**ALGORITMO DE TRAZAMIENTO DE LA RUTA CON MENOS ACOSO CALLEJERO EN LA CIUDAD DE MEDELLÍN**

|  |  |  |  |
| --- | --- | --- | --- |
| Alejandro Acevedo  Universidad Eafit  Colombia  jaacevedom@eafit.edu.co | David Lopera  Universidad Eafit  Colombia  dloperal2@eafit.edu.co | Andrea Serna Universidad Eafit Colombia asernac1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **RESUMEN**

## El acoso callejero es una gran problemática a nivel mundial, y la ciudad de Medellín no es la excepción, con una dificultad que es creciente día a día. Por esto, es necesario implementar, con ayuda de tecnología, soluciones que sean eficientes y seguras a la hora de tomar rutas dentro de la ciudad, teniendo como base ejemplos de algoritmos que hayan tratado de solucionar problemas similares. La finalidad de este proyecto es encontrar dentro de la ciudad mencionada tres caminos que puedan reducir distancia y problemas relacionados con el acoso.

## **Palabras clave**

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

# **1. INTRODUCCIÓN**

# El acoso callejero es uno de los principales problemas que puede presentar una persona a la hora de movilizarse a través de una ciudad, representando un riesgo constante. Este riesgo es mucho mayor si el transeúnte es mujer y puede llegar a variar dependiendo de la zona en que se encuentre. Según esto, se hace necesario poder evaluar y calcular que caminos son eficientes para el transporte y la movilización y que, a su vez, sean rutas con un mínimo riesgo de acoso. Esto toma relevancia ya que permite que las personas que sean o no de la ciudad, en este caso Medellín, puedan tener un control sobre los caminos y que tan seguros son con respecto al acoso sexual, esto con el objetivo de usar la información para fines propios, ya sea: Conocer la ciudad como turista, ir de la casa hacia el trabajo, salir a hacer ejercicio, entre otros casos.

# **1.1. Problema**

El problema que se trabajará consiste en calcular tres caminos diferentes que reduzcan tanto la distancia como el riesgo de acoso sexual callejero. El impacto que este problema genera es un impacto positivo, pues permite a las personas tener conocimiento acerca de rutas seguras y acoplarse a estas

**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 Rutas seguras para turistas motorizados basadas en datos abiertos y VGI.**

En la ciudad de Los Ángeles existen muchos distritos y barrios azotados por problemáticas como la delincuencia, conflictos políticos, problemas de drogas y prostitución. Estos barrios considerados “peligrosos” y estigmatizados por la sociedad, son lugares normalmente evitados por los locales y las personas que conocen rutas más seguras, pero este conocimiento no lo poseen turistas o personas que no vivan cerca de esas zonas por lo que las hace propensas a pasar por rutas muy riesgosas para su integridad personal. En este caso, usaron un algoritmo que utiliza tres datos puntuales: posiciones de las comisarias, cantidad de farolas o focos, y las carreteras. Mediante estos tres conjuntos de datos crearon una superficie de costes utilizando el método de interpolación IDW, el cual tiene en cuenta los puntos estimados y les da un peso a los valores medios teniendo en cuenta su distancia con los puntos. En este estudio utilizaron 100 datos de entrada dándole a la superficie de costes un rango de 100 metros. En el uso practico de este estudio se enrutó el camino que tomaría un turista sin conocimiento de la ciudad. Al comparar la ruta trazada por el programa con la ruta más rápida se puede notar una gran diferencia en los caminos trazados, también se pueden observar las desviaciones que da el programa en los focos de peligro de la ciudad, cosa que no hace la ruta más corta al destino.

**2.2 Prevención del acoso sexual mediante un algoritmo de búsqueda de trayectorias utilizando la búsqueda cercana.**

En este caso el problema era crear un algoritmo que permita la búsqueda de la ruta más segura frente al acoso sexual, esto utilizando mapas térmicos, que hacen referencia al acoso ocurrido en esos sitios los cuales permitan identificar los puntos más seguros en relación con el usuario y a sus coordenadas. Este algoritmo debe tomar decisiones por el usuario para de la manera más rápida y efectiva encontrar la ruta más segura.

