**ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA**

|  |  |  |  |
| --- | --- | --- | --- |
| Samuel Andrés Areiza  Universidad Eafit  Colombia  saareizat@eafit.edu.co | Sara Maria Cano  Universidad Eafit  Colombia  smcanom@eafit.edu.co | Andrea Serna Universidad Eafit Colombia asernac1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

**RESUMEN**

Para escribir un resumen, debe responder a las siguientes preguntas en un solo párrafo: ¿Cuál es el problema? ¿Por qué es importante el problema? ¿Cuáles son los problemas relacionados? ¿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 obtenidos con los tres caminos*).

**Palabras clave**

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

**1. INTRODUCCIÓN**

Explique la motivación, en el mundo real, que conduce al problema. Incluya algunos antecedentes de este problema. *(En este semestre, la motivación es por qué y para qué necesitamos calcular un camino que reduzca tanto la distancia como el riesgo acoso sexual callejeros).*

**1.1. Problema**

En pocas palabras, explique el problema, el impacto que tiene este problema en la sociedad y por qué es útil resolverlo. *(En este semestre, el problema consiste en calcular tres caminos diferentes que reduzcan tanto la distancia como el riesgo de acoso sexual callejero).*

**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?)*

**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 HarassMap: El uso de datos de origen colectivo para mapear el acoso sexual en Egipto**

El presente trabajo presenta una herramienta llamada HarassMap, la cual busca la prevención del acoso sexual en Egipto, mediante la recopilación de información de riesgo acerca de las regiones de este país.

Esta aplicación toma como base principal la tecnología crowdsourcing, donde es la misma sociedad quien va reportando la información acerca de cuáles son las regiones peligrosas; posteriormente, esta información es utilizada para cartografiar los incidentes y establecer las regiones que sean más o menos seguras, para así, notificar a sus usuarios cual es el recorrido más recomendado con énfasis en la seguridad de este. [1]

**2.2 Predicción de una ruta segura para las mujeres utilizando Google Maps**

Este trabajo, realizado por el Departamento de Ciencias Computacionales e Ingeniería del Instituto Internacional de Investigación e Estudios Manav Rachna, propone un algoritmo que permite al usuario elegir entre la ruta más corta o la más segura para llegar de un punto “a” a un punto “b”. El camino más corto y seguro es calculado a partir de registros de crimen en el área.

Además, este algoritmo recurre a Google Maps cuando el usuario prefiere consultar el camino más corto, sin tomar en cuenta la seguridad. Este algoritmo también compara los caminos más seguros según los datos de crimen y los más cortos según Google Maps, con el fin de encontrar un solo camino que sea el más corto y más seguro. Finalmente, si el usuario desea encontrar el camino más seguro, sin que este sea el más corto, el programa se basa únicamente en el registro de crímenes. Por último, para calcular la ruta más corta, el programa utiliza el algoritmo de Dijkstra. [2]

**2.3 MehfoozAurat: Transformar los teléfonos inteligentes en Dispositivos de seguridad contra el acoso de mujeres.**

Este articulo presenta un análisis de la herramienta MehfoozAurat y el impacto de esta en la sociedad pakistaní. Esta aplicación surge bajo el marco de la inseguridad que sufren las mujeres trabajadoras de clase media o baja en Pakistán, las cuales no tienen oportunidad o facilidad de usar transporte privado para su movilidad.

Esta aplicación tiene distintos mecanismos de ayuda para la movilidad, pero se hará principal enfoque en la elección de rutas seguras para los usuarios. Este se inició mediante reportes judiciales para elegir que rutas eran más y menos seguras, así, en el mapa de la aplicación se colocaban de color verde o rojo, respectivamente. Posteriormente, se iba complementando la información con base en reportes que los mismos usuarios fueran dejando en la aplicación después de recorrer ciertas zonas de las ciudades. Mas especificaciones del algoritmo no se encuentran en el artículo. [3]

**2.4 SafeStreet: Empoderar a las mujeres contra el acoso callejero mediante una aplicación basada en la privacidad de la ubicación**

SafeStreet es una aplicación móvil desarrollada por el Departamento de Ciencias Computacionales e Ingeniería de la Universidad de Daca para proteger a las mujeres del acoso sexual en público. Esta aplicación permite a las mujeres encontrar el camino más seguro para llegar a su destinación en cualquier hora del día. El usuario ingresa su posición actual, su destinación final y la hora en la que desea realizar este recorrido y la aplicación le muestra el camino más seguro y el más inseguro.

