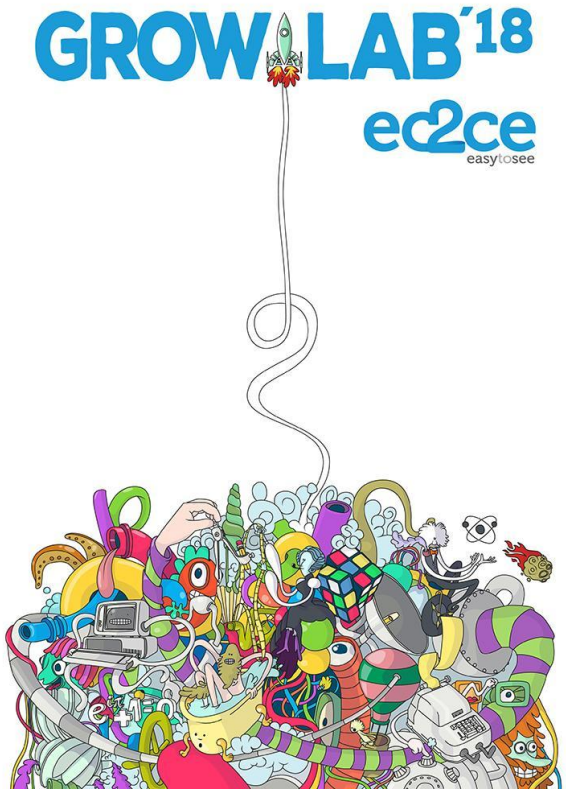
MATEMÁTICAS DISCRETAS

1ª Edición Growlab

Ámbito: Matemáticas Aplicadas



Trabajo realizado por:
Fernando Cáceres Plaza
Rafael Martín Arenas
Guillermo Ramón Soria
Juan Antonio Romero Muñoz

Tutor:
Juan José Jiménez Ruiz

ec2ce easytosee

Jesuitas
Fundación Loyola
Colegio Portaceli

Índice

1. Introducción:

• 1.1 ¿Por qué este concurso?

Decidimos participar en este concurso ya que se nos presentó en el colegio como una posibilidad de aprender a desarrollar un proyecto por nuestra cuenta mediante la investigación. Además, durante el proceso tendríamos la oportunidad de ampliar nuestros conocimientos en ámbitos de las matemáticas que normalmente no son objeto de estudios en la ESO ni en Bachillerato.

Por otro lado, nos llamó la atención la oportunidad de poder estar trabajando durante un tiempo en la empresa dedicándonos a estudiar un problema real e intentar darle una solución efectiva. A través de este proyecto pretendíamos también conseguir experiencia y práctica para futuros trabajos reales.

• 1.2 ¿Por qué este trabajo?

Cuando decidimos entrar a participar en este concurso, nuestro profesor nos ofreció una serie de posibles proyectos a realizar, de los cuales uno era este, un trabajo sobre grafos. Tras ver todos las opciones, esta fue la más interesante para nosotros ya que veíamos que podría ser utilizado para solucionar problemas de empresas reales y así poder investigar en una herramienta matemática muy útil para el mundo que nos rodea.

• 1.3 ¿Cómo hemos trabajado?

Para realizar este trabajo tuvimos que dividir las tareas asignando cada aspecto a uno de los componentes del grupo en función de sus preferencias y de sus facultades.

Los primeros días recopilamos información sobre grafos y matrices, investigando a la vez como operar con ellos. Mientras, fuimos redactando toda la información en word para llevar un registro de todos los datos para poder hacer uso de ellos a lo largo del desarrollo del trabajo.

Tras esto, buscamos las principales ciudades y las carreteras que las conectan además de las distancias, tiempo y coste de traslado entre ellas. Durante este tiempo hicimos algunas reuniones por las tardes para organizar el trabajo y poner en común los resultados que encontrábamos.

Tras recopilar toda la información empezamos a usar el programa Matlab en el que fuimos realizando las cuentas necesarias para resolver el problema.

