



Jonathan Atancuri

Objetivo:

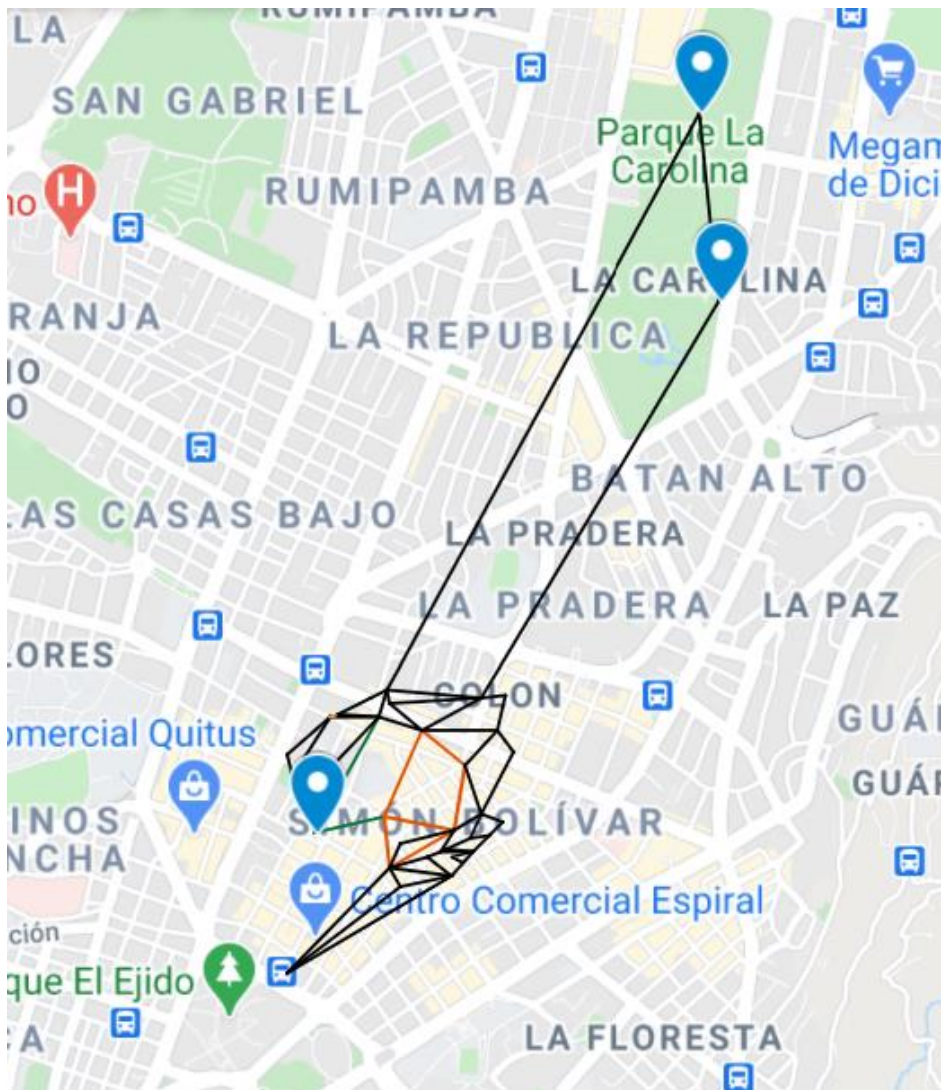
- Consolidar los conocimientos adquiridos en clase para los métodos de búsqueda.

Enunciado:

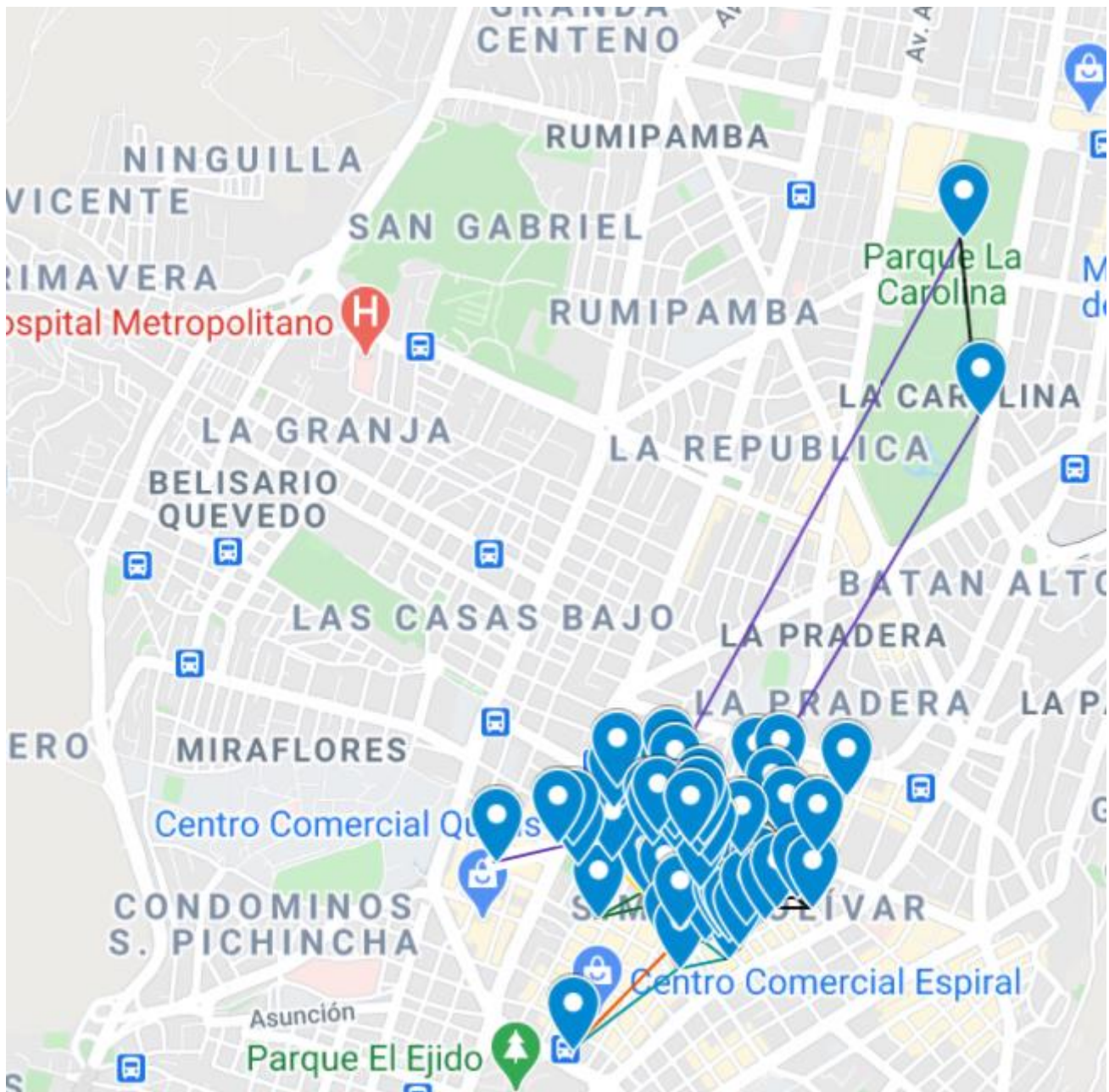
- Diseñe y desarrolle un mapa de nodos para encontrar la ruta mas corta de la iglesia central de cada ciudad a los hoteles y lugares turísticos al menos 50 para cada ciudad, para ello se debe seguir los siguientes pasos :
 - Se tiene los datos dentro de Google Maps (<https://www.google.com/maps/search/iglesias/@-2.891806,-79.013548,14.13z>), generar y agregar un captura de pantalla de la búsqueda y generación de los mapas:

Mapa de Quito con sus respectivos Nodos

◦



◦



- Seleccionar como nodo de partido la iglesia central y al menos dos nodos lejanos como la llegada.

Nodo Origen (Iglesia La Viña Quito)

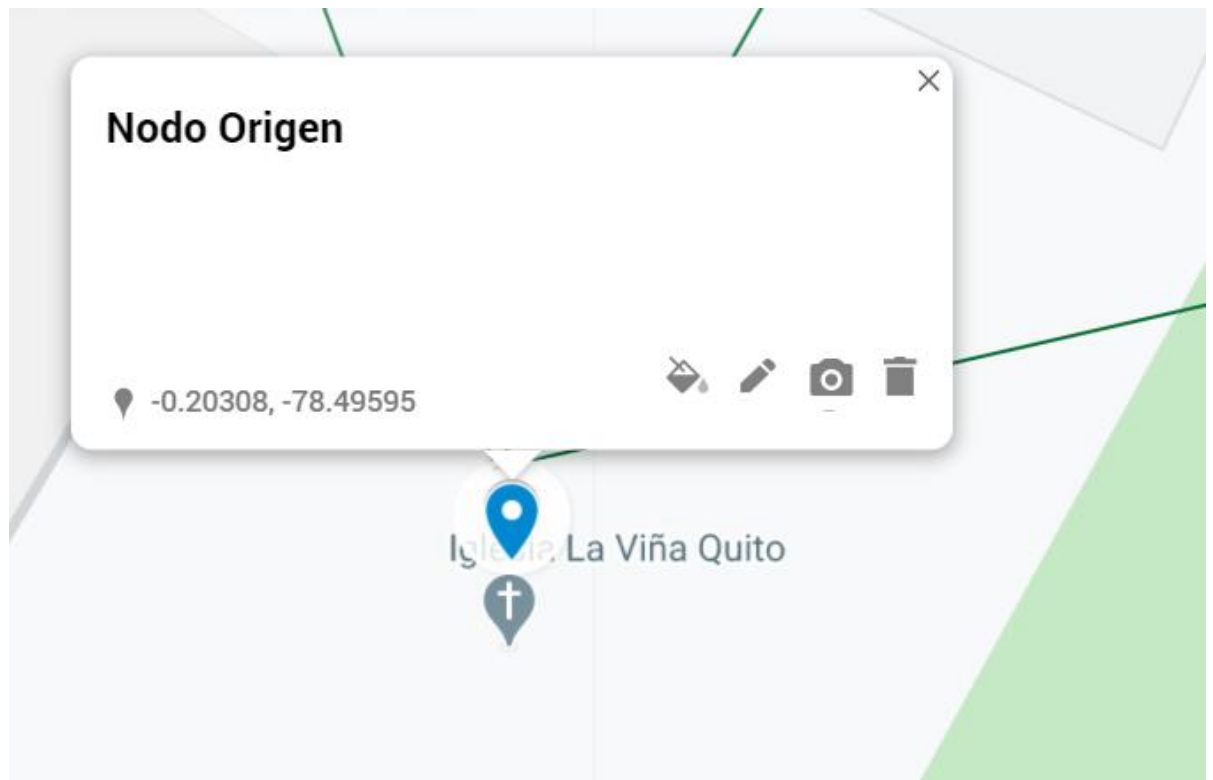


Simulación

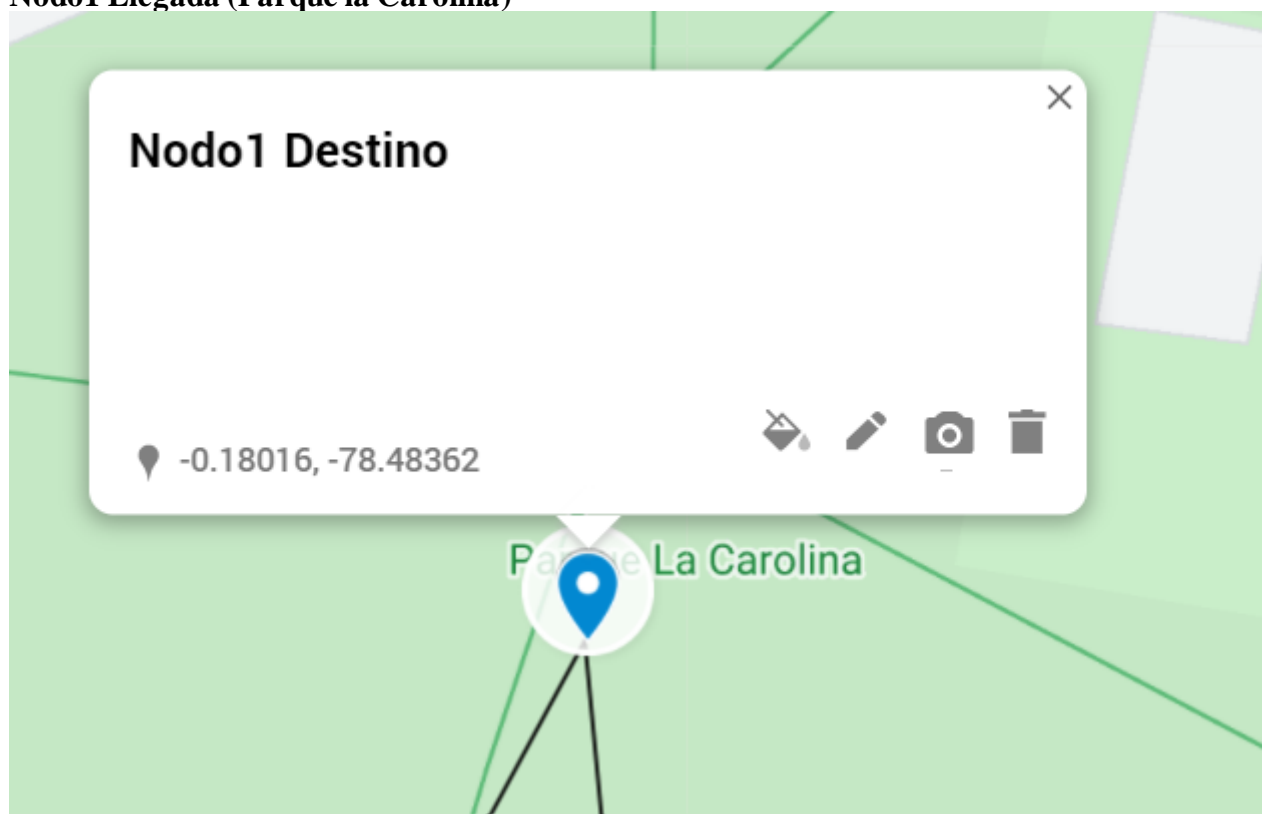
Tema: Inteligencia Artificial 1.

Prueba 2

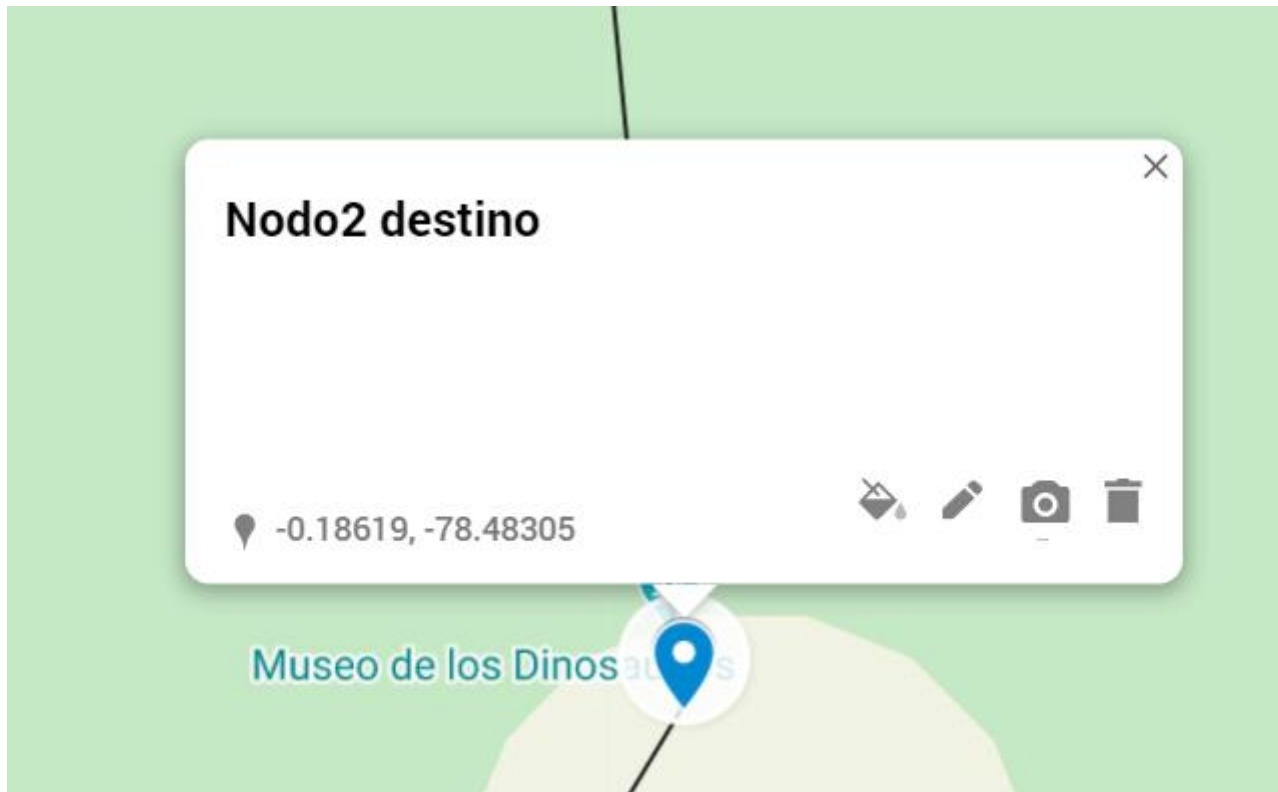
1/26/2021



Nodo1 Llegada (Parque la Carolina)

