

THE DISCRETE PATHFINDER

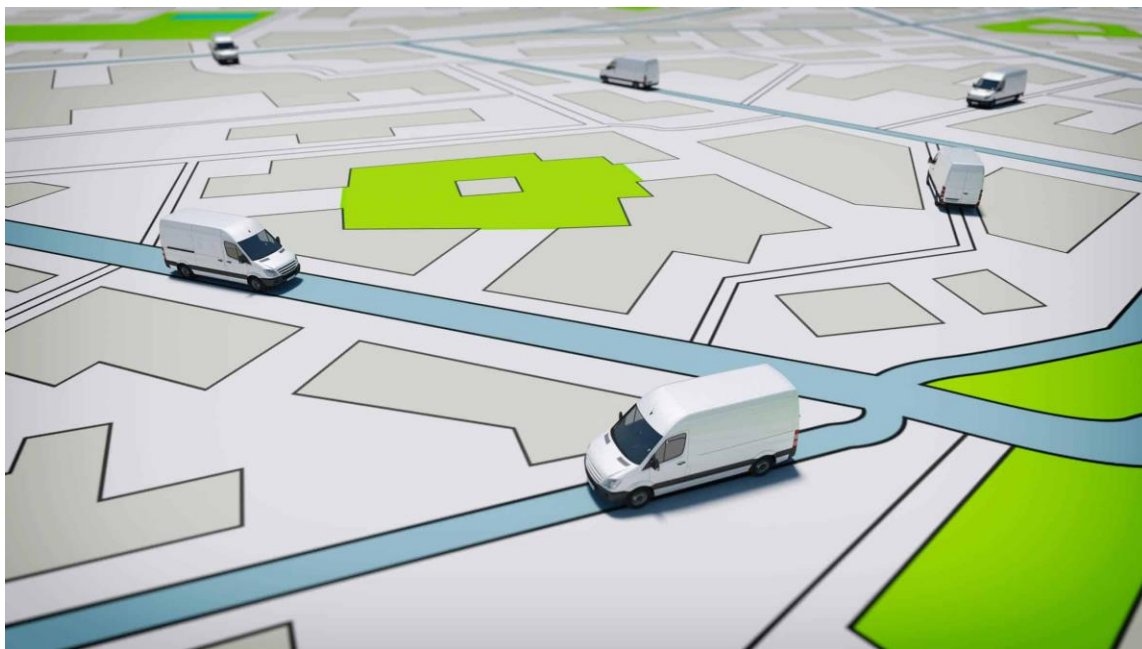


Nikolai Shelyn Jaimes Guerrero - 2201573
Ángel Fernando Ortega García - 2211583
Juan Esteban Sepúlveda Tapias - 2210095



Universidad
Industrial de
Santander





MOTIVACIÓN

- Para las empresas relacionadas con el transporte, crear rutas siempre se ha tratado de un problema, dadas las condiciones de tráfico, generalmente, se recurre a la intuición con base a la experiencia, incluso destinan recursos en contratar grupos de personas cuya única ocupación es crear rutas!

MOTIVACIÓN

- Ante lo anteriormente mencionado, surge la necesidad de crear un medio automatizado que pueda permitir obtener una ruta mínima que pase por todos los puntos indicados, con un formato de entrada sencillo de entender para el usuario.





OBJETIVO

- Determinar por medio de un algoritmo la ruta más corta para recorrer varios puntos dados en las ciudades de Bucaramanga y Floridablanca.