2. Objetivo

Nuestro objetivo en este proyecto ha sido basarnos en la realidad de una empresa que quiere establecer una central de distribución en Andalucía de la forma más eficaz posible, tardando el mínimo tiempo para llegar a las ciudades y que sea lo más barato posible. Para esto hemos tenido en cuenta:

- Ciudades más pobladas de Andalucía (más de 100000 habitantes).
- Carreteras principales entre las ciudades. Dentro de las cuales hemos tenido en cuenta:
    - Distancias recorridas en carretera entre las ciudades.
    - Tiempo que se tarda en llegar por cada carretera.
    - Coste de estos recorridos.

3. Base matemática

Para llevar a cabo este proyecto hemos necesitado una gran ayuda de las matemáticas, en concreto, del álgebra. Una rama de ésta son los grafos en los que nos centramos a continuación, ayudándonos de las matrices las cuales han sido vitales para llevar a cabo nuestro proyecto y que explicaremos más adelante.

3.1 Grafos

Definición: La palabra grafo etimológicamente significa "*grabar o escribir*". Consiste en un conjunto de **vértices o nodos** unidos por enlaces llamados **aristas o arcos**. Son objeto de estudio de la teoría de grafos.

La primera mención sobre los grafos fue en el Siglo XVIII (1736) por Leonhard Euler, matemático y físico que destacó por ser una de las figuras más importantes de su tiempo en esta materia.

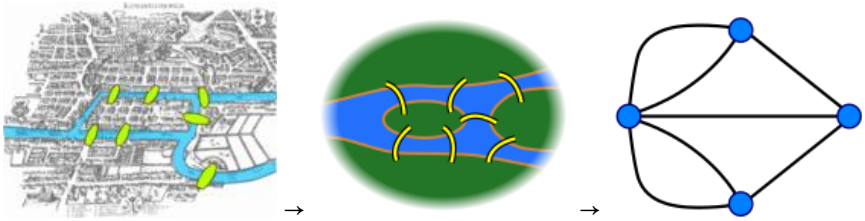
Problema de los puentes de Königsberg:

El problema de los puentes de Königsberg, también llamado más específicamente problema de los siete puentes de Königsberg, es un célebre problema matemático, resuelto por Leonhard Euler en 1736 y cuya resolución dio origen a la teoría de grafos. Su nombre se debe a Königsberg, la ciudad de Prusia Oriental y luego de Alemania que desde 1945 se convertiría en la ciudad rusa de Kaliningrado.Esta ciudad es atravesada por el río Pregel, en ruso «Pregolya», el cual se bifurca para rodear con sus brazos a la isla Kneiphof, dividiendo el terreno en cuatro regiones distintas, las que entonces estaban unidas mediante siete puentes llamados Puente del herrero, Puente conector,

Puente verde, Puente del mercado, Puente de madera, Puente alto y Puente de la miel. El problema fue formulado en el siglo XVIII y consistía en encontrar un recorrido para cruzar a pie toda la ciudad, pasando sólo una vez por cada uno de los puentes, y regresando al mismo punto de inicio.

La respuesta es negativa, es decir, no existe una ruta con estas características. El problema puede resolverse probando todos los posibles recorridos existentes. Sin embargo, Euler en 1736 en su publicación *«Solutio problematis ad geometriam situs pertinentis»* demuestra una solución generalizada del problema, que puede aplicarse a cualquier territorio en que ciertos accesos están restringidos a ciertas conexiones, tales como los puentes de Königsberg.

Para dicha demostración, Euler recurre a una abstracción del mapa, enfocándose exclusivamente en las regiones terrestres y las conexiones entre ellas. Cada puente lo representó mediante una línea que unía a dos puntos, cada uno de los cuales representaba una región diferente. Así el problema se reduce a decidir si existe o no un camino que comience por uno de los puntos azules, transite por todas las líneas una única vez, y regrese al mismo punto de partida.



3.2 Tipos de grafos

- Grafo simple: grafo que acepta una sola arista uniendo dos vértices cualesquiera.