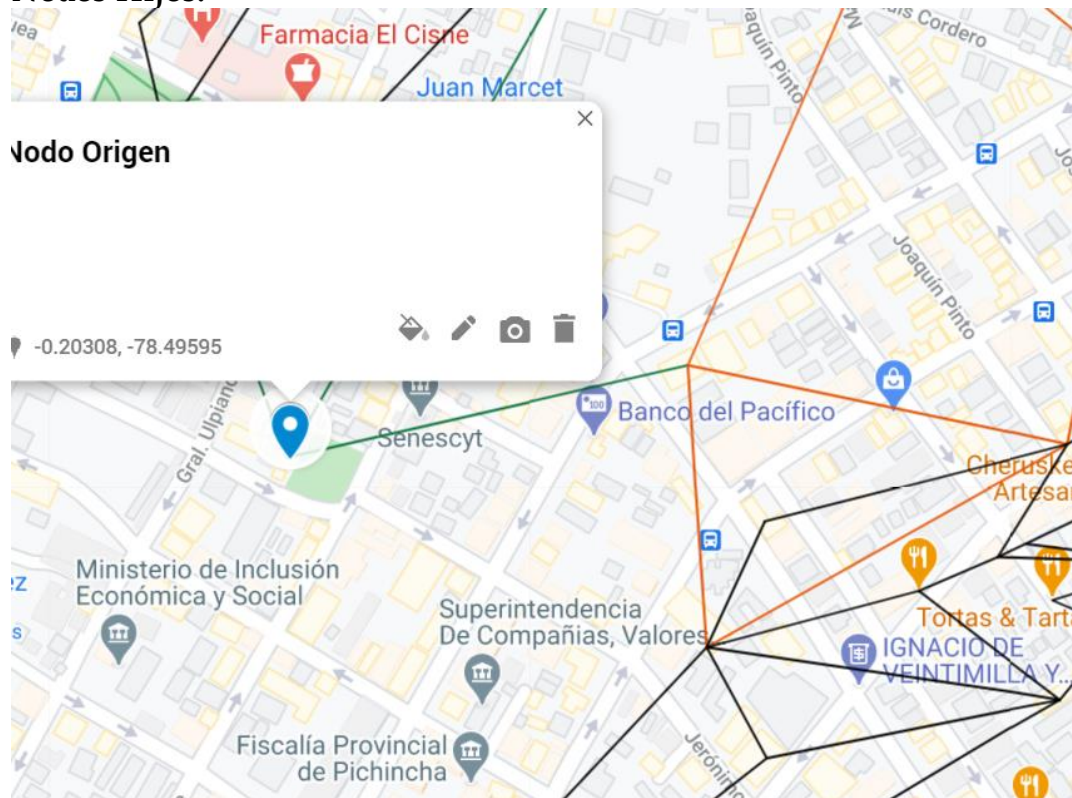


Nodo2 Llegada (Museo de los Dinosaurios)



- Agregar un grafico con los nodos conformados al menos cada nodo debe tener tres o mas hijos.

Nodos Hijos.



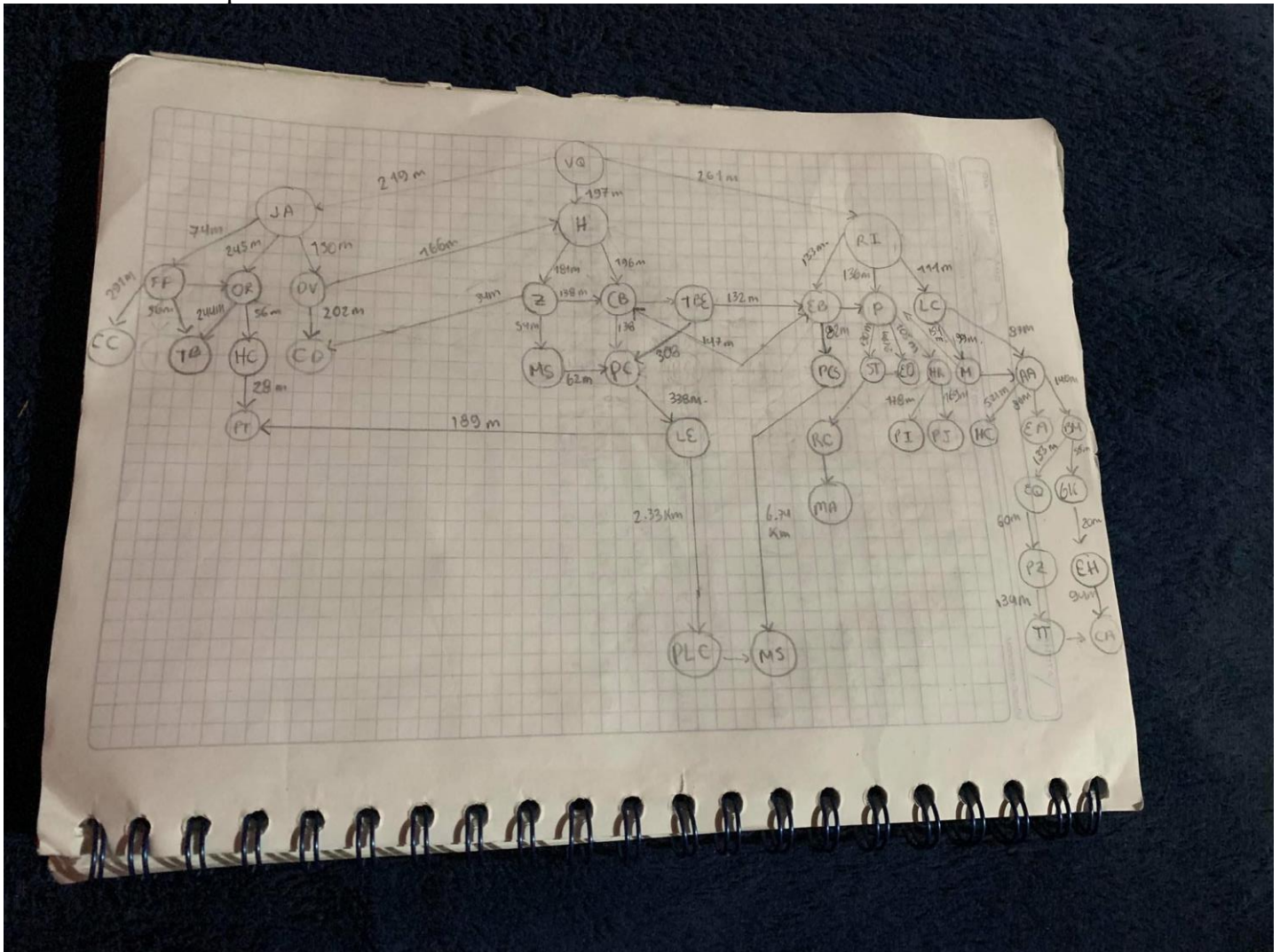
- Generar un árbol de nodos que represente los datos del mapa para realizar la



Prueba 2

1/26/2021

búsqueda.



- Agregar el tipo de medida, además de tomar los datos $h(n)$ = Medición con la herramienta de regla Google, $g(n)$ = Costo de llegar con vehículo.

QUITO			
Nodos	Lugares turísticos	$f(n)$	$g(n)$
1	Iglesia La Viña Quito		
2	Parque Gral. Julio Andrade	219 m.	
3	El Hornero	197 m.	
4	Reina Isabel	261 m.	
5	Hotel Gran Mariscal Quito	448 m.	