Adicionalmente, la aplicación le advierte al usuario en tiempo real sobre posibles riesgos cerca. El algoritmo de búsqueda para encontrar el camino más seguro se basa en los reportes anónimos de acoso de los otros usuarios. [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]](#footnote-1) y se descargó utilizando la API[[2]](#footnote-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]](#footnote-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.

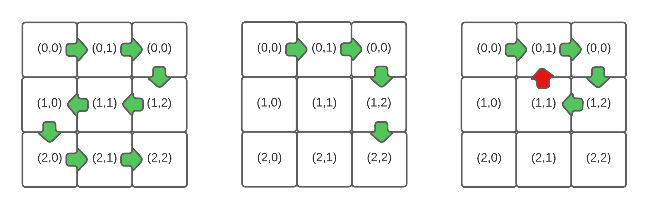
**3.2 Alternativas de caminos que reducen el riesgo de acoso sexual callejero y distancia**

A continuación, presentamos diferentes algoritmos utilizados para un camino que reduce tanto el acoso sexual callejero como la distancia. *(En este semestre, ejemplos de dichos algoritmos son DFS, BFS, Dijkstra, A\*, Bellman, Floyd, entre otros).*

**3.2.1 Algoritmo de Lee**

Este algoritmo, basado en el Breadth-First Search (BFS), busca cual es el camino más corto, dada una matriz rectangular, entre un punto de origen A y un punto de destino B. Para esto solo se permiten cuatro movimientos, subir, bajar, dirigirse a la izquierda o derecha.

Consiste en recorrer todos los caminos posibles para llegar del origen al destino y marcar cada posición de la matriz que ya se haya recorrido, así puede descartar, en el proceso recursivo, aquellos caminos que recorren un mismo trayecto más de una vez; de tal manera que, al final solo compara cual es el camino menos corto que si haya llegado al destino. Este tiene complejidad O(m\*n), donde m y n son las dimensiones de la matriz. [5]