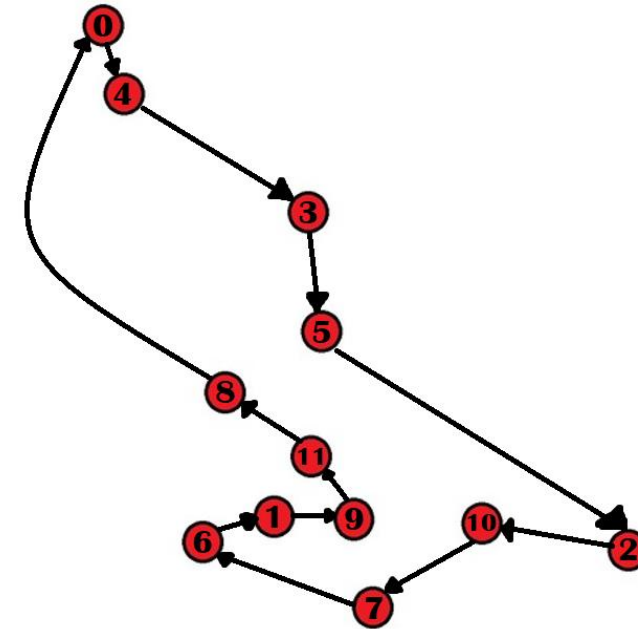
FORMATO DE ENTRADA

NUM	DIRECCION	Latitud	Longitud
0	Calle 18 No. 22-29 Colegio Franciscano del Virrey Solís	7,1308	-73,12361
1	Calle 200 No. 12-518 Olimpo	7,06123	-73,10651
2	CALLE 201A #25-26 URB.BAVIERA /FLORIDA	7,06019	-73,09349
3	CALLE 9 NO. 13-14 APTO. 402 VILLABEL	7,08077	-73,10184
4	CRA. 21 NO. 35-264 CASA 15 QUINTAS DEL CAMPESTRE	7,12801	-73,12247
5	CALLE 21 NO. 11C - 21 ROSALES	7,0748	-73,10177
6	CRA. 12 NO. 200-105 T. 3 AP. 801 MEDITERRANE	7,06024	-73,10887
7	Cra. 14 No. 200-293 Apto. 201 EDIF TERRARIUM	7,05974	-73,10459
8	CRA. 24 NO. 35-30 ALTOS CAÑAVERAL CAMPESTRE	7,06723	-73,10552
9	Calle 198 # 22b-13 VILLA MARGARITA Floridablanca	7,06168	-73,10478
10	CALLE 200 NO. 22B-645 T.5 AP.1420 VERSALLES	7,06138	-73,09904
11	CRA. 24 NO. 35-191 BL. 1 AP. 202 CAÑAVERAL	7,06562	-73,10274

- Se usa un formato en excel, en el cual se le pide al usuario la dirección, la latitud y longitud de los puntos a los cuales se tenga que dirigir. (El punto 0 corresponde al punto de inicio y llegada).

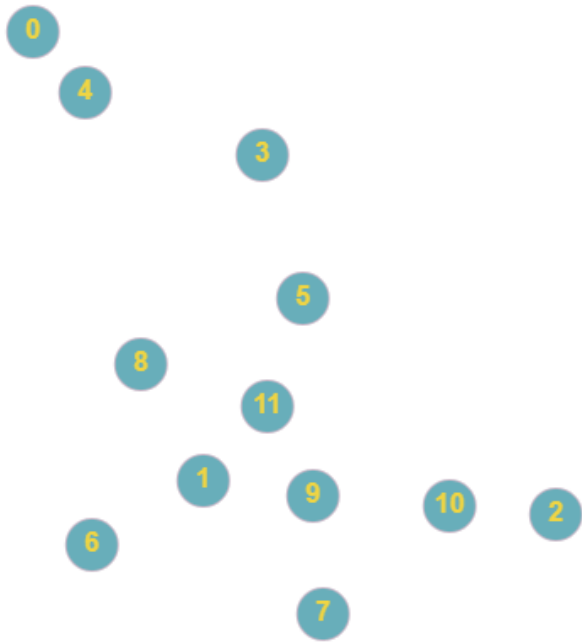
SOLUCIÓN PROBADA

```
1. Calle 18 No. 22-29 Colegio Franciscano del Virrey Solís (Número 0 en la base de datos)
2. CRA. 21 NO. 35-264 CASA 15 QUINTAS DEL CAMPESTRE (Número 4 en la base de datos)
3. CALLE 9 NO. 13-14 APTO. 402 VILLABEL (Número 3 en la base de datos)
4. CALLE 21 NO. 11C - 21 ROSALES (Número 5 en la base de datos)
5. CALLE 201A #25-26 URB.BAVIERA /FLORIDA (Número 2 en la base de datos)
6. CALLE 200 NO. 22B-645 T.5 AP.1420 VERSALLES (Número 10 en la base de datos)
7. Cra. 14 No. 200-293 Apto. 201 EDIF TERRARIUM (Número 7 en la base de datos)
8. CRA. 12 NO. 200-105 T. 3 AP. 801 MEDITERRANE (Número 6 en la base de datos)
9. Calle 200 No. 12-518 Olimpo (Número 1 en la base de datos)
10. Calle 198 # 22b-13 VILLA MARGARITA Floridablanca (Número 9 en la base de datos)
11. CRA. 24 NO. 35-191 BL. 1 AP. 202 CAÑAVERAL (Número 11 en la base de datos)
12. CRA. 24 NO. 35-30 ALTOS CAÑAVERAL CAMPESTRE (Número 8 en la base de datos)
13. Calle 18 No. 22-29 Colegio Franciscano del Virrey Solís (Número 0 en la base de datos)
```

