

Ejercicio: Búsqueda por costo uniforme

Nombre: Fabian

Armijos

Enunciado:

- Diseñe un grafo similar al que se ha presentado en este ejercicio partiendo de las siguientes coordenadas de latitud y longitud: - 2.8801604,-79.0071712. Para ello deberá realizar las siguientes tareas:

Emplear la herramienta Google Maps (R) con las coordenadas antes indicadas .

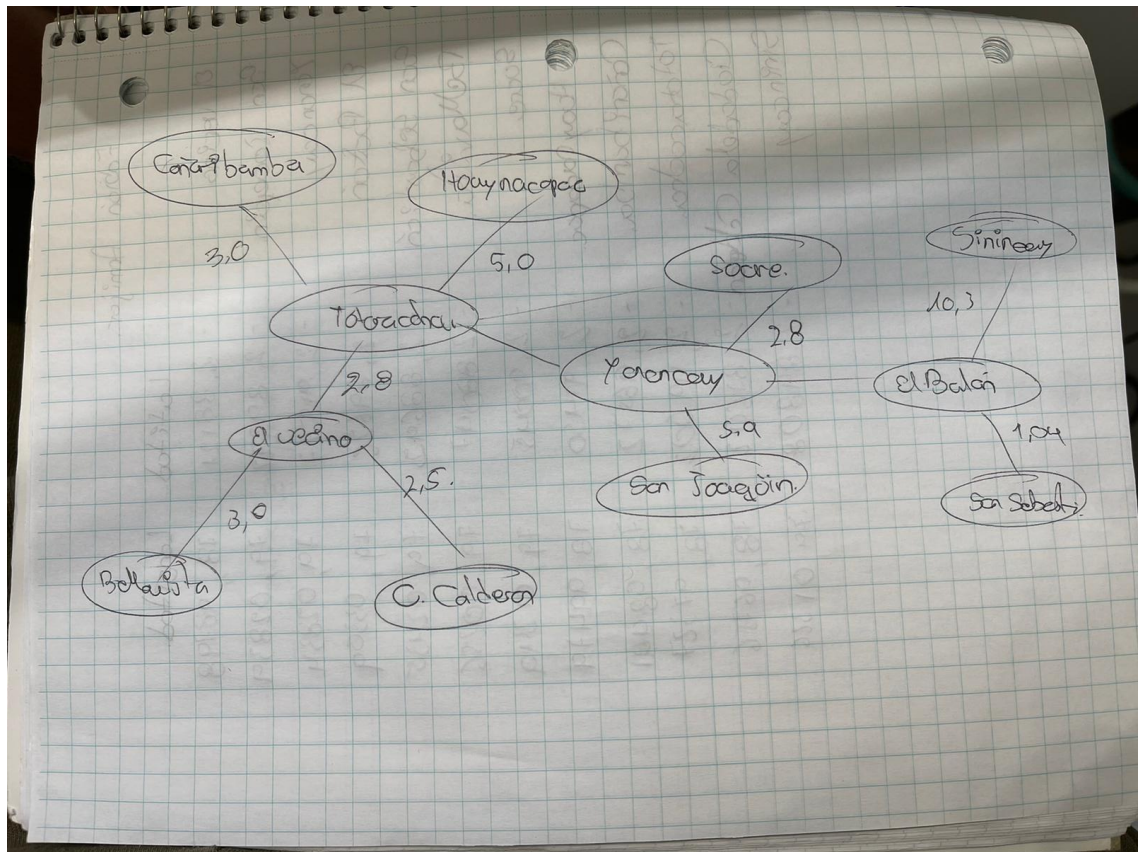
Definir 11 puntos de interés y armar el grafo.

11 puntos de interés

Fabian Armijos.

	Latitud	Longitud.
El vecino	- 2, 88121	78, 98798
San Joaquin.	- 2, 89372	79, 02839
Yorancay	- 2, 91577	79, 02834
El Batón	- 2, 89626	79, 03309
San Sebastián	- 2, 88692	79, 02435
Bellavista.	- 2, 88047	79, 00286
Sucre	- 2, 90045	79, 01349
Hay Pasaque.	- 2, 91450	78, 99479
Corralbamba.	- 2, 96572	78, 98441
Totoracondan.	- 2, 99002	78, 97327
Ciudadela Cabaon.	- 2, 83642	78, 96756
Sinincay	- 2, 84808	79, 01326

Grafo



Especificar como punto de partida al sector "San Sebastián" y como objetivo "Totoracocha".

