

**Uso de grafos para encontrar caminos cortos en ciudades, caso: Cuernavaca, Morelos, México**

Derick Axel Lagunes Ramírez

José Alberto Morales Morante



Fecha: 03 de diciembre de 2018

**Resumen**

Este documento presenta el desarrollo del proyecto de la unidad 5 de matemáticas discretas del programa de maestría en ciencias de la computación de Cenidet. La unidad 5 trata el tema de grafos y árboles estas son herramientas que en matemáticas y computación nos permiten representar estructuras de elementos relacionados. Más específicamente los grafos representan relaciones binarias, se representan gráficamente como un conjunto de puntos unidos por líneas.

Los grafos permiten representar distintos escenarios para resolver problemas variados. En este proyecto se realizó un grafo representando la ciudad de Cuernavaca (Morelos, México), con este grafo se pretende obtener las rutas más cortas (en tiempo gastado) entre uno de los distintos puntos de la ciudad y el Cenidet. De este modo darle a nuevos estudiantes y aspirantes una ruta más corta para llegar.

El documento describe a fondo la generación del grafo (definición de vértices y aristas, límites de la ciudad y ponderación de las distancias). Y algoritmo utilizado para calcular las distancias más cortas. Al final se obtuvo una matriz con todos los vértices definidos y la distancia de la ruta más corta entre pares de vértices.

**Contenido**

[1 Introducción 1](#_Toc531342259)

[1.1 Planteamiento del problema 1](#_Toc531342260)

[1.2 Objetivos 1](#_Toc531342261)

[1.2.1 General 1](#_Toc531342262)

[1.2.2 Específicos 1](#_Toc531342263)

[1.3 Estructura del documento 1](#_Toc531342264)

[2 Marco Teórico 2](#_Toc531342265)

[2.1 Grafos 2](#_Toc531342266)

[2.2 Algoritmo de Floyd-Warshall 3](#_Toc531342267)

[2.3 Pseudocódigo 3](#_Toc531342268)

[2.4 C# (Visual) 4](#_Toc531342269)

[3 Desarrollo del proyecto 6](#_Toc531342270)

[3.1 Realizando el grafo de la ciudad 6](#_Toc531342271)

[3.1.1 Parámetros para definir un vértice 6](#_Toc531342272)

[3.1.2 Conjunto de vértices obtenidos 6](#_Toc531342273)

[3.1.3 Unión de los vértices 10](#_Toc531342274)

[3.1.4 Ponderación de las aristas 11](#_Toc531342275)

[3.2 Obtención de las rutas más cortas 12](#_Toc531342276)

[3.2.1 Utilización del algoritmo de Floyd-Warshall en C# 12](#_Toc531342277)

[4 Resultados 13](#_Toc531342278)

[4.1 Matriz de tiempos 13](#_Toc531342279)

[4.2 Matriz de rutas 14](#_Toc531342280)

[5 Conclusiones y trabajo futuro 15](#_Toc531342281)

[5.1 Conclusiones 15](#_Toc531342282)

[5.2 Trabajo Futuro 15](#_Toc531342283)

[Bibliografía 16](#_Toc531342284)

# 1 Introducción

Un grafo es una representación de un conjunto de elementos llamados vértices que están relacionados, por medio de una unión llamada arista, con otros vértices. Permiten visualizar relaciones binarias entre pares de vértices y son utilices para la resolución de diversos problemas en el campo de las matemáticas y la computación.

Computacionalmente los grafos pueden ser representados como matrices bidimensionales, donde la unión de las filas y columnas representan si hay conexión entre 2 vértices y a veces también algún valor que exista entre ellos (como distancia, tiempo, etc.).

Una de las aplicaciones más importantes y más usadas de los grafos en los mapas, es para determinar e intentar disminuir el tiempo y camino en recorridos entre sitios distintos. El problema del camino más corto consiste en encontrar un camino entre dos vértices (o nodos) de tal manera que la suma de los pesos de las aristas que lo constituyen es mínima (como cuando se desea ir de una ciudad a otra en el menor tiempo posible).

## 1.1 Planteamiento del problema

En la ciudad de Cuernavaca, los caminos construidos y las rutas definidas por el transporte público no siguen patrones debido a la naturaleza del lugar (zona montañosa y dividida por barrancas) esto resulta en dificultad para encontrar lugares la primera vez que se visitan y más aún saber por dónde se puede llegar a esos lugares. Esta problemática impacta sobre todo a personas que comienzan a vivir en la ciudad por trabajo o estudios como es el caso de estudiantes foráneos de Cenidet.

## 1.2 Objetivos

### 1.2.1 General

Utilizar un grafo de la ciudad de Cuernavaca para obtener las rutas más cortas entre distintos puntos de la capital del estado.

### 1.2.2 Específicos

* Realizar un grafo de la ciudad de Cuernavaca y sus alrededores
* Utilizar el grafo generado y un algoritmo para encontrar las rutas más cortas entre 2 de los vértices
* Publicar los resultados para ayudar a alumnos y aspirantes a tomar caminos más cortos a Cenidet

## 1.3 Estructura del documento

Este trabajo se estructura en secciones, la sección 2, el marco teórico, trata de la base de conocimientos necesarios para el desarrollo del proyecto. En la sección 3, desarrollo del proyecto, se describen los métodos que se utilizaron para desarrollar la solución propuesta. La sección 4 abarca los resultados que se obtienen a partir de la utilización de la solución. Finalmente, en la sección 5 se exponen las conclusiones del proyecto desarrollado.

# 2 Marco Teórico

En esta sección se presenta la teoría fundamental y las herramientas en las que está basado el proyecto desarrollado.

## 2.1 Grafos

Un grafo es la representación simbólica de un sistema, su estructura se compone de elementos llamados vértices o nodos unidos por líneas también conocidas como aristas. Se puede representar matemáticamente como:

Ilustración 1 Representación matemática de un grafo

Donde V es la representación de un conjunto de vértices mientras que A representa a un conjunto de aristas. Se le conoce como grafo ponderado a aquel cuyas aristas poseen un peso o valor, representando el esfuerzo que toma trasladarse entre los vértices que conecta.

