# **Uso de Algoritmos de Búsqueda de Rutas para la Prevención de Acoso Sexual**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Daniel Palacio  Universidad Eafit  Colombia  Correo electrónico en Eafit | Miguel A. Martínez  Universidad Eafit  Colombia  mamartinef@eafit.edu.co | Pablo Micolta López  Universidad Eafit  Colombia  pmicoltal@eafit.edu.co | Andrea Serna Universidad Eafit Colombia asernac1@eafit.edu.co | Mauricio Toro  Universidad Eafit  Colombia  mtorobe@eafit.edu.co |

# **RESUMEN**

En este trabajo se va a intentar responder al problema de como encontrar la ruta más eficiente para evitar el acoso callejero, en la sociedad actual, estas forma asalto afecta a una gran cantidad de personas alrededor del mundo no solo de manera física, tanto a la violencia que viene relacionada con estos casos sino también psicológica con el trauma que estos eventos dejan en la victima y por lo tanto en afán de prevenir que estas atrocidades afecten a más personas está en el interés común dar solución a este problema, para lograr esto nos apoyaremos de trabajos de búsqueda de camino más corto, búsqueda del camino más seguro y búsqueda de camino con menor criminalidad, hemos encontrado que los caminos con menor riesgo siguen superando 10 puntos de riesgo en todos los casos revisados y centrarse en la reducción de distancias esta son significativamente menores a los presentados por el otro algoritmo (el centrado en riesgo) todos estos resultados se lograron manteniendo los tiempos de ejecución cercanos a 1 minuto

## **Palabras clave**

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

# **INTRODUCCIÓN**

Las tecnologías de la información a causa de su fácil acceso han generado un gran cambio a lo largo de la sociedad actual, todo se ha “acelerado” de manera notoria comparado con los tiempo antes del invenciones como el internet, un ejemplo de esto es el éxito que aplicaciones como “Waze” han visto en los últimos años aplicaciones cuya única función es encontrar rutas rápidas para llegar de un sitio a otro, pero estas aplicaciones tienen un gran problema y es que la ruta más rápida no siempre es la más segura en específico cuando viene acoso sexual que puede ser una práctica muy común en algunas comunidades, practica que afecta a las víctimas a un nivel tanto físico como mental con el fin de controlar esta enfermedad social es importante no tener en cuenta únicamente la velocidad a la hora de calcular una ruta sino que también las estadísticas relacionadas a este problema.

# **1.1. Problema**

El problema a afrontar es la falta a la hora de tener en cuenta estadísticas de problemas tales como el acoso sexual callejero en el proceso de encontrar las rutas más cortas, esto va a servir para asegurar la seguridad de los miembros de la sociedad (sobre todo las mujeres) con esto se busca mejorar la calidad de vida de los ciudadanos y minimizar la posibilidad de que sean víctimas de estos actos que tienen un gran impacto físico y mental

**1.2 Solución**

Para la solución del primer problema se decidió utilizar el algoritmo de Dijkstra pues algoritmos recursivos generarían problema por la magnitud del grafo y Dijkstra en especifico nos permite un fácil acceso desde cualquier punto del mapa hasta cualquier otro, por más “recóndita” que sea la ruta mientras mantiene en cuenta los parámetros que se dieron lo que nos permite prepararnos para cualquier consulta del usuario

**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.

## **3.1 Identificación de Crimen Potencial Búsqueda de Caminos Usando Proceso Jerárquico Analítico (AHP) en Prevención de Crimen Situacional**

En este trabajo a raíz de la popularidad de las tecnologías de computación personal tales como son los teléfonos inteligentes se buscó realizar un método para buscar rutas que cumplan con las necesidades específicas al peatón pues este es más vulnerable a factores externos que los conductores, sobre todo se centraron en usar indicadores propios de la prevención de crímenes situacional para encontrar la ruta más segura con el fin de reducir las oportunidades y dificultar la acción de los criminales, para esto usaron un modelo de decisión multi criterio (MCDM por sus siglas en inglés) con 3 categorías distancia entre peatones y lugares de crimen, visibilidad del criminal y frecuencia de crímenes en la zona, en base a esto realizaron un proceso de análisis jerárquico para encontrar la mejor ruta. Al poner a prueba estos algoritmos en Gong Badak se concluyó que los aspectos que afectan las rutas más satisfactorias son extremadamente objetivo y que el algoritmo usado podría ser ajustado para adaptarse a esta subjetividad [1]