- 1) Punto de partida: San Sebastián"
- 2) Punto objetivo: Totoracocha

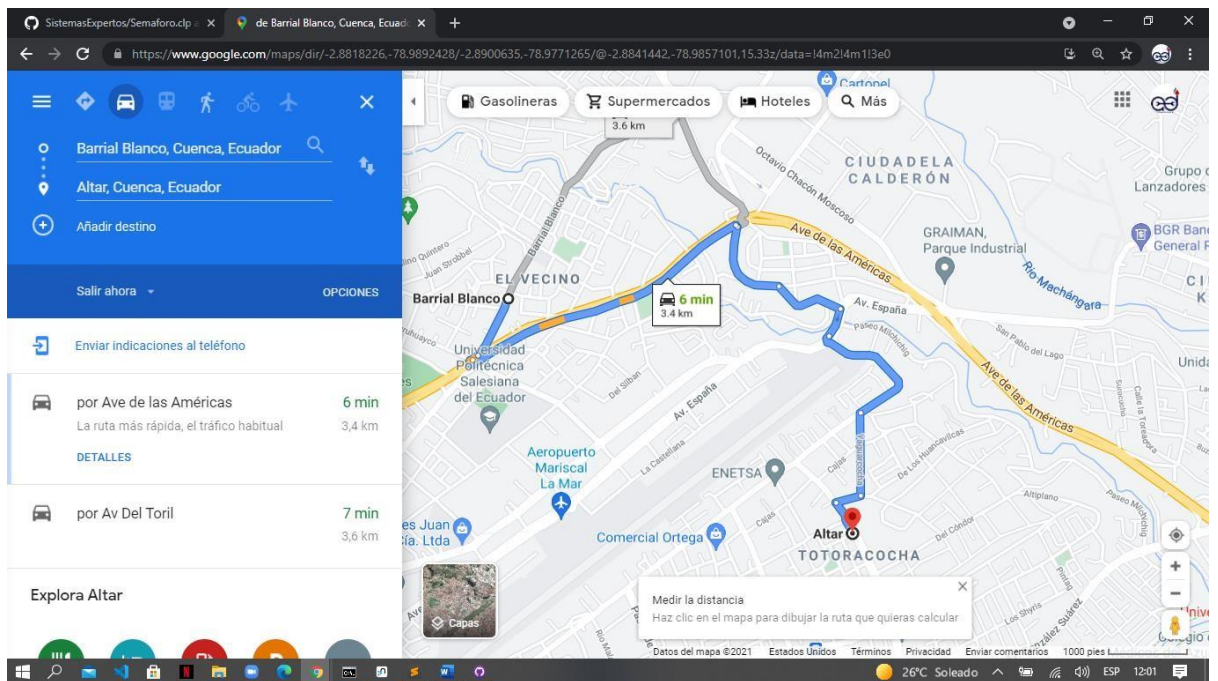
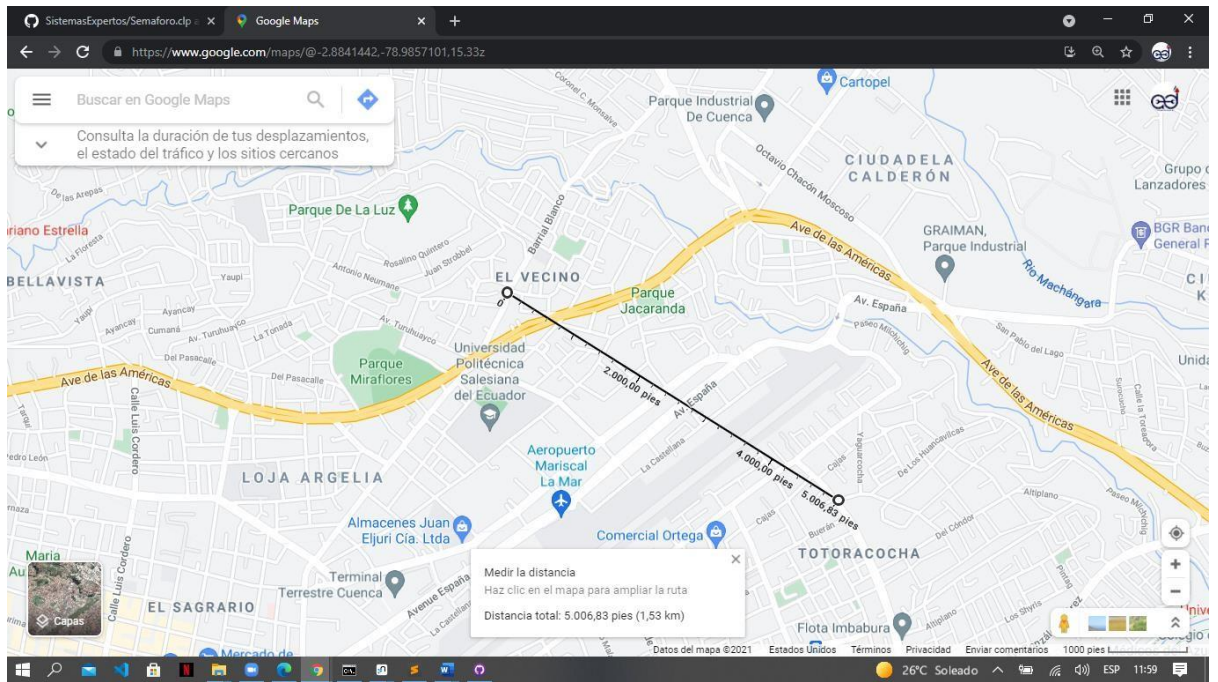
Calcular la distancia que existe entre los puntos de interés

