Mapping against sexual harassment

Sara Domínguez
Universidad EAFIT
Colombia
sdominguer@eafit.edu.co

Samuel Acosta Aristizábal
Universidad EAFIT
Colombia
sacostaa1@eafit.edu.co

Alejandro Arango
Universidad EAFIT
Colombia
aarangom1@eafit.edu.co

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

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

Red text = Commentsk

Black text = Andrea and Mauricio's contribution

Green text = To complete for the 1st deliverable

Blue text = To complete for the 2nd deliverable

Violet text = To complete for the 3rd deliverable

ABSTRACT

To write an abstract, you should answer the following questions in a single paragraph: What is the problem? Why is the problem important? Which are the related problems? Which is the algorithm you proposed to solve the problem? What QUANTITATIVE results did you achieve? What are the conclusions of this work? The abstract should have **at most 200 words**. (*In this semester, you should summarize here execution times, and results of lowest risk path and shortest path*).

Keywords

Constrainted shortest path, street sexual harassment, secure-path identification, crime prevention.

1. INTRODUCTION

Sexual harassment has been a social problem during decades, and it affects a large percentage of the female population. More exactly, nearly 70% of them has gone through this type of abuse [1]. However, there are three types of sexual abuse: physical, verbal and not verbal. Unfortunately, these actions can lead the victim to experience physical and psychological events that can affect them negatively in many aspects of their lives [2]. For this reason, our intention with this project is to make women feel more safe and secure by walking alone on the streets of Medellín. Furthermore, the way we are going to achieve this purpose is by calculating the shortest path with the lower probability of sexual harassment, recollecting as many data as we can about the number of women abused in a specific place. Thus, we can suggest our users (in this case female population) a path that doesn't exceed the limit of distanced wanted, and, at the same time, with the lowest risk of harassment possible.

1.1. Problem

As said before, sexual abuse has been part of the society for a long time, but what many people doesn't know is that this problem has been increasing for the last years. Beyond, cases of crimes against woman increased by 19% from 2015 to 2019 [4]. Additionally, there are many ways to make a woman feel uncomfortable or insecure, and all of them are denominated as sexual harassment. In fact, any sexual act, unwanted sexual comment or insinuation, the use of physical force, attempts to obtain sex under duress, assault on sexual organs, among many others are considered as abuse to another person. Furthermore, according to statistics made by the ONU, 1 out of 3 women has gone through this type of abuse at least one in their lives, and about 45% and 55% of them were around 15 years old [3]. Consequently, there have been many initiatives against this huge problem women have to live with day to day, but none of them have helped avoiding this situation 100%. Regardless of this, it is important for us to decrease as much as we can the number of cases occurring each day in our city. This way we can prevent many women to go through this situation, which can cause them, as mentioned before, an impact in their professional, physical, psychological, and sometimes sexual life. Finally, we also want to incentivize more women to speak out, since less than 40% of the women that suffer from sexual harassment makes the decision to tell someone, and that may affect them negatively [5].

1.2 Solution

We had two main solutions two create our program that helps to find the shortest path between a place and another with the lowest grade of sexual harassment. Our first one was A*, which we liked a lot since it has a very good time and memory complexity, but was more difficult to code. In the other hand, we had Dijkstra, which was also a great solution and easier to code, so we went for this one. Finally, we also made use of some python libraries which made the work much easier, such as numpy, pandas, pydeck and others.

1.3 Article structure

In what follows, in Section 2, we present related work to the problem. Later, in Section 3, we present the data sets and methods used in this research. In Section 4, we present the algorithm design. After, in Section 5, we present the results. Finally, in Section 6, we discuss the results and we propose some future work directions.

2. RELATED WORK

In what follows, we explain four related works to path finding to prevent street sexual harassment and crime in general.

3.1 Free to be

Free to be is a website created by Monash University with the purpose of helping women to avoid going through places in which others haven't feel save or have had a bad experience. It was the first app created with this purpose since it was launched in 2016. They thought it was a good idea to solve this problem by letting every woman share anonymously with others which places make them feel insecure and in which ones they feel well. Moreover, we don't have any information about the algorithm they used to create the map that supported this website, since they don't share any of this information in their website. Also, this website was launched in many places of the world including Madrid, Sydney, Delhi, Lima, and others. Finally, this website was very successful, since many women began to share their experiences and places which other woman should avoid going, and the outcome was that more women could see other ways to get to many places without suffering from harassment. Otherwise, as this project was launched so many years ago, it is important to consider that the technology at that time wasn't as used ads now, so the accessibility that people had to that website was less than how it would be known. For that reason, this website wasn't that recognized, and even though some women used it, it didn't have the desires scope.