## **3.2 Creación de rutas consiente de seguridad para turistas monitorizados basado en open data y VGI**

En este trabajo se presenta una forma de creación de rutas evitando los lugares de relativo peligro la información de lo peligroso que son las zonas urbanas se basa en el uso de información geográfica voluntariada (VGI) e información abierta (Open Data) gubernamental para detectar propiedades y sus funcionalidades de la infraestructura urbana e información de crimen histórico complementándola con información de la policía analizándolo con GIS, en base de esta información se propuso un índice de seguridad (usado para el peso de los arcos) usando lo lugares de crimen como obstáculos. En el estudio realizado en la ciudad de Los Ángeles (LA) se vio en primer lugar una diferencia al comparar las rutas propuestas por el algoritmo por las propuestas por un algoritmo basado únicamente en distancia y a su vez se especula que podría existir una diferencia entre las rutas mostradas por el algoritmo si se mide de día y de noche (pero el experimento solo se realizó en la noche), pero este modelo se beneficiaria de modelos más exactos de extracción de datos referentes al crimen [2]

## **3.3 Integrando seguridad y movilidad para búsqueda de rutas usando Big Data generada por vehículos conectados**

## Este articulo busca una manera de incluir seguridad a la hora de encontrar una ruta esto a través del desarrollo de la creación de un marco metodológico que integre la movilidad y seguridad a la hora de la búsqueda de ruta, para esto usaron Big Data e información sobre el tráfico en tiempo real para calcular tanto la seguridad y el tiempo de viaje, la seguridad es calculada en base a la volatilidad de los conductores el historial de choques y la volatilidad de velocidad y aceleración, esto junto con la distancia crea una función a partir del cual se calcula la ruta haciendo un algoritmo flexible y cambiante dependiendo de la situación actual, al final se concluyó no siempre la ruta que es recomendada es vista como la mejor por el usuario, además los datos con los que este programa está trabajando son sensible a cambio a lo largo del día pero aun con esto los datos que fueron analizados fueron consistentes.[3]

## **3.4 Un Sistema de Información Basado en Datos Munlti-Sansados e Información Oficial en Crimen para Encontrar Rutas Seguras: Estudio de Caso de la Ciudad De México**

Este trabajo busca solucionar la escasez de soluciones centradas en seguridad o estadísticas criminales, buscan presentar una posibilidad para generar rutas definidas por estas estadísticas, esta solución integra información censada por el público e información oficial, estos datos son procesadas semánticamente y clasificada por el algoritmo Bayes, un repositorio geoespacial fue usado para almacenar tweets relacionados con crimen en la ciudad de México y los reportes oficiales fueron geo codificados para obtener rutas seguras y finalmente se usó unas predicciones de posible crimen en la zonas, la frecuencia de crimen será usada como peso al escoger la ruta y luego los valores se asignaban a nodos correspondientes, también la probabilidad y la hora que vino de la predicción de crimen se procesa y se guarda en una tabla vectorial, al aplicar este algoritmo en la ciudad de México se concluyó que este método tiene una efectividad de 75% con esto se concluyó que el modelo propuesto puede no solo encontrar rutas seguras sino también adaptarse a la opinión de los usuarios. [4]

## **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 Alternativas de camino más corto con restricciones**

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

**3.2.1 CSP**

Existen varias versiones restringidas del problema básico del SP, que consiste en que el camino puede estar restringido para incluir nodos específicos, para incluir un numero especifico de nodos o incluir nodos dentro de una distancia de cobertura preespecifica de cada nodo en la red. En base a este problema llega el algoritmo mejorado CSP que permite demostrar que instancias de problemas muy grandes (40000 nodos y 800000 arcos) pueden resolverse en un tiempo razonable, obteniendo ventajas sobre otros métodos en términos de tiempo de solución y requisitos de memoria del ordenador.