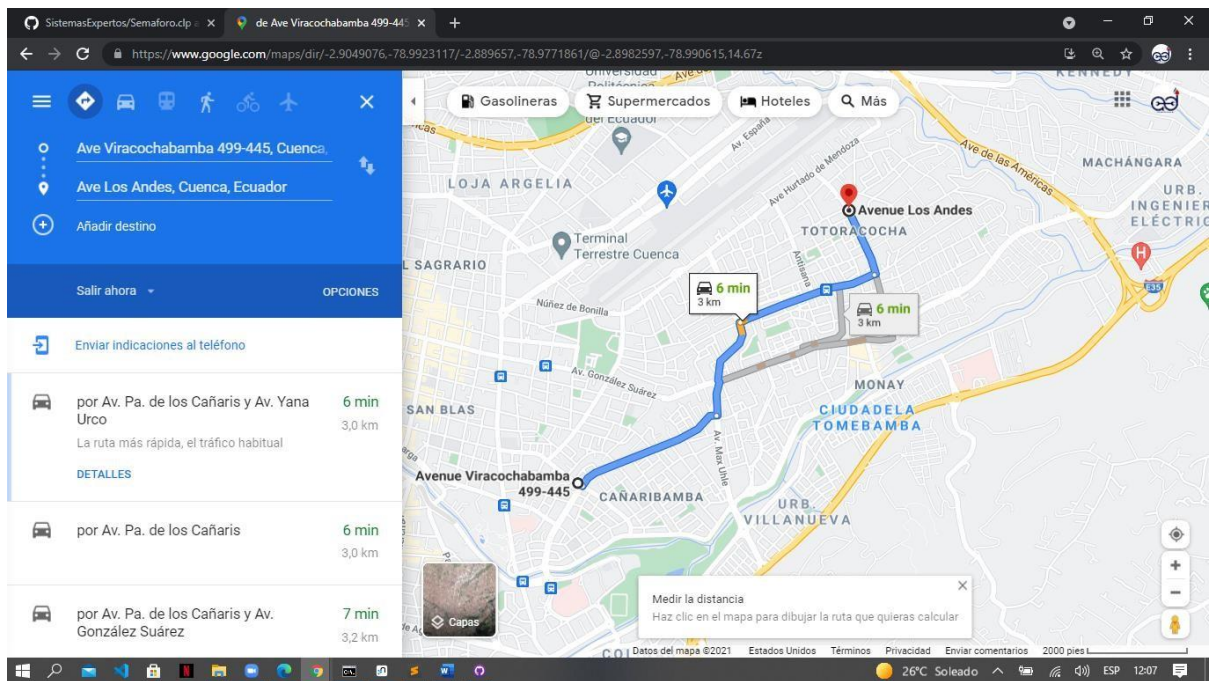
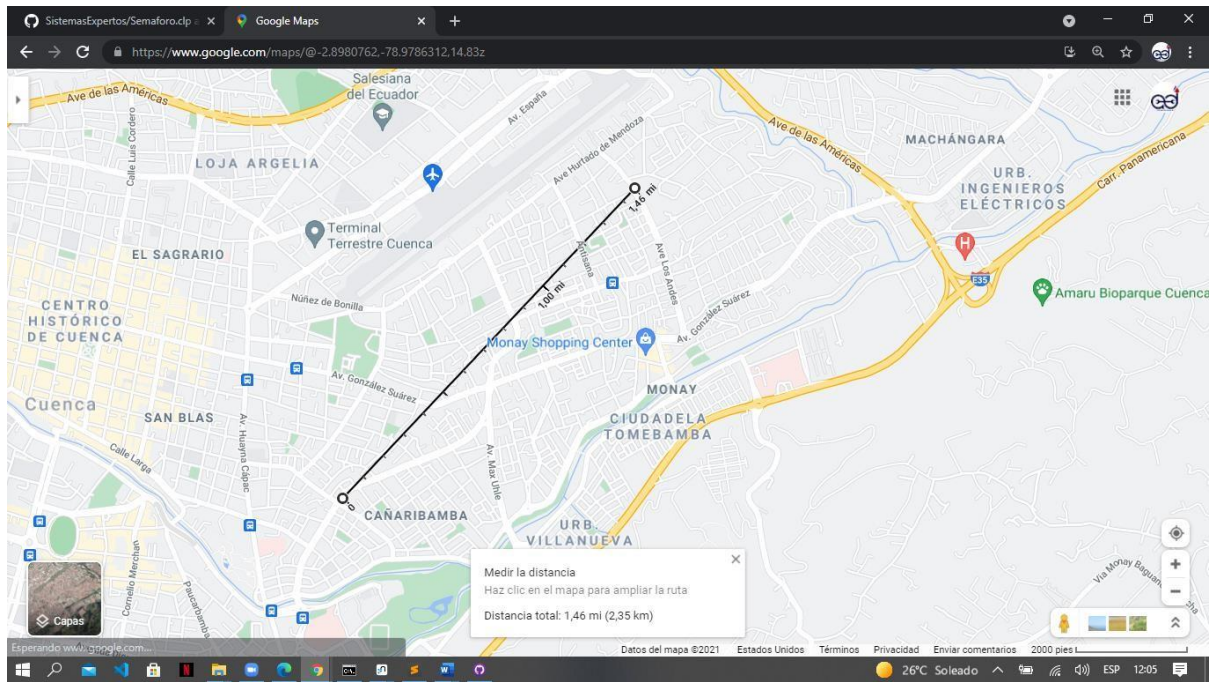
Para ello debe usar la "ir de un punto a otro" de Google Maps (Direcciones o Indicaciones).