- Multi grafo: grafo que acepta más de una arista entre dos vértices cualesquiera

- Pseudografo: grafo que incluye algún lazo

- Grafo orientado: grafos cuales tienen las aristas una orientación representadas por una flecha.

- Grafo etiquetado: grafos en el que se le ha añadido un *peso* a las aristas.

- Grafo aleatorio: grafos cuyas aristas están asociadas a una probabilidad

- Hipergrafo: grafos en los cuales tienen más de dos extremos.

- Grafo plano: grafos cuyos vértices y aristas pueden ser representados sin ninguna intersección entre ellos.

- Grafo regular: grafo donde todos sus vértices tienen el mismo grado de valencia.

3.3 Características de grafos

- Vértices/Nodos: Se asignan con un pequeño círculo y se nombran con un número o letra.
- Lados/aristas: Unen los vértices y también se les asigna un número o letra.
- Lados paralelos: Lados que unen los mismos vértices.
- Lazos/Bucles: Arista que sale de un vértice y regresa al mismo.

De esta forma en el ejemplo representado en el grafo tendríamos las siguientes características:

Vértices: {1,2,3,4,5} Aristas: {a,b,c,d,e,f,g,h} Lados paralelos: (c,g) Bucles: {h}

Pero es difícil trabajar con grafos tanto matemáticamente como computacionalmente. Para ayudarnos en esta tarea, las matemáticas nos ofrecen las matrices.

3.4 Matrices

Se trata de un conjunto de números y expresiones ordenadas en filas y columnas en el que cada número que forma parte de la matriz es un elemento.

El número de filas y columnas de una matriz es su dimensión y se nombra como m·n (m=número de filas,n=número de columnas). Si tiene el mismo número de filas que de columnas se dice que es de “orden 1,2,3...” -El conjunto de matrices con m filas y n columnas se denomina matriz Am·n.

Un elemento perteneciente a la fila i y columna j se denomina aij.

Para la aplicación de grafos nos es especialmente útil la llamada matriz de adyacencia.

3.5 Matriz de adyacencia

La matriz de adyacencia es una matriz cuadrada que se utiliza como una forma de representar relaciones binarias.

Construcción

1. Se crea una matriz cero, cuyas columnas y filas representan los *nodos* del grafo.
2. Por cada arista que une a dos nodos, se suma 1 al valor que hay actualmente en la ubicación correspondiente de la matriz.
3. Si tal arista es un bucle y el grafo es no dirigido, entonces se suma 2 en vez de 1.

Finalmente, se obtiene una matriz que representa el número de aristas (relaciones) entre cada par de nodos (elementos).

Existe una matriz de adyacencia única para cada grafo (sin considerar las permutaciones de filas o columnas), y viceversa.

De esta forma el ejemplo del grafo anterior, la matriz de adyacencia quedaría de la siguiente forma:

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 2 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$

3.6 Algoritmo de Dijkstra

Este algoritmo es el que se usa para encontrar el camino más corto desde un nodo a otro teniendo en cuenta el peso de las aristas (distancias entre nodos). Fue usado por primera vez por Edsger Dijkstra en 1959. El algoritmo consiste en lo siguiente.