En este caso de estudio, implementaron un sistema que calculaba la media de las puntuaciones de riesgo basadas en la cobertura de cada paso de la ruta para llegar al destino. Se utilizo el algoritmo de Bresenham que determina los puntos de una trama bidimensional para formar la aproximación de la línea recta entre dos puntos. Como resultado, el algoritmo calcula el riesgo de los pasos de cada ruta para obtener la mejor, siempre priorizando la seguridad de cada ruta antes que la distancia entre sí. También si dos rutas están empatadas en su nivel de seguridad se mostrará la ruta con menor distancia.

## **2.3 Ruta de la seguridad.**

## La seguridad es una de las principales preocupaciones a la hora de viajar para cualquier ciudadano, pero actualmente las aplicaciones de navegación son deficientes al momento de dar rutas seguras para las personas o ni siquiera cuentan con la opción de elegir rutas seguras si no que por defecto escogen la más corta. Se busca crear un algoritmo para la ciudad de Nueva York que de puntuaciones exactas del grado de seguridad de una ruta y determine la mejor. El algoritmo consiste en pre-calcular la puntuación de riesgo en las áreas de interés mediante los datos obtenidos del departamento de policía de Nueva York (barrios donde han ocurrido accidentes, latitud y longitud, números de heridos…) para posteriormente obtener las rutas más seguras y mediante estas obtener la más corta.

Al comparar el resultado arrojado por el algoritmo K means con una ruta sugerida por Google Maps, se puede observar que a pesar de ser más larga le brinda al usuario una seguridad mucho mayor fusionando la ruta más segura con la más rápida para llegar a su destino.

**2.4 Un sistema de integración y análisis de datos para la planificación de rutas seguras.**

En la India el acoso callejero hacia las mujeres es un problema exponencial que ha ido incrementando en gran medida la última década, este problema generalmente se da bajos ciertas condiciones que presentan las calles, pero como estas condiciones son muy difíciles de cambiar lo mejor es encontrar la manera de evitar los lugares que la presentan. Por esto es necesario desarrollar una aplicación que tenga en cuenta la distancia y la seguridad del camino con el fin de diseñar una ruta. Para crear el algoritmo se incluyeron diferentes variables que permitieron determinar la seguridad de cada zona, variables como el número de casos de acoso, estaciones de policía, hora del día, el desempleo, situación monetaria y cámaras en el sector. A partir de estos datos se determinó una función con el método Bottom up, y para finalizar se usó el algoritmo Dijkstra para la planificación de rutas. Como resultado se tiene un algoritmo que da a las mujeres la capacidad de escoger el camino más seguro o el más corto, este se diferencia de otros ya que no solo tiene en cuenta los accidentes pasados y las limitaciones que este conlleva, sino que tiene en cuenta multiples factores de riesgo.

## **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.

**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.

**3.2.1 Algoritmo de Búsqueda A**

*“El algoritmo de búsqueda A, o también llamado A estrella (A\*), fue presentado en el año 1968 por Peter E. Hart, Nils J. Nilsson y Bertram Raphael. Su función es encontrar el camino de menor costo entre un nodo origen y uno objetivo, siempre y cuando se cumplan ciertas condiciones”* [1]. Hace uso de la búsqueda heurística, que le permite cuantificar que tan bueno es un camino, teniendo como principal ventaja lograr optimización y completitud, dos valiosas propiedades de los algoritmos de búsqueda. Cuando un algoritmo de búsqueda tiene buena optimización, significa que está garantizado que encontrará la mejor solución posible. Cuando un algoritmo de búsqueda tiene la propiedad de completitud, significa que, si existe una solución para un problema dado, el algoritmo está garantizado para encontrarla[2].

A\* hace uso de la siguiente ecuación para llevar a cabo su proceso: f\*(n) = g(n) + h\*(n), donde **g(n)** es la distancia total que ha tomado para llegar desde la posición inicial a la ubicación actual, **h\*(n)** es la estimación de la distancia desde la posición actual con el destino, es una función heurística se utiliza para crear esta estimación sobre cuán lejos se está para alcanzar la meta y **f\*(n)** es la suma de g (n) y h' (n). Este es el camino actual estimado más corto. Donde f (n) es el verdadero camino más corto que no se descubrieron hasta que el algoritmo A \* ha terminado. [3]

