

METRO UIS

Matemáticas

Discretas

Santiago Torres

Santiago Gelvez

Juan José Rincón

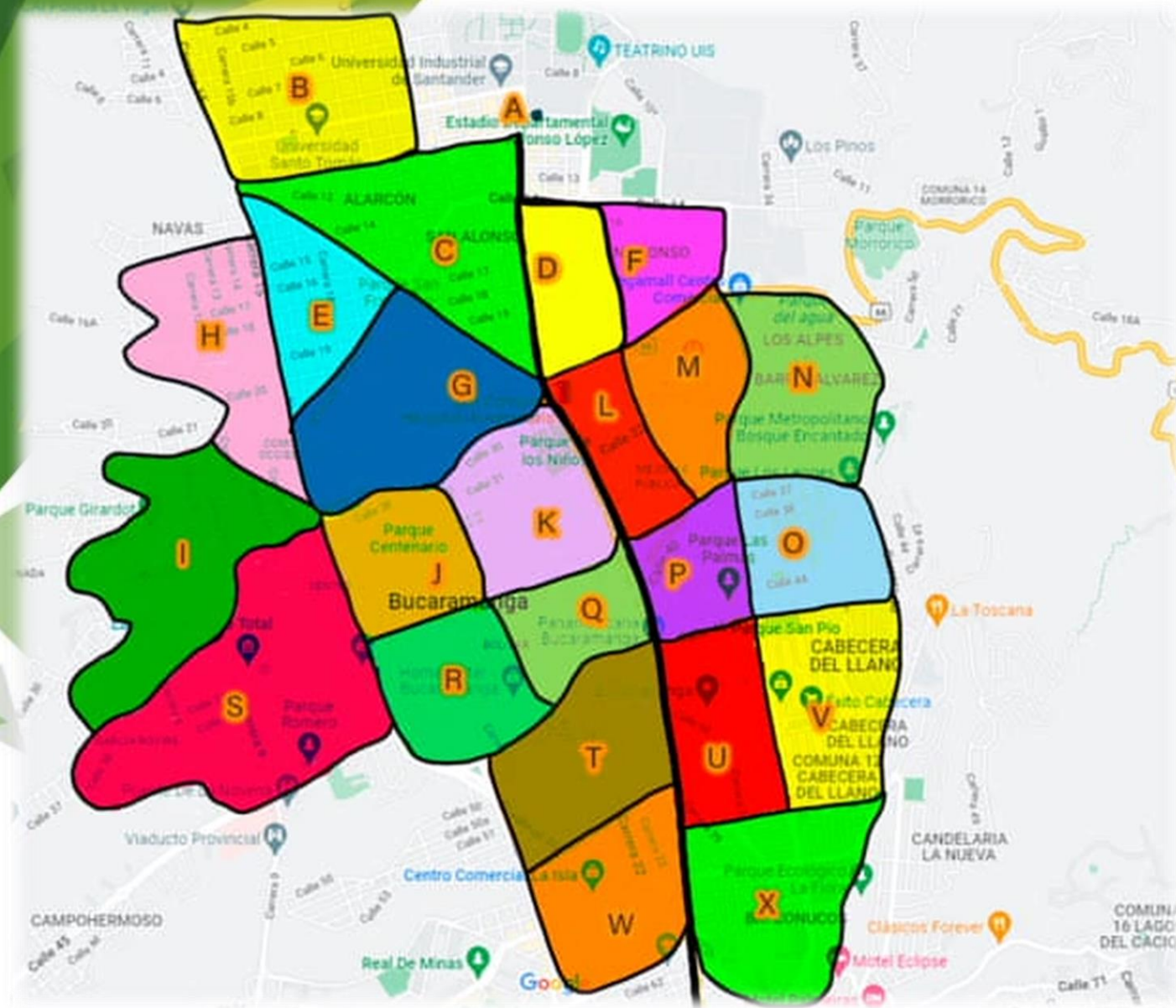


Universidad
Industrial de
Santander

Somos el mejor escenario
de creación e innovación
www.uis.edu.co

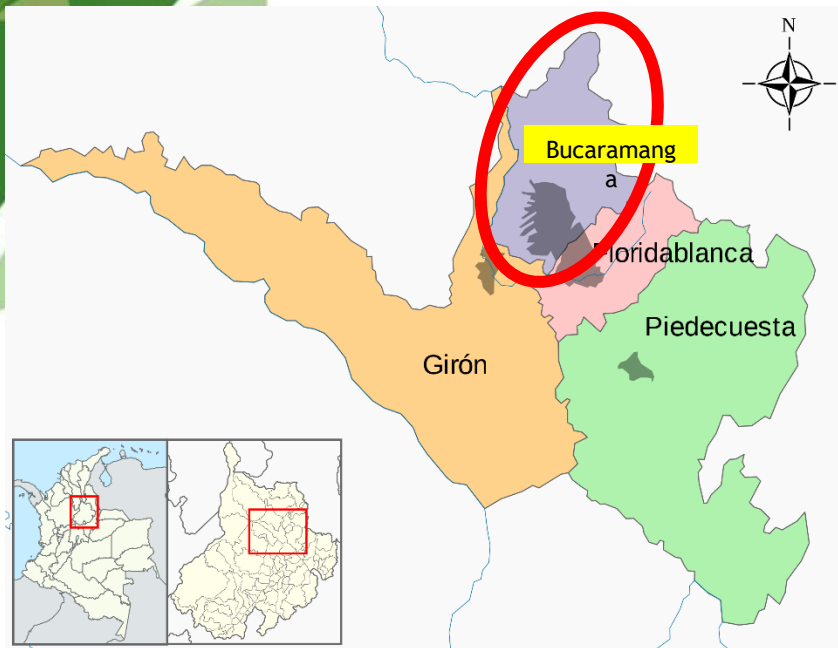


Universidad
Industrial de
Santander



Somos el mejor escenario
de creación e innovación
www.uis.edu.co

Mediante este proyecto se determinará la ruta más corta que lleve a los usuarios que se encuentran en Bucaramanga a la universidad industrial de Santander, en el menor tiempo posible. Esto haciendo uso de grafos, calculando el tiempo que hay entre vértices dependiendo de la hora.





Modo de trabajo

Origen

Destino

Tiempo

Para cada uno de los nodos.

8:00

12:00

18:00



