

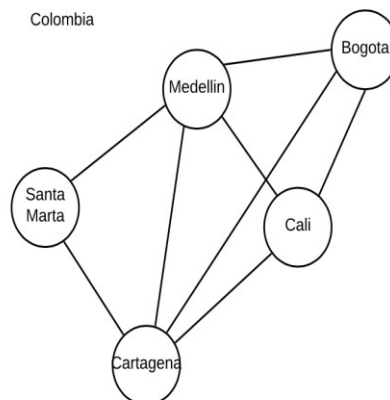
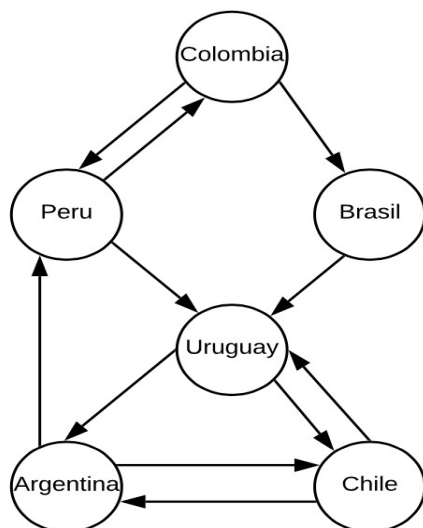
MÉTODO DE LA INGENIERÍA

CONTEXTO DEL PROBLEMA:

Una familia europea desea viajar hacia América del sur para viajar a algunos países y en ellos sus ciudades que antes no han tenido oportunidad de visitarlos. Este viaje se realizará de país a país en avión. Para el caso de las ciudades el viaje se hará por tierra, es decir, en carro.

En medio de una conversación que tenían sobre este viaje, la familia se da cuenta que no tienen una buena organización para elegir la ruta más eficiente y así ahorrar recursos en el viaje. Debido a esto han solicitado a la universidad Icesi que les brinde un programa con el cual puedan saber cuál es el mejor camino para cada visita que desean hacer.

En el momento en que deciden viajar a un país van a llegar a la ciudad capital del mismo, en ese momento van a poder elegir si desean visitar más ciudades de este país o si desean pasar al siguiente.



FASE 1 Identificación del problema:

- > El camino más corto de un país a otro.
- > Tiempo que demora de llegar de un país/ciudad a otro/a, teniendo en cuenta la velocidad ingresada
- >Mostrar las ciudades dentro de un país seleccionado que se puedan visitar.

Requerimientos:

Name	R.# 1 Conceder el camino más corto entre países
Summary	Se requiere conocer el camino más corto entre los vértices
Inputs	
<ul style="list-style-type: none">· País de inicio.· País de llegada.	
Results	
Camino mínimo entre los dos puntos .	

Name	R.# 2 Conceder el camino más corto entre ciudades
Summary	Se requiere conocer el camino más corto entre los vértices
Inputs	
<ul style="list-style-type: none"> · Ciudad de inicio. · Ciudad de llegada. 	
Results	
Camino mínimo entre los dos puntos .	

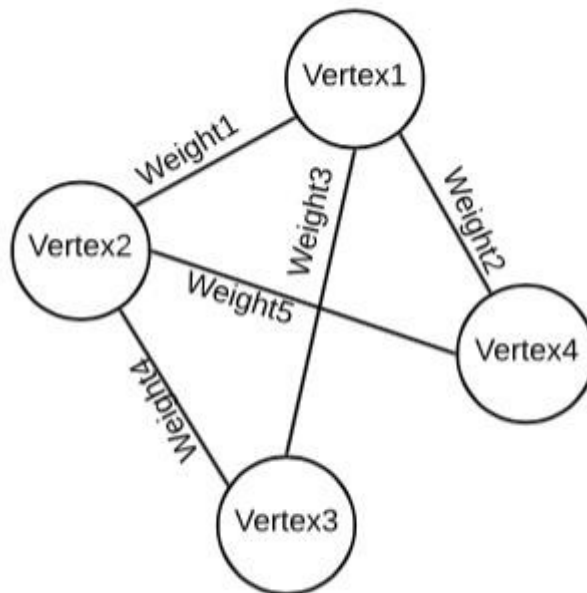
Name	R.# 3 Conocer el tiempo de demora de un país a otro al conocer la velocidad.
Summary	Se requiere estimar el tiempo de llegada de una ciudad a otra por medio de la velocidad promedio introducida por el usuario.
Inputs	
<ul style="list-style-type: none"> · Velocidad Promedio. · País de inicio. · País de llegada 	
Results	
Tiempo estimado de llegada.	

Name	R.# 4. Visualizar las ciudades del país seleccionado
Summary	Se debe visualizar las ciudades del País
Inputs	
<ul style="list-style-type: none"> • nombre país 	
Results	
Se ha desplegado los nombres de las ciudades	

Name	R.# 5. Agregar al País una Ciudad .
Summary	Se requiere agregar al país una ciudad.
Inputs	
<ul style="list-style-type: none"> • Nombre del país • Nombre de la ciudad 	
Results	
Se ha agregado la ciudad al país.	

