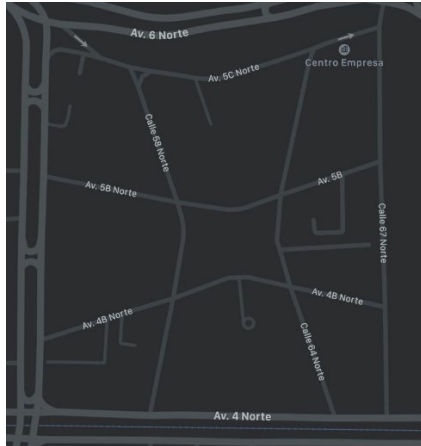


Enunciado

Los residentes del barrio la Flora, en la ciudad de Cali, han manifestado que su tranquilidad se ha visto afectada por la creciente inseguridad en el sector. En el mes de octubre del año 2021, se presentaron más de diez robos a habitantes. La mayoría de las víctimas coincidieron en que habían usado la aplicación *Waze* para seguir la ruta más rápida, pero lograron notar que no solo era la más rápida, sino que también la más peligrosa.



Este tipo de situaciones se siguen presentando y es por esto, que la comunidad del sector se ha reunido para analizar posibles soluciones a esta problemática. Se llegó a la conclusión de que sería de gran utilidad una aplicación que le permita al usuario elegir qué ruta quiere seguir para llegar a su destino, no solo teniendo en cuenta la distancia recorrida, sino que también la seguridad del trayecto.

Para esto, la comunidad ha decidido contratarlo a usted, para que desarrolle una aplicación con interfaz gráfica que le permita al usuario añadir un origen y un destino. De igual forma, el usuario podrá elegir si desea la ruta más rápida o lo más segura. El sistema analizará las diferentes rutas y le mostrará al usuario la ruta que debe seguir según los criterios elegidos.

Engineering Method

Paso 1. Identificación del problema

Identificación de necesidades y síntomas

- La inseguridad en el sector del norte de Cali y especialmente en la flora ha venido aumentando.
- Las personas que viven en estas urbanizaciones no saben cual es la mejor ruta para ir caminando de un lugar a otro.
- Los habitantes de la flora piden una manera de poderse mover entre urbanizaciones por la mejor ruta dependiendo sus intereses, si le dan prioridad a un camino más seguro o un camino más rápido pues su interés es llegar rápido al lugar de destino.

Definición del problema

Los habitantes de las urbanizaciones de la flora piden una solución que les permita saber los mejores caminos para moverse entre urbanizaciones dependiendo de los dos ítems que consideran más importante, distancia o seguridad.

Paso 2. Recopilación de información

Con el objetivo de entender a la empresa a la cual se le desarrollará el respectivo software se hace un análisis histórico superficial de la empresa, además de esto se hace imprescindible saber cuáles han sido históricamente las estadísticas de los jugadores que son evaluadas en los diferentes torneos.

¿Qué es un grafo?

Fuente:

<https://definicion.de/grafos/>

Para las ciencias de la computación y la matemática, un grafo es una representación gráfica de diversos puntos que se conocen como nodos o vértices, los cuales se encuentran unidos a través de líneas que reciben el nombre de aristas. Al analizar los grafos, los expertos logran conocer cómo se desarrollan las relaciones recíprocas entre aquellas unidades que mantienen algún tipo de interacción.

En este sentido no podemos pasar por alto el hecho de que el primer documento escrito que tenemos acerca de lo que son los grafos fue realizado en el siglo XVIII, y más concretamente en el año 1736, por Leonhard Euler. Este fue un matemático y físico, de origen suizo, que destacó por ser una de las figuras más importantes de su tiempo en la citada materia.

En concreto dicho autor realizó un artículo basándose en los puentes que existen en la ciudad de Kaliningrado. A partir de ellos, y mediante lo que es la teoría de los grafos, desarrolló una exposición acerca de los grafos y los vértices que se sustenta en el hecho de que es imposible regresar al vértice que ejerce como punto de partida sin antes no pasar por alguna de las aristas en dos ocasiones.

Percepción de Inseguridad en Cali

Fuente:

<https://www.elspectador.com/colombia/cali/seguridad-en-cali-los-delitos-que-persisten-en-la-ciudad/>

En la capital vallecaucana, el 73 % de las personas no se sienten seguras. Las altas cifras de hurtos, homicidios y ataques al sistema de transporte son parte del panorama.