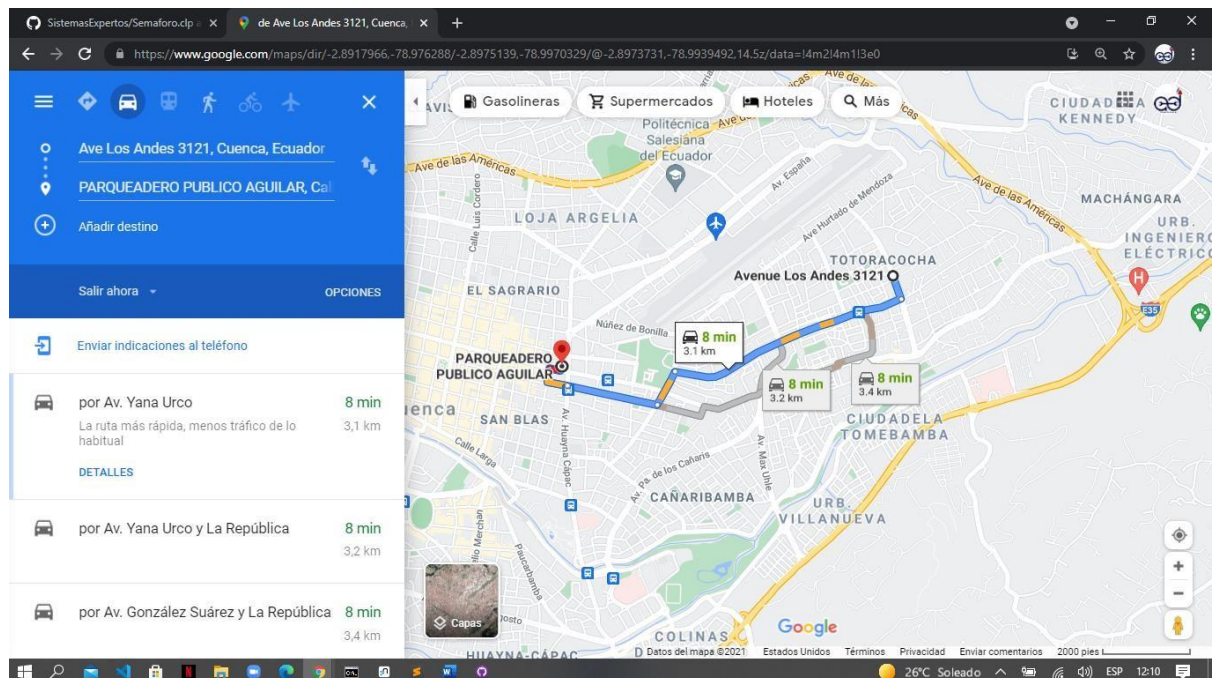
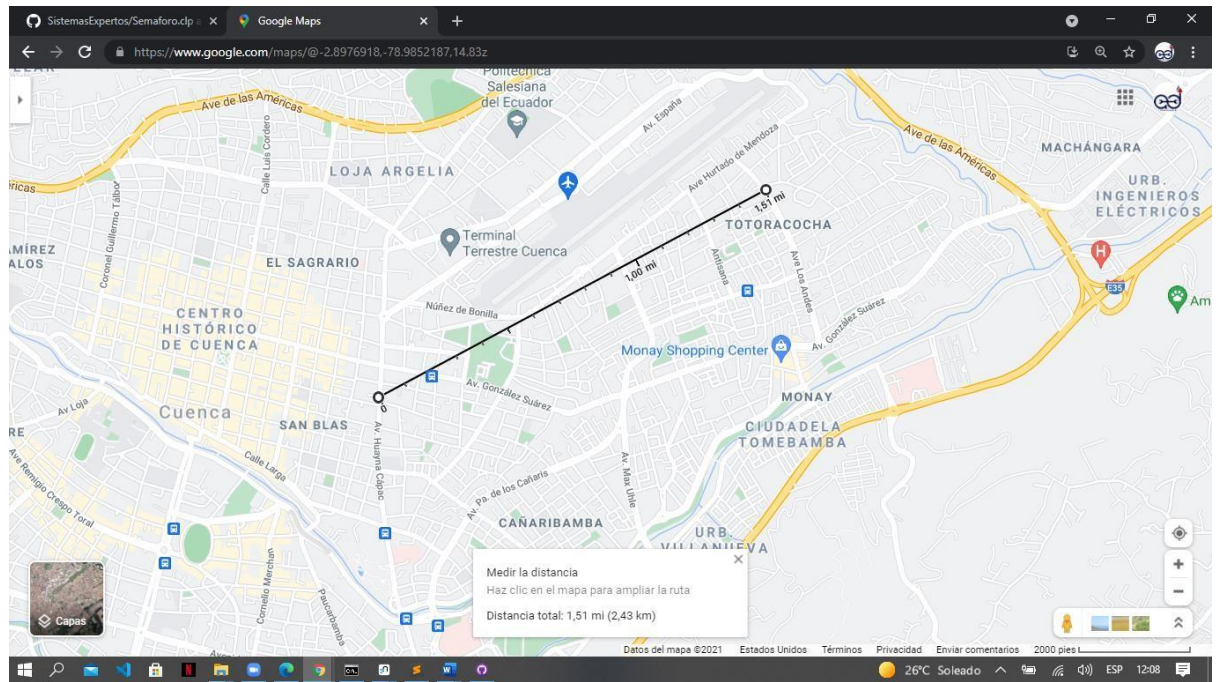
Tobón Armijo			
Bellavista - El vecino	1,89 3,09	$h(n)$ $g(n)$	
C. Calderon - El vecino	1,48 2,15	$h(n)$ $g(n)$	
Totoracocha - El vecino	1,66 2,18	$h(n)$ $g(n)$	
Peñarubamba - Totoracocha	2,07 3,10	$h(n)$ $g(n)$	
Huayra - cañe - Totoracocha	3,73 3,00	$h(n)$ $g(n)$	
Yanencay - Totoracocha	6,30 10,9	$h(n)$ $g(n)$	
Sacre - Yanencay	2,13 2,8	$h(n)$ $g(n)$	
Batán - Yanencay	2,27 4,2	$h(n)$ $g(n)$	
San Lorenzo - Yanencay	3,43 3,9	$h(n)$ $g(n)$	
San Sebastián - El Batán	1,04 1,5	$h(n)$ $g(n)$	
Sinincay - El Batán	5,65 10,3	$h(n)$ $g(n)$	
-2,8801604- 79,00712 - El vecino 2,16m			