TAD

Graph



Inv{any two vertex can only have one edge connecting them}

- **AddVertex: Vertex** → **Boolean**
- **AddEdge: Vertex X Vertex X weight** → **Boolean**
- **AdjacencyListGraph:** → **Graph**
- **SearchVertex: Texto** → **Vertex**
- **DepthFirstSearch: Graph X Vertex** → **List**

▪	BreadthFirstSearch: Graph X Vertex	→	List
▪	Dijkstra: Graph X Vertex	→	List
▪	FloydWarshall: Graph	→	List
▪	RemoveEdge: Vertex X Vertex	→	Boolean
▪	RemoveVertex Vertex	→	Boolean

AddVertex(Vertex)
 “adds a vertex object to the graph”
 {pre :Graph != null, Vertex \notin Graph}
 {post Vertex added, Vertex \in Graph}

AddEdge (Vertex 1 ,Vertex2)
 {pre :Graph != null, Vertex1 \wedge Vertex \in Graph}
 {post Edge added in between two vertexes}

SearchVertex (name)
 “from a text searches a vertex in the graph”
 {pre :Vertex \in Graph \wedge name \in Text }
 {post returns Vertex}

Diseño de pruebas unitarias

Escenarios:

nombre	clase	Escenario
SetupScenary()	MatrixWeight Graph	WeightMatrixGraph WG = new WeightMatrixGraph(false, 1)
SetupScenary1())	AdjencyListGra	AdjencyListGraph AG = new AdjencyListGraph(False)
SetupScenary2())	Algorithms	WeightMatrixGraph WG = new WeightMatrixGraph(false, 4) AdjencyListGraph AG = new AdjencyListGraph(False) GraphAlgorithms = GA = new GraphAlgorithms()

Objetivo de la prueba : Comprobar que se creo el grafo Correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado
MatrixWeight Graph	WeightMatrix Graph()	SetupScenary()		Se creo el grafo de manera adecuada

Objetivo de la prueba : Comprobar que se agrego un vertice de manera correcta				
Clase	Método	Escenario	Valores de Entrada	Resultado
MatrixWeight Graph	AddVertex	SetupScenario()	V vertex V vertex1	Se Agrego El vértice de manera correcta

Objetivo de la prueba : Comprobar que se agrego una arista al grafo de manera correcta				
Clase	Método	Escenario	Valores de Entrada	Resultado
MatrixWeight Graph	AddEdge	SetupScenario()	V vertex V vertex1 double w = 1	Se agrego la arista de manera correcta al grafo

Objetivo de la prueba : Comprobar que se quito la arista de manera correcta				
Clase	Método	Escenario	Valores de Entrada	Resultado
MatrixWeight Graph	RemoveEdge	SetupScenario()	V vertex V vertex1	Se elimino la arista para el par de vertices

AdjencyList :

Objetivo de la prueba : Comprobar que se creo el grafo Correctamente				
Clase	Método	Escenario	Valores de Entrada	Resultado
AdjencyList	AdjencyListGraph()	SetupScenariy1()		Se creo el grafo de manera adecuada

Objetivo de la prueba : Comprobar que se agrego un vertice de manera correcta				
Clase	Método	Escenario	Valores de Entrada	Resultado
AdjencyList	AddVertex	SetupScenariy1()	V vertex V vertex1	Se Agrego El vértice de manera correcta

Objetivo de la prueba : Comprobar que se agrego una arista al grafo de manera correcta				
Clase	Método	Escenario	Valores de Entrada	Resultado
AdjencyList	AddEdge	SetupScenariy()	V vertex V vertex1 double w = 1	Se agrego la arista de manera correcta al grafo

Algorithms:

Objetivo de la prueba : Comprobar que se encontró de manera correcta el recorrido mas corto

Clase	Método	Escenario	Valores de Entrada	Resultado
Algorithms	Dijkstra()	SetupScenar y2()	V vertex V vertex1 V vertex2 V vertex3	Se encontró de manera correcta el recorrido mas corto para los vertices

Objetivo de la prueba : Comprobar que se encontró de manera correcta el recorrido mas corto

Clase	Método	Escenario	Valores de Entrada	Resultado
Algorithms	FloydWarshall	SetupScenar y2()	V vertex V vertex1 V vertex2 V vertex3	Se encontró de manera correcta el recorrido mas corto para los vertices

Fase 2 Recopilación de información

Países de América del sur y sus ciudades principales:

El subcontinente sudamericano contiene muchos países que son reconocidos por su cultura, gastronomía y lugares turísticos. Debido a esto, muchas personas se interesan en viajar hacia este subcontinente y así poder conocer de lo que tanto se habla, la variedad de ambientes.