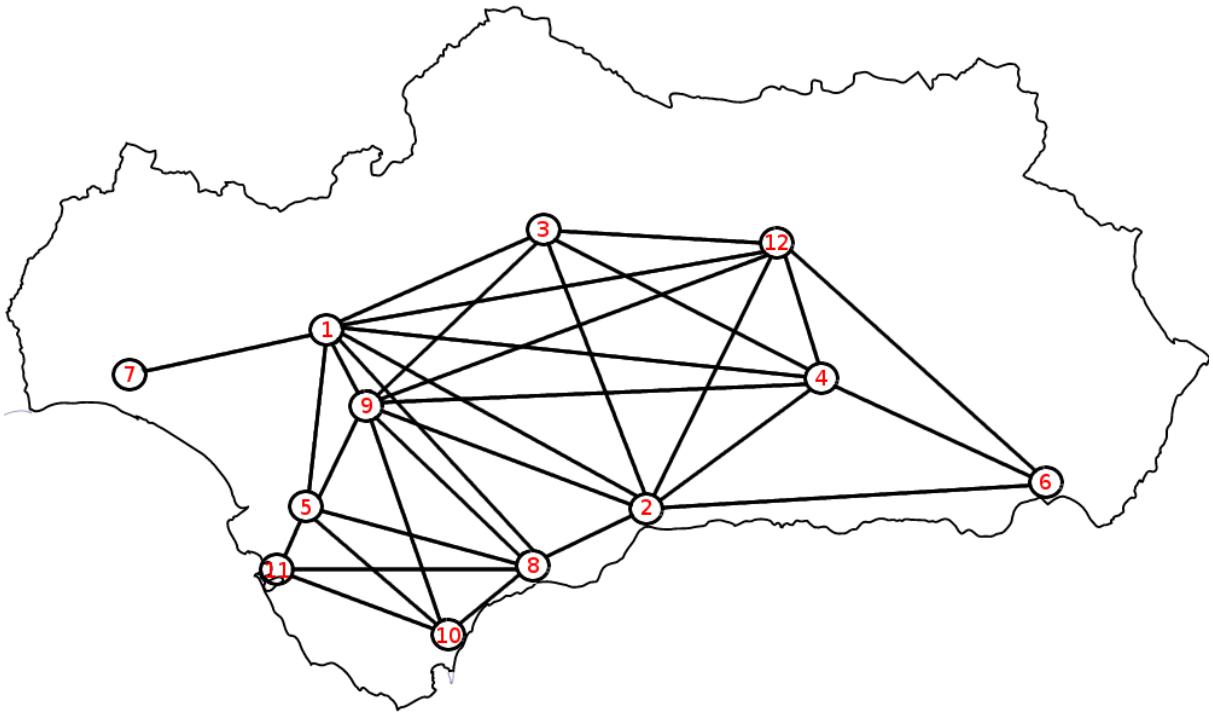
Primero marcamos todos los vértices como no utilizados. El algoritmo parte de un vértice origen que será ingresado, a partir de ese vértice evaluaremos sus adyacentes, como Dijkstra usa una técnica llamada Greedy.

La técnica Greedy utiliza el principio que consiste en que para que un camino sea óptimo, todos los caminos que contiene también deben ser óptimos.

Entre todos los vértices adyacentes, buscamos el que esté más cerca de nuestro punto origen, lo tomamos como punto intermedio y vemos si podemos llegar más rápido a través de este vértice a los demás. Después escogemos al siguiente más cercano (con las distancias ya actualizadas) y repetimos el proceso. Esto lo hacemos hasta que el vértice no utilizado más cercano sea nuestro destino. Al proceso de actualizar las distancias tomando como punto intermedio al nuevo vértice se le conoce como relajación (relaxation)

4. Andalucía en un grafo

Para poder aplicar todos estos conocimientos a una situación real tuvimos que utilizar los grafos y hacer uso de nuestro entusiasmo por aprender cosas nuevas. Así pudimos formar un mapa con los nodos enumerados de mayor a menor ya que hemos tenido en cuenta aquellas poblaciones mayores a 100000 habitantes, y las carreteras principales (autopistas, autovías y carreteras nacionales) que unían estos puntos sin pasar por otros. Por ejemplo: carreteras que unan Sevilla y Málaga sin pasar por otras poblaciones, todo esto teniendo en cuenta los kilómetros de dichas carreteras y el coste que conllevan. Para terminar y poder hacer los cálculos hemos formado una matriz con las carreteras que hemos seleccionado y los puntos.



1º *Sevilla* **2º** *Málaga* **3º** *Córdoba* **4º** *Granada* **5º** *Jerez de la frontera* **6º** *Almería*	**7º** *Huelva* **8º** *Marbella* **9º** *Dos Hermanas* **10º** *Algeciras* **11º** *Cádiz* **12º** *Jaén*

[1-2]: ·A-92 / 205km, 2h 21min.
·A-92 y A-357 / 199km, 2h 32min.

