

Algoritmo De Búsqueda Del Camino Mas Corto Evitando Zonas De Alto Indice De Acoso Sexual

Samuel Oviedo
Universidad Eafit
Colombia
sovedop@eafit.edu.co

Jerónimo Pérez Baquero
Universidad Eafit
Colombia
jperezb5@eafit.edu.co

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

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

Andres Gil
Universidad Eafit
Colombia
ajgiln@eafit.edu.co

RESUMEN

El acoso callejero es un problema que sufren las mujeres a diario en diferentes lugares públicos, este tipo de acciones vulgares realizadas por desconocidos pone en riesgo la integridad, comodidad, y seguridad de las mujeres por lo que es un problema que nos involucra a todos; sin hablar de las consecuencias que esto puede dejar en la víctima como lo son inseguridad, cambio obligado de ruta hacia su lugar de destino, cambio de vestimenta, ansiedad y miedo.

Palabras clave

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

1. INTRODUCCIÓN

Lo que nos llevó a realizar este proyecto se basa en la problemática que recae principalmente en las mujeres, en el acoso sexual recibido hacia ellas en distintos ámbitos de su cotidianidad, principalmente en la circulación por la ciudad. Se han visto muchos casos de acoso sexual, evaluados en la ciudad de Medellín que en donde se ha realizado este proyecto: Situación de acoso sexual hacia una mujer en el metro de Medellín, fue grabado un video por parte de un usuario donde se evidencia la escena y que luego se hizo viral en las redes sociales, según el articulo referenciado, casos como estos se han visto en un promedio de 1 cada dos días. El siguiente caso es de un supuesto caso de abuso sexual en Medellín menores de edad que habrían sido abusados por un auxiliar de nutrición en un centro infantil del barrio Santa Cruz, en el nororiente de la ciudad.

1.1. Problema

El problema del acoso callejero ha sido recurrente en nuestra sociedad desde hace años, por lo que nuestro grupo se ha propuesto reducir el índice de acoso sexual, que lo entendemos como cualquier acción vulgar por parte de un extraño hacia la persona acosada, para esto nos comprometimos a crear un programa que calcule el camino más seguro y corto hacia el destino que quiere llegar la persona.

1.2 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.

A continuación, explicamos cuatro trabajos relacionados con la búsqueda de caminos para prevenir el acoso sexual callejero y la delincuencia en general.

3. MATERIALES Y MÉTODOS

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos del camino más corto restringido para abordar el acoso sexual callejero.

3.1 Recogida y tratamiento de datos

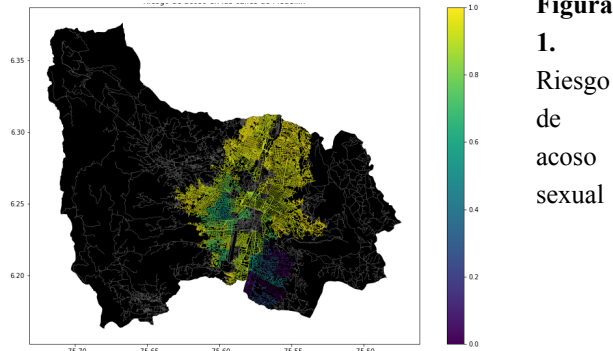
El mapa de Medellín se obtuvo de Open Street Maps (OSM) ¹ y se descargó utilizando la API ² OSMnx de Python. La (i) longitud de cada segmento, en metros; (2) la indicación de si el segmento es de un solo sentido o no, y (3) las representaciones binarias conocidas de las geometrías se obtuvieron de los metadatos proporcionados por OSM.

Para este proyecto, se calculó la 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 normaliza, utilizando el

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

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.



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, obtenida de la Encuesta de Calidad de Vida de Medellín, de 2017.

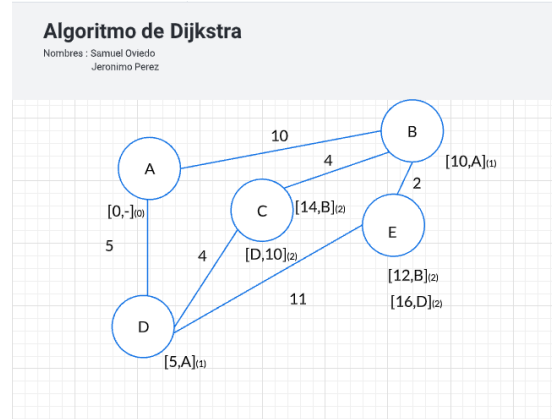
3.2 Alternativas de camino más corto con restricciones

A continuación, presentamos diferentes algoritmos utilizados para el camino más corto restringido.