Su aplicación permite modelar relaciones binarias entre elementos de un mismo conjunto, por esta razón es de ayuda para la realización de planos, circuitos eléctricos, topologías de red, redes sociales, isómeros, etc.

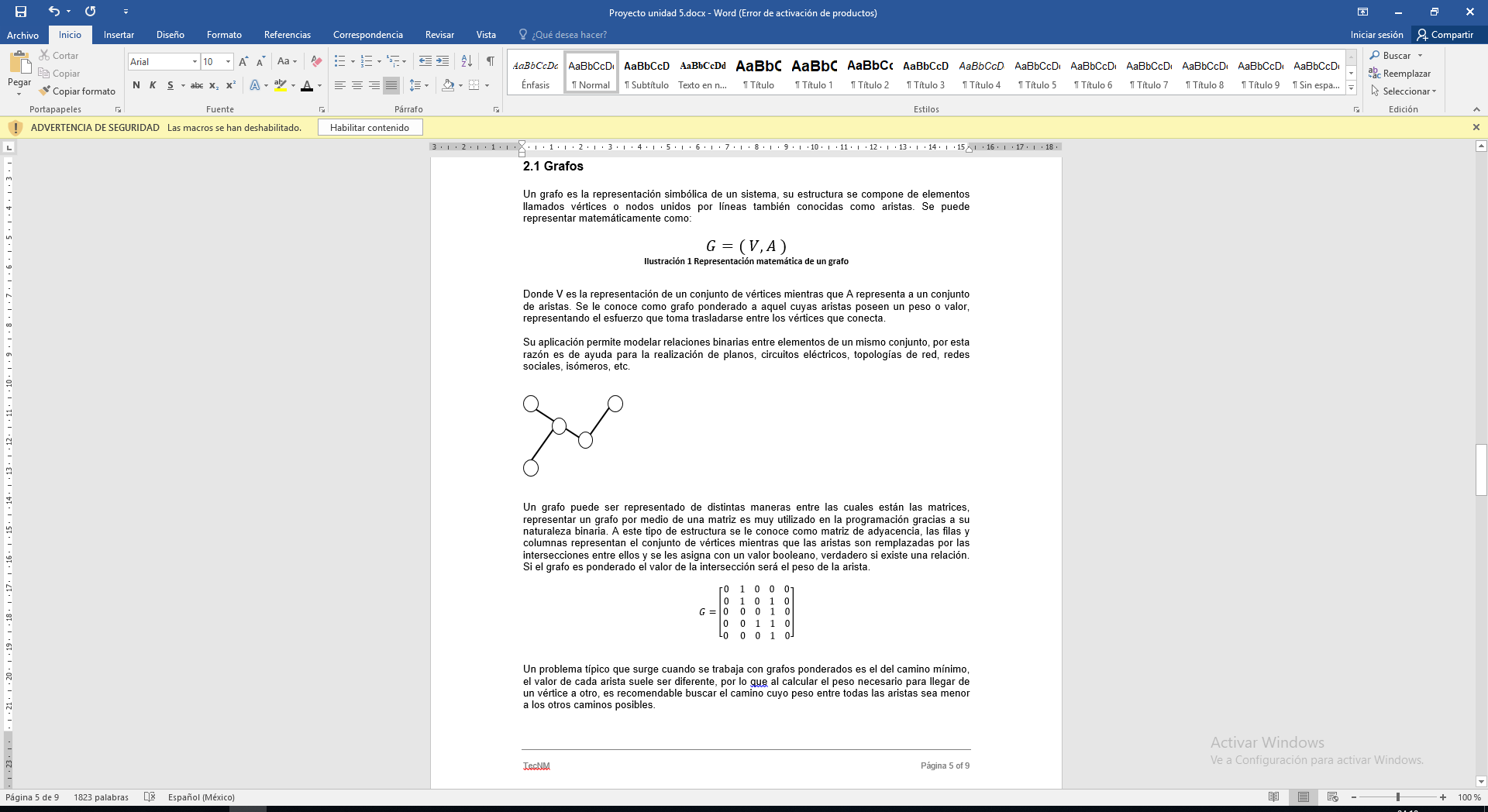


Ilustración 2 Representación gráfica de un grafo

Un grafo puede ser representado de distintas maneras entre las cuales están las matrices, representar un grafo por medio de una matriz es muy utilizado en la programación gracias a su naturaleza binaria. A este tipo de estructura se le conoce como matriz de adyacencia, las filas y columnas representan el conjunto de vértices mientras que las aristas son remplazadas por las intersecciones entre ellos y se les asigna con un valor booleano, verdadero si existe una relación. Si el grafo es ponderado el valor de la intersección será el peso de la arista.

Ilustración 3 Representación en matriz de un grafo

Un problema típico que surge cuando se trabaja con grafos ponderados es el del camino mínimo, el valor de cada arista suele ser diferente, por lo que, al calcular el peso necesario para llegar de un vértice a otro, es recomendable buscar el camino cuyo peso entre todas las aristas sea menor a los otros caminos posibles.

## 2.2 Algoritmo de Floyd-Warshall

Es un algoritmo de análisis sobre grafos que permite encontrar el camino mínimo en grafos dirigidos ponderados. El algoritmo encuentra el camino entre todos los pares de vértices en una única ejecución. Lo hace mejorando paulatinamente una estimación del camino más corto entre dos vértices, hasta que se sabe que la estimación es óptima.

Fue publicado en 1962 por Robert Floyd, es básicamente la síntesis de otros dos algoritmos publicados pocos años atrás, el algoritmo de Bernard Roy (1959) utilizado para conocer el camino mínimo en grafos ponderados y el algoritmo de Stephen Warshall (1962) usado para conocer el cierre transitivo en grafos dirigidos.

## 2.3 Pseudocódigo

El pseudocódigo es una forma de representar con un alto nivel y de forma compacta e informal el principio operativo de un programa informático u otro algoritmo. Utiliza las convenciones estructurales de un lenguaje de programación real, ​ pero está diseñado para la lectura humana en lugar de la lectura mediante máquina, y con independencia de cualquier otro lenguaje de programación.

