

RELACIÓN RUTA MÁS CORTA – ÍNDICE DE ACOSO: ALGORITMOS PARA REDUCIR EL ACOSO CALLEJERO EN MEDELLÍN

Miguel Sosa Villegas Universidad Eafit Colombia msosav@eafit.edu.co	Sergio A. Córdoba M. Universidad Eafit Colombia sacordobam@eafit.edu.co	Miguel Jaramillo A. Universidad Eafit Colombia mjaramil20@eafit.edu.co	Andrea Serna Universidad Eafit Colombia asernacl@eafit.edu.co	Mauricio Toro Universidad Eafit Colombia mtorobe@eafit.edu.co
--	--	---	--	--

RESUMEN

El problema o tema central del proyecto se basa en reducir el acoso callejero que se presenta en Medellín, por medio de algoritmos que encuentren el camino más corto con relación al acoso. Es importante resolverlo porque esta forma de agresión impide el desarrollo normal de la vida de las personas afectadas.

Del acoso callejero se deriva el miedo a salir nuevamente a las calles, miedo a socializar con las personas, incertidumbre e inseguridad. Es por las razones anteriormente mencionadas que es de suma importancia para nosotros encontrar soluciones eficientes para combatir este fenómeno.

Consideramos que el algoritmo de Dijkstra es una de esas soluciones eficientes, ya que nos permite hallar el camino más corto desde un vértice o nodo origen, hacia los demás vértices o destinos; cada arista entre nodos comprende un valor que representa su peso entre cada uno de ellos..

Básicamente la idea es explorar todos los caminos posibles desde un origen hasta un destino y brindarles a las personas un ruta tanto segura y eficiente en la que se sientan seguros al transitar hacia su destino final.

Palabras clave

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

1. INTRODUCCIÓN

Hoy en día, en cuanto al tema del acoso se trata; lo que se espera es que los niveles de este tipo de agresión tiendan a disminuir a medida que avanza el tiempo, para que así todas las personas; tanto afectadas como no, sientan con plena seguridad que pueden salir a las calles sin el temor de recibir ese irrespeto y abuso hacia su integridad física y emocional.

Pero verdaderamente se podría decir que estamos muy lejos de reducir esos números, ya que cada día el incremento de

acoso callejero hacia las personas es mayor por todas partes. En la sociedad que vivimos, ha tomado un auge muy masivo a tal punto que es “normal” ver ese tipo de comportamientos molestos e irreverentes en cada ocasión.

Es por esto que se necesita implementar con urgencia una manera útil a través de la cual las personas puedan evitar a toda costa las rutas con mayor índice de acoso y se movilicen por aquellas que son más efectivas y con menor presencia de esta modalidad. Lo importante es que se sientan seguros a la hora de salir y en todo su trayecto, desde su punto de inicio hasta su punto de destino. Ese es el objetivo de realizar este proyecto.

1.1 Problema

El problema es el acoso en Medellín y cómo evitar zonas donde esto ocurre con más frecuencia a través de algoritmos que hallen rutas con un riesgo medio ponderado y también sin que supere una cierta distancia. Este problema afecta a la sociedad, tanto a hombres como mujeres, ya que ambos pueden ser víctimas de abuso, por lo que hallar solución a este problema podría ayudar a que las personas tomen rutas más seguras donde haya menor probabilidad sufran cualquier caso de abuso.

1.2 Solución

Es debido a las razones anteriormente mencionadas que como solución a la problemática, proponemos la implementación de algoritmos para caminos más cortos restringidos, con el fin de que las personas eviten las rutas o caminos en los que se presencia más acoso callejero. Como parte de la solución, hemos elegido algoritmos tales como Dijkstra, porque nos permite calcular todos los caminos partiendo desde un punto hasta los demás. Una vez se ha encontrado el más corto procede a retornarlo.

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”

Siempre Seguras es una aplicación creada por un grupo de mujeres, todas con algo en común: habían sufrido de acoso sexual callejero en México. Debido a la alta incidencia de acoso callejero que recibían las mujeres frecuentemente en la ciudad de Querétaro se implementó esta estrategia para mapear el acoso en todas las zonas de la ciudad.