Simulación

Tema: Inteligencia Artificial 1.

Prueba 2

1/26/2021

6	Hotel Rio Amazonas Quito	596 m.	
7	Hotel Majestic Quito	506 m.	
8	Parque San Juan	1.72 km	
9	Parque el Ejido	326 m.	
10	Casa San Nicolas	1.16 km	
11	Parque el Arbolito	810 m.	
12	Parque Itchimbia	1.65 km	
13	Museo Jacinto Jijón y Caamaño	364 m.	
14	Museo Abya Yala	608 m.	
15	Parque Cumbayá	6.58 km	
16	Parque los Algarrobos	8.05 km	
17	Parque Guápulo	2.18 km	
18	Parque Metropolitano de Quito	3.42 km	
19	Parque la Carolina	2.30 km	



Simulación

Tema: Inteligencia Artificial 1.

Prueba 2

1/26/2021

20	Parque Arqueológico Ru- mipamba	2.51 km	
21	Parque Gral. Ju- lio Andrade	617 m	
22	Parque de la Mu- jer	1.65 km	
23	Vulkano Park	2.27 km	
24	La Vecindad de los Cuates	556 m.	
25	Mono Bolon Ecua- toriana	978 m.	
26	Pollo Campero	759 m.	
27	Cevicheria Pepe3	490 m.	
28	Apapacho Comida Mexicana	496 m.	
29	Casona Mama Rosa	117 m.	
30	Cofee Burguer	1.09 km	
31	Los Cuys Asados de Mama Olga	1.45 kmn	
32	Resturant el Ricon Manaba	522 m.	
33	Teleferico Quito	246 m.	



Simulación

Tema: Inteligencia Artificial 1.

Prueba 2

1/26/2021

34	Postres Carmita	123 m.	
35	RioFrio Oe 8-48	261 m.	
36	Centro de Arte Contemporaneo de Quito	918 m.	
37	Caravana Fast Food	876 m.	
38	Museo Alberto Mena Caamaño	1.33 km	
39	Museo Casa de Sucre	1.39 km	
40	La Guarida Wings	393 m.	
41	Caldos de Gallina Criolla	249 m.	
42	Vista Hermosa(It-chimbia)	459 m.	
43	Mirador Don Paco	523 m.	
44	Mirador Las Orquideas	1.84 km	
45	Museo de Historia Natural Gustavo Orces	281 m.	
46	Los 3 Monjes	691 m.	
47	Maki Sushi Bar	544 m.	
48	Establo de Alejo	124 m.	



Simulación

Tema: Inteligencia Artificial 1.

Prueba 2

1/26/2021

49	Servis Santafe	244 m.	
50	Resturant Rio San Pedro	82 m.	
51	Tradiciones Manabitas	156 m.	
52	La Terraza	357 m.	
53	Dr Pizzas	930 m.	

- Realizar la búsqueda por Amplitud, Profundidad, Costo Menor, A* y Ascenso por colinas $h(n)$ (Manualmente).

Búsqueda por Amplitud



53

La educación y la cortesía abren todas las puertas

Búsqueda por Amplitud

Lista de Nodos = {JA, H, RI}

Visitados = {VQ}

Lista de Nodos = { }

Visitados = {VQ, JA}

Diagrama de árbol de búsqueda:

```
graph TD
    VQ((VQ)) --- JA((JA))
    VQ --- H((H))
    VQ --- RI((RI))
    JA --- FF((FF))
    JA --- OR((OR))
    JA --- DV((DV))
    H --- Z((Z))
    H --- CB((CB))
    H --- TBE((TBE))
    RI --- EB((EB))
    RI --- P((P))
    RI --- LC((LC))
    FF --- CC((CC))
    FF --- TB((TB))
    OR --- HC((HC))
    DV --- CD((CD))
    DV --- MS((MS))
    Z --- PC((PC))
    Z --- PS((PS))
    CB --- EO((EO))
    TBE --- ST((ST))
    TBE --- HR((HR))
    TBE --- MM((MM))
    TBE --- MA((MA))
    EB --- RC((RC))
    EB --- PI((PI))
    EB --- PS((PS))
    EB --- CA((CA))
    P --- BM((BM))
    P --- EQ((EQ))
    P --- GK((GK))
    LC --- PZ((PZ))
    LC --- TT((TT))
    LC --- EH((EH))
    LC --- CA((CA))
```

Lista de Nodos = {VQ, JA}

Visitados = {VQ, JA}

Lista de Nodos = {VQ, JA, H, RI}

Visitados = {VQ, JA, H, RI}

Lista de Nodos = {VQ, JA, H, RI, FF, OR, DV, Z, CB, TBE, EB, P, LC}

Visitados = {VQ, JA, H, RI, FF, OR, DV, Z, CB, TBE, EB, P, LC}

Lista de Nodos = {VQ, JA, H, RI, FF, OR, DV, Z, CB, TBE, EB, P, LC, CC, TB, HC, CD, MS, PC, PS, EO, ST, HR, MM, MA, RC, PI, PS, CA, BM, EQ, GK, PZ, TT, EH, CA}

Visitados = {VQ, JA, H, RI, FF, OR, DV, Z, CB, TBE, EB, P, LC, CC, TB, HC, CD, MS, PC, PS, EO, ST, HR, MM, MA, RC, PI, PS, CA, BM, EQ, GK, PZ, TT, EH, CA}

$n = 41 \rightarrow$ total nodos

$d = 7 \rightarrow$ profundidad

$$n = \frac{b^{d+1} - 1}{b - 1}$$

$$41 = \frac{b^{7+1} - 1}{b - 1}$$

$$41 = \frac{b^8 - 1}{b - 1}$$

$$b = 1.44863 \rightarrow$$
 factor de ramificación

$$O(b^d) = (1.44863)^4 = 4.40382$$



Simulación

Tema: Inteligencia Artificial 1.