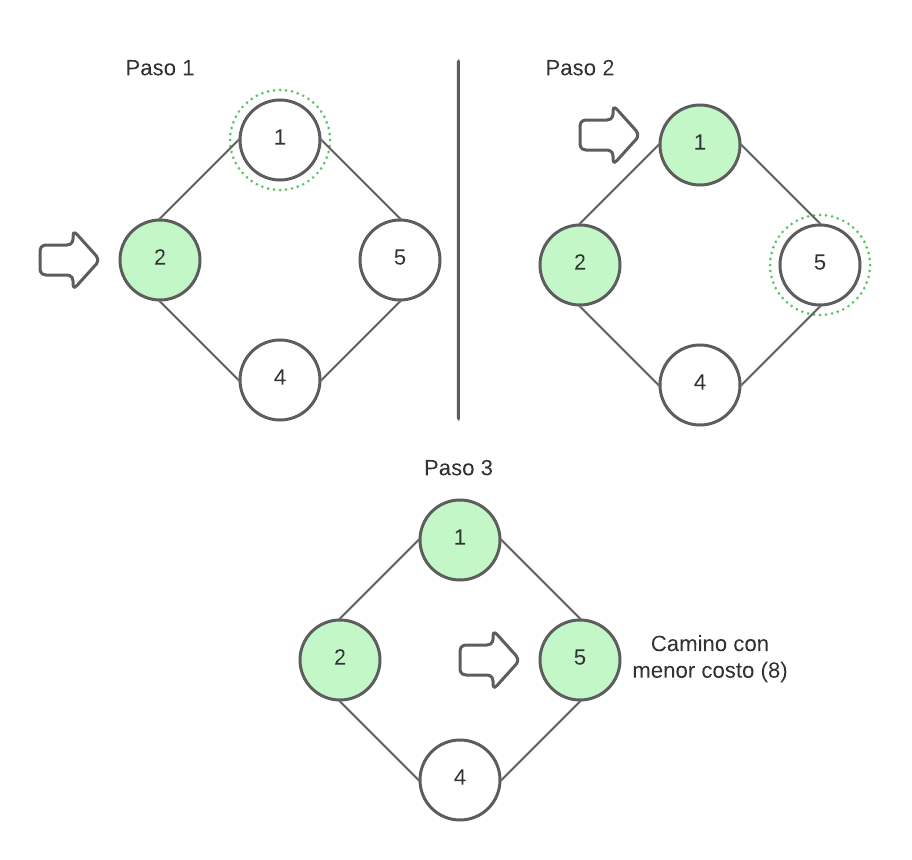


*Figura 1. Algoritmo de Lee*

**3.2.2 Algoritmo de Dijkstra**

El algoritmo de Dijkstra es utilizado para encontrar el camino más corto en un grafo dado G(V,E), donde V es el número de vértices y E es el número de aristas. Además, cada vértice tiene un costo asignado. El objetivo de este algoritmo es calcular el camino para recorrer el grafo desde un punto “a” a un punto “b” con el menor costo.

El primer paso llevado a cabo por el algoritmo es comparar los costos de cada vértice adyacente al vértice inicial y elegir el vértice con el menor costo. Luego, suma los costos del vértice inicial con el vértice seleccionado, calcula el costo de los vértices adyacentes al vértice seleccionado en el paso 1 y elige el de menor costo. El algoritmo repite el paso anterior hasta llegar al último vértice, obteniendo así el camino con el menor costo. La complejidad de este algoritmo es O(|E| + |V|log|V|). [6][7]

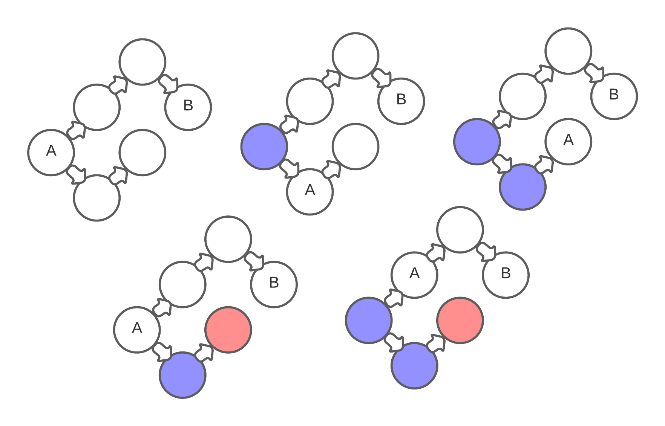


*Figura 2. Algoritmo de Dijkstra*

**3.2.3 Algoritmo de Búsqueda en Profundidad (DFS)**

Este algoritmo es utilizado para encontrar si hay una forma de ir desde un punto de origen A a un punto de destino B, generalmente en grafos, pero no necesariamente encuentra el camino más corto. Este es relativamente simple e intuitivo, se basa en primero cuestionarse si es posible ir a un nodo adyacente al que se encuentra el recorrido en ese instante y cada que pasa por un vértice marca el anterior como ya recorrido, en caso de haber más de una posibilidad elige un camino al azar y guarda en una pila el vértice del que partió.

Sigue de esta manera hasta que se encuentra en una situación donde ya no puede dirigirse a otro vértice, entonces retrocede hasta la última bifurcación que haya tomado el camino. Repitiendo este proceso hasta llegar al destino, retornando un camino posible, pero no el óptimo. En este caso, la complejidad O(V\*E) donde V es la cantidad de vértices y E la cantidad de aristas. [8]



*Figura 3. Algoritmo DFS*

**3.2.4 Algoritmo geométrico A-Star**

El algoritmo A-Star (A\*) es un algoritmo de búsqueda heurística con el fin de planear trayectorias. Este algoritmo utiliza una función *f(n)* que representa el costo final entre cada celda de una cuadrícula, *f(n)* está definida de la siguiente manera:

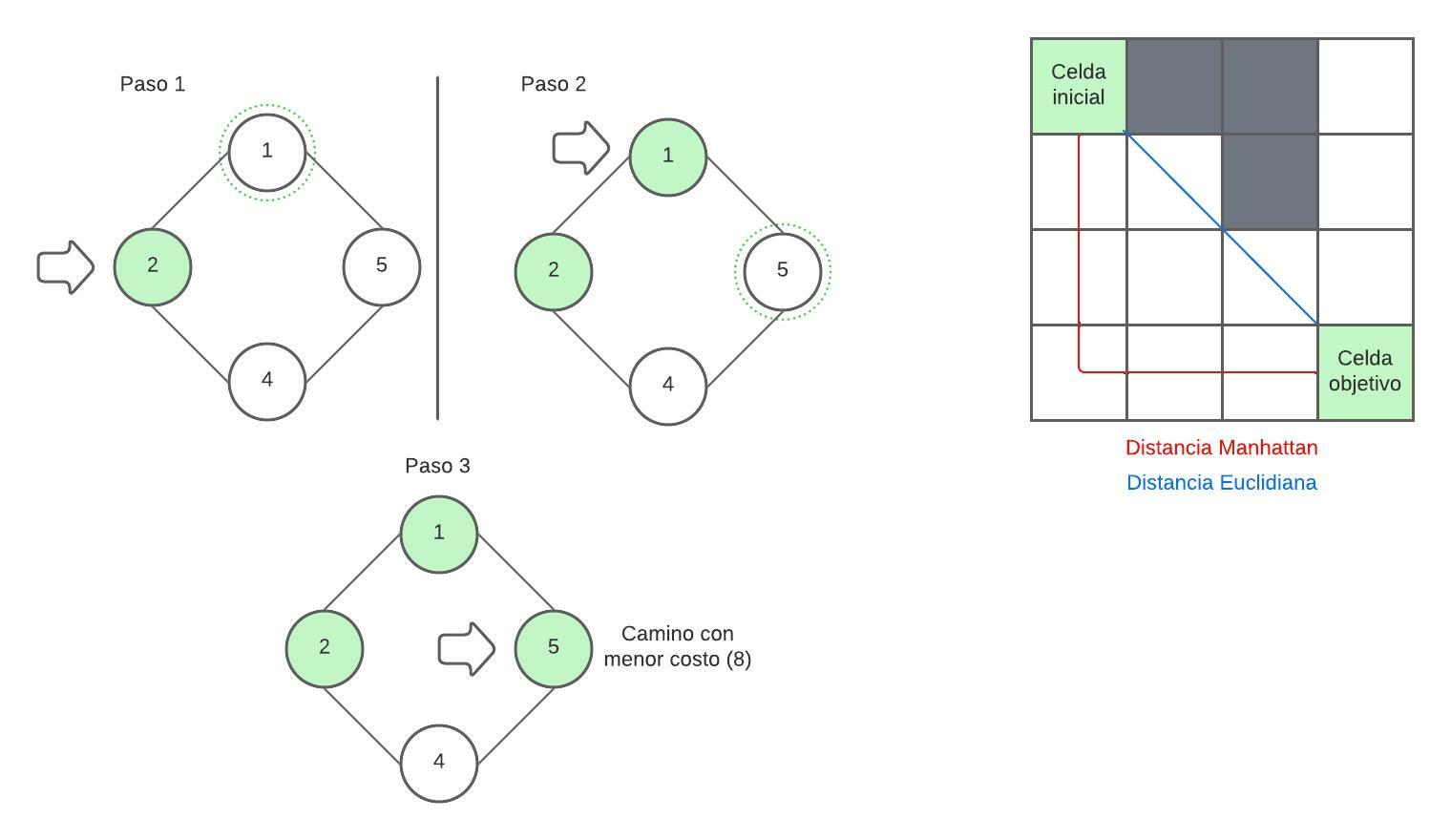
*f(n) = g(n) + h(n)*

Donde *g(n)* representa el costo desde de la celda inicial a la celda actual y *h(n)* representa el costo estimado de la celda actual a la celda objetivo. Existen dos métodos para calcular *h(n)*: Distancia Euclidiana *(hE)* o Distancia Manhattan *(hM).*

(xa ,ya): coordenadas de la celda actual

(xf ,yf): coordenadas de la celda objetivo

El algoritmo solo puede buscar en cuatro direcciones: arriba, abajo, derecha e izquierda, utilizando la Distancia Manhattan. Mientras que con la distancia Euclidiana puede buscar la distancia en todos los nodos adyacentes. La función *h(n)* permite priorizar las celdas para las cuales *f(n)* retorna el menor valor. Por lo tanto, este algoritmo no necesariamente recorre todas las celdas. Pero en el peor de los casos, es decir cuando recorra todas las celdas, el algoritmo tiene una complejidad de O(n), donde n es el número de celdas. [9]



*Figura 4. Algoritmo geométrico A-Star*

**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]](#footnote-4).

**4.1 Estructuras de datos**

Explica la estructura de datos que se utilizó para representar el mapa de la ciudad de Medellín. Haga una figura que lo explique. No utilice figuras de Internet. *(En este semestre, los ejemplos de las estructuras de datos son la matriz de adyacencia, la lista de adyacencia, la lista de adyacencia utilizando un diccionario).* 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). (Por *favor, siéntase libre de cambiar esta gráfica si utiliza una estructura de datos diferente*).