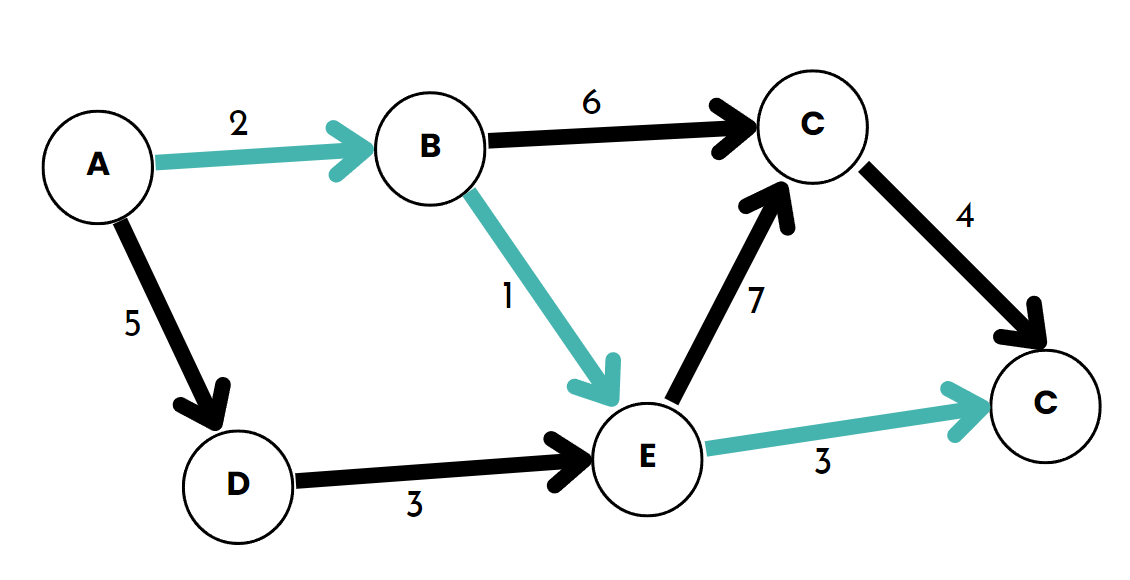
Diagrama

Descripción generada automáticamente con confianza baja

**Figura 2.** A la izquierda la proyección de la ruta hacia la celda objetivo, a la derecha el algoritmo aplicado con restricciones, pero con el mismo objetivo.

**3.2.2 Algoritmo de Dijkstra**

Este algoritmo hace uso y define etiquetas a partir del vértice de origen y también para cada uno de los siguientes vértices. Estas etiquetas contienen información relacionada con un valor acumulado del tamaño de los arcos y con la procedencia más próxima de la ruta [4]. El algoritmo irá explorando todos los caminos a partir del vertice de origen, usando la técnica greedy. La técnica greedy utiliza el principio donde para que un camino sea óptimo, todos los caminos que contiene también deben ser óptimos. De los vértices vecinos, buscamos el que esté más cerca de nuestro punto origen, lo tomamos como punto intermedio y vemos si podemos llegar más rápido a través de este vértice a los demás. Después escogemos al siguiente más cercano (con las distancias ya actualizadas) y repetimos el proceso. Esto lo hacemos hasta que el vértice no utilizado más cercano sea nuestro destino. Al proceso de actualizar las distancias tomando como punto intermedio al nuevo vértice se le conoce como **relajación** [5].

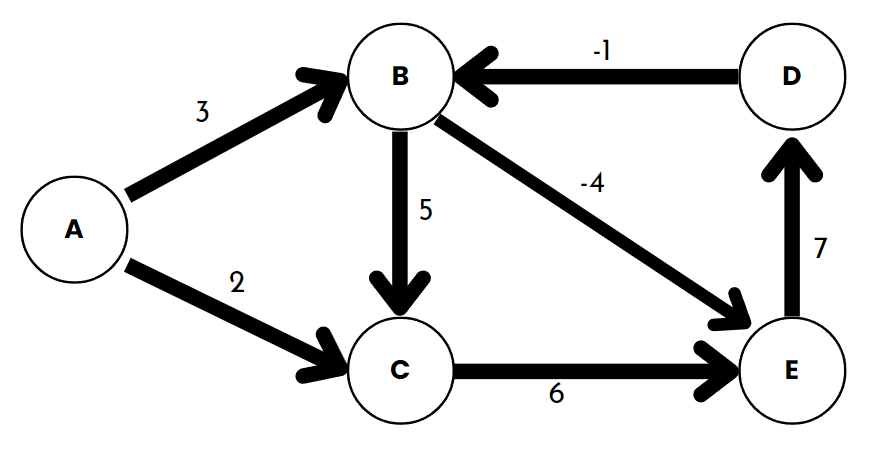


**Figura 3.** Ejemplo de cómo el algoritmo escogería un camino para ir de A hasta C teniendo en cuenta el valor de cada arco.