Archivo	Inicio	Insertar	Disposición de página	Fórmulas
5:00 am	9:00 am	5:00pm		
a 8:59 am	a 4:59pm	a 9:00pm		
C66	C62	C47		
20	10	18		
A	B	C	D	
Origen	Destino	Tiempo		
1				
2	A	B	5	
3	A	D	20	
4	B	A	9	
5	B	C	5	
6	B	H	10	
7	C	B	6	
8	C	D	15	
9	C	E	7	
10	D	A	25	
11	D	C	16	
12	D	F	4	
13	D	L	17	
14	D	G	7	
15	E	C	5	
16	E	H	5	
17	F	D	5	
18	F	M	5	
19	F	N	9	
20	G	H	15	
21	G	D	17	
22	G	K	17	
23	G	J	12	
24	H	B	9	
25	H	E	5	
26	H	G	8	
27	H	I	5	
28	I	H	6	
29	I	S	10	
30	J	G	15	
31	J	K	12	
32	J	R	10	
33	J	S	12	
34	K	G	15	
35	K	L	14	
36	K	J	11	
37	K	P	11	
38	L	D	18	

Obtención de datos

Para poder obtener los datos necesarios para poder realizar el algoritmo, tuvimos que utilizar Excel donde organizamos por horarios el tiempo que toman todos los vértices de cada uno de los caminos representados en el mapa.



