

# RUTAS PEATONALES QUE CONSIDERAN EL ACOSO CALLEJERO Y EL TIEMPO DE RECORRIDO

Isabela Ortega  
Universidad Eafit  
Colombia  
iortegav@eafit.edu.co

Paulina Cerón  
Universidad Eafit  
Colombia  
pceronm@eafit.edu.co

## RESUMEN

El acoso callejero es una problemática que vivencia una gran parte de la población del país, provoca que las mujeres tengan miedo de salir a la calle por evitar ser acosadas, lo que impide que cumplan con sus actividades diarias y vivan con una preocupación constante sobre tener que estar alerta en las calles y buscar una ruta adecuada para llegar a su destino.

Para esto se propone la implementación del algoritmo de Dijkstra, el cual calcula aquella ruta adecuada, teniendo en cuenta la distancia y el riesgo.

## Palabras clave

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

## 1. INTRODUCCIÓN

Vivimos en una sociedad machista e insegura, en las calles las mujeres se ven expuestas constantemente a ser acosadas si no tienen algún hombre que las acompañe. Debido a esto, surge la necesidad de plantear una solución inmediata, y desde el campo de la tecnología, se puede satisfacer esta necesidad mediante un algoritmo que reduzca tanto la distancia como el riesgo de acoso sexual callejero.

### 1.1. Problema

El acoso sexual callejero es una forma de violencia de género, además del riesgo que supone salir en determinadas horas del día, cuyas consecuencias negativas son a nivel psicológico y social, como cambiar la forma de vestir e incluso el lugar de residencia. En lugar de probar diferentes rutas para evitar el acoso, utilizar un algoritmo que genere varios caminos que reduzcan la distancia y el riesgo de acoso sería una solución viable para esta problemática.

### 1.2 Solución

La solución que se propone a través del software es un algoritmo que indique el mejor camino para peatones, teniendo en cuenta la distancia y el riesgo entre las vías a recorrer en Medellín. En este caso, el algoritmo elegido ha sido Dijkstra, el cual permite calcular el camino mas corto entre los nodos.

### 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 Siempre Seguras

Esta es una aplicación desarrollada en México, la cual a través de la recolección de tuits que denuncian acoso sexual, genera un mapeo de puntos críticos donde hay mayor incidencia de acoso sexual. A partir de la versión beta, que es la disponible, se ha logrado identificar más a fondo puntos específicos de la ciudad donde se es mas propenso vivir una situación de acoso sexual [8]

### 2.2 El camino más seguro a través de zonas seguras

Este es un proyecto desarrollado en la Universidad de Melbourne, donde, dado un conjunto de zonas seguras, calcula el camino que minimiza la distancia. Aquí, transformaron el mapa con las zonas seguras en un grafo, para así aplicar el algoritmo de Dijkstra y obtener el camino más corto [7]

### 2.3 Solución de seguridad inteligente para mujeres que viajan

Desarrollado en el Instituto de Ingeniería y Tecnología de Sri Shakthi, el objetivo de este proyecto es facilitar protección a mujeres que viajan largas distancias diariamente en caso de que se encuentren en una situación crítica, usando un sistema GPS. Además, hace uso del algoritmo de Dijkstra para que las autoridades logren localizar y atender de manera más eficiente la situación de acoso [1]

### 2.4 Algoritmo para prevenir acoso sexual usando *nearby search*

A través de un mapa, el cual muestra los puntos con alto riesgo de acoso sexual, se combina con la búsqueda del camino más corto, para así encontrar los puntos seguros más cercanos y generar las indicaciones para movilizarse desde el punto de origen al destino deseado. En este se usó el método de Bresenham, el cual se encarga de generar una aproximación de una línea recta entre dos puntos [6]

