



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

El problema al cual nos enfrentamos se basa en crear un algoritmo que nos permita calcular el camino más corto sin superar un riesgo medio ponderado de acoso  $r$  y el camino

Como se puede ver, hay varias ubicaciones de hospitales en un radio de 800 metros; sin embargo, es bastante intuitivo

evitar las ubicaciones superiores, ya que las puntuaciones de riesgo son más altas (ilustradas por colores más cálidos). Mirando esto, se puede simplemente tomar una decisión sobre por dónde pasar: Priorizando la seguridad general de la ruta y eligiendo el lugar más cercano.

### 3.2 La ruta segura en NY

El problema que resolvieron fue predecir las rutas más seguras utilizando datos sobre delitos y accidentes. En este trabajo se propone una solución que sugiere a las personas el camino más seguro para viajar desde el origen hasta el destino. La tecnología de la API de Google combinada con modelos de aprendizaje automático, concretamente el algoritmo de agrupación K Means y el algoritmo KNN Regressor, se utilizan en esta solución propuesta. Se tienen en cuenta conjuntos de datos de arrestos y accidentes de la ciudad de Nueva York para predecir la ruta más segura.

En contraste con la sugerencia de rutas Google maps, que se basa puramente en la distancia más corta; esta solución es modificada y recomienda rutas que son seguras para viajar. La ruta segura en esta solución significa la ruta con la puntuación de riesgo más baja y se calcula en base a los accidentes y delitos ocurridos en esa ruta o en las regiones cercanas.

Esta solución es muy útil especialmente para las personas que son nuevas en la ciudad o son turistas.

### 3.3 Búsqueda de rutas para viajeros

Este estudio ha desarrollado un método para incorporar la seguridad en el proceso de búsqueda de rutas, mediante el desarrollo y la aplicación de un nuevo índice de seguridad, este estudio ha desarrollado una metodología para incorporar aspectos de seguridad en el proceso de búsqueda de rutas de los viajeros.

El proceso de búsqueda de rutas seguras consta de dos partes principales: un índice de riesgo de seguridad específico de la ruta y un algoritmo de búsqueda de rutas que tiene en cuenta tanto el tiempo de viaje como la seguridad. El índice de riesgo de seguridad se formuló utilizando el mecanismo de colisión a lo largo del segmento de carretera y en la intersección, la información específica de los automovilistas (por ejemplo tipo de vehículo, edad, estado del pavimento), y en el modelo de índice de seguridad para poder tener en cuenta las necesidades individuales de los viajeros.

El algoritmo de búsqueda de rutas, que combina movilidad y seguridad, tiene tres objetivos: Reducir el tiempo de viaje, reducir el índice de riesgo de seguridad de la ruta y evitar los lugares con mayor índice de riesgo para la seguridad a lo largo de la ruta. Ya que se debe asignar una puntuación de seguridad a instalaciones de tráfico individuales (por ejemplo, segmento de carretera, intersección o a toda la ruta), utilizaron Dijkstra, el cual calculó las puntuaciones de cada ruta utilizando criterios propuestos, como la minimización de

las transiciones entre categorías de carreteras, la minimización del número de giros a la izquierda y la densidad de las intersecciones lo más baja posible.

### 3.4 Seguridad para los turistas motorizados

En este trabajo se construyó una aplicación de enrutamiento para que los turistas motorizados eviten conducir en zonas relativamente peligrosas.

Se presenta un enfoque de enrutamiento que evita las zonas relativamente peligrosas de una ciudad. La información sobre la peligrosidad de algunas zonas urbanas se obtiene utilizando información geográfica voluntaria (VGI), datos abiertos gubernamentales para detectar propiedades y funcionalidades de la infraestructura urbana y datos históricos de delincuencia de los departamentos de policía para detectar puntos calientes de delincuencia. Por lo tanto, se presenta los fundamentos de la cartografía y el análisis de la delincuencia con SIG, el uso práctico de la VGI para el enrutamiento y se describe la contribución dentro del campo de las soluciones de enrutamiento.