Lo más interesante de este proyecto es cómo estas mujeres implementan el algoritmo para conocer en qué zonas se evidencia mayor movimiento. Angelina se encargó de crear un algoritmo que descarga los tuits de mujeres relacionados con acoso o violencia. Se le ocurrió esta idea ya que las personas suelen compartir en las redes sociales sus experiencias vividas. Por medio de estos tuits se localiza su ubicación y así se crea un mapa que se encarga de evidenciar en qué zonas se genera mayor acoso.

Como resultado de este proyecto, se consiguió el apoyo de la Secretaría de Desarrollo Social del Estado de Querétaro y el Instituto Municipal de las Mujeres para dar a conocer esta problemática en el país y así prevenir a cuantas más personas como sea posible.

2.2 bSafe

bSafe es una aplicación para teléfonos móviles, que usa la ubicación en tiempo real de la persona, para que su familia o amigos la puedan localizar.

Lo que más llama la atención de este proyecto son las funcionalidades que este tiene integradas, una, la más útil, es que en caso de que la persona sea víctima de acoso, el decir una palabra clave, aún con el celular en el bolsillo, va a hacer que se mande una señal de emergencia a los contactos que dicha persona ha elegido junto con la ubicación de la persona, además automáticamente se empezará a grabar el audio y el video del dispositivo móvil, y, aunque el agresor destruya el teléfono móvil, esta grabación no será destruida.

Otra de las funcionalidades que este tiene es que le permite al usuario poner un tiempo estimado de llegada a los lugares que vaya, así si este se demora más de lo que puso, se mande una señal de emergencia a los contactos.

Lo que esta aplicación resuelve es la inseguridad que las personas pueden tener al caminar solas, por miedo que les roben o las acosen.

2.3 Viomapp

Viomapp es una aplicación que muestra las agresiones y el acoso callejero que sufren las mujeres en tiempo real. Esta aplicación fue lanzada por Joaquín Vázquez, un ingeniero que trabajó para aplicaciones de la Consejería de Justicia e Interior de la Junta de Andalucía. Esta app nace luego de que múltiples amigas de él le comentaran que fueron víctimas de violencia de género y de acoso callejero.

Lo que más resalta de esta aplicación es que es muy fácil abrir incidencias, es decir describir en qué lugar fuiste víctima de acoso y esto automáticamente se actualiza en tiempo real. Además, es totalmente anónimo y permite contactarse con la persona por medio de la aplicación en caso de que seas un testigo visual y quieras corroborar cierto evento o también se pueden contactar para brindar apoyo psicológico.

En caso de que una mujer quiera llegar a cierta parte, puede consultar su ruta y la aplicación le mostrará si la zona es segura o si debe de tener cuidado. Así sabrán de antemano si la ruta que tenían pensada si era la más adecuada y segura.

Esta aplicación ayuda tanto a saber si la ruta que deseas ir es la más segura y además permite hacer reportes en tiempo real de abuso o acoso.

2.4 Safe & the city

Safe & the city es una aplicación creada en el 2018 por Kowalchuk una canadiense que al llegar a Londres fue víctima de un incidente donde fue acosada verbalmente en un callejón. Desde ahí empezó a desarrollar la app que usa datos de la policía para saber las zonas con más índice de abuso y acoso.

Entre las funcionalidades de la aplicación se encuentran, un algoritmo que encuentra la mejor ruta y notifica si la zona es conocida por crímenes o abuso, todo esto utilizando datos de la policía. Además, tiene un botón de emergencia que automáticamente llama a la policía. También, toda la información que brindan los usuarios va a ser guardada anónimamente y ayudarán para reconocer patrones de crímenes en la ciudad.

Esta aplicación ayuda a las mujeres a estar más seguras y saber en qué zonas es más peligroso transitar, y ayuda a contactar a la policía de una manera más rápida.

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 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.