Modo de trabajo

Se importaron las tablas Excel de tiempo de un punto a otro según los horarios establecidos

```
[ ] from google.colab import files  
files.upload()
```

Examinar... No se han seleccionado archivos. Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving 8am.xlsx to 8am.xlsx

```
{'8am.xlsx': b'PK\x03\x04\x14\x00\x06\x00\x08\x00\x00\x00!\x00b\xee\x9dh^\x01\x00\x00\x90\x04\x00\x00\x13\x00\x08\x02[Content_Type
```




Modo de trabajo

Se importan los datos y se convierten en un DataFrame, tomando las diferentes variables (origen, destino y tiempo) implementando cada una de las diferentes jornadas (mañana, medio y tarde)

```
def rutas(ori, desti, time):  
  
    leeman = pd.read_excel("8am.xlsx", index_col=None)  
  
    MANANA = nx.from_pandas_edgelist(leeman, source='Origen', target='Destino', edge_attr='Tiempo')  
  
    leemid = pd.read_excel("12pm.xlsx", index_col=None)  
  
    MEDIO = nx.from_pandas_edgelist(leemid, source='ORIGEN', target='DESTINO', edge_attr='TIEMPO')  
  
    leetar = pd.read_excel("6pm.xlsx", index_col=None)  
  
    TARDE = nx.from_pandas_edgelist(leetar, source='Origen', target='Destino', edge_attr='Tiempo')
```



Modo de trabajo

Se trabajara haciendo uso de 3 caminos implementando **source** (nodo de origen), **target** (destino) y **weight** (cada una de las aristas que tendrá como atributo el tiempo que separa a la distancia en metros) donde se le pide al usuario los puntos para crear el grafo más corto.

```
camino1=nx.dijkstra_path(MANANA, source= ori, target= desti, weight='Tiempo')
ruta1 = MANANA.subgraph(camino1)

camino2=nx.dijkstra_path(MEDIO, source= ori, target= desti, weight='Tiempo')
ruta2 = MEDIO.subgraph(camino2)

camino3 = nx.dijkstra_path(TARDE, source= ori, target= desti, weight='Tiempo')
ruta3 = TARDE.subgraph(camino3)
```


Algoritmo de Dijkstra

Es eficiente para calcular caminos de longitud mínima entre dos vértices cualesquiera en un grafo.

Va explorando los caminos más cortos que parten del vértice origen y que lleva a todos los demás vértices, y cuando se obtiene el camino más corto desde el vértice origen al resto de vértices que componen el grafo, el algoritmo se detiene.

El algoritmo es uniforme y no funciona con aristas negativas.

Algoritmo de Dijkstra

A partir del algoritmo de Dijkstra se implementaron las siguientes condiciones:

if

```
if(time>=5 and time<=8):  
    print("Manana")  
  
print(camino1)  
  
nx.draw(ruta1, with_labels=True)
```

```
if(time>8 and time <=16):  
    print("Medio dia")  
  
print(camino2)  
  
nx.draw(ruta2, with_labels=True)
```

elif

```
elif(time>16):  
    print("Tarde")  
  
print(camino3)  
  
nx.draw(ruta3, with_labels=True)
```

¿Por qué?