Retrieved from: https://www.plan.org.au/news/youth/free-to-be-a-youth-activists-reflection/

3.2 Purple save

This initiative´s intention was to create an app that shows a map with the exact places where women have been abused in real time in Bogotá-Colombia. They wanted to decrease the cases of abuse in their city by letting women know which route they should take that's safer and shortest. In the other hand, this app also counts with legal advice for their users, which can provide them information about what to do if you are sexually abused. Furthermore, it also has virtual dialogs that can help every victim of abuse to vent and talk to someone without fear. In the other hand, this app was made mainly with Java, but we don't have the exact algorithm that was used to create the map of this app. Lastly, after asking many women about their opinion of this project, they showed a lot of interest on it and agreed that it was a very innovative and necessary project.

Castillo, B., & Quevedo, A. 2020. Aplicación contra el acoso callejero. https://repository.usta.edu.co/handle/11634/28790

3.3 Street harassment mapping

Technology has advanced exponentially, which can be a great ally in addressing violence caused to women. A clear example of this is what the city of Glasgow, Scotland is doing, where its main objective is to make the streets safer for women. How are they doing it? It is an online map software where women collect their data about their experiences of violence, harassment, and abuse on the streets, including stalking, intimidation and sexual assault. This project is led by Wise Women, a community safety net, with the purpose of identifying the main places where harassment incidents occur. They hope to influence politicians and urban planners to make the city a safer space for women citizens. The data collected during 3 months until March 1, 2022, and the initial results will be shared to the public on March 8 (International Women's Day), with this they hope to be extended nationally and even in the future to the United Kingdom.

It should be clarified that mapping or uploading geographic data to create a digital map has already been used to combat street harassment. In 2010, women in Egypt created HarassMap, an application that allows anonymously reporting abuse in public spaces. Finally, we couldn't find the algorithm used for this initiative, since they don´t mention that in any of their publications or sources.

[1]_ONU, 2022. Facts and figures: Ending violence against women. https://www.unwomen.org/es/what-we-do/ending-violence-against-women/facts-and-figures

2 Orell, H. 2022. Esta es la manera en que las mujeres usan la tecnología para poner fin al acoso en las calles. https://www.lanacion.com.ar/el-mundo/esta-es-la-manera-en-que-las-mujeres-usan-la-tecnologia-para-poner-fin-al-acoso-en-las-calles-nid10022022/

Retrieved from: *https://www.wisewomen.org.uk*

4 ElUniverso, 2022. Cómo las mujeres alrededor del mundo están usando la tecnología para poner fin al acoso en las calles. https://www.eluniverso.com/noticias/internacional/como-las-mujeres-alrededor-del-mundo-estan-usando-la-tecnologia-para-poner-fin-al-acoso-en-las-calles-nota/

3.4 ELSA genderLab

GenderLab is seeking to create spaces of comfort for women, these spaces must clearly be free from harassment, which is why they have created ELSA, and app that they describe as a commitment to innovation and technology to recognize and take action against sexual harassment directed towards the labor sector. Elsa is a comprehensive tool that helps companies respond to workplace sexual harassment problems, it does this through self-assessment that lasts only 10 minutes, and with this technology allows early response and improvements that can offer in indicators.

Elsa measures the indicators through the surveys of the work staff, measuring 4 indicators:

1 Tolerance: Explore people's personal thinking about workplace sexual abuse

2 Prevalence: builds based on behaviors identified by victims and witnesses

3 Trust: measures the perception that staff have about commitment to sexual harassment

4 Myths: look for the myths that staff have about sexual harassment in the organization

Finally, we weren´t able to find this apps algorithm.

https://elsa.genderlab.io

Banco Interamericano De Desarrollo, 2021. Tecnología al servicio de la reducción de la violencia contra las mujeres. *https://gestion.pe/blog/bid/2021/02/tecnologia-al-servicio-de-la-reduccion-de-la-violencia-contra-las-mujeres.html/* (CITADO EN ACM)

3. MATERIALS AND METHODS

In this section, we explain how data was collected and processed and, after, different constrained shortest-path algorithm alternatives to tackle street sexual-harassment.