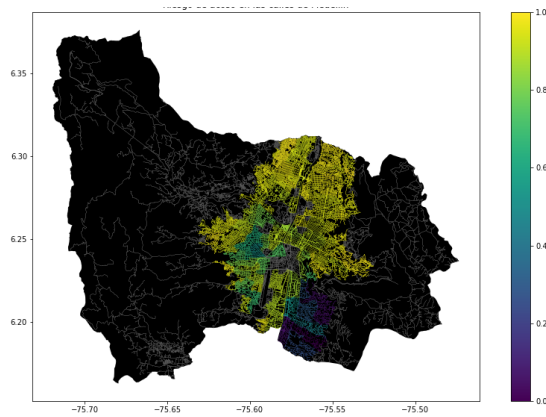


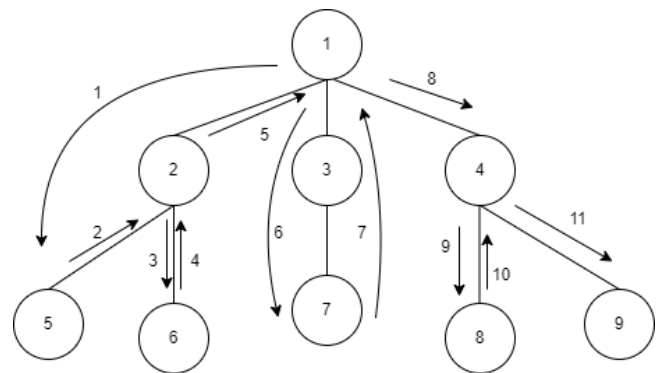
Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenida de la Encuesta de Calidad de Vida de Medellín, de 2017.

3.2.1 Algoritmo DFS

El algoritmo DFS se encarga de recorrer todos los vértices desde un origen o raíz. Funciona de tal forma que primero recorre los hijos de la raíz, luego procede a recorrer los

hijos de aquellos nodos hasta que completa el camino. Una vez hecho, se devuelve o se lleva a cabo lo que denominamos “backtracking” para seguir recorriendo los otros caminos formados a partir de los hijos de la raíz.

En la búsqueda de un elemento particular se puede dar el caso de que recorra todas las posiciones comparando los elementos hasta encontrar o no el indicado. Su complejidad está dada por: $O(V + E)$ donde “V” representa los vértices del grafo y “E” representa las aristas; esto debido a que para saber los vecinos de un vértice es necesario hacer un recorrido lineal. Por eso se suma vértices y aristas.



3.2.2 Algoritmo BFS

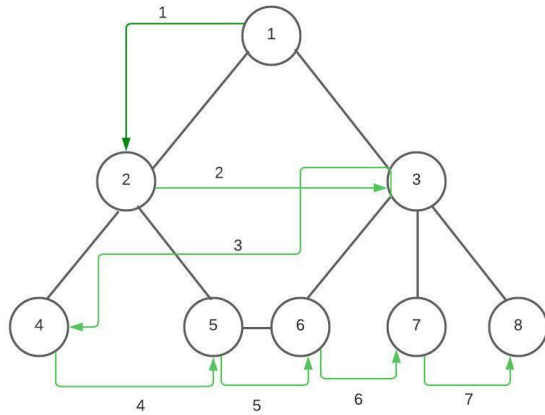
La búsqueda en anchura o BFS, es un algoritmo de búsqueda que recorre todos los nodos de un grafo, este comienza desde la raíz y luego va pasando a los nodos vecinos. Luego, para cada vecino se va explorando los nodos adyacentes y esto se repite hasta recorrer todo el grafo.

Si el grafo está representado como una lista de adyacencia la complejidad es $O(V + E)$ y si el grafo está representado como una matriz de adyacencia la complejidad es $O(V * V)$, siendo V el número de vértices y E la cantidad de arcos. Esto ya que en una lista de adyacencia ya sabemos los que son estrictamente adyacentes al vértice por lo que no hay que recorrer todos.

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

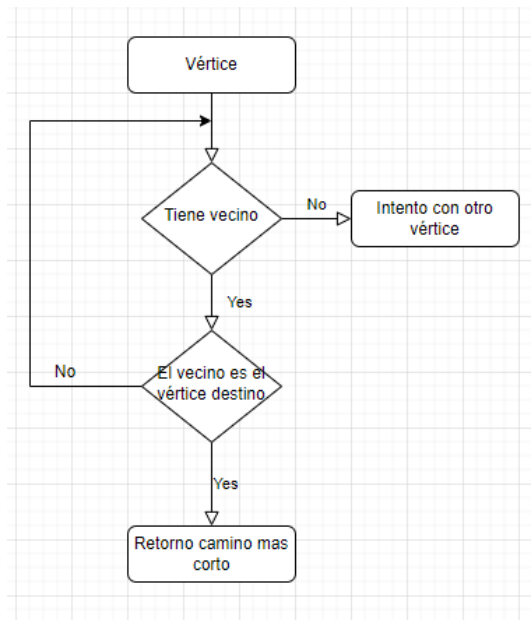
³ <https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets/>



3.2.3 Dijkstra

El algoritmo de Dijkstra lo que hace es encontrar el camino más corto entre un vértice y otro cuando las aristas tienen pesos positivos. Este explora todos los posibles caminos de un vértice a otro, hasta que encuentra el más corto y el algoritmo se detiene.