Prueba 2

1/26/2021

Búsqueda por Profundidad

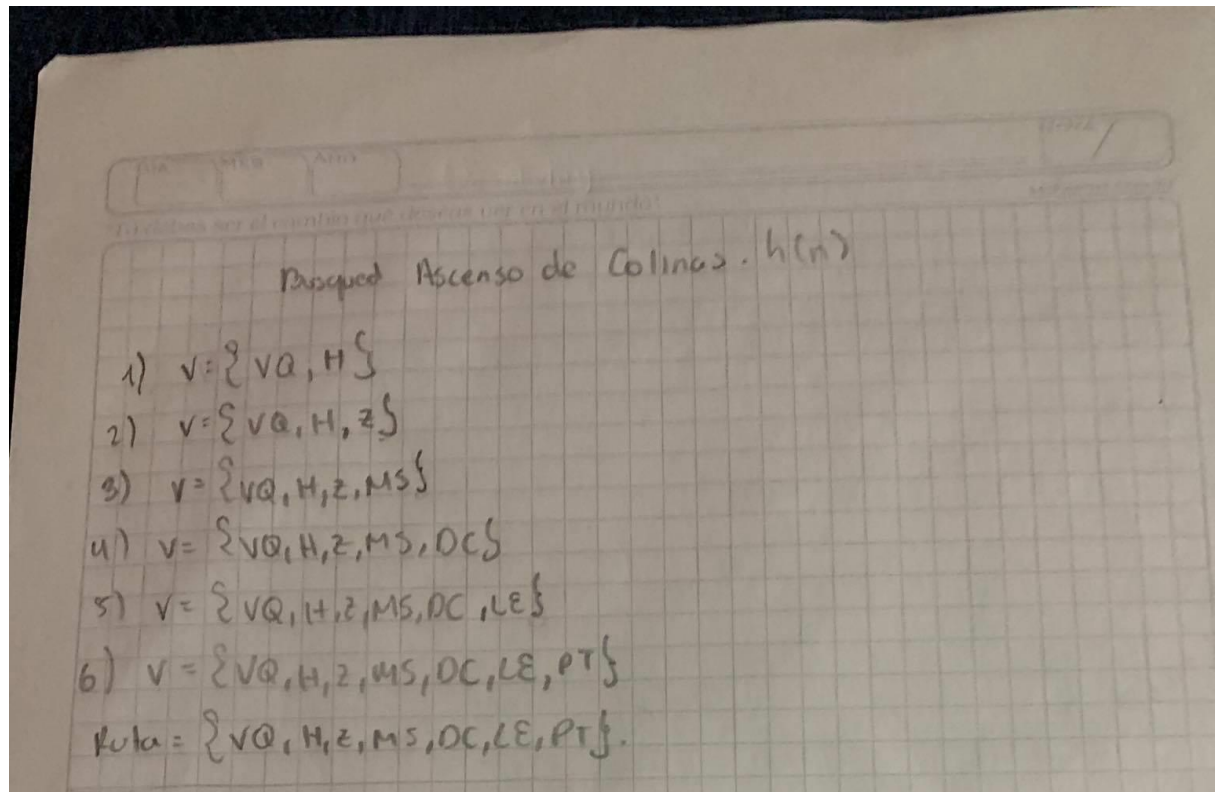
Inicio = Viña Quito - Meta = Parque La Carolina

- $CC = 0$
 $LV = \text{Viña Quito}$
 $LxV = \text{El Horno}$
- $CC = 0 + 197 = 197$
 $LV = \text{Viña Quito} + \text{El Horno}$
 $LxV = \text{Zentrum}$
- $CC = 197 + 181 = 378$
 $LV = \text{Viña Quito, El Horno, Zentrum}$
 $LxV = \text{La Casa del Cangrejo}$
- $CC = 378 + 34 = 412$
 $LV = \text{Viña Quito, El Horno, Zentrum, La Casa del Cangrejo}$
 $LxV = \text{Mesón San Gabriel}$
- $CC = 412 + 54 = 466$
 $LV = \text{Viña Quito, El Horno, Zentrum, La Casa del Cangrejo, Mesón San Gabriel}$
 $LxV = \text{Los Pinchos de la Colón}$
- $CC = 631 + 62 = 693$
 $LV = \text{Viña Quito, El Horno, Zentrum, La Casa del Cangrejo, Mesón San Gabriel, Los Pinchos de la Colón}$
 $LxV = \text{La Empanadería}$
- $CC = 693 + 338 = 1031$
 $LV = \text{Viña Quito, El Horno, Zentrum, La Casa del Cangrejo, Mesón San Gabriel, Los Pinchos de la Colón, La Empanadería}$
 $LxV = \text{Posada Tumbaca}$
- $CC = 1031 + 189 = 1219$
 $LV = \text{Viña Quito, El Horno, Zentrum, La Casa del Cangrejo, Mesón San Gabriel, Los Pinchos de la Colón, La Empanadería, Posada Tumbaca}$
 $LxV = \text{Parque La Carolina}$

Solución = Viña Quito, Zentrum, Mesón San Gabriel, Posada Tumbaca, La Empanadería, Parque La Carolina