[1-3]: ·A-4 / 141km, 1h 37min.
·A-431 / 133km, 1h 54min.

[1-4]: ·A-92 / 250km, 2h 42min.

[1-5]: ·AP-4 / 90'9km, 1h 11min (peaje)
·AP-4 y CA-31 / 91'9km, 1h 18min. (peaje)

[1-7]: ·A-49 / 92'9km, 1h 8min.

[1-8]: ·A-92 / 209km, 2h 36min.

[1-9]: ·A-8032 / 14,1km, 28min.
·Av. de las Universidades / 15'6km, 27 min.

[1-12]: ·A-4 y A-316 /246km, 2h 43min.

[2-3]: ·A-45 / 158km, 1h 50min.
·A-92 / 212km, 2h 34min.

[2-4]: ·A-92 / 126km, 1h 36min.

[2-6]: ·A-7 / 202km, 2h 13min.

[2-8]: ·AP-7 / 60'7km, 53min.
·A-357 y A-355 / 62'7km, 1h 2min.

[2-9]: ·A-92 / 208km, 2h 17min.
·A-357 y A-92 / 203km, 2h 29min.

[2-12]: ·A-45 / 190km, 2h 17min.

[3-4]: ·N-432 / 172km, 2h 28min.

[3-9]: ·A-4 / 152km, 1h 43min.
·A-92 / 164km, 1h 58min.

[3-12]: ·A-306 / 113km, 1h 27min.
·A-4 y A-311 / 120km, 1h 28min.

[4-6]: ·A-92 / 167km, 1h 48min.

[4-9]: ·A-92 / 250km, 2h 42min.

[4-12]: ·A-44 / 94'1km, 1h 5min.

[5-8]: ·A-381 y Ap-7 / 171km, 1h 49min (peaje)
·A-384 y A-397 / 174km, 2h 31min (peaje)

[5-9]: ·AP-4 / 77'3km, 56min (peaje)

[5-10]: ·A-381 / 97'8km, 1h 6min.

[5-11]: ·AP-4 / 36'7km, 35min.

[6-12]: ·A-92 / 225km, 2h 18min.

[8-9]: ·A-397 / 180km, 2h 36min. (peaje)
·A-92 / 213km, 2h 38min.

[8-10]: ·AP-7 y A-7 / 80'5km, 59min.

[8-11]: ·AP-7 y A-381 / 179km, 2h 4min.

[9-10]: AP-4 y A-381 / 170 km, 1h 41 min.

[9-12]: ·A-92 / 251km / 2h 56min.

[10-11]: N-340 y A-48 / 125km, 1h 38min.
·A-381 / 106km, 1h 17min.

A continuación presentamos dos matrices que resumen los datos que hemos recopilado.
La primera matriz es la matriz de adyacencia, y la segunda es una matriz en la que el elemento a_{ij} son los kilómetros que habría que recorrer (directamente o pasando por algún otro nodo) para llegar desde el nodo i al nodo j.

Matriz de adyacencia:

$$\begin{pmatrix} 0 & 2 & 2 & 1 & 2 & 0 & 1 & 1 & 2 & 0 & 0 & 1 \\ 2 & 0 & 2 & 1 & 0 & 1 & 0 & 2 & 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 1 & 1 & 0 \\ 2 & 2 & 2 & 1 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 2 & 0 & 0 \\ 1 & 1 & 2 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Matriz kilómetros