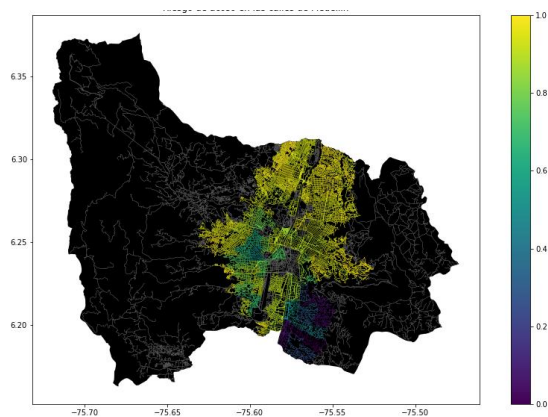
### 3. MATERIALES Y MÉTODOS

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)<sup>1</sup> y se descargó utilizando la API<sup>2</sup> 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<sup>3</sup>.



**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.

<sup>1</sup> <https://www.openstreetmap.org/>

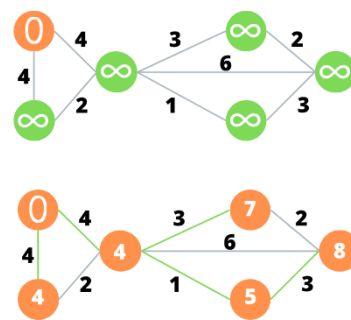
<sup>2</sup> <https://osmnx.readthedocs.io/>

#### 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 Dijkstra

Este algoritmo nos permite encontrar el camino más corto entre dos vértices de un grafo. Su funcionamiento se basa en que cualquier subtrayecto B -> D del trayecto más corto A -> D, entre A y D, es también el trayecto más corto entre B y D [5] Complejidad:  $V^2$



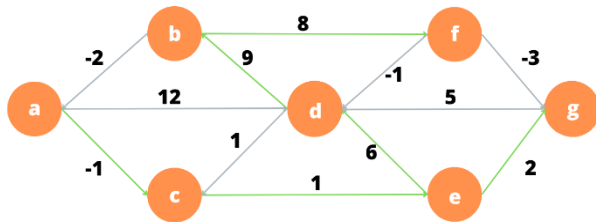
Paulina Cerón | Isabela Ortega



##### 3.2.2 Bellman Ford's

Este algoritmo nos permite encontrar el camino más corto de un vértice a los otros vértices de un grafo ponderado. Es similar al algoritmo Dijkstra, pero este puede trabajar con grafos que tienen cargas negativas. Su funcionamiento se basa en la sobreestimación de la longitud del trayecto desde el vértice inicial hacia los demás vértices. Después, por medio de la iteración, encuentra nuevos caminos que son más cortos que los estimados previamente.[2] Complejidad:  $O(V^3)$

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

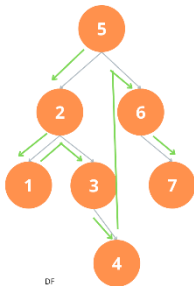


### 3.2. DFS (Depth First Search | Búsqueda profunda primero)

Es un algoritmo recursivo para buscar todos los vértices de un grafo. Este visita todos los nodos del grafo. Su implementación categoriza cada vértice en una de estas categorías: Visitada o No visitada. [4] Complejidad:  $O(V)$

El algoritmo funciona así:

1. Ubica uno de los vértices en el tope de una pila
2. Toma el elemento del tope y lo añade la lista de "visitado"
3. Crea una lista de los nodos adyacentes al vértice. Añade los "no visitados" al tope de la pila
4. Repite los pasos 2 y 3 hasta que la pila esté vacía



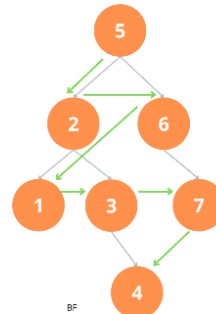
### 3.2.4 BFS (Breadth first search | Primera búsqueda en amplitud)

Es un algoritmo recursivo para buscar todos los vértices de un grafo. Este visita todos los nodos del grafo. Su implementación categoriza cada vértice en una de estas categorías: Visitada o No visitada. [3]

Complejidad:  $O(V+E)$

El algoritmo funciona así:

1. Ubica cualquier vértice del grafo al final de una cola
2. Toma el elemento del frente de la cola y lo ubica en la lista de "visitados"
3. Crea una lista de los nodos adyacentes al vértice. Añade los "no visitados" al final de la cola
4. Repite los pasos 2 y 3 hasta que la pila esté vacía



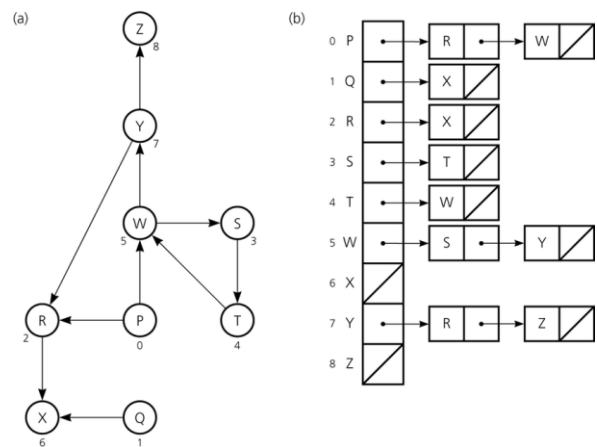
## 4. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

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<sup>4</sup>.

### 4.1 Estructuras de datos

Inicialmente, los datos se almacenan en un DataFrame a través de la librería de Pandas, y la información de esta es almacenada posteriormente como una lista de adyacencia, teniendo en cuenta los puntos de origen, de destino, y si es una calle de doble vía, y el cálculo a tener en cuenta relacionando la distancia y el riesgo

La estructura de los datos se presenta a continuación



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

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

4.2 Algoritmos

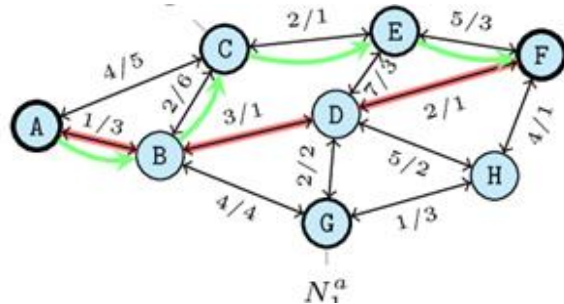
En este trabajo, proponemos un algoritmo para un camino que minimiza tanto la distancia como el riesgo de acoso sexual callejero.

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

El algoritmo elegido ha sido el algoritmo de Dijkstra. Su funcionamiento se trata de recorrer cada nodo y así, ir evaluando si la ruta que tiene actualmente es la óptima en términos del riesgo y la distancia, es decir, que esta variable tenga el valor más pequeño posible.

Para empezar, se asigna el valor 9999999999999999 a cada nodo, de tal manera que empiece a disminuir en cada evaluación de la relación distancia-riesgo, la cual se hace por cada nodo y va variando, hasta que llegue al resultado adecuado. Finalmente, cuando se haya llegado al nodo de destino, y este tenga el valor óptimo (por lo menos menor a 9999999999999999), se dará como exitosa la búsqueda de la ruta y la mostrará.

El algoritmo se ejemplifica a continuación



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

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

Para generar otras alternativas de rutas, se eligió considerar para una de las rutas solamente la distancia. Para la otra, se considera solamente la distancia, y una tercera prioriza el riesgo, siendo esta la única variable que interviene.

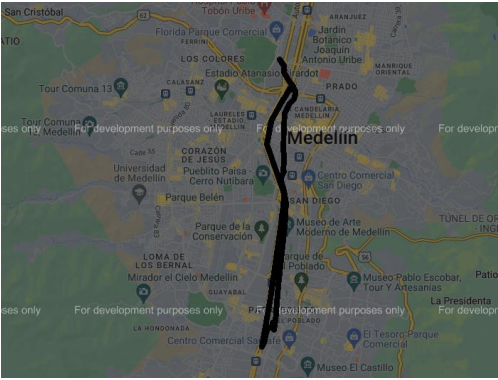


Figura 4: Mapa de la ciudad de Medellín donde se presentan tres caminos para peatones que reducen tanto el riesgo de acoso sexual como la distancia en metros entre la Universidad EAFIT y la Universidad Nacional.

