas relacionados. Se puede ver como mejorar su tiempo de ejecución o cómo hacer para que lo haga en menor tiempo, o poder generar una mejor interfaz más personalizada para el usuario por medio de desarrollo web, o por medio de una IA poder generar el mejor camino en tiempo real. Si a esto le sumamos que el proyecto tiene una gran cantidad de datos, para relacionar estos datos con gráficas y trabajos estadísticos para ver que ocurre en los lugares menos recorridos y poder generar estrategias para que haya mayor circulación por estas zonas.

AGRADECIMIENTOS

Esta investigación ha sido parcialmente apoyada por la Universidad EAFIT mediante la beca parcial 30%, y los padres de los autores por haber patrocinado el restante de la matrícula.

Los autores agradecen a los estudiantes Samuel Rico y Jacobo Zuluaga , de la Universidad EAFIT, por haber ayudado en la optimización de nuestro código, y la información brindada que mejoró en gran medida este reporte técnico.

REFERENCIAS

1. Siriaraya, P., Zhang, Y., Wang, Y. and Wakamiya, S., 2020. *Beyond the Shortest Route: A Survey on Quality-Aware Route Navigation for Pedestrians*. [online] ReasearchGate. Available at: <https://www.researchgate.net/publication/343251733

_Beyond_the_Shortest_Route_A_Survey_on_Quality-Aware_Route_Navigation_for_Pedestrians> [Accessed 20 August 2022].

2. Chalkia, E., Salanova, J., Bekiaris, E., Ayfandopoulou, G., Ferarini, C. and Mitsakis, E., 2014. *Routing algorithms for the safe transportation of pupils to school using school buses*. [online] ResearchGate. Available at: <https://www.researchgate.net/profile/Evangelos-Mitsakis/publication/270571926_Routing_algorithms_for_the_safe_transportation_of_pupils_to_school_using_school_buses/links/56ec306208aea35d5b981e3b/Routing-algorithms-for-the-safe-transportation-of-pupils-to-school-using-school-buses.pdf> [Accessed 20 August 2022].

3. Chalkia, E., Salanova, J., Bekiaris, E., Ayfandopoulou, G., Ferarini, C. and Mitsakis, E., 2014. *Routing algorithms for the safe transportation of pupils to school using school buses*. [online] ResearchGate. Available at: <https://www.researchgate.net/profile/Evangelos-Mitsakis/publication/270571926_Routing_algorithms_for_the_safe_transportation_of_pupils_to_school_using_school_buses/links/56ec306208aea35d5b981e3b/Routing-algorithms-for-the-safe-transportation-of-pupils-to-school-using-school-buses.pdf> [Accessed 21 August 2022].

4. Chalkia, E., Salanova, J., Bekiaris, E., Ayfandopoulou, G., Ferarini, C. and Mitsakis, E., 2014. *Routing algorithms for the safe transportation of pupils to school using school buses*. [online] ResearchGate. Available at: <https://www.researchgate.net/profile/Evangelos-Mitsakis/publication/270571926_Routing_algorithms_for_the_safe_transportation_of_pupils_to_school_using_school_buses/links/56ec306208aea35d5b981e3b/Routing-algorithms-for-the-safe-transportation-of-pupils-to-school-using-school-buses.pdf> [Accessed 21 August 2022].

5. Algoritmo Depth first Search - Encora. Encora.com.

https://www.encora.com/es/blog/dfs-vs-bfs#:~:text=Una%20b%C3%BAsqueda%20en%20profundidad%20(DFS,padre%20hacia%20el%20nodo%20hijo).

6. Algoritmo Breadth first search - https://jariasf.wordpress.com/2012/03/02/algoritmo-de-busqueda-depth-first-search-parte-1/

7. Algoritmo A Star - https://www.ecured.cu/Algoritmo_de_B%C3%BAsqueda_Heur%C3%ADstica_A*

8. Algoritmo de Dijkstra - EcuRed. Ecured.cu. https://www.ecured.cu/Algoritmo_de_Dijkstra.