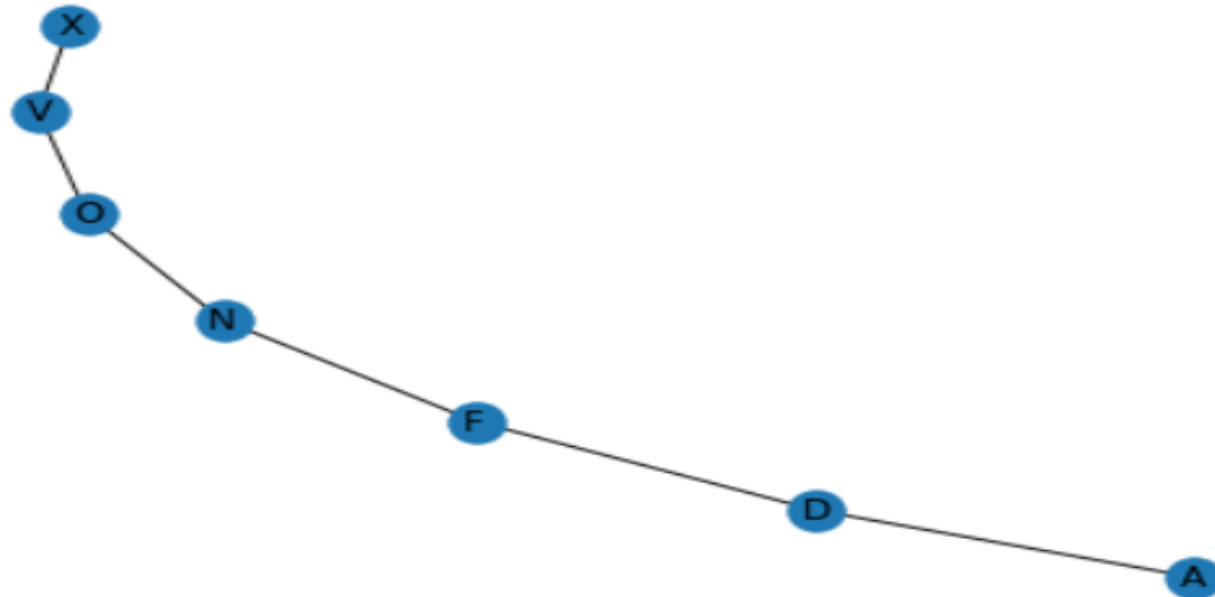
Porque nos permite ejecutar bloques de código según el valor de algunos datos y la condición se evalúa como verdadera.

```
rutas('X', 'A', 18)
```

X A 18

Tarde

```
['X', 'V', 'O', 'N', 'F', 'D', 'A']
```

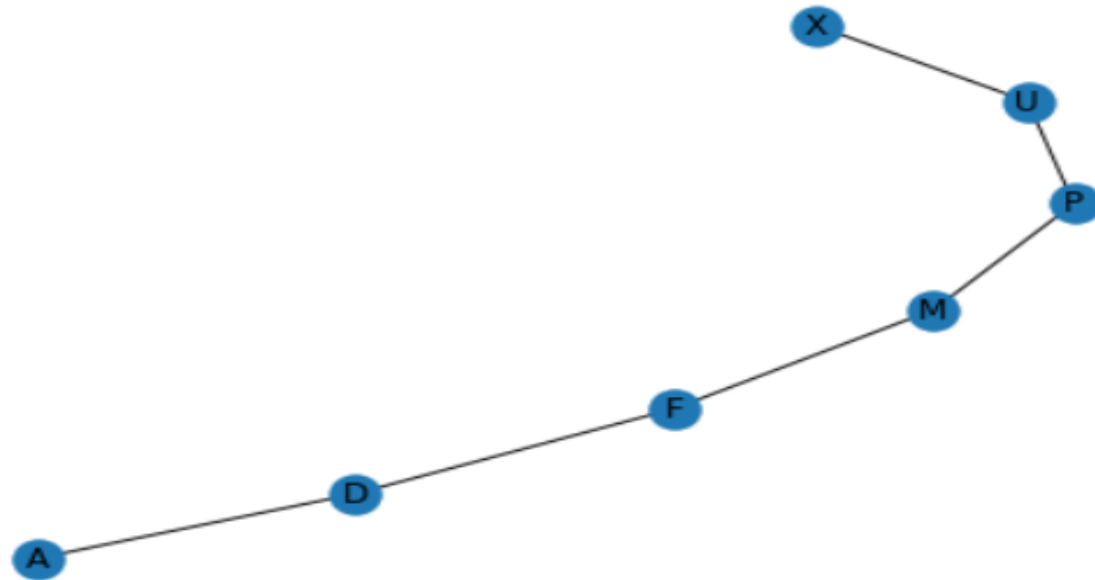



```
rutas('X', 'A', 12)
```

```
X A 12
```

```
Medio día
```

```
['X', 'U', 'P', 'M', 'F', 'D', 'A']
```





MUCHAS

GRACIAS

Santiago Torres
Santiago Gelvez
Juan José Rincón

METRO UIS
nos acompaña en el camino
de creación e innovación
www.uis.edu.co



Universidad
Industrial de
Santander