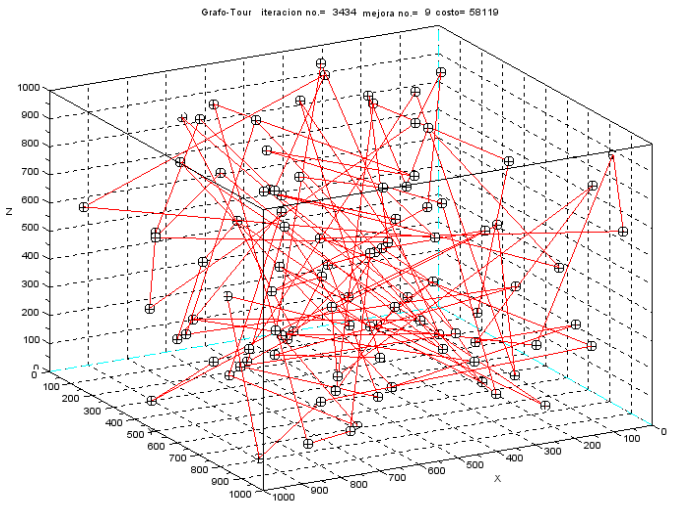
$$\begin{pmatrix} 000 & 199 & 133 & 250 & 091 & 417 & 093 & 210 & 014 & 184 & 128 & 246 \\ 199 & 000 & 158 & 126 & 232 & 202 & 310 & 061 & 203 & 141 & 240 & 190 \\ 133 & 158 & 000 & 172 & 229 & 339 & 226 & 219 & 152 & 322 & 266 & 113 \\ 250 & 126 & 072 & 000 & 327 & 167 & 343 & 187 & 250 & 267 & 366 & 094 \\ 091 & 232 & 229 & 327 & 000 & 434 & 184 & 171 & 077 & 098 & 037 & 349 \\ 417 & 202 & 339 & 167 & 434 & 000 & 510 & 263 & 417 & 343 & 442 & 225 \\ 093 & 310 & 226 & 343 & 184 & 510 & 000 & 303 & 107 & 277 & 221 & 339 \\ 210 & 061 & 219 & 187 & 171 & 263 & 303 & 000 & 180 & 080 & 179 & 251 \\ 014 & 203 & 152 & 250 & 077 & 417 & 107 & 180 & 000 & 170 & 114 & 251 \\ 184 & 141 & 322 & 267 & 098 & 343 & 277 & 080 & 170 & 000 & 106 & 331 \\ 128 & 240 & 266 & 366 & 037 & 442 & 221 & 179 & 114 & 106 & 000 & 386 \\ 246 & 190 & 113 & 094 & 349 & 225 & 339 & 251 & 251 & 331 & 386 & 000 \end{pmatrix}$$

5. Resolución con Matlab

Matlab es una herramienta de cálculo, simulación y modelado matemático. Entre sus funciones básicas se encuentra:

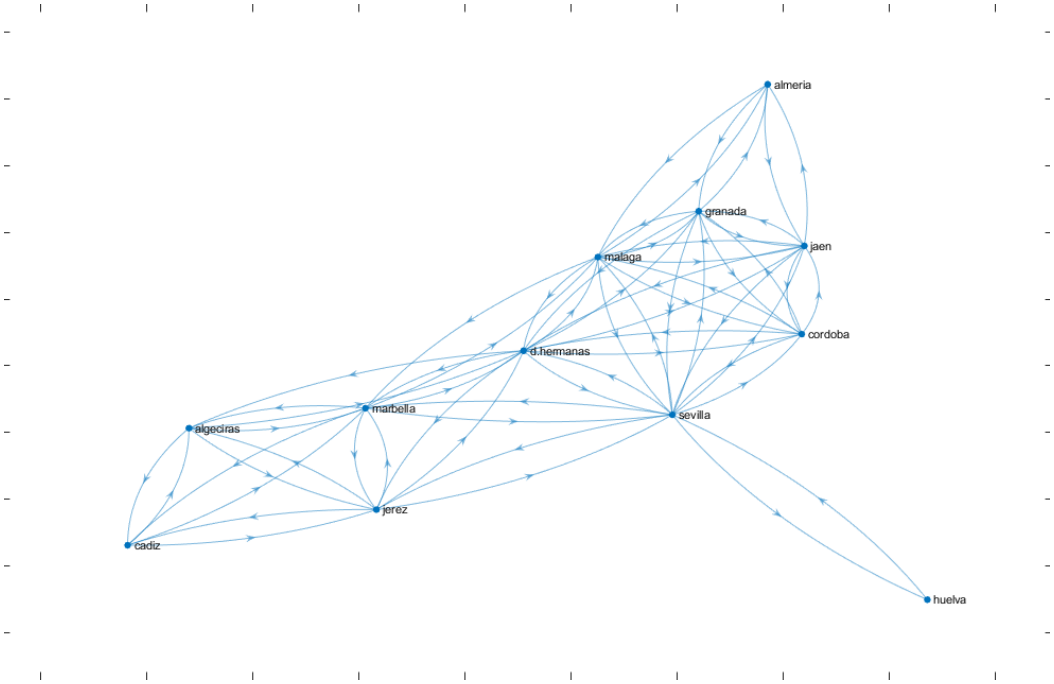
- ❏ Manipulación de Matrices.
- ❏ La representación de datos y funciones.
- ❏ Implementación de algoritmos.
- ❏ Creación de interfaces de usuario (GUI).
- ❏ Comunicación con programas en otros lenguajes y con otros dispositivos Hardware.