Normalmente, el pseudocódigo omite detalles que no son esenciales para la comprensión humana del algoritmo, tales como declaraciones de variables, código específico del sistema y algunas [subrutinas](https://es.wikipedia.org/wiki/Subrutina). Se utiliza pseudocódigo pues este es más fácil de entender para las personas que el código del lenguaje de programación convencional, ya que es una descripción eficiente y con un entorno independiente de los principios fundamentales de un algoritmo.

El algoritmo de Floyd-Warshall puede ser representado mediante un pseudocódigo (Código 1 Pseudocódigo del algoritmo de Floyd-WarshallCódigo 1).

/\* La función pesoArista devuelve el coste del camino que va de i a j

infinito si no existe). y n es el número de vértices y pesoArista(i,i) = 0 \*/

**int** camino[][];

/\* Una matriz bidimensional. En cada paso del algoritmo, camino[i][j] es el camino mínimo de i hasta j usando valores intermedios de (1..k-1). Cada camino[i][j] es inicializado a pesoArista(i,j) \*/

**procedimiento** FloydWarshall ()

para k: = 0 hasta n − 1

*camino*[i][j] = mín (camino*[i][j]*, c*amino[i][k]*+*camino[k][j]*)

fin para

Código 1 Pseudocódigo del algoritmo de Floyd-Warshall

## 2.4 C# (Visual)

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Aunque C# forma parte de la plataforma .NET, ésta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma.

Se pueden programar varios algoritmos y soluciones a problemas con las herramientas que brinda este lenguaje y utilizando la lógica de programación. Por ejemplo, una implementación del algoritmo de Floyd-Warshall en lenguaje de programación C# (Código 2).

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Diagnostics;

namespace FloydWarshallAlgorithm

{

class FloydWarshallAlgo

{

public const int cst = 9999;

private static void Print(int[,] distance, int verticesCount)

{

Console.WriteLine("Distancias mas cortas entre un par de vewrices:");

for (int i = 0; i < verticesCount; ++i)

{

for (int j = 0; j < verticesCount; ++j)

{

if (distance[i, j] == cst)

Console.Write("cst".PadLeft(7));

else

Console.Write(distance[i, j].ToString().PadLeft(7));

}

Console.WriteLine();

}

}

public static void FloydWarshall(int[,] graph, int verticesCount)

{

int[,] distance = new int[verticesCount, verticesCount];

for (int i = 0; i < verticesCount; ++i)

for (int j = 0; j < verticesCount; ++j)

distance[i, j] = graph[i, j];

for (int k = 0; k < verticesCount; ++k)

{

for (int i = 0; i < verticesCount; ++i)

{

for (int j = 0; j < verticesCount; ++j)

{

if (distance[i, k] + distance[k, j] < distance[i, j])

distance[i, j] = distance[i, k] + distance[k, j];

}

}

}

Print(distance, verticesCount);

}

static void Main(string[] args)

{

int[,] graph = {

{ 0, cst, cst, cst, cst, 343, cst, 1435, 464, cst },

{cst, 0, cst, cst, cst, 879, 954, 811, cst, 524},

{cst, cst, 0, cst, 1364, 1054, cst, cst, cst, cst},

{cst, cst, cst, 0, cst, cst, 433, cst, cst, 1053},

{cst, cst, 1364, cst, 0, 1106, cst, cst, cst, 766},

{343, 879, 1054, cst, 1106, 0, cst, cst, cst, cst},

{cst, 954, cst, 433, cst, cst, 0, 837, cst, cst},

{1435, 811, cst, cst, cst, cst, 837, 0, cst, cst},

{464, cst, cst, cst, cst, cst, cst, cst, 0, cst},

{cst, 524, cst, 1053, 766, cst, cst, cst, cst, 0}

};

FloydWarshall(graph, 10);

Console.ReadKey();

}

}

}

Código 2 código en C# del algoritmo de Floyd-Warshall

# 3 Desarrollo del proyecto

En esta sección se describe como fue el desarrollo del proyecto.

## 3.1 Realizando el grafo de la ciudad

### 3.1.1 Parámetros para definir un vértice

Los parámetros que se tomaron en cuenta para decidir qué puntos de la ciudad (puntos con una latitud y una longitud) se utilizarían como vértices del grafo se describen a continuación:

* Deben ser lugares comunes de interés
  + Centros comerciales
  + Plazas
  + Restaurantes
  + Monumentos
  + Centro de la ciudad
* Lugares donde comúnmente habitan estudiantes del Cenidet
* Deben de abarcar la mayoría de la ciudad y caminos principales
* Se consideran vértices en los alrededores de la ciudad como:
  + Emiliano Zapata
  + Temixco
  + Jiutepec
  + El aeropuerto de Cuernavaca

### 3.1.2 Conjunto de vértices obtenidos

Utilizando los parámetros anteriores se obtuvo la siguiente lista de vértices (Tabla 1):