4.3 Análisis de la complejidad del algoritmo

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

Tabla 1: Complejidad temporal del nombre de su algoritmo, donde V es el número de vértices y E es el número de aristas

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

Tabla 2: Complejidad de memoria lista de adyacencia

4.4 Criterios de diseño del algoritmo

Lo intuitivo y la eficiencia del algoritmo para encontrar el mejor camino (el objetivo principal) entre los nodos de un grafo; además del almacenamiento del orden de nodos que conforman el camino, fueron los puntos principales para la elección de este algoritmo.

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	3905	
Eafit	Unal	7744	

Eafit	Unal		17.9
-------	------	--	------

**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

Tiempos medios por ejecución	Tiempo total
60s	180 s

Tabla 4: Tiempos de ejecución del nombre del *algoritmo*

## 6. CONCLUSIONES

Con la implementación de este algoritmo teniendo en cuenta diferentes variables se logró visualizar que no hay un único camino correcto, y dependiendo de los puntos de prioridad de cada persona, se pueden tomar diferentes alternativas. En el caso de esta implementación, se observa que la implementación no es lo suficientemente eficiente para usarla en tiempo real, por lo que se apunta a una optimización del algoritmo, para así poder ser usada en tiempo real por medio de una aplicación móvil y contribuir a la reducción del riesgo del acoso sexual callejero al cual se encuentran expuestas las mujeres diariamente.

### 6.1 Trabajos futuros

Para implementaciones futuras de este trabajo, se busca primero la optimización del algoritmo, para luego ser trasladado a una aplicación web y una aplicación móvil, para tener interconectividad entre diferentes equipos. Además, se propone el uso de machine learning de tal manera que halle el camino más corto y menos peligroso de una mejor forma dependiendo de los caminos comunes de cada usuario.

## AGRADECIMIENTOS

Esta investigación ha sido apoyada por la beca Mejores Bachilleres de Sapiencia para la primera autora, y también ha contado con el apoyo y financiación del círculo familiar para la segunda autora.

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. Arumugam, S., Kannammal, K., Sonia, E. and Shanmugavadivel, B., 2022. Smart Safety Solution for Travelling Womens. [online] Ijrpr.com. Recuperado de:

<https://ijrpr.com/uploads/V3ISSUE9/IJRPR6919.pdf>

2. Bellman Ford's Algorithm Recuperado en Agosto 15, 2022, de Programiz. <https://www.programiz.com/dsa/bellman-ford-algorithm>
3. Breadth first search. Recuperado en Agosto 15, 2022, de Programiz. <https://www.programiz.com/dsa/graph-bfs>
4. Depth First Search (DFS). Recuperado en Agosto 15, 2022, de Programiz. <https://www.programiz.com/dsa/graph-dfs>
5. Dijkstra's Algorithm. Recuperado en Agosto 15, 2022, de Programiz. <https://www.programiz.com/dsa/dijkstra-algorithm>
6. Ma, D., 2020. Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search - Omdena. [online] Omdena | Building AI Solutions for Real-World Problems. Recuperado de: <https://omdena.com/blog/path-finding-algorithm>
7. S. Aljubayrin, J. Qi, C. S. Jensen, R. Zhang, Z. He and Z. Wen. 2015. The safest path via safe zones. IEEE 31st International Conference on Data Engineering, 2015, pp. 531-542
8. Sergio Adrián Ángeles. 2021. Siempre Seguras, una app para mapear acoso sexual callejero en México. SWI *swissinfo.ch*. Recuperado de SWI: [https://www.swissinfo.ch/spa/m%C3%A9xico-violencia-machista\\_siempre-seguras--una-app-para-mapear-acoso-sexual-callejero-en-m%C3%A9xico/46537134](https://www.swissinfo.ch/spa/m%C3%A9xico-violencia-machista_siempre-seguras--una-app-para-mapear-acoso-sexual-callejero-en-m%C3%A9xico/46537134)