3.1 Data Collection and Processing

The map of Medellín was obtained from Open Street Maps (OSM)[1] and downloaded using Python OSMnx API[2]. The (i) length of each segment, in meters; (2) indication whether the segment is one way or not, and (3) well-known binary representation of geometries were obtained from metadata provided by OSM.

For this project, we calculated the linear combination that captures the maximum variance between (i) the fraction of households that feel insecure and (ii) the fraction of households with income below one minimum wage. These data were obtained from the quality of life survey, Medellín, 2017. The linear combination was normalized, using the maximum and minimum, to obtain values between 0 to 1. The linear combination was obtained using principal components analysis. The risk of harassment is defined as one minus the normalized linear combination. Figure 1 presents the risk of harassment calculated. Map is available at Github[3].

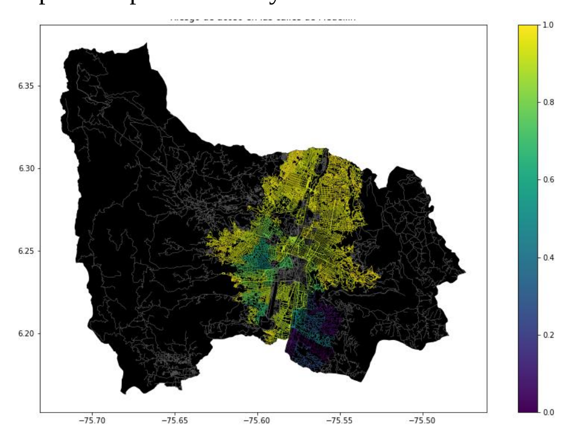


Figure 1. Risk of sexual harassment calculated as a lineal combination of the fraction of households that feel insecure and the fraction of households with income below one minimum wage, obtained from Life Quality Survey of Medellín, in 2017.

3.2 Constrained Shortest-Path Alternatives

In what follows, we present different algorithms used for constrained shortest path. *(In this semester, examples of such algorithms are DFS, BFS, a modified version of Dijkstra, a modified version of A*, among others).*

3.2.1 Shortest path in a binary maze using backtracking

This algorithm works with a matrix with binary numbers, and the intention is to find the shortest path through the “1” numbers from an initial point to an end point. There are only four types of movement:

Go up: (x , y) -> (x – 1 , y)

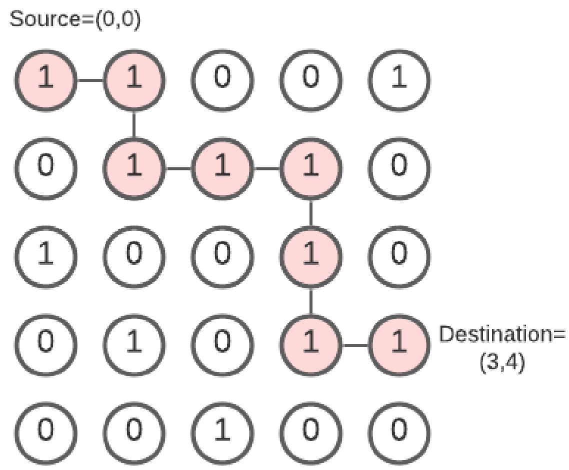
Go left: (x , y) -> (x , y – 1)

Go Down: (x , y) -> (x + 1 , y)

Go Right: (x , y) -> (x , y + 1)

Consequently, the algorithm will start by analyzing each of the four possibilities of movement and recursively checks which one will lead to the destination. Otherwise, if a path doesn’t reach the destination wanted, we will use the backtrack method to see other alternatives. The output of this algorithm will be an integer giving the exact number of “1” you must go over from the starting point to the destination given. Finally, it is also important to keep a track of which paths were already analyzed, this way the software can ignore them when analyzing the rest.