| **Vértice** | **Latitud** | **Longitud** | **Lugar** |
| --- | --- | --- | --- |
| **1** | 18.934496 | -99.228986 | Plaza Cuernavaca |
| **2** | 18.936084 | -99.192424 | Galerías |
| **3** | 18.951631 | -99.232711 | La Salle, Universidad |
| **4** | 18.956004 | -99.237742 | Wallmart domingo diez |
| **5** | 18.965268 | -99.246819 | Glorieta de Zapata |
| **6** | 18.968068 | -99.24021 | Paloma de la paz |
| **7** | 18.958054 | -99.208528 | Base ruta 10 Sta. Rosa |
| **8** | 18.958896 | -99.212776 | Base ruta 4 Ahuatepec |
| **9** | 18.950135 | -99.22546 | Base ruta interescolar teopanzolco |
| **10** | 18.937779 | -99.225587 | Glorieta diana cazadora |
| **11** | 18.945258 | -99.242954 | Sanatorio santa Mónica |
| **12** | 18.982232 | -99.237399 | UNAM |
| **13** | 18.945826 | -99.230039 | Mercado Lomas de la selva |
| **14** | 18.930147 | -99.230908 | Pullman Casino de la Selva |
| **15** | 18.945816 | -99.233397 | Soriana súper- Cuernavaca |
| **16** | 18.948586 | -99.222132 | IEBEM |
| **17** | 18.935891 | -99.233612 | Hospital general José Parres |
| **18** | 18.968799 | -99.228559 | Panteón comunal Ocotepec |
| **19** | 18.955624 | -99.242018 | Hospital bella vista |
| **20** | 18.940834 | -99.208174 | Ciclopista Cuernavaca |
| **21** | 18.965524 | -99.210926 | Ahuatepec |
| **22** | 18.966465 | -99.220436 | Ocotepec |
| **23** | 18.981042 | -99.25403 | Santa María Ahuacatitlán |
| **24** | 18.951594 | -99.19688 | Antonio Barona |
| **25** | 18.952938 | -99.224835 | Lomas de Cortes |
| **26** | 18.957289 | -99.26121 | Lomas de Ahuatlán |
| **27** | 18.949601 | -99.245145 | Parque tlaltenango |
| **28** | 18.948319 | -99.25759 | Bugambilias II |
| **29** | 18.961474 | -99.186486 | Frac. Universo |
| **30** | 18.940953 | -99.186863 | Milpillas |
| **31** | 18.935838 | -99.217407 | Vista hermosa |
| **32** | 18.934296 | -99.236186 | Miraval |
| **33** | 18.943371 | -99.228076 | Sam’s Club Cuernavaca |
| **34** | 18.932837 | -99.232335 | Mega Soriana |
| **35** | 18.966724 | -99.234336 | CAPUFE |
| **36** | 18.911158 | -99.171953 | Civac |
| **37** | 18.894918 | -99.173584 | Jiutepec |
| **38** | 18.858372 | -99.239073 | Temixco |
| **39** | 18.860971 | -99.210148 | Tres de Mayo |
| **40** | 18.833028 | -99.261646 | Aero puerto |
| **41** | 18.847812 | -99.184141 | Emiliano Zapata |
| **42** | 18.914812 | -99.208775 | Parque barranca Chapultepec |
| **43** | 18.91193 | -99.178562 | Plaza Cedros |
| **44** | 18.925489 | -99.19796 | Fórum Cuernavaca |
| **45** | 18.92216 | -99.205685 | Hospital General Regional 1 |
| **46** | 18.876443 | -99.220115 | Cenidet Campus Electrónica |
| **47** | 18.93249 | -99.23951 | Carolina |
| **48** | 18.92873 | -99.2438 | Panteón municipal La Leona |
| **49** | 18.92603 | -99.23837 | Terminal de autobuses Cuernavaca |
| **50** | 18.9216 | -99.2333 | Palacio de cortes |
| **51** | 18.92108 | -99.23771 | Jardín Borda |
| **52** | 18.91029 | -99.23359 | Editorial Periódico "Diario de Morelos" |
| **53** | 18.91994 | -99.2198 | Hospital de trasplantes IMT |
| **54** | 18.87347 | -99.21885 | Glorieta reforma |
| **55** | 18.9196 | -99.19359 | Parque Alameda |
| **56** | 18.92142 | -99.20251 | Policía Federal |
| **57** | 18.92447 | -99.22202 | Gasolinera Pemex |
| **58** | 18.92584 | -99.22786 | Clínica "Imagen Medica" |
| **59** | 18.91046 | -99.23157 | Secretaria de ingresos |
| **60** | 18.90657 | -99.23245 | Niño Artillero |
| **61** | 18.89249 | -99.22855 | Hotel "La Iguana" |
| **62** | 18.91542 | -99.23407 | Plaza Juárez |
| **63** | 18.9159 | -99.23662 | Ayuntamiento de Cuernavaca |
| **64** | 18.88721 | -99.21885 | Glorieta Palmira |
| **65** | 18.91559 | -99.23242 | Procuraduría Agraria |
| **66** | 18.90659 | -99.2279 | Colegio Hamilton |
| **67** | 18.89914 | -99.2247 | Fraccionamiento Palmira |
| **68** | 18.89186 | -99.22226 | Condominios Rinconada |
| **69** | 18.88234 | -99.22041 | Secundaria No. 1 |
| **70** | 18.88644 | -99.23061 | Fabrica Coca Cola |
| **71** | 18.87903 | -99.22866 | Polvorín |
| **72** | 18.87944 | -99.22139 | CENIDET Campus Mecánica |

Tabla 1 Listado de vértices de Cuernavaca

Estos vértices se representan gráficamente en el mapa de la ciudad de la ciudad de la siguiente manera:

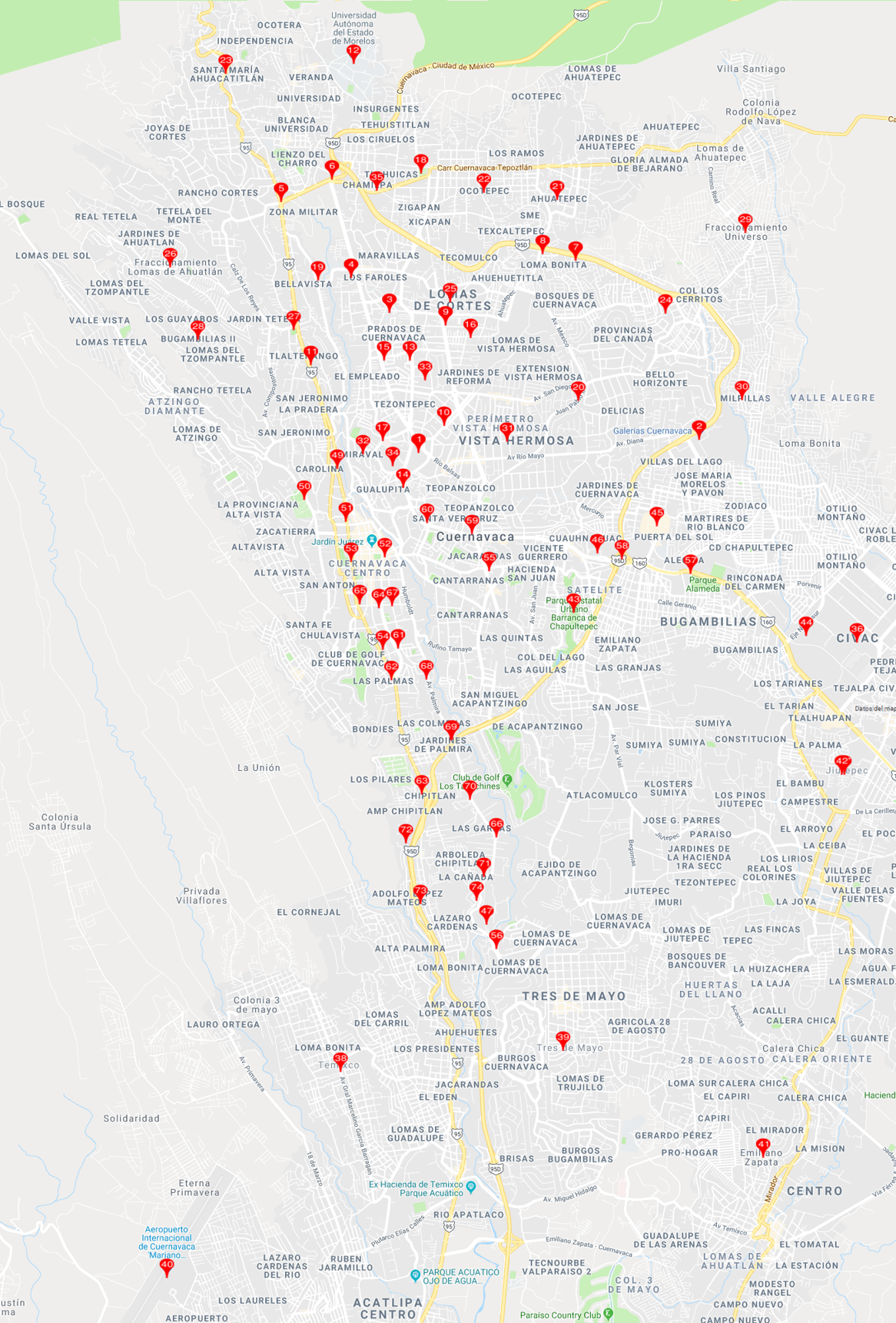


Ilustración 4 Mapa de Cuernavaca con los vértices definidos

### 3.1.3 Unión de los vértices

Ya que el número de vértices es elevado, no se puede realizar una representación gráfica de la unión de estos por lo que el grafo se representa por la siguiente lista de adyacencia donde los vértices de la izquierda están unidos con los de la derecha (Tabla 2):

|  |  |  |  |
| --- | --- | --- | --- |
| **Vértice** | **Relaciones** | **Vértice** | **Relaciones** |
| **1** | 10,31,33,13,15,17,34,14,59,2,20 | **37** | 36,44,57,41 |
| **2** | 20,30,24,58,45,31,10,1,14,60,59,46 | **38** | 40,73,56,39,41 |
| **3** | 9,25,4,15,13,22,18,35,6 | **39** | 41,38,40,56,73 |
| **4** | 6,19,3,25,9,8,22,35,6,15 | **40** | 41,38,39,56,73 |
| **5** | 6,19,27,26,28,12,23,11 | **41** | 42 y 37,39,38,40 |
| **6** | 5,35,4,12 | **42** | 58,46,55,59,68,61,67,52 |
| **7** | 8,24,20,16,9,25 | **43** | 36,57,45,2,30 |
| **8** | 7,21,22,35,2,25,9,16,20 | **44** | 2,57,58 |
| **9** | 25,3,16,33,13,15,4,22,18,35,6 | **45** | 58,43,55,59,45,31,2 |
| **10** | 33,16,9,13,20,,31,1,17 | **46** | 56,74,73,38,39 |
| **11** | 27,19,49 | **47** | 70,63,68 |
| **12** | 23,5 | **48** | 32,11,50,51 |
| **13** | 15,3,18,35,22,9,16,33,1,17 | **49** | 49,53,28,26,11,27 |
| **14** | 1,34,60,59,51,52 | **50** | 49,60,14,53,52 |
| **15** | 3,4,13,33,10,1,17, | **51** | 53,67,64,65,60,55,14,49,51 |
| **16** | 9,25,13,33,10,31,20 | **52** | 65,52,51 |
| **17** | 1,34,10,33,15,13,32 | **53** | 61,62,65 |
| **18** | 35,22,21,29 | **54** | 59,46,43,68,61,67,52 |
| **19** | 4,5,27,49 | **55** | 47,39,73,38,42 y 37 |
| **20** | 30,24,2,31,46,55,59,60,14,1,10,33,16,9 | **56** | 58,45,44,36 |
| **21** | 22,29,8 | **57** | 46,45,57,2,43,69 y 48 |
| **22** | 18,21,29,8 | **58** | 60,31,1,14,55,46,43 |
| **23** | 12,5,26 | **59** | 14,51,52,59,55 |
| **24** | 7,8,30,2,20,16,9,25 | **60** | 68,62,54,67 |
| **25** | 9,8,3,4,22,35 | **61** | 63,68,61,54 |
| **26** | 5,27,28,11,49,50,23 | **62** | 72,69 y 48 |
| **27** | 19,5,11,26,28,50,49 | **63** | 67,54,65 |
| **28** | 23,5,27,11,49,50 | **64** | 64,54,53 |
| **29** | 21,22,24,30 | **65** | 70,71 |
| **30** | 24,20,2,29 | **66** | 61,64,68,52,60,55 |
| **31** | 20,2,10,9,16,33,13,1,14,60,59,46,45 | **67** | 69 y 48,61,43,55,61,67,52 |
| **32** | 17,34,14,49,51 | **68** | 66,63,69 y 48 |
| **33** | 13,15,9,10,20,31,10,1,17 | **69** | 74,66 |
| **34** | 1,14,17,32 | **70** | 73,63 |
| **35** | 6,18,4,22,3,25,8 | **71** | 72,38,39,40,41 |
| **36** | 44,57,45,2,42 y 37 | **72** | 47,71 |

Tabla 2 Lista de adyacencia de los vértices

### 3.1.4 Ponderación de las aristas

La ponderación de las aristas se calculó utilizando la herramienta de GoogleMaps, al poner dos de los vértices en la herramienta se puede calcular cuánto tiempo se tarda en llegar del primer vértice al segundo. Se tomó el tiempo que tarda el recorrido en auto ya que es la única forma de limitar las opciones de caminos entre los vértices.