A continuación, se mencionan los países sudamericanos más visitados:

- Argentina.
- Brasil.
- Chile.
- Perú.
- Colombia.
- Costa Rica.

Por cada país se mencionan 5 ciudades consideradas las más importantes:

Argentina: Buenos Aires, Córdoba, Rosario, La plata y San Miguel de Tucumán.

Brasil: São Paulo, Río de Janeiro, Brasília, Salvador de Bahía y Belo Horizonte

Chile: Santiago, Gran Valparaíso, Gran Concepción, Gran La Serena y Antofagasta.

Perú: Lima, Cuzco, Iquitos, Arequipa y Puno.

Colombia: Bogotá, Medellín, Cali, Barranquilla y Cartagena de Indias.

Costa Rica: San José, Alajuela, Cartago, Heredia y Guanacaste.

También debemos tener en cuenta de qué forma se relacionan los territorios entre los países en cuanto a los viajes en avión. También hay que tener en cuenta entre ciudades cuales están conectadas con las otras mediante carretera ya que en ellas solo se viaja por tierra.

Representación gráfica:

En temas de programación existe una estructura de datos llamada Grafo la cual ha ayudado y facilitado mucho el diseño de soluciones en temas relacionados con generación de caminos en diferentes ámbitos.

Algoritmos para el mejor camino:

Durante bastante tiempo se han buscado algoritmos para encontrar el mejor camino entre dos puntos, es decir, el camino más cerca que se puede haber. Este problema ha sido muy explorado y se han encontrado muchas soluciones como lo son los algoritmos BFS(Breadth first search), el DFS(Depth first search), el algoritmo de Dijkstra y el algoritmo de Floyd-Warshall. Estos algoritmos en general son una solución para el problema mencionado, y particularmente nos permiten ejercer algunas funciones específicas y diferentes formas de implementación.

Fase 3 Búsqueda de soluciones creativas:

Alternativa 1: Utilizar una página web.

Esta opción consiste en utilizar una página en la que estén almacenadas toda la información acerca de las relaciones entre los vuelos de los países, es decir, cuáles serían los vuelos más cortos para llegar al destino deseado, sin mencionar que tendría que tener absolutamente toda la información de los vuelos.

Alternativa 2: Mapa.

Crear un mapa que indique las rutas de vuelo disponibles entre países, y que además permite ver la conexión de todas las ciudades en cada país.

Alternativa 3: Programa.

Esta alternativa consiste en diseñar un programa el cual mediante grafos nos permita observar las rutas disponibles ya sea en países o ciudades, calcular y elegir cual es el camino más eficiente entre diferentes puntos.

Fase 4: Diseños preliminares

A continuación, se van a analizar las opciones de solución que tenemos.

La alternativa 1 puede llegar a ser muy eficiente por toda la información que tendría y gran rango de decisión que podrían tener las personas. Sin embargo, al ser tantos países y ciudades, la información sobrepasará nuestra capacidad y se perdería mucho tiempo que no tenemos.

La alternativa 2 También cumpliría con el propósito, pero son demasiadas rutas por el hecho de la variedad de países, el tiempo para hacerlo se sale de las manos. Además, el mapa podría llegar a ser muy difícil de entender.

En la alternativa 3, el hecho de utilizar grafos nos facilita muchos procesos, por ejemplo, en el problema de encontrar el camino más corto, con grafos se pueden implementar algunos de los algoritmos antes mencionados. También hay que aclarar que por cada país se tendría que crear un grafo diferente o modificar un grafo ya determinado por la variedad de ciudades.

Fase 5: Evaluación y selección.

A continuación, se encuentra un análisis de las alternativas respecto a algunos criterios.

→ Criterio 1: Percepción de utilización.

- ◆ 3. Intuitivo
- ◆ 2. Entendimiento común ◆
- 1. Difícil entendimiento.

→ Criterio 2: Complejidad en implementaciones.

- ◆ 3. Fácil de crear
- ◆ 2. Medianamente asequible
- ◆ 1. Complejo para implementar

	Criterio 1	Criterio 2	Total
Alternativa 1	3	1	4
Alternativa 2	1	1	2
Alternativa 3	3	2	5

La alternativa 3 es la más asequible a implementar por temas anteriormente mencionados, además, según los criterios evaluados para la comodidad y eficiencia, esta alternativa está por encima de las demás.

Fase 6: Preparación de reportes y especificaciones.

Problema: Encontrar el camino o ruta más eficiente entre países y en ellos, definir la ruta más adecuada en vehículo terrestre para transportarse entre ciudades.

Para la realización del programa con grafos hay que tener en cuenta que el manejo de los países se hará con un grafo dirigido. Por cada país se tendrá un grafo diferente que contiene las ciudades de dicho país, este grafo será un grafo simple.

Las ciudades que los clientes deseen agregar deben tener especificado su nombre y hacia que otras ciudades están conectadas.