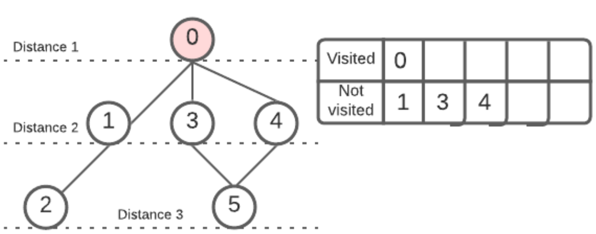
[3]https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets/



[1] https://www.techiedelight.com/find-shortest-path-in-maze/

3.2.2 BFS

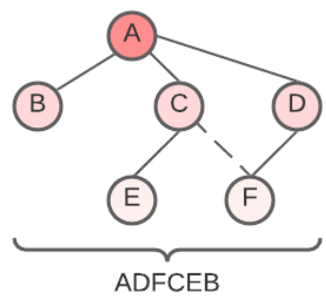
BFS algorithm works with undirected graphs, and its vertex are directly connected with others. Also, this type of algorithm categorizes each vertex of the graph into two categories: visited and not visited. The code starts by visiting the first node chosen and puts in the queue the ones connected to it, and this process repeats until every vertex was visited. In other words, it starts by checking all the nodes on distance 1, then on distance 2, and so on. Moreover, the BFS uses the queue method, so all the nodes that aren't visited and are at the same distance as the ones being visited or are connected to the node it is currently analyzed go to the queue. Ones the code is done with those nodes, they go out of the queue list.



[1] https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph/

3.2.3 DFS algorithm

First off, we start by selecting any random node as the starting vertex. Then it will mark it as visited and continue exploring it as far as possible and putting its unvisited neighbors in the stack. When it is done analyzing all the connectors of one node, it goes back again to the node using backtracking. Then the process continues with the next node until all of them are covered. For example, in the tree below the code would start by analyzing letter "A" and continue processing its data from right to left (in this case). So, it will first visit the connector "D" and follow all its connectors until the end. When it is done processing "F" (which doesn't have more connectors bellow) it goes all the way up to "A", and the process starts all over again until you get to the last connector "E".



[1] https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/

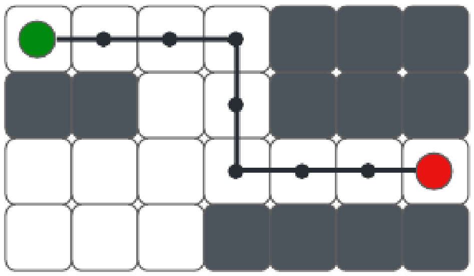
3.2.4 A* algorithm

This algorithm is used in 2D games and in real life maps to find the shortest path to a place. For this reason, this algorithm also contains obstacles in their grids, as if they were places people can't go through (may be interpreted as buildings or streets). Moreover, this algorithm works by choosing a starting and a destination, which we will have to create the shortest path to get from one to another. Basically, what this algorithm does is choosing the next node according to a value "C". This value equals to the sum of two other parameters "A" and "B", which can be defined as:

A-> The distance from the starting point to a given place "x" of the grid.

B-> The distance from that given place "x" to the final destination.

It´s important to know that these distances won't be exact unless we know the paths in which we are working on, since they can have obstacles that can make the distance much larger.



[1] https://www.geeksforgeeks.org/depth-first-search-or-dfs-for-a-graph/

ALGORITHM DESIGN AND IMPLEMENTATION

In what follows, we explain the data structures and the algorithms used in this work. The implementations of the data structures and algorithms are available at GitHub[4].

4.1 Data Structures

For this project, we will use graphs as our data structure. Graphs are a group of objects called vertices or nodes joined by links called edges, which allow to represent binary relationships between the elements of that group. Furthermore, graphs allow studying the interrelationships between units that interact with each other.

Data structure is presented in Figure 2.

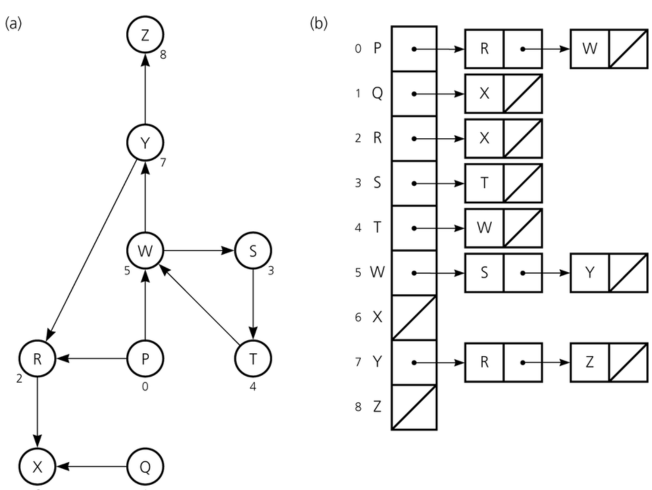


Figure 2: An example of a street map is presented in (a) and its representation as an adjacency list in (b). (*Please, feel free to change this Figure if you use a different data structure*).