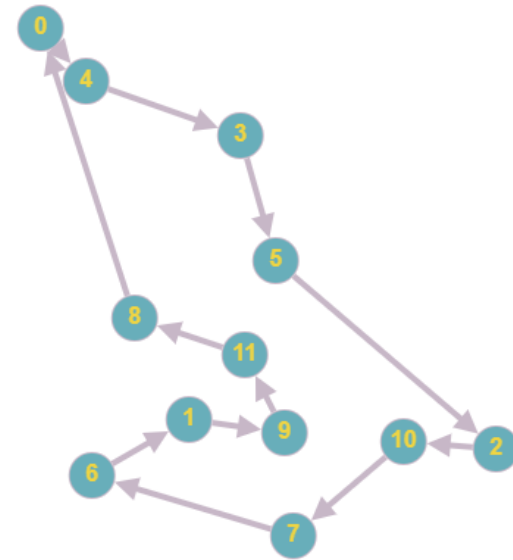


- La siguiente es la ruta más corta creada con las direcciones y coordenadas dadas en la diapositiva anterior, además de un grafo dirigido con el camino que se debería seguir.

GRAFOS DIBUJADOS CON "GRAPHONLINE"



Vértices del grafo



Grafo creado con la ruta
dada por el programa.

MÉTODO A SEGUIR PARA SOLUCIONAR EL PROBLEMA



Fase 1: Entender el problema

Para entender el programa tenemos que tener en cuenta el objetivo, el cual es: Encontrar la ruta más corta pasando por todos los puntos dados y vuelva al punto inicial.

Fase 2: Graficar el problema

Ya que tenemos un problema de grafos, debemos graficarlos vertices para poder crear los puentes de la forma más óptima posible.



0

4

3

5

8

11

6

1

9

10

2

7

Fase 3: Resolver el problema

Este problema es muy parecido al problema del vendedor ambulante el cual podemos resolver con un método de optimización.

CONCLUSIÓN

- Fue posible generar un algoritmo funcional y eficiente que permita a un usuario, por medio de una entrada, obtener una salida de datos correspondientes a una lista ordenada de direcciones que tenga como principal característica ser la ruta más corta para recorrer todos los puntos dados.

EXPLICACIÓN DEL CÓDIGO

1. Función para determinar la distancia entre dos coordenadas

```
#Función para medir la distancia entre dos puntos del planeta según sus coordenadas, el resultado está dado en metros.  
def distanciaEntreCoordenadas(lat1,long1,lat2,long2):  
    rad = math.pi/180  
    difLat = lat2-lat1  
    difLong = long2-long1  
    contenidoRaiz = (math.sin(rad * difLat / 2))**2 + math.cos(lat1*rad) * math.cos(lat2*rad) * ((math.sin(difLong*rad/2)))**2  
    distancia = 2 * 6372000 * math.asin(math.sqrt(contenidoRaiz))  
  
    return distancia
```

- Se toma en cuenta una fórmula matemática, que con base a las coordenadas entre dos puntos, calcula la longitud del arco que une a los dos.

EXPLICACIÓN DEL CÓDIGO

2. Diccionario de Python con los puntos y las distancias entre estos

```
arregloNodos = [i for i in range (0,len(baseDeDatos))]  
matrizDistancias = {(i,j): distanciaEntreCoordenadas(baseDeDatos.at[i,'Latitud'],baseDeDatos.at[i,'Longitud'],baseDeDatos.at[j,'Latitud'],baseDeDatos.at[j,'Longitud']) for i in arregloNodos for j in arregloNodos if i!=j}  
matrizArcos = {(i,j) for i in arregloNodos for j in arregloNodos if i != j}
```

