# **UNA MEDELLÌN MÀS SEGURA**

|  |  |  |
| --- | --- | --- |
| David Londoño Sánchez  Universidad Eafit  Colombia  dlondonos2@eafit.edu.co | Andrea Serna Universidad Eafit Colombia asernac1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **RESUMEN**

Desde siempre ha existido cierto riesgo al salir a la calle a solo en determinados lugares, existe una alta probabilidad de ser acosado sexualmente, por ende, es de suma importancia encontrar la forma de identificar las rutas más seguras para evitar estas situaciones de riesgo y al mismo tiempo de no demorarnos mucho tiempo para hacer cierto recorrido.

# **1. INTRODUCCIÓN**

La razón por la que en este semestre nos centramos en este problema concreto es debido a que en resurgimiento de la sociedad luego de la pandemia algunos lugares están tendiendo a volverse peligrosos, en los que aún no hay vigilancia, pero si muy mala energía y un riesgo muy alto de ser acosados, además de por supuesto, solucionar la necesidad existente de llegar lo más rápido posible a algún lugar. Por esto es que necesitamos poder calcular la menor ruta sin superar un riesgo de acoso dado, y la ruta con menor acoso posible sin superar una distancia d dada. El algoritmo que he decidido usar para encontrar la respuesta a esta incógnita es el algoritmo de Dijkstra, solo que no pude utilizarlo efectivamente, por lo que no obtuve resultados.

# **1.1 Problema**

El problema consiste en crear un algoritmo capaz de calcular la ruta más corta con un máximo de posibilidad de acoso, y la ruta con menor acoso posible sin superar cierta distancia. Lograr esto es particularmente útil para una sociedad asustadiza luego de estar encerrada por 2 años, preferirán encerrarse o tomar rutas muy largas por miedo a tener confrontaciones; un algoritmo que solucione esto puede devolverles la tranquilidad a las personas y concientizaría a la gente lentamente volviendo la ciudad más segura.

# **1.2 Solución**

El modelo de algoritmo que decidí usar para el proyecto del camino más corto posible sin sobrepasar un promedio de acoso fue el de Dijkstra, debido a que me pareció mas fácil de escribir y a la vez es más eficiente que otros algoritmos similares como el back tracking.

**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 Mayor seguridad mediante algoritmos de búsqueda cercana.**

Usaron el algoritmo de línea de Bresenham para encontrar la ruta más corta de un punto a otro, el problema a solucionar era encontrar el camino más rápido al hospital más cercano desde un punto dado, sin pasar por zonas peligrosas por acoso sexual. Lograron encontrar la ruta con menos acoso, y en caso de encontrar dos rutas con igual riesgo el sistema selecciona el más corto

Ma, D 2020 *Preventing Sexual Harassment Through a Path Finding Algorithm Using Nearby Search*. Nueva York, Nueva York.

## **2.2 La ruta más segura posible.**

Se resolvió un problema de seguridad de las calles de NYC, en este usaron un algoritmo de múltiples capaz, que primero carga un mapa, hace un análisis de riesgos con la base de datos de accidentes de transito y arrestos de la policía, luego da como resultado cual es la ruta con menor riesgo para ir de un punto ¨a¨ a un punto ¨b¨; en caso de haber varias rutas con el mismo valor de riesgo mínimo, el sistema selecciona la opción más corta. Como resultado este programa arroja una ruta bastante similar a la de maps, pero siendo esta potencialmente más segura por el análisis que la respalda.

Soni S, V, Gauri y S, Chaurasia 2019 *Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data*. International Journal of Advanced Science and Technology, Jaipur, Rajasthan.

## **2.3 Nueva esperanza para la mujer india**

Se resuelve un problema de posible acoso sexual en las calles de India, usaron un algoritmo ¨Bottom up” para el análisis de riesgo, en el que se avanza creando graficas por prueba y error. Para la creación de la ruta se usó el Dijkstra's algorithm. Obtuvieron un algoritmo que le da a las mujeres indias una ruta segura con riesgo bajo más cercana que las que ya conoce. Aryan G y B, Khetan 2018 *A Data Integration and Analysis System for Safe Route Planning.* International Journal of Science and Research (IJSR),India.

## **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]](#footnote-1)y se descargó utilizando la API[[2]](#footnote-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 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, obtenida de la Encuesta de Calidad de Vida de Medellín, de 2017.