Una solución a un CSP es una asignación (a1, a2,…,an) de valores a todas sus variable, de tal manera que se satisfagan todas las restricciones del CPS, es decir, una solución es una tupla consistente que contiene todas las variables del problema. Una solución parcial es una tupla consistente que contiene alguna de las variables del problema, un CSP es consistente, si tiene al menos una solución, es decir, una tupla consistente. Además, de asume que dos CSPs son equivalentes si ambos representan el mismo conjunto de soluciones.[5]

Diagrama

Descripción generada automáticamente

**3.2.2 DFS**

Una búsqueda de profundidad (DFS) consiste en un algoritmo que nos permite buscar a partir de un recorrido de nodos de un grafo, con su funcionamiento de ir expandiendo cada uno de los nodos que va localizado, de forma recurrente (desde el nodo padre hasta en nodo hijo). Además, cuando ya no quedan más nodos que visitar en dicho camino, regresa el nodo procesador, de modo que repite el mismo proceso con cada uno de los vecinos del nodo, y cabe resaltar que, si se encuentra el nodo antes de recorrer todos los nodos, queda concluida la búsqueda.

En la mayoría de los casos, este algoritmo se utiliza cuando queremos probar si una solución entre varias posibles cumple con ciertos requisitos. Como puede suceder en algún problema donde tengamos que recorrer un tablero para recorrer todas sus casillas.

Para verificar su complejidad debemos tener en cuenta que cada vértice se lo mira a lo sumo una vez, y en una visita al nodo se lo agrega/saca de la pila, y se hacen operaciones de tiempo constante, por lo que tenemos O(V) tiempo para los vértices.

Diagrama

Descripción generada automáticamente

Como ejemplo, tenemos un grafo no conectado, con ocho nodos, donde las flechas naranjas indican el recorrido del algoritmo DFS sobre los nodos.[6]

**3.2.3 BFS**

Una búsqueda en anchura (BFS) es un algoritmo de búsqueda que nos ayuda a recorrer los nodos de un grafo, comenzando desde la raíz, de manera que eligiendo un nodo como elemento raíz en el caso de un grafo, para luego explorar todos los vecinos de este nodo. Además, para cada uno de sus vecinos se tienen que explorar sus respectivos vecinos adyacentes, y así hasta que se recorra el grafo, y si se encuentra el nodo antes de recorrer todos los nodos, queda como concluida la búsqueda.

Este algoritmo se utiliza para resolver problemas donde resulta critico elegir el mejor camino posible en cada momento del recorrido.

Diagrama

Descripción generada automáticamente

La complejidad del algoritmo es O(|V|+|E|), donde |V| es el número de vértices y |E| es el número de aristas, y como en el peor de los casos, cada vértice y cada arista será recorrido por el algoritmo.[6].

**3.2.4 A\***

Este algoritmo de búsqueda es empleado para el cálculo de caminos mínimos en una red. Una de sus principales características es que hará uso de la función evaluación heurística, mediante la cual se encargará de etiquetar los diferentes nodos de la red y debido a esto, ayudará a determinar la probabilidad de dichos nodose n pertenecer al camino más optimo.

La función de evaluación que etiquetara a los nodos de la red esta compuesta a su vez por otras dos funciones, siendo asi una de ellas, la indicación de la distancia actual desde el nodo origen hasta el nodo a etiquetar, y la otra expresara la distancia estimada desde este nodo a etiquetar hasta el nodo destino, hasta el que se pretende encontrar un camino mínimo.[7]

Diagrama

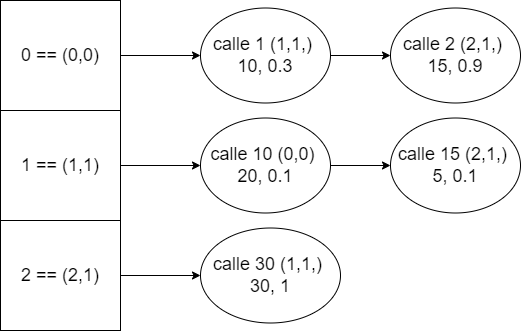
Descripción generada automáticamente

## **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**

A cada coordenada que se presento en el mapa se le asigno una ID la cual se guardo en un diccionario que luego se asocio a una lista que contenía todas la conexiones que tiene ese lugar incluyendo las coordenadas de llegada, el nombre de la ruta, la distancia y el riesgo, la razón de esto es por que nos permite un fácil acceso a cada una de las rutas a partir de su coordenada de inicio, cosa necesaria para la aplicación de los algoritmos de ruta mas corta, así podremos recorrer las rutas posibles de manera mas rápida