Coste = $1219 + 2330 = 3549$

Búsqueda por Ascenso por colinas

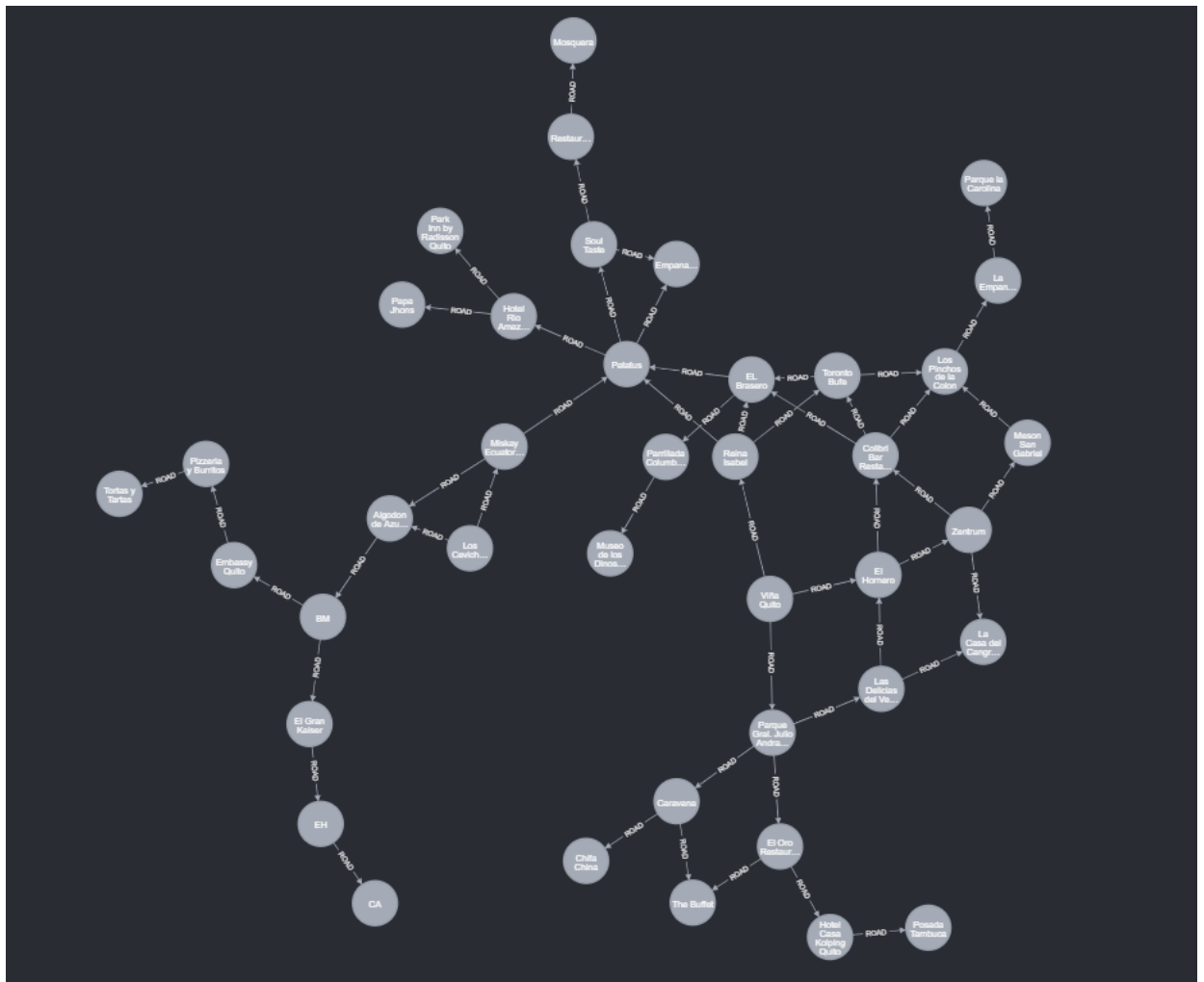


- Programar y presentar los resultados mediante los algoritmos de búsqueda en Neo4j (Amplitud, Profundidad, A*, CostoMenor).
- El proceso de programación desarrollado deberá considerar los siguientes aspectos:
 - Se deberá tener un archivo que tenga todos los procesos o códigos de búsqueda y datos de Neo4j (<https://neo4j.com/docs/labs/apoc/current/export/cypher/>).
 - Los datos de entrada serán los mismo solo cambia el llamar al método.
 - Deben presentar cada algoritmo las siguientes características:
 - Árbol de ingreso.
 - Árbol de nodos resultado.
 - Amplitud, Profundidad, Costo (búsqueda por costo) y A*.
- Distribución de ciudades por estudiante:

Desarrollo en Neo4j

Árbol de Ingreso

a. Creación del grapho en Neo4j y sus relaciones



b. Desarrollo de los algoritmo en cypher

Búsqueda por amplitud

#Busqueda por amplitud

```
MATCH (start:Station {name: 'Viña Quito'}), (end:Station {name: 'Parque la Carolina'})
CALL gds.alpha.shortestPath.stream({
  nodeProjection: 'Station',
  relationshipProjection: {
    ROAD: {
      type: 'ROAD',
      properties: 'time',
      orientation: 'UNDIRECTED'
    }
  },
  startNode: start,
  endNode: end,
  relationshipWeightProperty: 'time'
})
YIELD nodeId, cost
RETURN gds.util.asNode(nodeId).name AS name, cost
```



Simulación

Tema: Inteligencia Artificial 1.

Prueba 2

1/26/2021

```
MATCH (start:Station {name: 'Viña Quito'}), (end:Station {name: 'Parque la Carolina' }) CALL gds.alpha.shortestPath.stream({ nodeProjection: 'Station', relati...
```

name	cost
"Viña Quito"	0.0
"El Homero"	197.0
"Zentrum"	378.0
"Meson San Gabriel"	432.0
"Los Pinchos de la Colon"	494.0
"La Empanaderia"	832.0
"Parque la Carolina"	3162.0

Búsqueda por profundidad

La siguiente declaración creará el gráfico y lo almacenará en el catálogo de gráficos.

```
CALL gds.graph.create('myGraph' , 'Station' , 'ROAD' , { relationshipProperties: 'time'})
```

```
neo4j$ CALL gds.graph.create('myGraph' , 'Station' , 'ROAD' , { relationshipProperties: 'time'})
```

nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	createMillis
{ "Station": { "properties": { }, "label": "Station" } }	{ "ROAD": { "orientation": "NATURAL", "aggregation": "DEFAULT", "type": "ROAD", "properties": { "time": { "property": "time", "aggregation": "DEFAULT", "defaultValue": null } } } }	"myGraph"	42	53	7

```
MATCH (a:Station{name:'Viña Quito'})  
WITH id(a) AS startNode  
CALL gds.alpha.bfs.stream('myGraph', {startNode: startNode})  
YIELD path  
UNWIND [ n in nodes(path) | n.name ] AS tags  
RETURN tags as places  
ORDER BY tags
```



Simulación

Tema: Inteligencia Artificial 1.

Prueba 2

1/26/2021

```
neo4j$ MATCH (a:Station{name:'Viña Quito'}) WITH id(a) AS startNode CALL gds.alpha.bfs.stream('myGraph', {startNode: startNode})
```

places	
1	"Algodon de Azucar"
2	"BM"
3	"CA"
4	"Caravana Fast Food"
5	"Chifa China"
6	"Colibri Bar Restaurante"
7	"El Homero"

Started streaming 42 records after 1 ms and completed after 4 ms.