4.2 Algorithms

In this work, we propose algorithms for the constrained shortest-path problem. The first algorithm calculates the shortest path without exceeding a weighted-average risk of harassment *r*. The second algorithm calculates the path with the lowest weighted-average risk of harassment without exceeding a distance *d*.

4.2.1 First algorithm

We will be using Dijkstra's algorithm, since it was the easiest and fastest way we found to get the shortest path. Dijkstra is a designed to find the shortest paths between nodes in a graph. Firstly, we'll create a table of visited vertices, to keep track of all the vertices that have been assigned their shortest path. We will also need to set "costs" of all vertices in the graph (lengths of the current shortest path that leads to it).

All the values of each node will be set to 'infinity' at the beginning, to make sure that every other cost we may compare it to would be smaller than the starting one. The only exception is the cost of the first node or starting vertex, since this one will have a 0 assigned to it, because it has no path to itself.

Then, we repeat two main steps until the graph is traversed (as long as there are vertices without the shortest path assigned to them):

- We pick a vertex with the shortest current cost, visit it, and add it to the visited vertices set.
- We update the costs of all its adjacent vertices that are not visited yet.

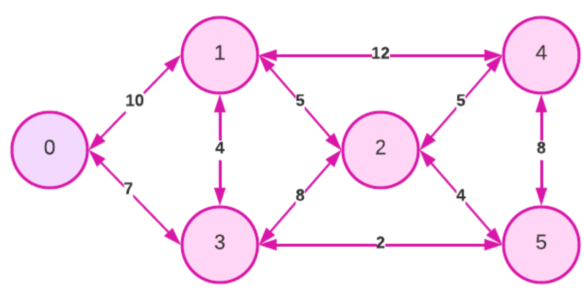


Figure 3: Solving the constrained shortest-path problem with Deep First Search (DFS).

4.2.2 Second algorithm

[4] https://github.com/saradrl/ST0245-001

Explain the design of the algorithm to calculate calculates the path with the lowest weighted-average risk of harassment without exceeding a distance *d* and make your own figure. Do not use figures from the Internet, make your own. *(In this semester, the algorithm could be DFS, BFS, a modified version of Dijkstra, a modified version of A*, among others).* Algorithm is exemplified in Figure 4.

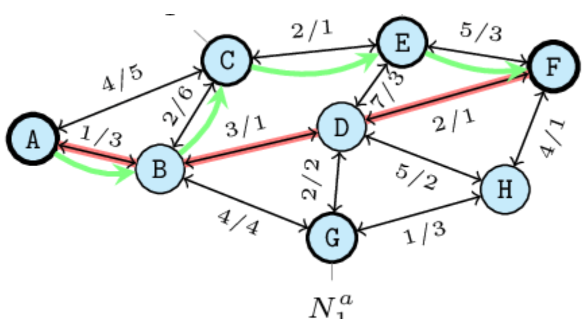


Figure 4: Solving the constrained shortest-path problem with Deep First Search (DFS). (Please, feel free to change this Figure if you use a different algorithm).

4.4 Complexity analysis of the algorithms

Explain, in your own words, the analysis, for the worst case, using O notation. How did you calculate such complexities? Please explain briefly.

Algorithm	Time Complexity
Name of the algorithm	$O(V^{2}*E^{2})$
Name of the second algorithm (in case you tried two)	$O(E^{3}*V*2^{V})$

Table 1: Time Complexity of the name of your algorithm, where V is… E is... *(Please explain what do V and E mean in this problem).*

Data Structure	Memory Complexity
Name of the data structure	$O(V*E*2^{E})$
Name of the second data structure (in case you tried two)	$O(2^{E*}2^{V})$

Table 2: Memory Complexity of the name of the data structure that your algorithm uses, where V is… E is... *(Please explain what do V and E mean in this problem).*

4.5 Design criteria of the algorithm

Explain why the algorithm was designed that way. Use objective criteria. Objective criteria are based on efficiency, which is measured in terms of time and memory. Examples of non-objective criteria are: “I was sick”, “it was the first data structure that I found on the Internet”, “I did it on the last day before deadline”, “it’s easier”, etc. Remember: This is 40% of the project grading.