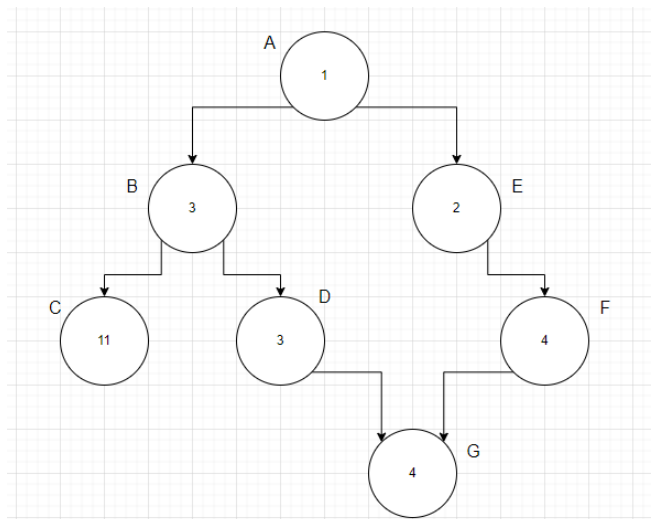
La complejidad que este algoritmo tiene es de $O(V^2)$ sin utilizar cola de prioridad, es decir, si no se priorizan algunos elementos antes que otros, y con cola de prioridad la complejidad se vuelve $O(A \log V)$.



3.2.4 A*

El A* (A estrella) se basa en encontrar el camino más óptimo de un nodo a otro, usando una heurística óptima, un ejemplo de esto puede ser cuando se viaja entre dos ciudades, el recorrido puede ser mayor o igual a su distancia en línea recta, esta última es la heurística óptima, es decir, el caso óptimo entre los dos nodos.

Hablando de su complejidad, esta llega a ser exponencial debido a que tiene que evaluar todos los siguientes nodos de cada estado y así sucesivamente, depende de una buena heurística para que demore menos.



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 se utilizó fue una lista enlazada, ya que permite un mejor manejo en temas de memoria a la hora de usar un gran número de vértices. Ya que al usar una matriz de adyacencia con un número elevado de vértices podemos presentar problemas a la hora de su creación y además consume más memoria que las listas de adyacencia.

⁴<https://github.com/msosav/ST0247-002/tree/master/proyector>

(a)

(b)

4.2 Algoritmos

4.2.1 Primer y Segundo algoritmo

4.4 Análisis de la complejidad de los algoritmos

Por esto se concluye con que la notación de la complejidad del algoritmo es de $O(V^2)$ como se evidencia en la tabla 1.

Algoritmo	Complejidad temporal
Dijkstra	$O(V^2)$

Estructura de datos	Complejidad de la memoria
Lista enlazada	$O(V^2 * E)$

4.5 Criterios de diseño del algoritmo

Esto es de gran utilidad, debido a que con otras estructuras de datos, se pueden crear demasiadas combinaciones o procesos que hacen que el programa no sea tan efectivo y no funcione correctamente en la práctica.

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

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

Origen	Destino	Distancia	Sin
--------	---------	-----------	-----

		más corta	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

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		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

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	
De la Universidad de Antioquia a la Universidad Nacional	
De la Universidad Nacional a la	

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

Tabla 4: Tiempos de ejecución del algoritmo Dijkstra para las consultas presentadas en la Tabla 3.

6. CONCLUSIONES

Explique los resultados obtenidos. ¿Son los caminos más cortos significativamente diferentes de los caminos con menor riesgo de acoso? ¿Qué utilidad tiene esto para la ciudad? ¿Son razonables los tiempos de ejecución para utilizar esta implementación en una situación real?

6.1 Trabajos futuros

A futuro nos gustaría convertir este proyecto en una realidad; una aplicación web con la que los usuarios verdaderamente puedan interactuar y hacer uso de ella en su cotidianidad porque este es un problema que se evidencia en todo el mundo.