**3.2.3 Algoritmo de Bellman-Ford**

Este algoritmo, al igual que el resto, determina la ruta más corta desde un nodo origen hacia los demás nodos, siendo necesaria una entrada que sea un grafo, donde las aristas tengan pesos. La gran diferencia con los otros algoritmos es que Bellman-Ford permite valores negativos, siendo posible detectar la existencia de un ciclo negativo [6].

El algoritmo parte de un vértice origen que será ingresado, a diferencia de Dijkstra que utiliza una técnica voraz para seleccionar vértices de menor peso y actualizar sus distancias mediante el paso de relajación, Bellman-Ford simplemente relaja todas las aristas y lo hace |V| – 1 veces, siendo |V| el número de vértices del grafo. Para la detección de ciclos negativos realizamos el paso de relajación una vez más y si se obtuvieron mejores resultados es porque existe un ciclo negativo, para verificar porque tenemos un ciclo podemos seguir relajando las veces que queramos y seguiremos obteniendo mejores resultados [6].



**Figura 4.** Ejemplo como el algoritmo Bellman-Ford trabajaría con los vértices y arcos.

**3.2.4 Algoritmo de Floyd**

Es un algoritmo que analiza el grafo de manera que encuentra el camino más corto de forma dirigida y ponderada, o sea, de forma equilibrada. El algoritmo encuentra la ruta entre todos los pares de vértices en una única ejecución. Compara todos los posibles trayectos a través del grafo, siendo capaz de hacerlo solo con V3 comparaciones [7]. Por otra parte, este algoritmo no funciona para grafos de ciclo negativo.

Diagrama

Descripción generada automáticamente

**Figura 5.** Ejemplo como el algoritmo de Floyd trabajaría con los vértices en un grafo.

# **REFERENCIAS**

[1] Algoritmo de Búsqueda Heurística A\*. *Recuperado el 11 de agosto, 2022, de EcuRed:* [https://www.ecured.cu/Algoritmo\_de\_B%C3%BAsqueda\_Heur%C3%ADstica\_A\*](https://www.ecured.cu/Algoritmo_de_B%C3%BAsqueda_Heur%C3%ADstica_A*).

[2] Roy, B. A-Star (A\*) Search Algorithm, 2019. Recuperado el 11 de agosto, 2022, de Medium: <https://towardsdatascience.com/a-star-a-search-algorithm-eb495fb156bb>.

[3] Rihawi, I. 6.5.2 Búsqueda heurística: Algoritmo A\*, 2010. Recuperado el 11 de agosto, 2022, Poiritem, metro de Madrid: <https://poiritem.wordpress.com/2010/01/14/6-5-2-busqueda-heuristica-algoritmo-a/>.

[4] López, B. Algoritmo de Dijkstra, 2019,. Recuperado el 11 de agosto, 2022, de Ingeniería Industrial Online: <https://www.ingenieriaindustrialonline.com/investigacion-de-operaciones/algoritmo-de-dijkstra/>.

[5] Arias, J. CAMINO MAS CORTO: ALGORITMO DE DIJKSTRA, 2012. Recuperado el 11 de agosto, 2022, de *Algorithms and More:* <https://jariasf.wordpress.com/2012/03/19/camino-mas-corto-algoritmo-de-dijkstra/>.

[6] Arias, J. CAMINO MAS CORTO: ALGORITMO DE BELLMAN-FORD, 2013. Recuperado el 12 de agosto, 2022, de *Algorithms and More:* <https://jariasf.wordpress.com/2013/01/01/camino-mas-corto-algoritmo-de-bellman-ford/>.

[7] Peñaranda, M., Osca, J., López, M., Civera, C. and Tortosa, F. FUNCIONAMIENTO, REPRESENTACIÓN Y  
COMPORTAMIENTO DE DIFERENTES ALGORITMOS FRENTE AL CÁLCULO DE UN SMALL WORLD EN CIENCIA, 2009. Recuperado el 12 de agosto, 2022, de Congreso Isko-Spain: http://www.iskoiberico.org/wpcontent/uploads/2014/09/99-110\_Pe%C3%B1aranda-Ortega.pdf.

1. <https://www.openstreetmap.org/> [↑](#footnote-ref-1)
2. https://osmnx.readthedocs.io/ [↑](#footnote-ref-2)