Ejecución del algoritmo Profundidad con nodos de destino

```
MATCH (a:Station{name:'Viña Quito'}), (d:Station{name:'Reina Isabel'}), (e:Station{name:'Caravana Fast Food'})
WITH id(a) AS startNode, [id(d), id(e)] AS targetNodes
CALL gds.alpha.bfs.stream('myGraph', {startNode: startNode, targetNodes: targetNodes})
YIELD path
UNWIND [ n in nodes(path) | n.name ] AS tags
RETURN tags as places
ORDER BY tags
```

Dado que no se especifica ninguna de las opciones para la terminación anticipada, se visita hasta el nodo especificado

```
neo4j$ MATCH (a:Station{name:'Viña Quito'}), (d:Station{name:'Reina Isabel'}), (e:Station{name:'Caravana Fast Food'}) WITH i...
```

places	
1	"El Homero"
2	"Parque Gral. Julio Andrade"
3	"Reina Isabel"
4	"Viña Quito"

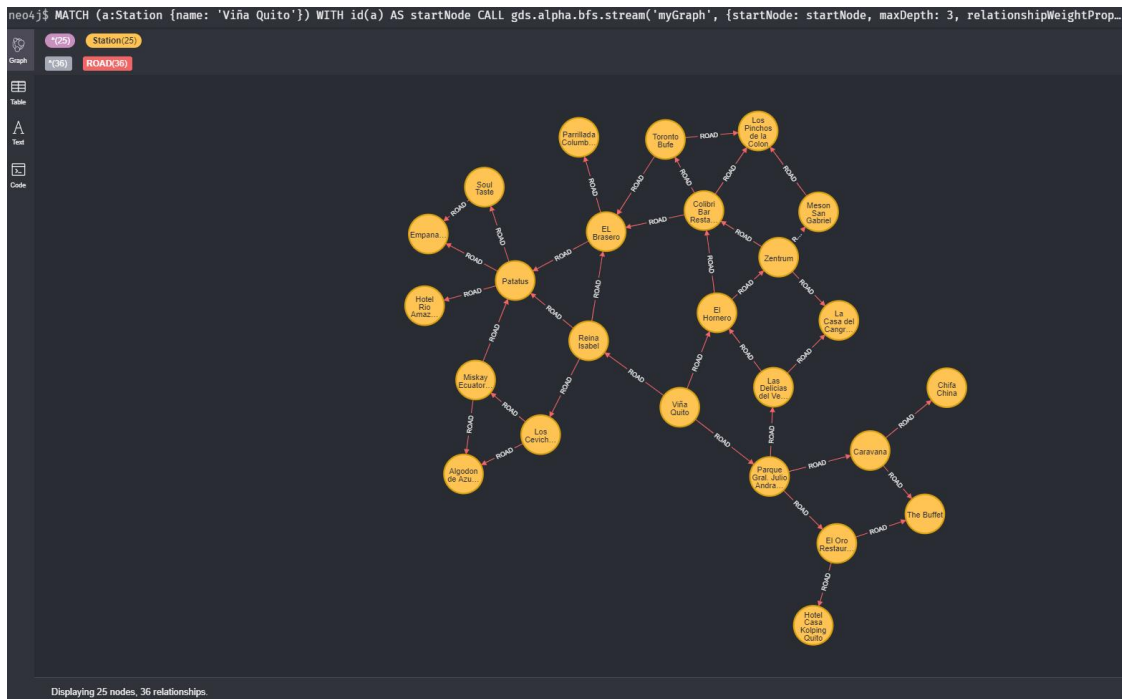
Started streaming 4 records after 1 ms and completed after 4 ms.

Ejecutando el algoritmo de profundidad con maxDepth (nivel profundidad), probamos con maxDepth=3



```
MATCH (a:Station {name: 'Viña Quito'})
WITH id(a) AS startNode
CALL gds.alpha.bfs.stream('myGraph',
    {startNode: startNode,
     maxDepth: 3,
     relationshipWeightProperty: 'time'})
YIELD path
UNWIND [n in nodes(path) | n] AS tags
RETURN tags
ORDER BY tags
```

Grafico



Búsqueda por costo

```
#Busqueda por costo
MATCH (start:Station {name: 'Viña Quito'}), (end:Station{name: 'Parque la Carolina'})
  CALL gds.alpha.kShortestPaths.stream({
    nodeProjection: 'Station',
    relationshipProjection: {
      ROAD: {
        type: 'ROAD',
        properties: 'time'
      }
    }
  },
  startNode: start,
  endNode: end,
  k: 1,
  relationshipWeightProperty: 'time'
  )
YIELD index, nodeIds, costs
RETURN [node IN gds.util.asNodes(nodeIds) | node.name] AS places,costs,
reduce(acc = 0.0, cost IN costs | acc + cost) AS totalCost
```




Simulación

Tema: Inteligencia Artificial 1.

Prueba 2

1/26/2021

```
neo4j$ MATCH (start:Station {name: 'Viña Quito'}), (end:Station{name: 'Parque la Carolina'}) CALL gds.alpha.kShortestPaths.st...
```

	places	costs	totalCost
1	"Viña Quito", "El Homero", "Zentrum", "Meson San Gabriel", "Los Pinchos de la Colon", "La Empanaderia", "Parque la Carolina"	[197.0, 181.0, 54.0, 62.0, 338.0, 2330.0]	3162.0

Búsqueda A*

```
MATCH (start:Station {name: 'Viña Quito'}), (end:Station {name: 'Parque la Carolina'})
CALL gds.alpha.shortestPath.astar.stream({
  nodeProjection: {
    Station: {
      properties: ['longitude', 'latitude']
    }
  },
  relationshipProjection: {
    ROAD: {
      type: 'ROAD',
      orientation: 'UNDIRECTED',
      properties: 'time'
    }
  },
  startNode: start,
  endNode: end,
  propertyKeyLat: 'latitude',
  propertyKeyLon: 'longitude'
})
YIELD nodeId, cost
RETURN gds.util.asNode(nodeId).name AS station, cost
```

```
neo4j$ MATCH (start:Station {name: 'Viña Quito'}), (end:Station {name: 'Parque la Carolina'}) CALL gds.alpha.shortestPath.as...
```

	station	cost
1	"Viña Quito"	0.0
2	"Reina Isabel"	1.0
3	"EL Braserero"	2.0
4	"Toronto Bufo"	3.0
5	"Los Pinchos de la Colon"	4.0
6	"La Empanaderia"	5.0
7	"Parque la Carolina"	6.0

Started streaming 7 records after 1 ms and completed after 27 ms.

Fecha de Entrega: 28/01/21 – 14:00

Python: Cuaderno de Python.



Simulación

Tema: Inteligencia Artificial 1.



Prueba 2

1/26/2021