3.2.1 Algoritmo de Dijkstra

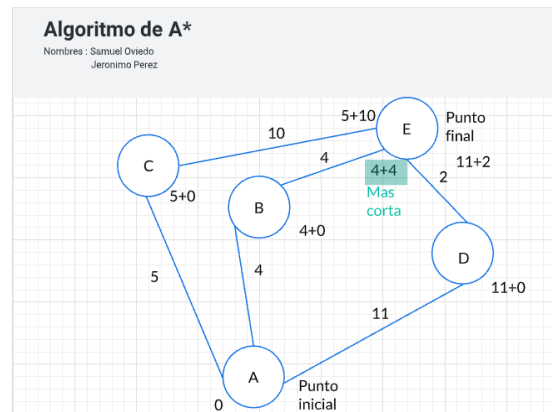
Dado un vértice origen y unos nodos correspondientes, el algoritmo se encarga de explorar cada camino en busca del más corto, al encontrarlo el programa se detiene. Su primera parte se basa en el etiquetado de los nodos teniendo como nodo permanente a $[0, -](0)$ y los demás nodos se van descubriendo a partir de éste, teniendo como esquema $[(\text{distancia acumulada de nodos}), (\text{nodo$

predecesor)](iteraciones)[1] su complejidad



3.2.2 Algoritmo de búsqueda A

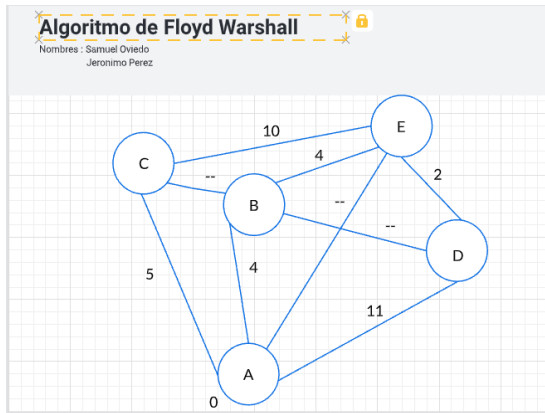
Dado un nodo origen y uno objetivo, el algoritmo tratará de encontrar la ruta más corta, el algoritmo expresado en función como $f(n) = g(n) + h(n)$ sus partes son $g(n)$: es el costo de las movidas realizadas. $h(n)$: Representa el costo estimado del mejor camino.



3.2.3 Algoritmo Floyd-Warshall

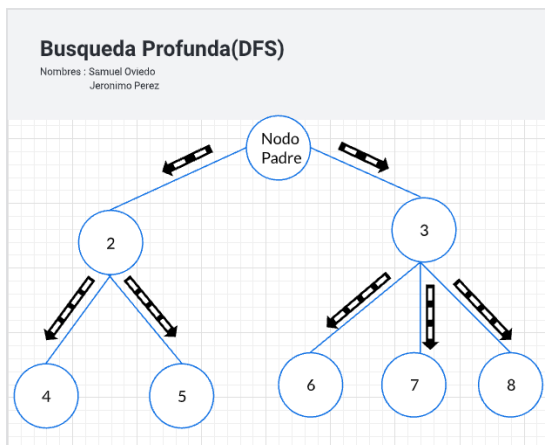
Este algoritmo a diferencia de Dijkstra está diseñado para calcular los caminos cortos de todos los pares, sin la necesidad de un nudo inicial o de partida, aparte los datos deben estar representados en una matriz y los nudos no entrelazados tienen un valor nulo o muy cercano a infinito para hacer más fácil la reconstrucción de los caminos, su complejidad es de $O(n^3)$ [2]

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



3.2.4 Búsqueda en profundidad(DFS)

algoritmo utilizado para recorrer grafos o nodos,tomando como punto de partida un nodo padre hasta el nodo hijo ,y cuando acaba con ese camino y prueba con otro recorrido su complejidad es $O(V)[3]$



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 que usaremos para resolver el problema del camino más corto restringido son las matrices de adyacencia con grafos no dirigidos ,ya que consideramos que el problema del acoso se da mientras la persona está caminando, por lo que en una calle podría dirigirse tanto para atrás como para adelante en su camino(no hay restricción en este sentido) ,el funcionamiento de esta estructura de datos se basa en que dado un grafo o un conjunto de pares ordenados en una matriz podemos llenar estos datos de la siguiente forma : si existe una relación entre nodos se coloca un 1 si no es así se coloca un 0;junto con esto usaremos la estructura de datos llamada cola ,la cual su funciona desencolado el elemento con mayor prioridad antes que un elemento de menor prioridad.. La estructura de los datos se presenta en la Figura 2 y 3.