Para el uso práctico, se simplifica toda la información de la infraestructura urbana y se propone un índice de seguridad, que representa la seguridad relativa en el área de investigación. Además, se detectan los puntos conflictivos históricos y se utilizan como obstáculos de enrutamiento. Los arcos de la red de carreteras se ponderan en función del índice de seguridad y los puntos conflictivos históricos se introducen como polígonos de obstáculos.

En este trabajo se utiliza un algoritmo de dijktra modificado para resolver la problemática.

## 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

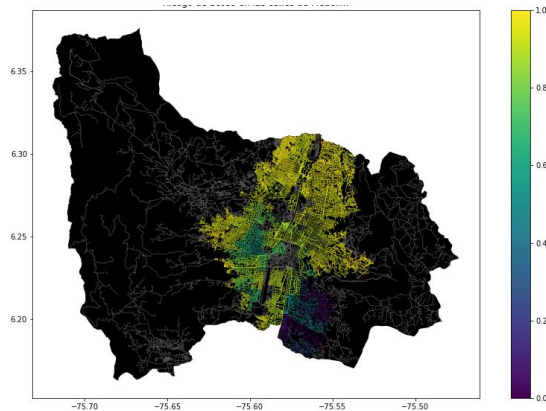
El mapa de Medellín se obtuvo de Open Street Maps (OSM) <sup>1</sup>y se descargó utilizando la API <sup>2</sup> OSMnx de Python. La (i) longitud de cada segmento, en metros; (2) la indicación de si el segmento es de un solo sentido o no, y (3) las representaciones binarias conocidas de las geometrías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

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

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

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

### 3.2 Alternativas de camino más corto con restricciones

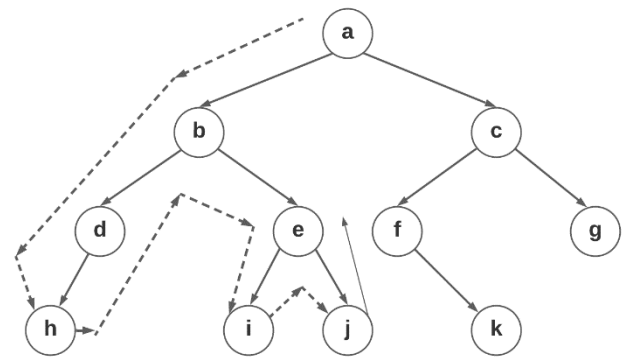
A continuación, presentamos diferentes algoritmos utilizados para el camino más corto restringido.

#### 3.2.1 DFS

Una búsqueda en profundidad (DFS) es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo. Su funcionamiento consiste en ir expandiendo cada uno de los nodos que va localizando, de forma recurrente (desde el nodo padre hacia el nodo hijo). Cuando ya no quedan más nodos que visitar en dicho camino, regresa al nodo predecesor, de modo que repite el mismo proceso con cada uno de los vecinos del nodo. Cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

La búsqueda en profundidad se usa cuando se quiere probar si una solución entre varias posibles cumple con ciertos requisitos. [2]

Ejemplo de su figura vectorial:

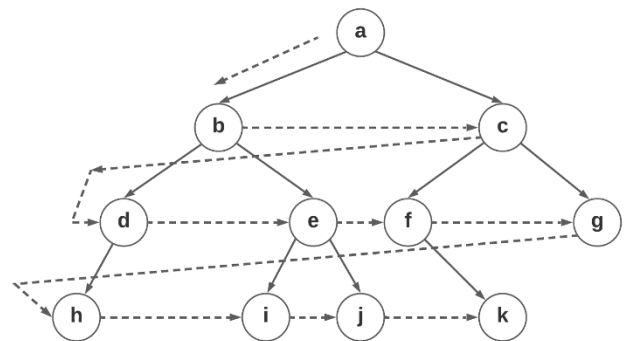


#### 3.2.2 BFS

Una búsqueda en anchura (BFS) es un algoritmo de búsqueda para lo cual recorre los nodos de un grafo, comenzando en la raíz (eligiendo algún nodo como elemento raíz en el caso de un grafo), para luego explorar todos los vecinos de este nodo. A continuación, para cada uno de los vecinos se exploran sus respectivos vecinos adyacentes, y así hasta que se recorra todo el grafo. Cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, concluye la búsqueda.