Inseguridad en Cali

Fuente:

http://www.concejodecali.gov.co/Publicaciones/inseguridad_en_cali_no_cede_persiste_el_aumento_en_tasa_de_homicidios_y_hurtos_harvy_mosquera

“Para este mes de julio de 2021, llegamos a 747 homicidios; 168 más que los registrados en el mismo periodo anterior (enero a julio de 2020); la tasa de homicidios aumentó al 56.5 por cada 100 mil habitantes, es decir, que vamos en retroceso, cuando en 4 años teníamos un porcentaje menor, A ello, dice, hay que sumarle el incremento al 31% de los hurtos, que para julio se sitúan en 9.283 casos; 2.199 más que el reportado en el mismo periodo anterior. “No veo disminuciones, se anuncian programas y apuestas que no generan impacto sobre la delincuencia y bandas criminales. Mientras que entre la población persiste el temor y la zozobra”

Paso 3. Búsqueda de Soluciones Creativas

Para la búsqueda de soluciones creativas se hicieron pequeñas entrevistas para una lluvia de ideas acerca de la problemática, además de esto se revisaron medidas de seguridad implementadas a lo largo de la historia para analizar si pueden solucionar o apoyar para solucionar el problema expuesto.

Alternativa 1. Apoyo policial

Pedir apoyo de la policía nacional para tener más uniformados encargados de la seguridad en el barrio la flora y las personas puedan movilizarse entre urbanizaciones por donde quieran.

Alternativa 2. Experimentos con las personas

Mediante la estadística hacer experimentos durante un tiempo guardando los datos de atracos y hurtos para decirle a las personas por donde es más inseguro caminar.

Alternativa 3. Desarrollo de Software

Desarrollar un software que permita mostrarle al usuario la mejor ruta posible dependiendo del interés sobre la seguridad o la rapidez de llegada mediante un mapa que le indique el camino a seguir.

Alternativa 4. Base de datos y Waze

Hacer estudios acerca de la peligrosidad de las calles y presentarles a los usuarios una base de datos que contenga los datos de la peligrosidad para que junto con waze ellos tomen la decisión de por donde irse.

Alternativa 5. Prohibición de circulación

Prohibir caminar por este sector hasta que no baje la inseguridad, todo debe ser por medios de transporte como carros o motos.

Alternativa 6. Grupos de seguridad ciudadana.

Crear grupos de seguridad ciudadana que permitan una libre circulación de las personas por el sector.

Paso 4. Transición de las ideas a los diseños preliminares.

En primer lugar se opta por descartar la alternativa 2 ya que es una alternativa que afecta a las personas pues se les utiliza como experimento para saber la cantidad de hurtos y en qué calles se dieron en vez de dar una solución a las cosas, además de esto no es una solución inmediata pues requiere de mucho tiempo en ser realizada.

Así mismo se descarta la alternativa 5 ya que prohíbe a las personas su libertad y no soluciona ni el problema de decir el camino más seguro ni el camino más rápido.

También se opta por descartar la alternativa 6 puesto que esta alternativa solo soluciona un problema de seguridad que no es del todo legal puesto que se presta para la creación de organizaciones al margen de la ley como el paramilitarismo.

Esta revisión cuidadosa nos conduce a quedarnos con las siguientes alternativas:

La alternativa 1. Es una alternativa que necesita del apoyo de la policía nacional y que tiene un gran gasto constante, pues los uniformados deben ser pagados. Además, soluciona el problema en poca medida puesto que si bien genera seguridad y no es un problema más para caminar por las calles, las personas siguen sin saber el camino más rápido para ir de una urbanización a otra.

La alternativa 3. Es una alternativa que permite al usuario tener la mejor solución para el problema expresado, pues ellos necesitan algo que les muestre el mejor camino dependiendo sus prioridades de seguridad y rapidez más no un incremento en la seguridad. Además de esto, esta alternativa se puede desarrollar rápidamente para darle solución a las personas lo más rápido posible.

La alternativa 4. Es una alternativa que realiza seguimientos de la peligrosidad de las calles y le presenta una solución al usuario ya que muestra esta base de datos que junto a la aplicación waze permite que las personas elijan por donde irse.

Paso 5. Evaluación y elección de la mejor solución

Criterios:

Criterio A: Afecta a los usuarios?

- [0] Afecta mucho a los usuarios
- [1] Afecta a los usuarios en poca medida
- [2] No afecta a los usuarios

Criterio B: Que tan inmediata es la solución

- [0] No es para nada inmediata, se demora mucho tiempo
- [1] Es relativamente inmediata pero se demora cierto tiempo
- [2] Es inmediata, se demora muy poco tiempo

Criterio C: Soluciona ambas peticiones

- [0] No permite solucionar ninguna petición
- [1] Permite solucionar la seguridad pero no mostrarle el camino más rápido
- [2] Permite solucionar ambas al mostrar el camino más seguro o el más rápido

Criterio D: El equipo de trabajo cuenta con la capacidad, conocimiento y herramientas para llevarla a cabo.