Figura

2:

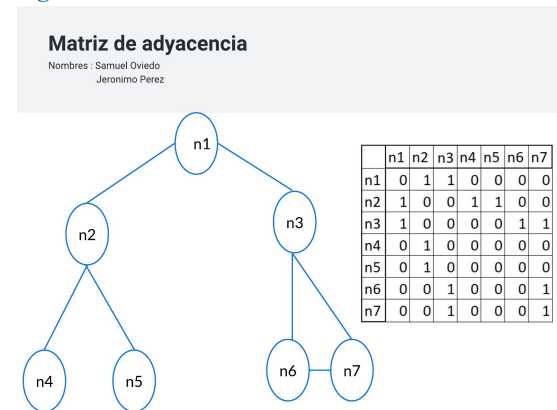
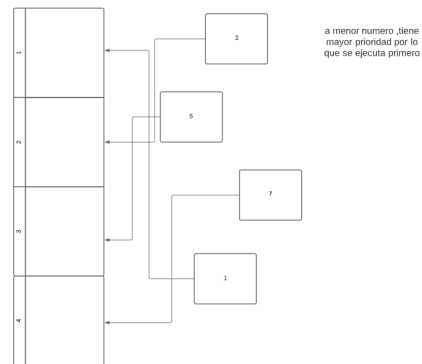


Figura 3:



⁴ <http://www.github.com/ JeroZp/proyecto/>

4.2 Algoritmos

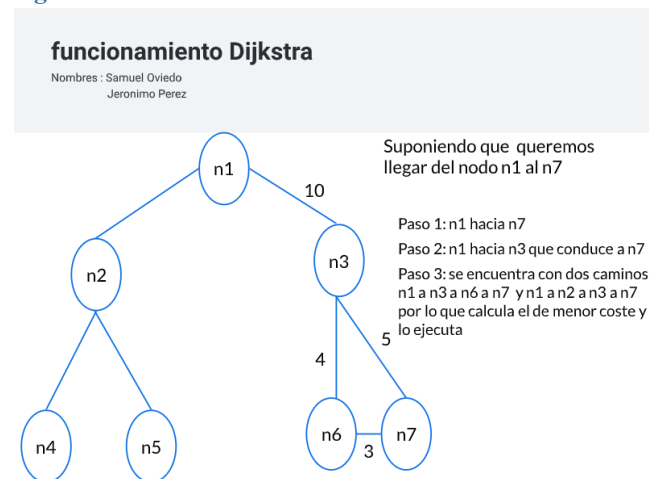
En este trabajo, proponemos algoritmos para el problema del camino más corto restringido. El primer algoritmo calcula el camino más corto sin superar un riesgo medio ponderado de acoso. El segundo algoritmo calcula el camino con el menor riesgo medio ponderado de acoso sin superar una distancia d .

4.2.1 Algoritmo Dijkstra

Para nuestro proyecto utilizaremos el algoritmo de Dijkstra el cual funciona de la siguiente manera Suponiendo que $A[i, j]$ representa la matriz de adyacencia, tenemos: si $i = j$ $A[i, j] = -1$, si no existe un camino de i a j , y $A[i, j] =$ longitud del camino de i hasta j . Este algoritmo presenta una complejidad de n^2 donde n es el número de nodos ,aparte de esto su manera de optimizar esto es ir cortando por trozos el camino desde un nodo inicial y uno final si se presentan varias opciones escoge el que menor coste de distancia le genere .

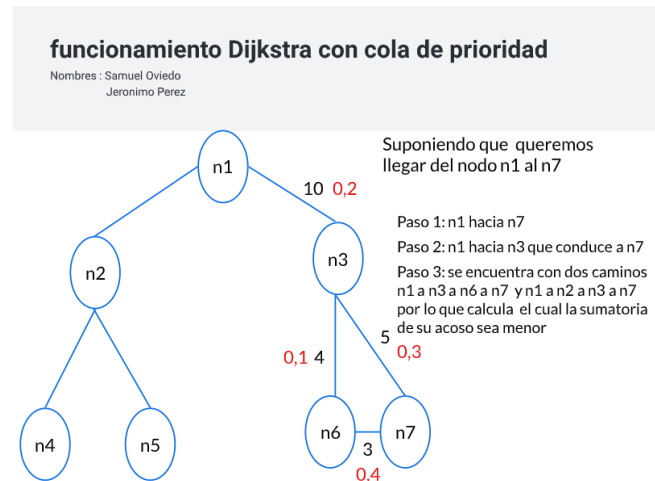
Figura

4:



4.2.1 Algoritmo Dijkstra con una cola de prioridad

Su funcionamiento es exactamente igual al de dijkstra ,su único cambio viene en el último paso donde entre las opciones de menor camino encontrado ,toma esa variable mas la cual es para nuestro caso “harassment” ,donde prioriza el camino el cual la suma de su promedio de acoso sexual sea menor . con una complejidad de $O = (E \log V)$ ya que se utiliza restricciones como un montículo binario



4.2.2 Dijkstra con uso de peso y cola de prioridad

Para lograr promediar el problema del acoso con la distancia del camino a encontrar;usamos la conocida solución de “knapsack”[4] el cual consiste en que una mochila no debe soportar mas de cierta cantidad de peso ,para esto en este caso se multiplica cada objeto por su peso y se realiza la sumatoria de esta operación para cada caso;en nuestro problema de acoso se multiplicará el acoso por la distancia entre dos nodos ,generandonos una manera de optimizar y mejorar la certeza del programa.

El algoritmo se ejemplifica en la Figura 4.

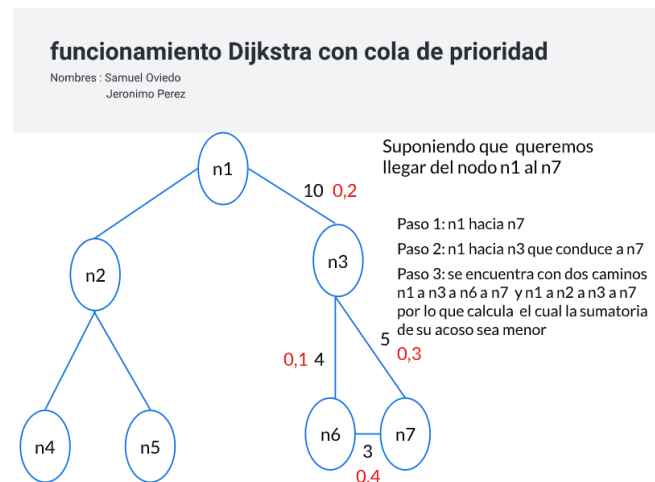


Figura 4: Resolución del problema del camino más corto restringido con la Búsqueda Dijkstra y uso de una cola de prioridad

4.4 Análisis de la complejidad de los algoritmos

la complejidad del algoritmo Dijkstra se basa en la cantidad de nodos que deba recorrer donde V representa la cantidad de nodos y el elevado a la dos se debe a que para comparar

y encontrar el peso mínimo entre dos nodos se usa un for anidado [1]

| Algoritmo | Complejidad temporal |
|---------------------------------------|----------------------|
| algoritmo de dijkstra con uso de peso | $O(V^2)$ |
| | |

Tabla 1: Complejidad temporal del algoritmo de dijkstra con uso de peso donde N significa la cantidad de nodos [1]

| Estructura de datos | Complejidad de la memoria |
|-----------------------------------|---------------------------|
| grafo en el algoritmo de dijkstra | $O(V + E)$ |
| | |

Tabla 2: Complejidad de memoria del grafo con uso del algoritmo de dijkstra es $O(|V| + |E|)$, donde V es la cantidad de nodos y E es la cantidad de aristas.[4]

4.5 Criterios de diseño del algoritmo

Después de analizar las opciones de algoritmos descritas al principio del informe y comparar las complejidades de cada uno escogimos el que menor coste de recursos y tiempo de ejecución nos causará al ser procesado (por ejemplo el marshall tenía una complejidad de V^3 mientras que el Dijkstra es menor con V^2), aparte de tener una puerta abierta hacia mejoras de optimización como por ejemplo el próximo uso de colas de prioridad.

5. RESULTADOS

En esta sección, presentamos algunos resultados cuantitativos sobre el camino más corto y el camino con menor riesgo.