**Figura 2:** Un ejemplo de nuestra representación del mapa de Medellín usando diccionarios y listas enlazadas

**4.2 Algoritmos**

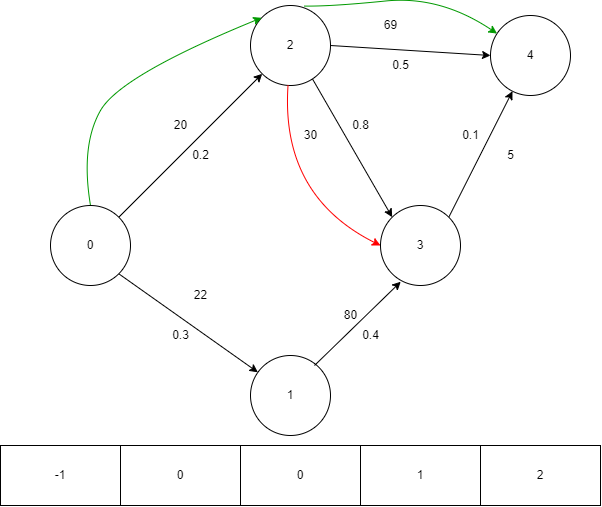
En este trabajo, proponemos algoritmos para el problema del camino más corto restringido. El primer algoritmo calcula el camino más corto sin superar un riesgo medio ponderado de acoso *r*. El segundo algoritmo calcula el camino con el menor riesgo medio ponderado de acoso sin superar una distancia *d*.

**4.2.1 Primer algoritmo**

**El algoritmo** Dijkstra es un algoritmo voraz que funciona revisando en cada nodo que visita cual es el siguiente paso con menor peso, en nuestro caso estamos añadiendo la restricción de que si se sobrepasa un cierto nivel de “peligro” se desvíe a otra opción esto hasta tener una ruta desde el nodo de inicio hasta todos los otros nodos accesibles desde este, la ruta se guarda en un arreglo donde cada nodo tiene como valor el nodo previo a el en la ruta teniendo el nodo inicial un valor de -1.

Este algoritmo tiene la ventaja de no ser recursivo, lo que evitara que, en un grafo de muchos nodos, no se sobrecargue la cantidad de búsquedas que se hacen, además, como tiene una ruta desde el nodo inicial hasta cualquier otro nodo nos asegura que siempre hay una ruta disponible que

Una desventaja es que el algoritmo es muy demandante en recursos lo que hace que no cualquier equipo pueda ejecutarlo.



**Figura 3:** Resolución del problema del camino más corto restringido con la Búsqueda Primera Profunda (DFS). (Por favor, siéntase libre de cambiar esta figura si utiliza un algoritmo diferente).

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

El peor de los casos para tanto memoria como tiempo es que todos los estén conectados con todos los otros, en este caso la estructura debe guardar todos los nodos en la lista de los otros nodos a esto se añaden la cantidad de aristas que se presentan (pues estas también deben ser almacenadas) y para el tiempo el algoritmo deberá revisar cada nodo cada vez que aparezca en los sucesores de un nodo de ahí los resultados mostrados en las siguientes tablas

|  |  |
| --- | --- |
| **Algoritmo** | **Complejidad temporal** |
| Dijkstra | O(V2) |

**Tabla 1:** Complejidad temporal del Dijkstra, donde V es los vértices del grafo

|  |  |
| --- | --- |
| **Estructura de datos** | **Complejidad de la memoria** |
| Dijkstra | O(V2E) |

**Tabla 2:** Complejidad de memoria del arreglo de listas, donde V es los vértices del grafo E es las aristas

**4.5 Criterios de diseño del algoritmo**

En algoritmo se desarrollo de esta manera, utilizando métodos voraces porque si se hace por la ruta de fuerza bruta entonces la cantidad de combinaciones que se crearían seria tal que, nuestros computadores no podrían procesarlas, pero, aunque pudieran, este proceso tomaría una gran cantidad de tiempo, y por backtraking la cantidad de revisiones que tendría que hacer es extremadamente alta, por lo tanto, lo mas seguro es que genere un error a causa de una sobrecarga de la pila. Por lo tanto, los métodos que nos permiten una menor cantidad de revisiones son los únicos viables para un problema tan masivo en cuanto a tamaño, y en tanto al porque se escogió a Dijkstra, dado que, dentro de los algoritmos voraces, hay una mayor simpleza a la hora de añadir restricciones y da una facilidad de acceso a diferentes caminos posibles.