Se obtuvo la siguiente matriz donde la celda que relaciona el vértice en la fila y el vértice en la columna son los minutos que toma recorrer ese camino (Tabla 3):

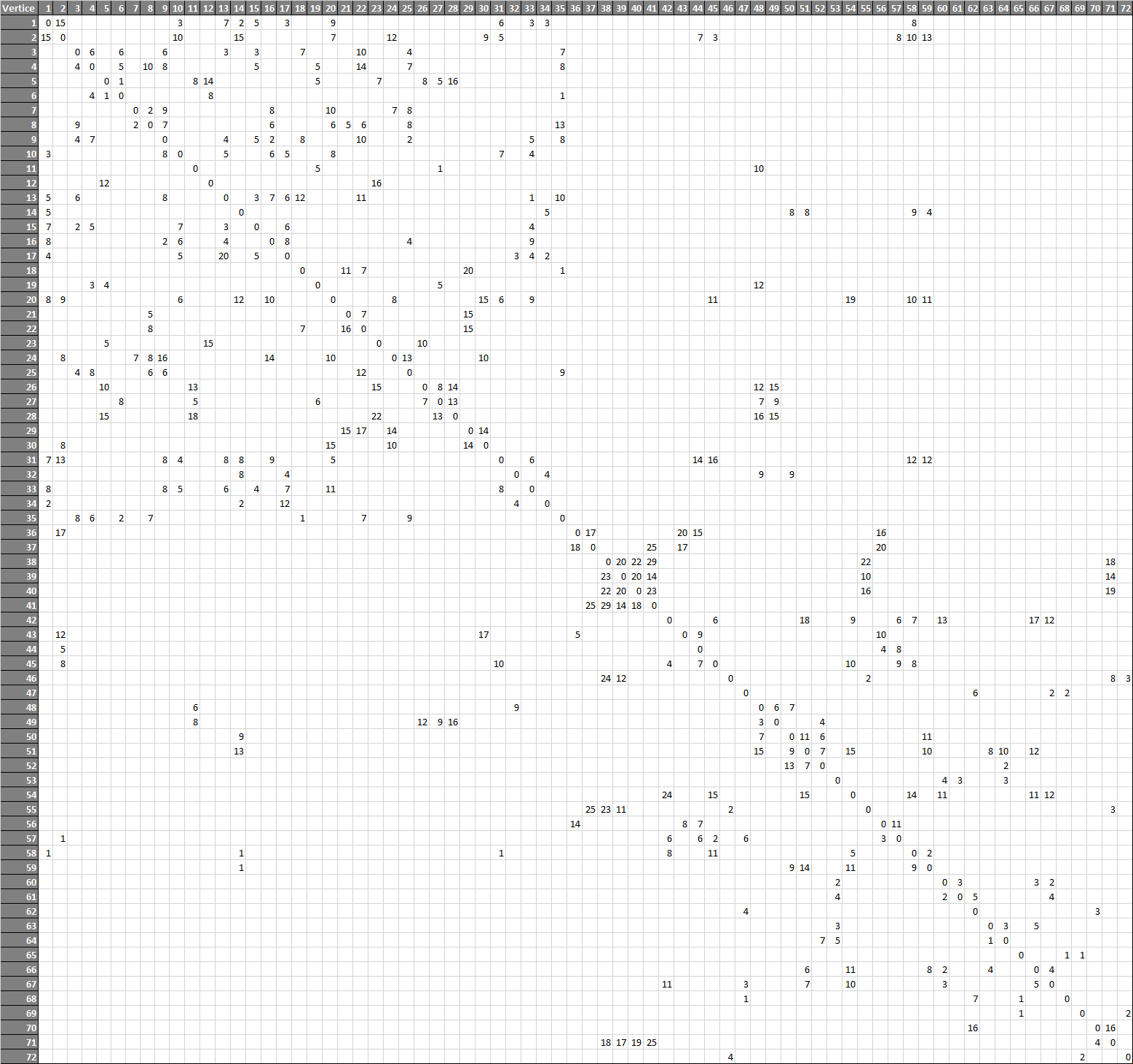


Tabla 3 Matriz del grafo ponderada

## 3.2 Obtención de las rutas más cortas

Para obtener las rutas que hacen falta en la matriz, aquellas que están en blanco y representan rutas que pasan por más de un par de vértices, Se utilizó una implementación del algoritmo de Floyd-Warshall en C#

### 3.2.1 Utilización del algoritmo de Floyd-Warshall en C#

Al alimentar el código en Código 2 con la matriz de 72 vértices de la ciudad de Cuernavaca se procesan los tiempos más corto en minutos que tomaría en llegar de un vértice a otro y se llenan los espacios en blanco de la matriz, además, al definir una nueva matriz y una condición en el algoritmo (Código 3) se extrae otra matriz donde se visualiza el camino que se debe que seguir.

public static void FloydWarshall(int[,] graph, int[,] pathS, int verticesCount)

{

int[,] distance = new int[verticesCount, verticesCount];

int[,] path = new int[verticesCount, verticesCount];

for (int i = 0; i < verticesCount; ++i)

for (int j = 0; j < verticesCount; ++j)

{

distance[i, j] = graph[i, j];

path[i, j] = pathS[i, j];

}

for (int k = 0; k < verticesCount; ++k)

{

for (int i = 0; i < verticesCount; ++i)

{

for (int j = 0; j < verticesCount; ++j)

{

if (distance[i, k] + distance[k, j] < distance[i, j])

{

if (distance[i, j] != distance[i, k] + distance[k, j])

{

path[i, j] = k+1;

}

distance[i, j] = distance[i, k] + distance[k, j];

}

}

}

}

Print(distance, verticesCount);

Console.Write("----------------------------");

Print(path, verticesCount);

}

Código 3 Modificación al código en C# de Floyd-Warshall

El código en C# del algoritmo de Floyd-Warshall que se utilizó en este proyecto puede ser implementado a cualquier grafo representado en una matriz ponderada.

# 4 Resultados

En esta sección se presentan los resultados del proyecto sobre el grafo de la ciudad de Cuernavaca: la matriz de tiempos y la matriz de rutas que representan los caminos más cortos entre los vértices del grafo.