**3.2.1 Techie Delight**

Este algoritmo usa una matriz en Python, lo que hace es que recibe una posición inicial y una final, y casilla por casilla revisa sus 4 posiciones adyacentes (arriba, abajo, izc, der) hasta llegar a la posición final, una vez llega allí retorna cuentas casillas tuvo que recorrer. A mi parecer tiene un orden O(MxN), puesto que en el peor de los casos (que deba ir de esquina a esquina) este algoritmo tendrá que usar todas las casillas para llegar a la posición final

**3.2.2 Lee Algorithm**

Este algoritmo usa una matriz en Python, lo que hace es que recibe una posición inicial y una final, crea una cola en cada movimiento y añade la nueva casilla a la que se haya movido (se mueve a casillas que existan y sean distintas de 0), en caso de que encuentre su casilla destino retorna el valor que haya sumado cada vez que crea una cola (siempre que avanza, a la cola que crea le añade un +1).Al igual que el anterior este algoritmo tiene un orden O(MxN), puesto que en el peor de los casos tendrá que recorrer todas las casillas de la matriz.

**3.2.3 Dijkstra algorithm**

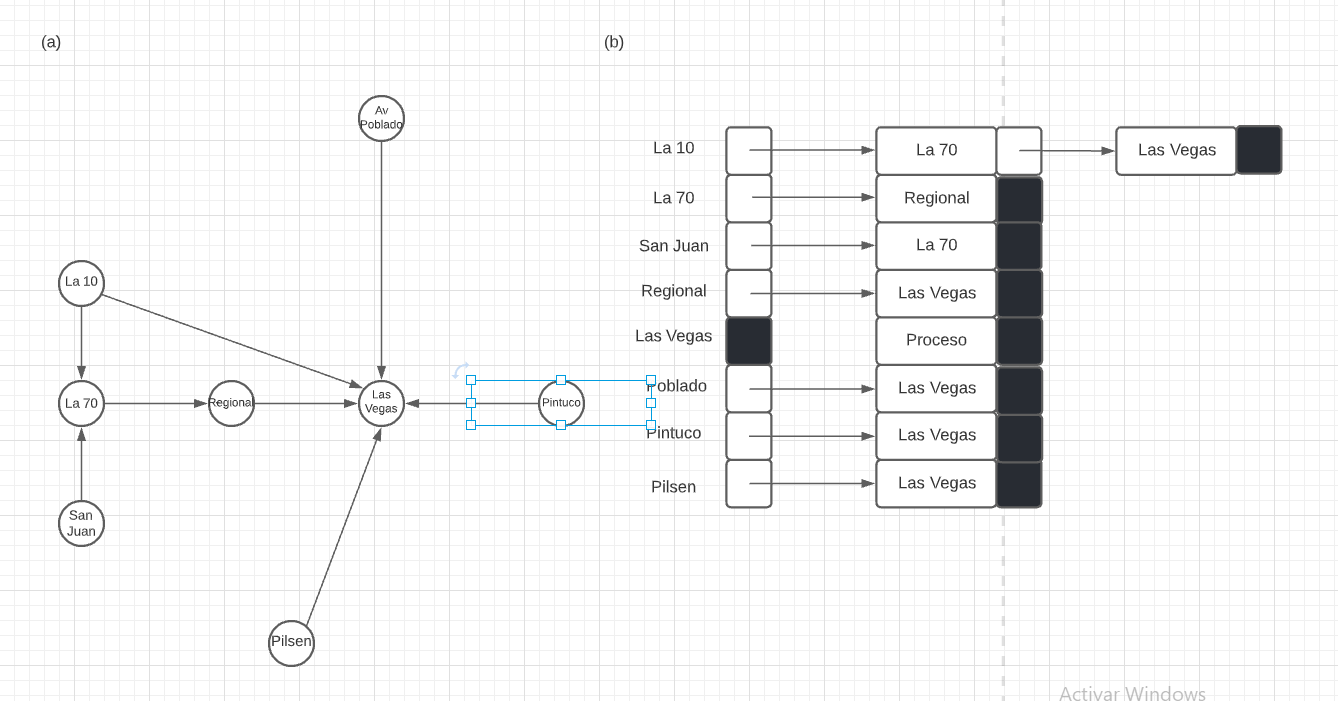
Este algoritmo usa el peso de las aristas entre nodos para encontrar el camino más rápido a un nodo final, siempre escogiendo el nodo más cercano al nodo inicial (por esto puede que el camino final no sea el más rápido, puesto que podría tener que escoger algún segmento más pesado que el resto para conseguirlo. En el peor de los casos este algoritmo tendrá un costo de O(n)

## **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**

## La estructura usada para el proyecto fue una lista de adyacencia. La estructura de los datos se presenta en la Figura 2.



**Figura 2:** Un ejemplo de el mapa de calles se muestra en (a) y su representación como lista de adyacencia se presenta en (b).