Esta potente herramienta de software matemático nos resulta muy útil por su relativa simpleza de su lenguaje de programación (Lenguaje M) y sus prestaciones en la creación de matrices, implementación de algoritmos y representación de grafos de adyacencia.



5.1 Planteamiento del algoritmo con MatLab

Con ayuda de MatLab y el algoritmo de Dijkstra transformamos la matriz de Andalucía en un digrafo al principio para obtener sus sentidos y más tarde aplicamos el algoritmo para averiguar el camino más corto en Kilómetros, el más corto en tiempo y el más barato, y así de este modo poder calcular la ruta más corta y barata que podría hacer un camión a la hora de repartir un producto.



Por ejemplo si se quiere llevar un producto/paquete de Sevilla a Almería podemos calcular la ruta más corta y la más barata siendo este el camino que pasa por Granada recorriendo 417 Kilómetros, tardando 4 horas y 32 minutos, consumiendo un total de 125 litros de gasolina gastando 162.5 euros en el viaje. La función que utilizamos en matlab para hallar dicho resultado es la siguiente:

```
function [e, L] = dijkstra(D,s,t)
if s==t
    e=0;
    L=[s];
else
D = setupgraph(D,inf,1);
if t==1
    t=s;
end
D=exchangenode(D,1,s);
lengthA=size(D,1);
W=zeros(lengthA);
for i=2 : lengthA
    W(1,i)=i;
    W(2,i)=D(1,i);
end
for i=1 : lengthA
    D(i,1)=D(1,i);
    D(i,2)=i;
end
D2=D(2:length(D),:);
L=2;
```

```
while L<=(size(W,1)-1)
   L=L+1;
   D2=sortrows(D2,1);
   k=D2(1,2);
   W(L,1)=k;
   D2(1,:)=[];
   for i=1 : size(D2,1)
      if D(D2(i,2),1)>(D(k,1)+D(k,D2(i,2)))
         D(D2(i,2),1) = D(k,1)+D(k,D2(i,2));
         D2(i,1) = D(D2(i,2),1);
      end
   end
   for i=2 : length(D)
      W(L,i)=D(i,1);
   end
end
if t==s
   L=[1];
else
   L=[t];
end
e=W(size(W,1),t);
L = listdijkstra(L,W,s,t);
end
```

Como punto de equilibrio el resultado después de usar el algoritmo con los demás nodos nos dió Dos Hermanas debido a su posición centralizada en el mapa permitiendo no tener que pasar por más de un nodo antes de llegar al nodo objetivo. Estableciendo la central de distribución en esta ciudad conseguimos que los kilómetros totales que se tendrían que recorrer hasta el resto de nodos sea mínimo y por tanto se optimiza tanto el tiempo de distribución como el gasto que supondría en suministro de combustible.

6. Conclusión y mejora

Empezamos este proyecto con la duda de como acabaría nuestro trabajo, cuestionandonos de lo que podíamos llegar a hacer. Todo lo que se necesita para conseguir desarrollar un proyecto es tener claro el objetivo, un buen trabajo en equipo y mostrar devoción por aprender y descubrir sobre el mundo laboral de las empresas.