EXPLICACIÓN DEL CÓDIGO

3. Modelo de optimización

```
#Modelo del algoritmo
model = Model("Solucion")

#Variables de decisión
x = model.addVars(matrizArcos, vtype = GRB.BINARY, name = 'x')
u = model.addVars(arregloNodos, vtype = GRB.CONTINUOUS, name = 'u')

#Función del objetivo definida
model.setObjective(quicksum(matrizDistancias[n] * x[n] for n in matrizArcos), GRB.MINIMIZE)

#Restricciones para el modelo
model.addConstrs(quicksum(x[i,j] for j in arregloNodos if j != i) == 1 for i in arregloNodos)
model.addConstrs(quicksum(x[i,j] for i in arregloNodos if j != i) == 1 for j in arregloNodos)
model.addConstrs((x[i,j] == 1) >> (u[i]+1 == u[j]) for i,j in matrizArcos if j!=0)

model.optimize()
```

¿POR QUÉ UN MODELO DE OPTIMIZACIÓN?

- Se decidió tomar en cuenta, dado que este tipo de algoritmos, pertenecientes a la programación lineal, resultan muy útiles a la hora de resolver problemas donde hay recursos limitados, y se tienen una serie de restricciones que se especifican al momento de diseñar el modelo.

EXPLICACIÓN DEL CÓDIGO

Variables de decisión: Son las variables que se utilizarán en el modelo, se hará referencia a estas en la función del objetivo y las restricciones.

Función del objetivo: Es aquella función del algoritmo que se va a optimizar, se puede realizar de varias formas, ya sea maximizando o minimizando, en el caso del código, se puede ver que se eligió la opción de minimizar dada la constante "GRB.MINIMIZE"

Restricciones: Son una serie de condiciones que se deben cumplir a cabalidad a la hora de llevar a cabo el modelo.

TEORÍA APLICADA AL CÓDIGO

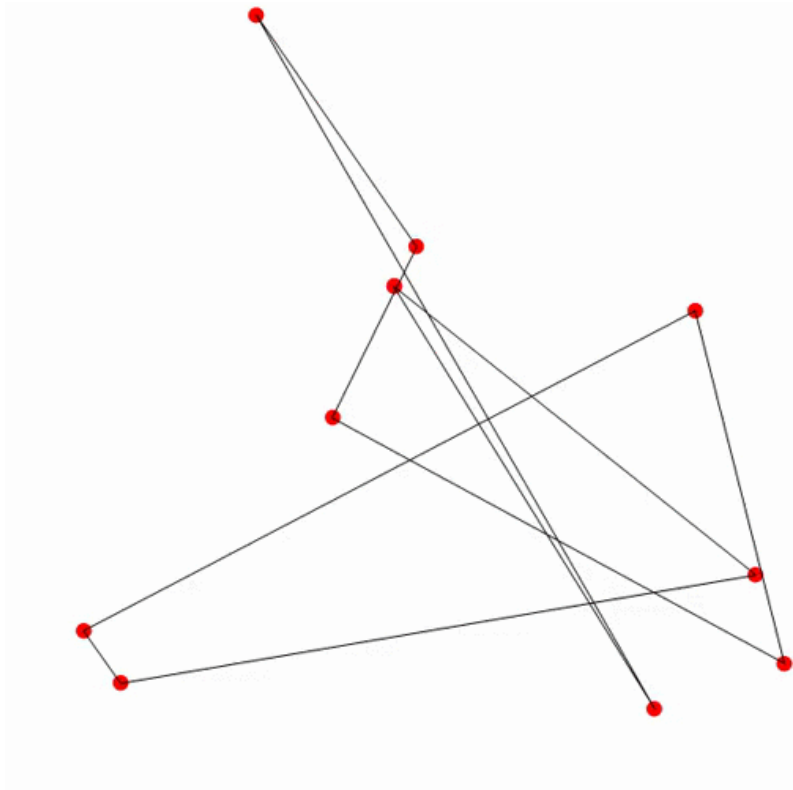
Variables de decisión: En este caso, "x" hace referencia a una variable booleana, en esta, se describe si se decidió ir de un punto i a un punto j. Por otro lado, "u" se trata de una variable continua y describe el número del nodo que se está visitando.