**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**

Explica el diseño del algoritmo para calcular un camino que reduce tanto la distancia como el riesgo de acoso y haga su propia gráfica. No utilice gráficas de Internet, haga las suyas. *(En este semestre, el algoritmo podría ser DFS, BFS, Dijkstra, A\*, Bellman, Floyd entre otros ).* El algoritmo se ejemplifica en la Figura 3.

**Figura 3:** Cálculo de un camino que reduce tanto la distancia como el riesgo de acoso. (Por favor, siéntase libre de cambiar esta figura si utiliza un algoritmo diferente).

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

Explica los otros dos caminos que reducen tanto la distancia como el riesgo de acoso sexual callejeroy haga su propia gráfica. No utilice gráficas de Internet, haga las suyas. *(En este semestre, el algoritmo podría ser DFS, BFS, Dijkstra, A\*, entre otros).* 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**

Explica, con tus propias palabras, el análisis, para el peor caso, utilizando la notación O. ¿Cómo ha calculado esas complejidades? Explique brevemente.

|  |  |
| --- | --- |
| **Algoritmo** | **Complejidad temporal** |
| Nombre del algoritmo | O(V2\*E 2) |
| Nombre del segundo algoritmo (en caso de que haya probado dos) | O(E 3\*V\*2V) |

**Tabla 1:** Complejidad temporal del nombre de su algoritmo, donde V es... E es... *(Por favor, explique qué significan V y E en este problema). No, no use ‘n’.*

|  |  |
| --- | --- |
| **Estructura de datos** | **Complejidad de la memoria** |
| Nombre de la estructura de datos | O(V\*E\*2E ) |
| Nombre de la segunda estructura de datos (en caso de que haya intentado dos) | O(2E\*2 V) |

**Tabla 2:** Complejidad de memoria del nombre de la estructura de datos que utiliza su algoritmo, donde V es... E es... *(Por favor, explique qué significan V y E en este problema). No, no sive ‘n’. Es decir, no usar ‘n’. No ‘n’.*

**4.4 Criterios de diseño del algoritmo**

Explique por qué el algoritmo fue diseñado de esa manera. Utilice criterios objetivos. Los criterios objetivos se basan en la eficiencia, que se mide en términos de tiempo y memoria. Ejemplos de criterios NO objetivos son: "estaba enfermo", "fue la primera estructura de datos que encontré en Internet", "lo hice el último día antes del plazo", "es más fácil", etc. Recuerde: Este es el 40% de la calificación del proyecto.

**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 | ?? | ?? |
| Eafit | Unal | ??? | ?? |
| Eafit | Unal | ?? | ?? |

**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.

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

|  |  |
| --- | --- |
| **Cálculo de v** | **Tiempos medios de ejecución (s)** |
| v = ?? | 100000.2 s |
| v = ?? | 800000.1 s |
| v = ?? | 8450000 s |

**Tabla 4:** Tiempos de ejecución del nombre del *algoritmo (Por favor, escriba el nombre del algoritmo, por ejemplo, DFS, BFS, A\*)* para cada uno de los tres caminos calculadores entre EAFIT y Universidad Nacional.

**6. CONCLUSIONES**

Explique los resultados obtenidos. ¿Son los caminos significativamente diferentes? ¿Qué utilidad tiene esto para la ciudad? ¿Son razonables los tiempos de ejecución para utilizar esta implementación en una situación real? ¿Qué camino recomendaría para una aplicación móvil o web?

**6.1 Trabajos futuros**

Responda, ¿qué le gustaría mejorar en el futuro? ¿Cómo le gustaría mejorar su algoritmo y su aplicación? ¿Continuará este proyecto trabajando en la optimización? ¿En estadística? ¿Desarrollo web? ¿Aprendizaje automático? ¿Realidad virtual? ¿Cómo?