(No están realmente orientadas así, pero se hizo así para mostrar un ejemplo sencillo)

**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 *r*. El segundo algoritmo calcula el camino con el menor riesgo medio ponderado de acoso sin superar una distancia *d*.

**4.1**

Para ambas situaciones se usa el mismo algoritmo de Dijkstra con distntos parámetros. Diagram

Description automatically generated

**4.4 Análisis de la complejidad de los algoritmos**

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** |
| Dijkstra | O(V\*A) |
| Dijkstra | O(V\*A) |

**Tabla 1:** Complejidad temporal del algoritmo de camino más corto es V\*A siendo V los vértices y A las aristas. En el peor de los casos la complejidad será de V\*V, que significa que hay igual número de aristas que de vertices

|  |  |
| --- | --- |
| **Estructura de datos** | **Complejidad de la memoria** |
| Lista de adyacencia | O(V\*A) |
| Lista de adyacencia | O(V\*A) |

**Tabla 2:** Complejidad de memoria de las listas de adyacencia será el tamaño total del grafo A\*V, siendo V los vértices y A las aristas. En el peor de los casos serà V\*V puesto que habrá igual numero de vértices que aristas

**4.5 Criterios de diseño del algoritmo**

El algoritmo de Dijkstra es el algoritmo de camino más corto del cual hay mas información por lo que este es mas fácil de comprender y aplicar para la solución del problema, además de que este tiene un orden cercano a O(N) en el peor de los casos (serian nAristas+nVertices) mientras que otros algoritmos similares tienen un orden ligeramente mayor.

**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**

Mi código no funcionó y trabajare en el durante las vacaciones, por ello no hay datos aqui

|  |  |  |  |
| --- | --- | --- | --- |
| **Origen** | **Destino** | **Distancia más corta** | **Sin exceder *r*** |
| Universidad EAFIT | Universidad de Medellín | ?? | 0.84 |
| Universidad de Antioquia | Universidad Nacional | ??? | 0.83 |
| Universidad Nacional | Universidad Luis Amigó | ?? | 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**

Mi código no funcionó y trabajare en el durante las vacaciones, por ello no hay datos aqui

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

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

**5.2 Tiempos de ejecución del algoritmo**

Mi código no funcionó y trabajare en el durante las vacaciones, por ello no hay datos aqui

## 

|  |  |
| --- | --- |
|  | **Tiempos medios de ejecución (s)** |
| Universidad EAFIT a Universidad de Medellín | 100.2 s |
| De la Universidad de Antioquia a la Universidad Nacional | 800.1 s |
| De la Universidad Nacional a la Universidad Luis Amigó | 845 s |

## **6. CONCLUSIONES**

Estas conclusiones son sacadas a apartir del trabajo de mis compañeros puesto que no tengo código para la entrega 3.

Realmente no hubo una diferencia significativa de distancia entre los caminos mas cortos y los que tenían menor posibilidad de acoso, esto debido a que realmente no eran rutas particularmente inseguras por lo que el algoritmo no tuvo que trabajar extra para encontrar una mejor ruta.

Este algoritmo es particluarmente útil, no solo para la ciudad y no solo para las mujeres, en general la posibilidad de conocer una ruta por la que sabes que no te pasara nada es un alivio gigante, sobretodo para nosotros que vivimos en una ciudad tan peligrosa como lo es Medellin, donde el mal te podría estar esperando en cualquier esquina.

Aun así con las estructuras de datos que tenemos el algoritmo se vuelve poco eficiente, y en una situación desesperada no tendría ninguna utilidad, puesto que este demora demasiado tiempo para arrojar resultados. Hasta no mejorar su tiempo de respuesta a solo unos segundos (como maps) este solo servirá para casos my específicos en la vida real

**6.1 Trabajos futuros**

Durante las vacaciones y el próximo semestre pienso entender a fondo como funciona el algoritmo y mejorarlo para que su tiempo de procesamiento se reduzca. Puesto que me parece que este es un algoritmo que realmente puede ser de ayuda en la vida cotidiana me gustaría cambiar el riesgo de acoso por la criminalidad en la zona (de ser posible sacaria los datos de la base de datos de la policía y/o fiscalía): esto para implementarlo en una pagina web o una aplicación móvil sencilla, que me permita a mi (o en caso de funcionar muy bien, a otras personas también) a tener una idea de por cuales zonas podría circular con tranquilidad y por cuales no

# **AGRADECIMIENTOS**

Esta investigación fue llevada a cabo gracias al apoyo económico de mis padres, y gracias a la temática de proyecto desarrollado en la asignatura Datos Y Algoritnos 1 de la universidad de EAFIT.

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)