Función del objetivo: En la rama de la optimización y programación lineal, hay principalmente dos procesos, minimizar y maximizar, en este caso, se minimizará, y lo podemos ver cuando se llama a la constante "GRB.MINIMIZE".

Restricciones: En este caso hay dos principales:

- No se puede llegar a un nodo desde más de un nodo.
- Solo se puede trazar un camino desde un nodo, hacia un y solo un nodo.

ILUSTRACIÓN DE COMO FUNCIONA



```
Explored 1531324 nodes (1794690 simplex iterations) in 101.00 seconds (42.93 work units)  
Thread count was 2 (of 2 available processors)
```

```
Solution count 10: 19004.5 19079.4 19107 ... 19339.7
```

```
Optimal solution found (tolerance 1.00e-04)  
Best objective 1.900448342458e+04, best bound 1.900448342458e+04, gap 0.0000%
```


EXPLICACIÓN DEL CÓDIGO

4. Funciones para organizar la salida

```
arcos_ruta = [i for i in matrizArcos if x[i].x > 0.9]
print(arcos_ruta)

[(10, 7), (7, 6), (0, 4), (3, 5), (6, 1), (8, 0), (9, 11), (1, 9), (11, 8), (2, 10), (4, 3), (5, 2)]

matriz_nueva = []
no_ordenado = True
contador = 0
while(no_ordenado):
    for i in range(0, len(arcos_ruta)):
        if(arcos_ruta[i][0] == contador):
            print(arcos_ruta[i][0])
            contador = arcos_ruta[i][1]
            matriz_nueva.append(arcos_ruta[i])
            if(arcos_ruta[i][1] == 0):
                no_ordenado = False

0
4
3
5
2
10
7
6
1
9
11
8

print(matriz_nueva)

[(0, 4), (4, 3), (3, 5), (5, 2), (2, 10), (10, 7), (7, 6), (6, 1), (1, 9), (9, 11), (11, 8), (8, 0)]
```

EXPLICACIÓN DEL CÓDIGO

```
print(matriz_nueva)

[(0, 4), (4, 3), (3, 5), (5, 2), (2, 10), (10, 7), (7, 6), (6, 1), (1, 9), (9, 11), (11, 8), (8, 0)]

ruta_numeros = []

for i in range(0, len(matriz_nueva)):
    ruta_numeros.append(matriz_nueva[i][0])
    if(i == len(matriz_nueva)-1):
        ruta_numeros.append(matriz_nueva[i][1])

print(ruta_numeros)

[0, 4, 3, 5, 2, 10, 7, 6, 1, 9, 11, 8, 0]

ruta_direcciones = ""
contador = 0
for i in ruta_numeros:
    contador += 1
    ruta_direcciones = ruta_direcciones + f"{contador}. {baseDeDatos.at[i,'DIRECCION']} (Número {i} en la base de datos) \n"

print(ruta_direcciones)

1. Calle 18 No. 22-29 Colegio Franciscano del Virrey Solís (Número 0 en la base de datos)
2. CRA. 21 NO. 35-264 CASA 15 QUINTAS DEL CAMPESTRE (Número 4 en la base de datos)
3. CALLE 9 NO. 13-14 APTO. 402 VILLABEL (Número 3 en la base de datos)
4. CALLE 21 NO. 11C - 21 ROSALES (Número 5 en la base de datos)
5. CALLE 201A #25-26 URB.BAVIERA /FLORIDA (Número 2 en la base de datos)
6. CALLE 200 NO. 22B-645 T.5 AP.1420 VERSALLES (Número 10 en la base de datos)
7. Cra. 14 No. 200-293 Apto. 201 EDIF TERRARIUM (Número 7 en la base de datos)
8. CRA. 12 NO. 200-105 T. 3 AP. 801 MEDITERRANE (Número 6 en la base de datos)
9. Calle 200 No. 12-518 Olimpo (Número 1 en la base de datos)
10. Calle 198 # 22b-13 VILLA MARGARITA Floridablanca (Número 9 en la base de datos)
11. CRA. 24 NO. 35-191 BL. 1 AP. 202 CAÑAVERAL (Número 11 en la base de datos)
12. CRA. 24 NO. 35-30 ALTOS CAÑAVERAL CAMPESTRE (Número 8 en la base de datos)
13. Calle 18 No. 22-29 Colegio Franciscano del Virrey Solís (Número 0 en la base de datos)
```