## 4.1 Matriz de tiempos

La matriz de tiempos (Tabla 4) muestra cuanto tiempo en minutos tarda el recorrido de un vértice en una fila a un vértice en una columna.

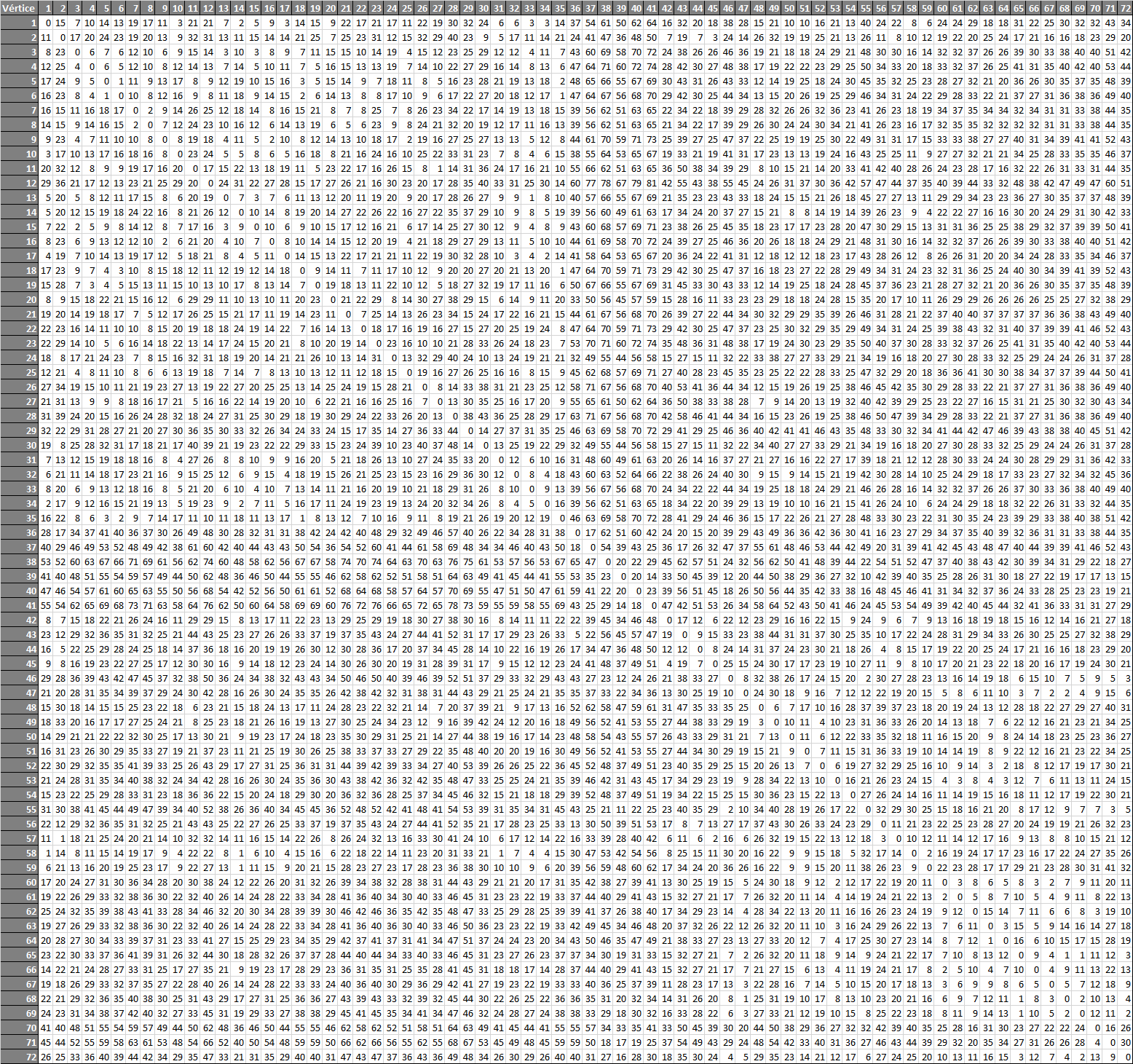


Tabla 4 Matriz de tiempos obtenida del algoritmo de Floyd-Warshall

## 4.2 Matriz de rutas

La matriz de rutas (Tabla 5) muestra por cuales nodos hay que pasar para ir de un vértice en una fila hacia un vértice en una columna. Por ejemplo, para ir del vértice 1 al 3 se debe pasar por el 15.