La búsqueda por anchura se usa para aquellos algoritmos en donde resulta crítico elegir el mejor camino posible en cada momento del recorrido. [3]

Ejemplo de su figura vectorial:



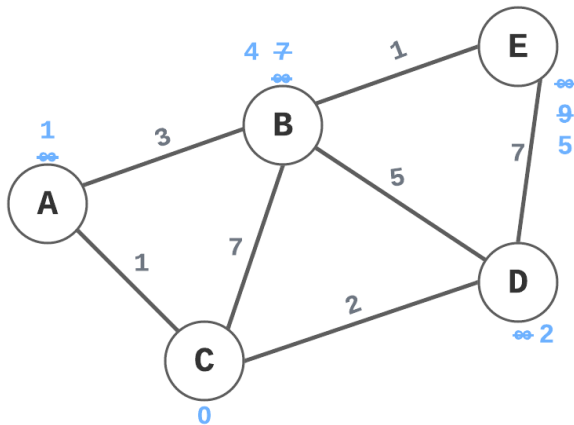
#### 3.2.3 Dijkstra

El algoritmo de Dijkstra. También llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del

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

camino más corto dado un vértice origen al resto de vértices en un grafo con pesos en cada arista. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959. [4]

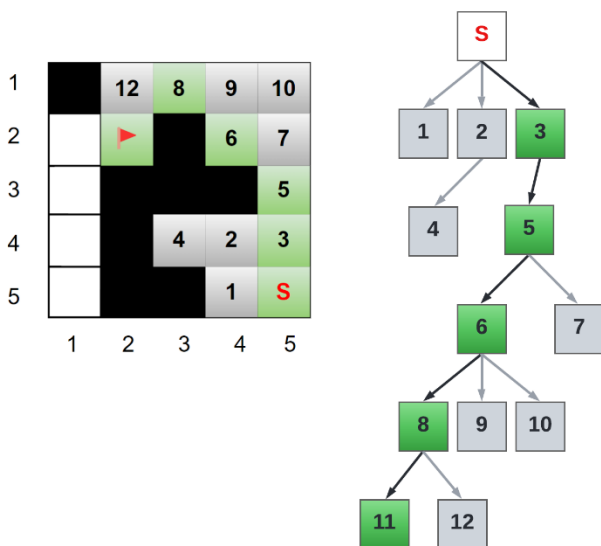
Ejemplo de su figura vectorial:



### 3.2.4 A\*

El algoritmo A\* es un algoritmo de búsqueda que puede ser empleado para el cálculo de caminos mínimos en una red. Se va a tratar de un algoritmo heurístico, ya que una de sus principales características es que hará uso de una función de evaluación heurística, mediante la cual etiquetará los diferentes nodos de la red y que servirá para determinar la probabilidad de dichos nodos de pertenecer al camino óptimo. [5]

Ejemplo de su figura vectorial:



## REFERENCIAS

1. Ruta N. Reducir el acoso callejero: el reto de la secretaría de mujeres. Recuperado de <https://www.rutanmedellin.org/es/programas-vigentes/2-uncategorised/592-reto-de-mujeres>
2. López, M. 2020. DFS vs BFS. Recuperado de Encora <https://www.encora.com/es/blog/dfs-vs-bfs>
3. López, M. 2020. DFS vs BFS. Recuperado de Encora <https://www.encora.com/es/blog/dfs-vs-bfs>
4. EcuRed. Algoritmo de Dijkstra. Recuperado de [https://www.ecured.cu/Algoritmo\\_de\\_Dijkstra](https://www.ecured.cu/Algoritmo_de_Dijkstra)
5. IDELab. Algoritmo A\*. Recuperado de <http://idelab.uva.es/algoritmo>