**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 | 6914 | 100 |
| Universidad de Antioquia | Universidad Nacional | 2698 | 45 |
| Universidad Nacional | Universidad Luis Amigó | 556 | 10 |

**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 | 95.6 | 8000 |
| Universidad de Antioquia | Universidad Nacional | 30.7 | 3800 |
| Universidad Nacional | Universidad Luis Amigó | 7.16 | 565 |

**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.

Calcule el tiempo de ejecución de las consultas presentadas en la Tabla 3. Indique los tiempos de ejecución medios.

## 

|  |  |
| --- | --- |
|  | **Tiempos medios de ejecución (s)** |
| Universidad EAFIT a Universidad de Medellín | 61.71 s |
| De la Universidad de Antioquia a la Universidad Nacional | 54.19 s |
| De la Universidad Nacional a la Universidad Luis Amigó | 51.09 s |

## **Tabla 4:** Tiempos de ejecución de Dijkstrapara las consultas presentadas en la Tabla 3.

## **6. CONCLUSIONES**

Efectivamente los caminos propuestos por ambos algoritmos son bastamente diferentes siendo que los que se centraron a reducir el acoso son más largos y contienen una mayor cantidad de nodos, este tipo de aplicaciones son de extrema utilidad para los transeúntes por la cantidad de violencia y asaltos callejeros que se presentan, lamentablemente para que este proyecto resulte en una aplicación mas practica deben mejorarse los tiempos de ejecución

**6.1 Trabajos futuros**

Para trabajos en el futuro, sería una posible mejora reducir el tiempo de ejecución, esto se podría lograr utilizando en lugar de simples listas enlazadas, utilizar colas de prioridad, además de una posible aplicación web y funcionalidades para el aporte de los usuarios; Este tipo de cambios y programas podrían ser útiles en otros campos y materias como lo sería la estadística o inclusive en otras formas de estudio social.

# **AGRADECIMIENTOS**

Agradecemos la ayuda con técnica particular y asesoría general a Kevin Alejandro Sossa Chavarría, Monitor, EAFIT por los comentarios que mejoraron en gran medida este manuscrito.

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 Wan Mohd F. Bin Wan, Noor Maizura and Masta A. Jalil. Identification of Potential Crime Tactical Path-Finding Using Analytical Hierarchy Process (AHP) in Situational Crime Prevention in *The 7th International Conference on Information Technology*(Hangzhou, China, 2021), ResearchGate.

2. Andreas Keler and Jean D. Mazimpaka. Safety-aware routing for motorised tourists based on open data and VGI.   
*Journal of Location Based Services***,** 10 (1). 64-77

3. Nima Hoseinzadeh, Ramin Arvin, Asad J. Khattak and LeeD. Han. Integrating safety and mobility for pathfinding using big data generated by connected vehicles. *Journal of Location Based Services*, 24(4), 404-420

4. Félix Mata, Miguel Torres-Ruiz, Giovanni Guzmán, Rolando Quintero, Roberto Zagal-Flores, Marco Moreno-Ibarra, Eduardo Loza, "A Mobile Information System Based on Crowd-Sensed and Official Crime Data for Finding Safe Routes: A Case Study of Mexico City", *Mobile Information Systems*, vol. 2016, Article ID 8068209, 11 pages, 2016. <https://doi.org/10.1155/2016/8068209>

5. Miguel A. Salido and Federico Barber. Inteligencia Artificial: Técnicas, métodos y aplicaciones, McGraw-Hill Interamericana, , 2008

6. Miguel López DFS vs BFS. Recuperado 19 de febrero de 2022 de <https://www.encora.com/es/blog/dfs-vs-bfs>

7. Instituto Tecnológico de Nuevo Laredo. Algoritmo A\*

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)