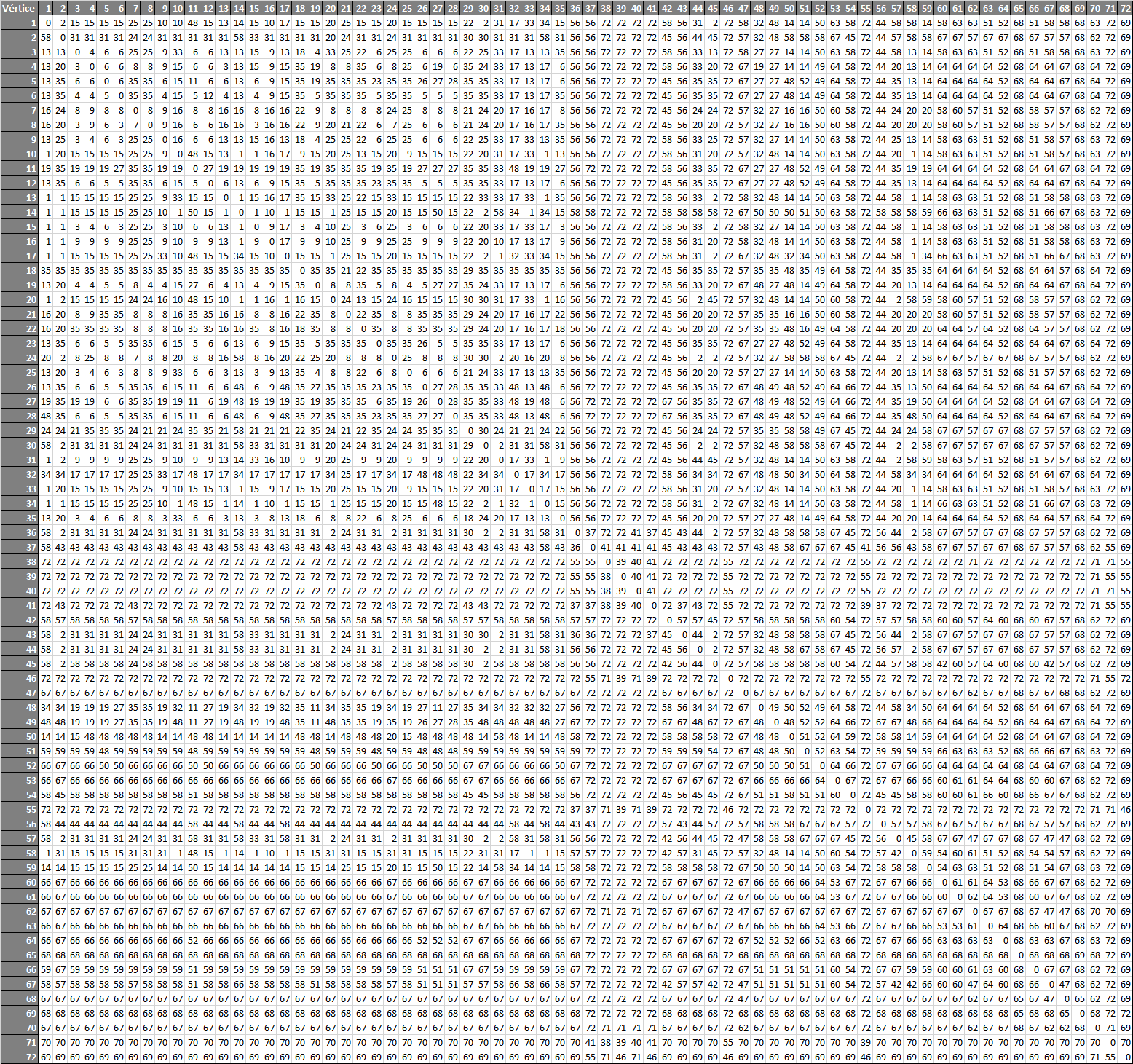


Tabla 5 Matriz de rutas obtenida del algoritmo de Floyd-Warshall

# 5 Conclusiones y trabajo futuro

## 5.1 Conclusiones

Los grafos son una herramienta muy útil para mapear y encontrar rutas optimas en cualquier tipo de situación, solo se necesitan definir puntos y obtener las distancias que existen entre ellos. En este caso se utilizó encontrar y explotar las rutas más cortas en la ciudad de Cuernavaca en Morelos, México. El objetivo principal del proyecto es que esta información pueda ser de utilidad a personas que quieran visitar el Centro Nacional de Investigación y Desarrollo Tecnológico por primera vez (aspirantes de estudios de postgrado).

Gracias al algoritmo de Floyd-Warshall se logra obtener las distancias entre pares de vértices no conectados, esto reduce mucho la carga de trabajo ya que solo se deben obtener manualmente aquellos caminos conectados directamente. Además, el mismo algoritmo se usa para obtener y saber cuáles las rutas de vértices que definen ese camino más corto.

Aunque el proyecto utilizo promedios de tiempos que toman los recorridos en automóvil entre un punto y otro, otros factores como la actualización de la infraestructura de la ciudad, el tráfico, el estado de los caminos, el clima y los accidentes pueden hacer que los tiempos que se muestran en el proyecto no sean exactos.

## 5.2 Trabajo Futuro

Las tareas que más tiempo consumieron en el proyecto fueron la obtención de los vértices y la ponderación de las aristas entre ellos, Se podría desarrollar una herramienta web que al presionar sobre un mapa defina un vértice y automáticamente calcule el tiempo que toma el recorrido de ese vértice a los vértices vecinos. De este modo la generación de grafos de mapas de ciudades sería más rápida.

Se debe introducir manualmente las matrices para el método que se utilizó (Algoritmo implementado en C#). Esto puede ser mejorado para recibir archivos CSV que contengan las matrices generadas en Excel y de esta forma prepara el código para funcionar sin tener que modificarlo.

Al utilizar las matrices de rutas y tiempos que fueron generadas en el proyecto se podría crear una aplicación que proporcione los caminos más cortos para llegar de un lugar a otro en la ciudad, estos caminos pueden ser dibujados con APIs de mapas disponibles en Internet.

# Bibliografía

CsharpStar. (30 de Marzo de 2017). C# – Floyd–Warshall Algorithm. Recuperado el Noviembre de 2018, de https://www.csharpstar.com/floyd-warshall-algorithm-csharp/

EcuRed. (Noviembre de 2018). Floyd-Warshall. Obtenido de https://www.ecured.cu/Floyd-Warshall

Ramirez, A., Diana, O., & David, O. (1 de Noviembre de 2016). Aplicaciones de los Grafos. Recuperado el Noviembre de 2018, de https://sites.google.com/site/aplicaciongrafos/

Voroncovs, A. (2015). The Floyd-Warshall Algorithm. Recuperado el Noviembre de 2018, de https://www-m9.ma.tum.de/graph-algorithms/spp-floyd-warshall/index\_en.html

Wikipedia. (9 de Octubre de 2018). C Sharp. Recuperado el Noviembre de 2018, de https://es.wikipedia.org/wiki/C\_Sharp

Wikipedia. (9 de Octubre de 2018). Floyd–Warshall algorithm. Recuperado el Noviembre de 2018, de https://en.wikipedia.org/wiki/Floyd–Warshall\_algorithm

Wikipedia. (11 de Noviembre de 2018). Grafo. Recuperado el Noviembre de 2018, de https://es.wikipedia.org/wiki/Grafo

Wikipedia. (12 de Noviembre de 2018). Pseudocódigo. Recuperado el Noviembre de 2018, de https://es.wikipedia.org/wiki/Pseudocódigo

Wikipedia. (4 de Noviembre de 2018). Teoría de grafos. Recuperado el Noviembre de 2018, de https://es.wikipedia.org/wiki/Teoría\_de\_grafos