- [0] El equipo no cuenta con el conocimiento ni las herramientas
- [1] El equipo no cuenta con el conocimiento y las herramientas

Evaluación

<u>Alternativas</u>	<u>Criterio A</u>	<u>Criterio B</u>	<u>Criterio C</u>	<u>Criterio D</u>	<u>Total</u>
<u>1. Apoyo policial</u>	2	1	1	0	4
<u>4. Base de datos y Waze</u>	2	0	2	1	5
<u>3. Desarrollo de Software</u>	2	2	2	1	7

Decisión final: Elección de la alternativa 3, construcción de un software

Paso 6. Preparación de informes y especificaciones

Especificación del problema:

El problema será subdividido en dos problemas, el primero es la creación del mapa del sector de la flora estableciendo aristas y vértices puesto que la solución a utilizar se hará por medio de grafos, a estas aristas se les asignarán ciertos pesos los cuales dependen de la peligrosidad de la calle o del largo de la calle. El segundo problema se dará y será el pilar del proyecto puesto que se encargará de darle solución a mostrar el mejor camino para el usuario dependiendo su interés, o la seguridad o la distancia.

Problema 1: Creación del mapa de la flora

Entradas: Información existente del mapa de la flora

Salidas: Mapa creado con vértices y aristas, y cada arista con su peso indicando su peligrosidad y distancia.

Problema 2: Camino ideal

Entradas: Parámetro de importancia del usuario

Salida: Dibujo del mejor camino a realizar por el usuario dependiendo de sus prioridades o parámetros de interés

Paso 7. Implementación de la solución.

Para la implementación de la solución se decidió hacerlo en el lenguaje de programación java ya que es el lenguaje en el que el equipo de trabajo posee mayor conocimiento y hay mejor manejo para la resolución del problema, así mismo se utilizarán grafos, pues la modelación de un mapa y de las calles con peligrosidad y la muestra del mejor camino se pueden solucionar de manera muy eficiente con esta estructura de datos. De igual forma, se hará uso de javafx para realizar la interfaz del usuario.

Para esto hay que tener en cuenta diferentes aspectos:

- El programa debe contar con interfaz gráfica que permita la interacción con el usuario
- El programa debe permitirle a la persona seleccionar sus prioridades, camino con mayor seguridad, camino con menor distancia, camino más seguro y más corto posible
- Se utilizarán grafos.

Grafo TAD

TAD Grafo			
Grafo = {Vertex = <Vertex>, Edge = <Edge>}			
{inv: Debe tener un vértice de partida y otro vértice de llegada}			
Primitive operations:			
Methods	Operation Type	Input	Output
Grafo	Constructor		→Grafo
createVertex	Constructor	T(start)	
createEdge	Modifier	Vertex X Vertex X weight	
adjacent	Analyzer	T(value)	→ArrayList<Vertex>
getVertex	Analyzer	T(value)	→ MyLinkedList<T>
getEdge	Analyzer	weight	→ Edge
isEmpty	Analyzer		→ Boolean
deleteVertex	Modifier	T(value)	→ Vertex
deleteEdge	Modifier	weight	

Grafo ():
*Create an empty Grafo. *
{pre:}
{post: Grafo was created}

createVertex (value):
*Create a new Vertex. *
{pre: TRUE}
{post: Vertex created}

createEdge (vertex1, vertex2):
*Create an Edge between two vertices *
{pre: Vertex1 must to exist} {pre: Vertex2 must to exist}
{post: Edge between two vertices created}

adjacent (vertex, grafo):

*It's a method to find the adjacent vertices of one vertex *

{pre: Vertex must to exist}

{pre: It must exist an edge between the adjacent vertices and the original vertex}

{post: An ArrayList with the adjacent vertices was returned}

getVertex (vertex, value):

*This method allows to get a Vertex from the Grafo without deleting it. The vertex is obtained from the Grafo as long as it exists *

{pre: Grafo must to be initialized}

{pre: vertex must to exist (It must be different from null because we need to compare somehow)}

{pre: vertex must to exist in the Grafo}

{post: Returns the Vertex to get}

getEdge (edge, value):

*This method allows to get an Edge from the Grafo without deleting it. The edge is obtained from the Grafo as long as it exists. *

{pre: Grafo must to be initialized}

{pre: edge must to exist (It must be different from null because we need to compare somehow)}

{pre: edge must to exist in the Grafo}

{post: Return the Edge to get}

isEmpty ():

