

# Reflexión Individual 2

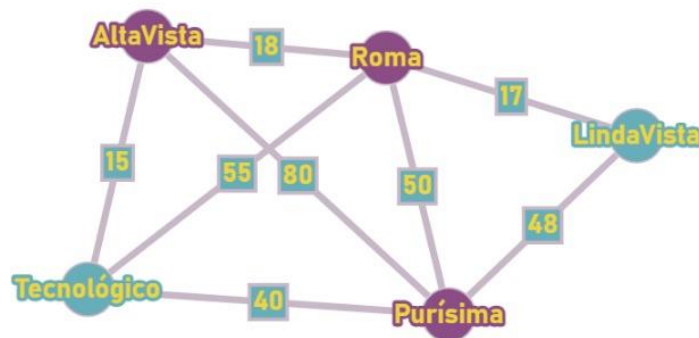
Erik Josías González Lucas | A01424067 | Campus Monterrey

## Punto 1 – Algoritmo de Kruskal MST

El punto consistía en desplegar la información para cablear todo con fibra óptica, y en base a eso entendimos que podíamos utilizar el algoritmo de Kruskal que busca la sumatoria mínima para la conexión de un grafo entre sí. No utilizamos una modificación considerable, nos guiamos por lo que vimos en clase haciendo leves modificaciones.

## Punto 2 – Algoritmo de Dijkstra.

Primero es digno de considerar que la sugerencia fue usar el algoritmo de TSP (Travelling Salesman Problem), y así fue en un inicio cuando comenzamos a trabajar con la actividad, y su funcionamiento fue el esperado. Dado que el algoritmo de TSP es para encontrar la ruta más corta para visitar a todas las ciudades, y en este caso colonias, al menos una vez y regresar al nodo de origen. El algoritmo funcionó correctamente, el problema es que visitábamos, precisamente, todas las colonias, dígame centrales o no, y según la condición del punto dos es visitar todas las colonias no centrales, con la posibilidad de pasar por colonias que sí son centrales, no es óptimo utilizar TSP ya que considera sí o sí, no como una posibilidad, pasar por colonias centrales. Ante tal situación consideramos un acercamiento un tanto más sencillo.



Primeramente, nos preguntamos cómo pasar por las colonias no centrales, y a partir de eso decidimos hacer un vector donde almacenáramos todas las colonias no centrales, a partir de ahí nos preguntamos ¿cuál debe de ser el camino más corto de un nodo a otro?, en este caso las distancias más cortas entre los elementos del vector. Una modificación del algoritmo de Dijkstra nos permite saber la distancia más corta entre un nodo y otro, buscando por todos los caminos posibles. A partir de ese caso base podemos iterar a través del vector, hasta que hallamos pasado por todos los elementos del vector, que en este caso serían todas las colonias no centrales. Así mismo dentro del algoritmo vamos añadiendo las rutas dentro de un vector *path* de tipo string, y si en dado caso sí pasa por una colonia que sí es central también se añade dentro de *path* para finalmente obtener toda la ruta completa, así como todo el costo del viaje por el vector de colonias no centrales.

### **Punto 3 – Algoritmo de Floyd**

En el tercer punto teníamos que obtener la ruta más óptima para ir entre las centrales, en un principio consideramos que era semejante al punto anterior, pero al inverso, es decir, si en el punto dos teníamos que ir por todas las no centrales con la posibilidad de pasar por centrales, aquí teníamos que ir por colonias centrales con la posibilidad de pasar por no centrales, y en cierto sentido así es, aunque cabe destacar que en el punto dos solo imprime una ruta, y el enfoque que debíamos de darle fue era la ruta entre centrales por cada una de ellas. Se ocupó para sacar Floyd para obtener la mejor ruta entre los nodos de las colonias centrales, básicamente ocupamos un doble for para obtener todas las combinaciones posibles, añadiendo dos condicionales que verificaban que eran centrales las colonias por las cuales pasabas, evitando posibles redundancias entre las rutas.

### **Punto 4 – Mínima distancia entre dos puntos**

Este punto fue sencillo, ya que solo consistía en saber la mínima distancia entre dos puntos, hicimos uso de la fórmula matemática. Ya dentro de un for solo iterábamos para obtener la distancia más corta para finalmente regresarla.