En un principio nos repartimos los trabajos teniendo en cuenta los puntos fuertes de cada componente del grupo, para que cada uno realizase el trabajo que mejor se le diese. Hicimos nuestra parte además de apoyarnos a través de aplicaciones de comunicación (Skype, Discord) cuando algún compañero necesitaba ayuda.

Siempre que podíamos, quedamos en casa de uno de los componentes del grupo y aportamos ideas que pudieran aportar avances para el proyecto.

Cada dos semanas siempre había una reunión en el colegio para preguntar dudas a nuestro tutor que realizaba una revisión de nuestro trabajo y nos guiaba de cómo seguir.

Nos ha resultado tremendamente útil este proyecto estudiantil para promover el trabajo en equipo, el esfuerzo y las ganas de iniciar "startups" curiosas para un futuro.

Ante todo, este proyecto nos ha enseñado a utilizar herramientas del ámbito informático como el "MatLab" y la herramienta de diseño "Gimp 2" que seguro en un futuro nos será útil y práctico.

Como mejora de este proyecto, podríamos considerar la posibilidad de colocar la central de distribución en una arista, es decir, cerca de alguna carretera. En vez de dentro de una ciudad tal y como hemos realizado en este trabajo. Este cambio dificultaría la resolución del mismo pero le daría un toque más de realismo al problema y por tanto a su solución.

7. Bibliografía

- https://es.wikipedia.org/wiki/Problema_de_los_puentes_de_K%C3%B6nigsberg
- https://es.wikipedia.org/wiki/Grafo
- https://es.wikipedia.org/wiki/Teor%C3%ADa_de_grafos
- http://micaminomaster.com.co/grafo-algoritmo/dijkstra-y-floyd-warshall-algoritmo-mas-rapido-matlab/
- https://www.google.es/maps?hl=es&tab=wl
- https://jariasf.wordpress.com/2012/03/19/camino-mas-corto-algoritmo-de-dijkstra/
- https://es.wikipedia.org/wiki/Matriz_de_adyacencia
- http://interactivepython.org/runestone/static/pythoned/Graphs/UnaMatrizDeAdyacencia.html
- https://prezi.com/_0bkucbgzv_5/elementos-y-caracteristicas-de-los-grafos/
- https://www.vitutor.net/1/matrices.html
- https://es.mathworks.com/products/matlab.html
- https://www.youtube.com/watch?v=MEw9Ne1ALCA
- https://es.wikipedia.org/wiki/MATLAB
- https://es.mathworks.com/products/matlab.html
- https://www.uam.es/UAM/Licencia-Campus-y-Formaci%C3%B3n-en-MATLAB/1446757505518.htm?language=es&pid=1234886352083&title=Licencia%20Campus%20y%20Formaci%C3%B3n%20en%20MATLAB
- https://es.mathworks.com/matlabcentral/fileexchange/20025-dijkstra-s-minimum-cost-path-algorithm
- https://es.mathworks.com/matlabcentral/fileexchange/36140-dijkstra-algorithm

8. Agradecimientos

Gracias a EC2CE y su proyecto Growlab por brindarnos la oportunidad de llevar a cabo un proyecto en grupo y así aprender a construir un proyecto desde cero y aprender nuevos conocimientos. Y también gracias al Colegio Portaceli por darnos la oportunidad de entrar en este tipo de proyectos y así mejorar como personas y formarnos para el día del mañana.
Gracias a Juan José Jiménez Ruiz por meternos en este proyecto y sacrificar su tiempo libre para prestarnos su ayuda y así poder ir prosperando en el proyecto.
Gracias a Pablo Ramón Soria e Ignacio Cáceres Plaza por ayudarnos con los programas que apenas sabíamos usar. También agradecer a Antonio Marín Maqueda por ayudarnos a terminar el proyecto en el matlab. Y por último, gracias a nuestros padres por apoyarnos en este tiempo y darnos ánimos para seguir adelante.

Proyecto realizado por:
Fernando Cáceres Plaza
Rafael Martín Arenas
Guillermo Ramón Soria
Juan Antonio Romero Muñoz