**AGRADECIMIENTOS**

Identifique el tipo de agradecimiento que desea escribir: para una persona o para una institución. Tenga en cuenta las siguientes pautas: 1. El nombre del profesor no se menciona porque es un autor. 2. No debe mencionar a los autores de los artículos con los que no se ha puesto en contacto. 3. Debe mencionar a los alumnos, profesores de otros cursos que le han ayudado.

A modo de ejemplo: Esta investigación ha sido apoyada/parcialmente apoyada por [Nombre de la Fundación, Donante].

Agradecemos la ayuda con [técnica particular, metodología] a [Nombre Apellido, cargo, nombre de la institución] por los comentarios que mejoraron en gran medida este manuscrito.

Los autores agradecen al profesor Juan Carlos Duque, de la Universidad EAFIT, por facilitar los datos de la Encuesta de Calidad de Vida de Medellín, de 2017, procesados en un archivo *Shapefile*.

**REFERENCIAS**

1. Young, C. 2014. HarassMap: Using Crowdsourced Data to Map Sexual Harassment in Egypt. Technology Innovation Management Review, 4(3): 7-13. DOI: <http://doi.org/10.22215/timreview/770>
2. Bura, D., Singh, M. and Nandal, P. Predicting Secure and Safe Route for Women using Google Maps. En *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon),* (Faridabad, India, 2019), Institute of Electrical and Electronic Engineers, 103-108. DOI: 10.1109/COMITCon.2019.8862173
3. Muhammad Yasir Sarosh, Muhammad Abdullah Yousaf, Mair Muteeb Javed, and Suleman Shahid. 2016. MehfoozAurat: Transforming Smart Phones into Women Safety Devices Against Harassment. En *Proceedings of the Eighth International Conference on Information and Communication Technologies and Development* (ICTD '16). Association for Computing Machinery, New York, NY, USA, Article 61, 1–4. DOI: https://doi.org/10.1145/2909609.2909645
4. Mohammed Eunus Ali, Shabnam Basera Rishta, Lazima Ansari, Tanzima Hashem, and Ahamad Imtiaz Khan. 2015. SafeStreet: empowering women against street harassment using a privacy-aware location based application. En *Proceedings of the Seventh International Conference on Information and Communication Technologies and Development* (ICTD '15). Association for Computing Machinery, New York, NY, USA, Article 24, 1–4. DOI: <https://doi.org/10.1145/2737856.2737870>
5. N.D. Shortest path in a maze – Lee Algorithm. In Techie Delight Website. 2021. Given a maze in the form of the binary rectangular matrix, find the shortest path’s length in a maze from a given source to a given destination. DOI: https://www.techiedelight.com/lee-algorithm-shortest-path-in-a-maze/
6. Noto, M. and Sato, H. A method for the shortest path search by extended Dijkstra algorithm. En *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions'* (Nashville, TN, USA, 2000), Institute of Electrical and Electronic Engineers , 2316-2320 vol.3. DOI: 10.1109/ICSMC.2000.886462.
7. Barbehenn, M. A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices. En *IEEE Transactions on Computers*, 47 (2). 263-. Feb. 1998. DOI: 10.1109/12.663776.
8. Andrade, E. Núñez, F, J. and Tomás, V, T. Análisis de algoritmos de búsqueda en espacio de estados. N.D. En Universidad Autónoma del Estado de Hidalgo: Escuela Superior de Huejutla. Huejutla de Reyes, Hidalgo, México, C.P. 43000. DOI: <https://www.researchgate.net/publication/322348310_Analisis_de_algoritmos_de_busqueda_en_> espacio\_de\_estados
9. Tang, G., Tang, C., Claramunt, C., Hu, X. and Zhou, P. 2021. Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment. *IEEE Access, 9.* 59196-59210 DOI: 10.1109/ACCESS.2021.3070054

1. <https://www.openstreetmap.org/> [↑](#footnote-ref-1)
2. https://osmnx.readthedocs.io/ [↑](#footnote-ref-2)
3. [https://github.com/mauriciotoro/ST0245Eafit/tree/master/  
   proyecto/Datasets/](https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets)  [↑](#footnote-ref-3)
4. http://www.github.com/ ????????? /.../proyecto/ [↑](#footnote-ref-4)