Ejemplos:

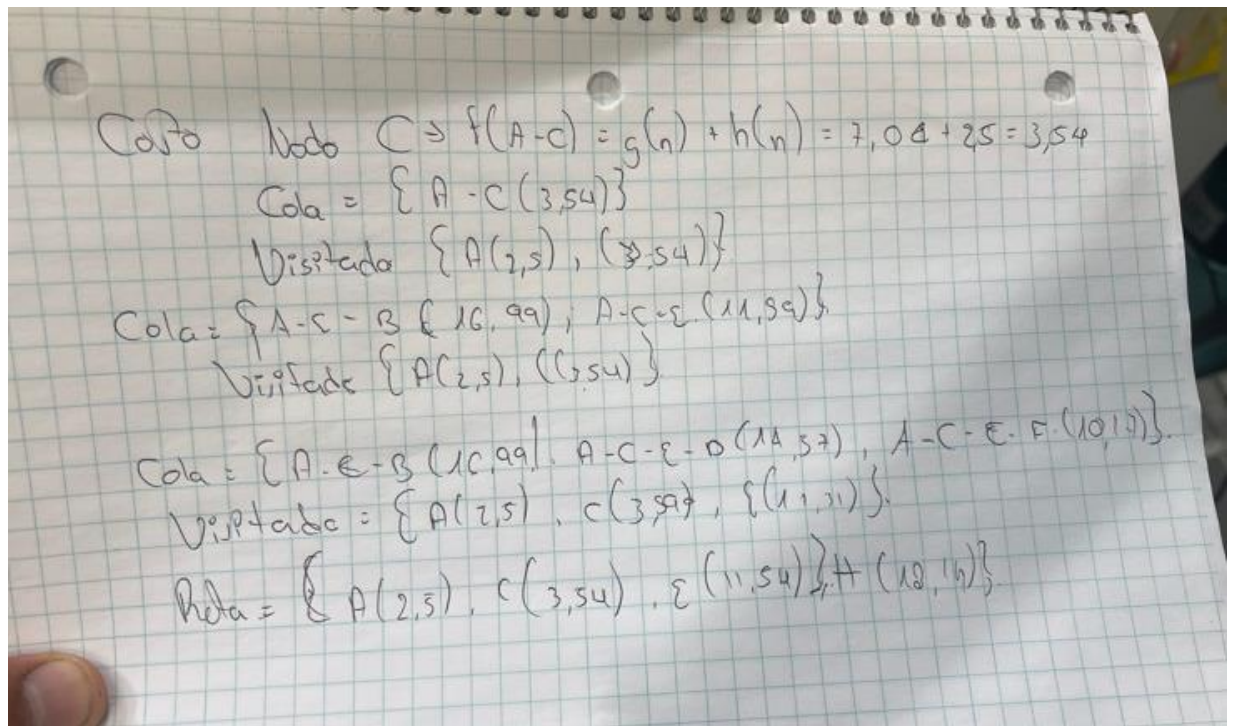
El vecino - Totoracocha







Realizar el proceso de búsqueda de forma similar a cómo se a explicado en este apartado, almacenando para ello los datos de la lista Visitados y de la Cola.