*Verify if the Grafo has elements (Vertices and Edges) or not. *

{pre: Grafo must to exist}

{post: True if it hasn't elements == null
False otherwise}

deleteVertex (vertex, value):

*It's a method to delete a vertex. *

{pre: Grafo must to be initialized}

<pre>{pre: vertex must to exist (It must be different from null because we need to compare somehow)} {pre: vertex must to exist in the Grafo}</pre>

<pre>{post: Vertex returned and removed}</pre>
--

deleteEdge (vertex1, vertex2):

<pre>*It's a method to delete an edge between two vertices. *</pre>

<pre>{pre: Grafo must to be initialized} {pre: Vertex1 must to exist} {pre: Vertex2 must to exist} {pre: Vertex1 must to exist in the Grafo} {pre: Vertex2 must to exist in the Grafo}</pre>
--

<pre>{post: Edge between two vertices returned and removed}</pre>

https://www6.uniovi.es/usr/cesar/Uned/EDA/Apuntes/TAD_apUM_07.pdf

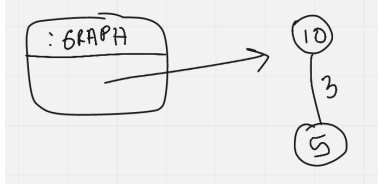

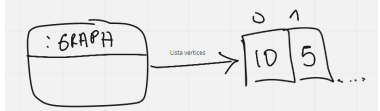
<http://informatica.utem.cl/~mcast/ESDATOS/GRAFOS/grafos.pdf>

[https://es.wikipedia.org/wiki/Grafo_\(tipo_de_dato_abstracto\)](https://es.wikipedia.org/wiki/Grafo_(tipo_de_dato_abstracto))

<https://www.fceia.unr.edu.ar/estruc/2005/graffund.htm>

Test Cases Design

GraphTest

Name	Class	Scene
sc1	Graph	
sc2	Graph	
sc3	Graph	 No hay aristas

Create Vertex Method Test

Objective: Validate the correct creation of a vertex when the graph is empty

Class	Method	Scene	Entries	Result
Graph	createVertex	sc2	Vertex(5)	After creating the vertex(5) the Graph is not going to be empty so the test will show false for finalEmptyState because it is not empty

Remove Vertex Method test

Objective: Validate if removing a vertex algorithm is working correctly.

Class	Method	Scene	Entries	Result
Graph	removeVertex	sc2	vertex(5)	The test creates the vertex first and then delete it, at the end the graph is not empty before removing the vertex, after that the graph has to be empty.

Get Vertex Method test

Objective: Validate if when getting a vertex depending on the vertex name it returns the correct one

Class	Method	Scene	Entries	Result
Graph	getVertex	sc1	none	The returned vertex is the same that the consulted.

Create edge Method test

Objective: Validate if the process to create a new edge is correct

Class	Method	Scene	Entries	Result
Graph	createEdge	sc3	none	The edge is created and the adjacentList is not empty but before creating it the adjacentList is empty cuz nothing have been added to it

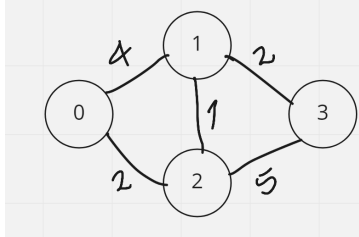
Remove edge Method test

Objective: Validate if the process to remove a edge is correct

Class	Method	Scene	Entries	Result
Graph	createEdge	sc1	adjacentList	The edge is removed from the adjacentList, so the adjacentList is empty now because when vertex 3 is eliminated vertexed 5 and 10 are not adjacent vertexes anymore.

Test Cases Design

Graphic

Name	Class	Scene
sc1	Graphic	

Dijkstra Method Test

Objective: Validate the correct working of the dijkstra.

Class	Method	Scene	Entries	Result
Graphic	dijkstra	sc1	none	After the method is done, the shortestDsitances array has to be=[0, 3, 2, 5]

Floyd- Marshall Method Test

Objective: Validate that floyd algorithm is working correctly

Class	Method	Scene	Entries	Result																
Graphic	floydWarshall	sc1	none	After the method is done, the final matrix has to be <table><tr><td>0</td><td>3</td><td>2</td><td>5</td></tr><tr><td>3</td><td>0</td><td>1</td><td>2</td></tr><tr><td>2</td><td>1</td><td>0</td><td>3</td></tr><tr><td>5</td><td>2</td><td>3</td><td>0</td></tr></table>	0	3	2	5	3	0	1	2	2	1	0	3	5	2	3	0
0	3	2	5																	
3	0	1	2																	
2	1	0	3																	
5	2	3	0																	