Optimizar este proyecto tanto como sea posible es una buena idea para nosotros, debido a que así se garantiza una respuesta efectiva en el menor tiempo posible y los usuarios no tendrían que esperar tanto por una respuesta por parte de la aplicación.

En cuanto al campo de la estadística, este proyecto sería muy útil en el caso de que se quiera hacer una recopilación en tiempo real de cuáles son las rutas que presentan mayor índice de acoso en una hora determinada. Lo que implica que las personas van a estar al tanto de en cuáles rutas pueden o no dirigirse hacia su destino.

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. Forbes Staff. 2021. Siempre Seguras: la app para mapear el acoso sexual callejero en México. *Forbes México*. Retrieved February 20, 2022 from <https://www.forbes.com.mx/tecnologia-siempre-se-guras-app-mapear-acoso-sexual-callejero-mexico/>
2. Xiana Siccardi. 2018. Nace Viomapp, una aplicación que muestra a tiempo real las agresiones y el acoso callejero a las mujeres. *La Vanguardia*. Retrieved February 20, 2022 from <https://www.lavanguardia.com/tecnologia/2018042/442694999230/nace-viomapp-aplicacion-muestra-a-real-agresiones-acoso-callejero-mujeres.html>
3. 2022. *Safeandthecity.com*. Retrieved February 20, 2022 from <https://www.safeandthecity.com/faq/>

4. BBC News Mundo. (2016, 14 noviembre). Las tecnologías que puedes usar para sentirte más seguro por la calle. Recuperado 20 de febrero de 2022, de <https://www.bbc.com/mundo/noticias-37950359>
5. bSafe. (s. f.). bSafe. Recuperado 20 de febrero de 2022, de <https://www.getbsafe.com/>
6. Mokhtar Ebrahim. 2020. Algoritmo Depth First Search en Python (múltiples ejemplos). *Like Geeks*. Retrieved February 21, 2022 from <https://likegeeks.com/es/algoritmo-depth-first-search/>
7. Algoritmo de Dijkstra. (2022, 6 enero). En Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra#Complejidad
8. Cola de prioridades. (2020, 19 abril). En Wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Cola_de_prioridades
9. Instituto Tecnológico de Nuevo Laredo. (2005). ALGORITMO A*. [http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/A-Star/A-Star\(2005-II-A\).pdf](http://www.itnuevolaredo.edu.mx/takeyas/apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/A-Star/A-Star(2005-II-A).pdf)
10. UNIVERSIDAD MAYOR DE SAN SIMÓN FACULTAD DE CIENCIAS Y TECNOLOGÍA. (2019). INTELIGENCIA ARTIFICIAL (BÚSQUEDA A ESTRELLA). http://www.cs.umss.edu.bo/doc/material/mat_gral_139/Busqueda%20A%20estrella-2.pdf
11. 2012. Búsqueda en amplitud primero. *Wikioes.icu*. Retrieved February 21, 2022 from https://wikioes.icu/wiki/Breadth-First_Search
12. Shortest Path in Maze using Backtracking. (s. f.). Pencil Programmer. Recuperado 20 de marzo de 2022, de <https://pencilprogrammer.com/algorithms/shortest-path-in-maze-using-backtracking/>
13. 3 FORMAS DE CALCULAR UN PROMEDIO PONDERADO EN PYTHON - BLOG. (s. f.). QUISH. Recuperado 20 de marzo de 2022, de <https://es.quish.tv/3-ways-compute-weighted-average-python>
14. Rodríguez, D. (2021, 24 abril). Pandas: Cómo convertir un Dataframe en una lista de listas por filas o columnas. Analytics Lane. Recuperado 20 de marzo de 2022, de <https://www.analyticslane.com/2021/04/26/pandas-como-convertir-un-dataframe-en-una-lista-de-listas-por-filas-o-columnas/>
15. ¿Qué es mejor, las listas de adyacencia o las matrices de adyacencia para los problemas de grafos en c++? Fuente: <https://www.iteramos.com/pregunta/73860/que-es-mejor-las-listas-de-adyacencia-o-las-matrices-de-adyacencia-para-los-problemas-de-grafos-en-c>. (2010, 7 febrero). magiix. Recuperado 20 de marzo

de 2022, de <https://www.iteramos.com/pregunta/73860/que-es-mejor-las-listas-de-adyacencia-o-las-matrices-de-adyacencia-para-los-problemas-de-grafos-en-c>