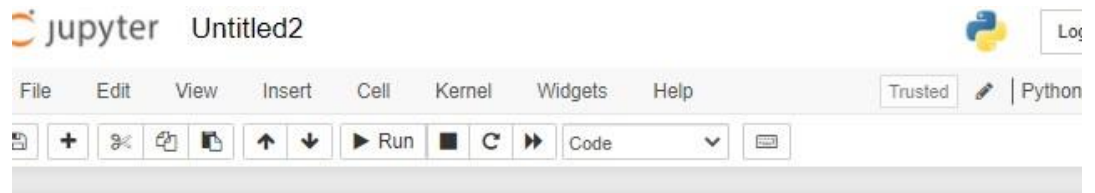
Importar la API py2neo

Para el ingreso de los datos que se encuentran dentro de la lista

Conexión con Neo4j

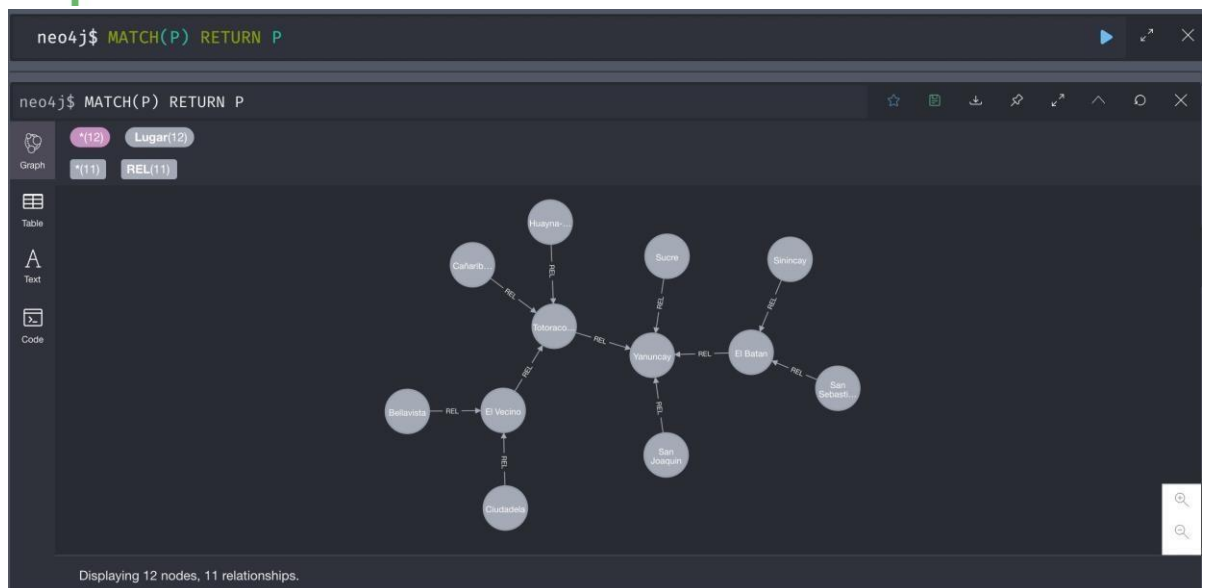
Configure la URL de conexión con la base de datos de Neo4j:

Creación de los 11 lugares con sus relaciones.



```
In [ ]: from py2neo import Node, Relationship, Graph
graph=Graph("bolt://localhost:11003", aut="neo4j", password="12345")
```

```
In [5]: graph("CREATE (a:Lugar{name:'El Vecino', latitude:-2.881 2
          (b:Lugar{name:'San Joaquin', latitude:-2.89372,longitud
          (c:Lugar{name:'Yanuncay', latitude:-2.89372,longitud
          (d:Lugar{name:'El Batan', latitude:-2.89372,longitud
          (e:Lugar{name:'San Sebastian', latitude:-2.89372,longitud
          (f:Lugar{name:'Bellavista', latitude:-2.89372,longitud
          (g:Lugar{name:'Sucre', latitude:-2.89372,longitud
          (h:Lugar{name:'Huayna - Capac', latitude:-2.89372,longitud
          (i:Lugar{name:'Cañaribamba', latitude:-2.89372,longitud
          (j:Lugar{name:'Totoracocha', latitude:-2.89372,longitud
          (k:Lugar{name:'Ciudadela Calderon', latitude:-2.89372,longitud
          (m:Lugar{name:'Sinincay', latitude:-2.89372,longitud
          (e)-[:REL{cost: 1.04}]->(d),"
          (m)-[:REL{cost: 10.3}]->(d),"
          (d)-[:REL{cost: 4.2}]->(c),"
          (b)-[:REL{cost: 5.9}]->(c),"
          (g)-[:REL{cost: 2.8}]->(c),"
          (j)-[:REL{cost: 10.8}]->(c),"
          (h)-[:REL{cost: 5.0}]->(j),"
          (i)-[:REL{cost: 3.0}]->(j),"
          (a)-[:REL{cost: 2.8}]->(j),"
          (f)-[:REL{cost: 3.0}]->(a),"
          (k)-[:REL{cost: 2.5}]->(a)).data().|
```



```
neo4j$ CALL gds.graph.create('amplitud', 'Lugar', 'REL', { relationshipProperties: 'cost' })
```

	nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	createMillis
1	{ "Lugar": { "properties": { }, "label": "Lugar" } }	{ "REL": { "orientation": "NATURAL", "aggregation": "DEFAULT", "type": "REL", "properties": { "cost": { "property": "cost", "aggregation": "DEFAULT", "defaultValue": null } } } }	"amplitud"	12	11	73

Started streaming 1 records after 2 ms and completed after 1585 ms.

```

1 MATCH (start:Lugar {name: 'San Sebastian'}), (end:Lugar {name: 'Totoracocha'})
2 CALL gds.alpha.shortestPath.astar.stream({
3   nodeProjection: {
4     Lugar: {
5       properties: ['longitude', 'latitude', 'h']
6     }
7   },
8   relationshipProjection: {
9     CONNECTION: {
10      type: 'CONNECTION',
11      orientation: 'UNDIRECTED',
12      properties: 'g'
13    }
14  })
15 YIELD nodeId, cost
16 RETURN gds.util.asNode(nodeId).name AS lugares, cost

```

	lugares	cost
1	"San Sebastian"	0.0
2	"El Batan"	1.0
3	"Yanuncay"	2.0
4	"Totoracocha"	3.0

Started streaming 4 records after 1 ms and completed after 9 ms.

```
In [12]: 1 graph.run("MATCH (start:Lugar {name: 'San Sebastian'}), (end:Lugar {name: 'Totoracocha'})
2 CALL gds.alpha.shortestPath.astar.stream({
3   nodeProjection: {
4     Lugar: {
5       properties: ['longitude', 'latitude', 'h']
6     }
7   },
8   relationshipProjection: {
9     CONNECTION: {
10      type: 'CONNECTION',
11      orientation: 'UNDIRECTED',
12      properties: 'g'
13    }
14  })
15 YIELD nodeId, cost
16 RETURN gds.util.asNode(nodeId).name AS lugares, cost").data()
```

```
Out[12]: [{'lugares': 'San Sebastian', 'cost': 0.0},
{'lugares': 'El Batan', 'cost': 1.0},
{'lugares': 'Yanuncay', 'cost': 2.0},
{'lugares': 'Totoracocha', 'cost': 3.0}]
```