5.1.1 Resultados del camino más corto

A continuación, presentamos los resultados obtenidos para el camino más corto, sin superar un riesgo medio ponderado de acoso r , en la Tabla 3.

| Origen | Destino | Distancia más corta | Sin exceder r |
|--------|---------|---------------------|-----------------|
| | | | |

| | | | |
|--------------------------|-------------------------|------|------|
| Universidad EAFIT | Universidad de Medellín | 7100 | 0.9 |
| Universidad de Antioquia | Universidad Nacional | 2600 | 0.85 |
| Universidad Nacional | Universidad Luis Amigó | 1700 | 0.85 |

Tabla 3. Distancias más cortas sin superar un riesgo de acoso medio ponderado r .

5.1.2 Resultados de menor riesgo de acoso

A continuación, presentamos los resultados obtenidos para el trayecto con menor riesgo de acoso medio ponderado, sin superar una distancia d , en la Tabla 4.

| Origen | Destino | Acoso más bajo | Sin exceder d |
|--------------------------|-------------------------|----------------|-----------------|
| Universidad EAFIT | Universidad de Medellín | 0.92 | 5,000 |
| Universidad de Antioquia | Universidad Nacional | 0.85 | 7,000 |
| Universidad Nacional | Universidad Luis Amigó | 0.89 | 6,500 |

Tabla 3. Menor riesgo de acoso ponderado sin superar una distancia d (en metros).

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.

| | Tiempos medios de ejecución (s) |
|--|---------------------------------|
| Universidad EAFIT a Universidad de Medellín | 92 s |
| De la Universidad de Antioquia a la Universidad Nacional | 62 s |
| De la Universidad Nacional a la | 56 s |

| | |
|------------------------|--|
| Universidad Luis Amigó | |
|------------------------|--|

Tabla 4: Tiempos de ejecución del nombre del *algoritmo (dijkstra)* para las consultas presentadas en la Tabla 3.

6. CONCLUSIONES

los caminos más cortos cambian significativamente en la práctica real del uso del programa ya que evita grandes recorridos o altos niveles de acoso debido al uso del “peso” y el algoritmo que usamos .aparte de esto damos una solución a una problemática social que sufre Medellín y que refleja la falta de educación y comportamiento ciudadano .en una práctica real al programa le falta eficiencia en su ejecución ,ya que no da resultados lo suficientemente rápidos para su ejecución en la vida cotidiana.

6.1 Trabajos futuros

despues de que hayamos tenido la experiencia de haber realizado un progrma junto con algoritmos y estructuras de datos podemos mejorar nuestro metodo de optimizacion de tiempo en la ejecucion aparte de la seccion estettica del mapa y la forma interactiva con el usuario .aparte si el proyecto continua podriamos desarrollar una base de datos donde reportemos estadisticas de acoso sexual actualizadas y sobretodo utiles para la seguridad de las personas en la ciudad igualmente en el aprendizaje automatico para buscar mejoras en los datos del programa pa que se adpate mejor al usaurio , apendiedno patrones de acoso , usuario promedio ,uso de la aplicacion,etc.

AGRADECIMIENTOS

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 *Shapefile*.

REFERENCIAS

1. Delfstack,2021.Algoritmo de Dijkstra.

www.delftstack.com/es/howto/python/dijkstra-algorithm-python/#:~:text=El%20algoritmo%20de%20Dijkstra%20se,desde%20el%20v%C3%A9rtice%20de%20origen.

2. Simar Enrique,2014.Eficencia algorimica en aplicaciones de grafos
<http://www.scielo.org.co/pdf/rfing/v23n36/v23n36a09.pdf>

3. simplilearn,Simplilearn.

www.simplilearn.com/tutorials/data-structure-tutorial/dfs-algorithm#:~:text=Complexity%20Of%20Depth%20

[First%20Search%20Algorithm&text=If%20the%20entire%20graph%20is,all%20of%20its%20neighboring%20edges.](http://www.delftstack.com/es/howto/python/dijkstra-algorithm-python/#:~:text=First%20Search%20Algorithm&text=If%20the%20entire%20graph%20is,all%20of%20its%20neighboring%20edges.)

4.Wikipedia,2022,Problema de mochila.

https://es.wikipedia.org/wiki/Problema_de_la_mochila#:~:text=En%20algoritmia%2C%20el%20problema%20de,posibles%20soluciones%20a%20un%20problema.

5.stackOverflow,2021,Manmay Barot.

[https://stackoverflow.com/questions/50856391/what-is-the-space-complexity-of-dijkstra-algorithm#:~:text=The%20time%20complexity%20for%20Dijkstra,O\(E%20log%20V\).](https://stackoverflow.com/questions/50856391/what-is-the-space-complexity-of-dijkstra-algorithm#:~:text=The%20time%20complexity%20for%20Dijkstra,O(E%20log%20V).)