5. RESULTS

In this section, we present some quantitative results on the shortest path and the path with lowest risk.

5.1.1 Shortest-Path Results

In what follows, we present the results obtained for the shortest path without exceeding a weighted-average risk of harassment *r* in Table 3.

Origin	Destination	Shortest Distance	Without Exceeding *r*
Universidad EAFIT	Universidad de Medellín	??	0.84
Universidad de Antioquia	Universidad Nacional	???	0.83
Universidad Nacional	Universidad Luis Amigó	??	0.85

Table 3. Shortest distances without exceeding a weighted-average risk of harassment *r*.

5.1.2 Lowest Harassment-Risk Results

In what follows, we present the results obtained for the path with lowest weighted-average harassment risk without exceeding a distance *d* in Table 4.

Origin	Destination	Lowest Harassment	Without Exceeding *d*
Universidad EAFIT	Universidad de Medellín	??	5,000
Universidad de Antioquia	Universidad Nacional	???	7,000
Universidad Nacional	Universidad Luis Amigó	??	6,500

Table 3. Lowest weighted-average harassment risk without exceeding a distance *d* (in meters).

5.2 Algorithm Execution-Time

In Table 4, we explain the relation of the average execution times for the queries presented in Table 3.

Compute execution time for the queries presented in Table 3. Report average execution times.

	Average execution times (s)
Universidad EAFIT to Universidad de Medellín	100.2 s
Universidad de Antioquia to Universidad Nacional	800.1 s
Universidad Nacional to Universidad Luis Amigó	845 s

Table 4: Execution times of the algorithm name *(Please write the name of the algorithm, for instance, DFS, BFS, a modified A*)* for the queries presented in Table 3.

6. CONCLUSIONS

Explain the results obtained. Are shortest paths significantly different from paths with lowest harassment-risk? How is this useful for the city? Are execution times reasonable to use this implementation in a real-life situation?

6.1 Future work

Answer, what would you like to improve in the future? How would you like to improve your algorithm and its implementation? Will you continue this projects by further working on Optimization? Statistics? Web development? Machine learning? Virtual Reality? How?

ACKNOWLEDGEMENTS

Identify the kind of acknowledgment you want to write: for a person or for an institution. Consider the following guidelines: 1. Name of teacher is not mentioned because he is an author. 2. You should not mention authors of articles that you have not contacted. 3. You should mention students, teachers from other courses that helped you.

As an example: This research was supported/partially supported by [Name of Foundation, Grant maker, Donor].

We thank for assistance with [particular technique, methodology] to [Name Surname, position, institution name] for comments that greatly improved this manuscript.

The authors are greateful to Prof. Juan Carlos Duque, from Universidad EAFIT, for providing data from Medellín Life Quality Survey, from 2017, processed into a Shapefile.

REFERENCES

[1]

ONU, 2022. Facts and figures: Ending violence against women. https://www.unwomen.org/es/what-we-do/ending-violence-against-women/facts-and-figures (CITADA EN ACM)

[2]

Mininterior. Dirección de los derechos humanos, 2018. ABC sobre el acoso sexual. https://pruebaw.mininterior.gov.co/sites/default/files/documentos/cartilla_-_abc_sobre_acoso_sexual_1.pdf (CITADA EN ACM)

[3]

UNODC. 2018. Examine los hechos: Violencia contra las mujeres. *Interactive.unwomen.org*. https://interactive.unwomen.org/multimedia/infographic/violenceagainstwomen/es/index.html#trafficking-3. (CITADA EN ACM)

[4]

Justicia. R, 2021. Casos de delitos sexuales contra mujeres subieron 19% entre 2015 y 2019. https://www.eltiempo.com/justicia/delitos/violencia-contra-mujeres-medicina-legal-presenta-cifras-de-2015-a-2019-622279 (CITADA EN ACM)

[5]

ONU, 2022. Facts and figures: Ending violence against women. https://www.unwomen.org/es/what-we-do/ending-violence-against-women/facts-and-figures (CITADA EN ACM)

[6] **Enlace imagenes lucid chart:**

https://lucid.app/lucidchart/2e81c067-9327-4893-a48f-eadeeba56d45/edit?invitationId=inv_d49ad700-5037-4edd-a495-222129750a96

[7]

[8]

[9]

[10]