



Objetivo:

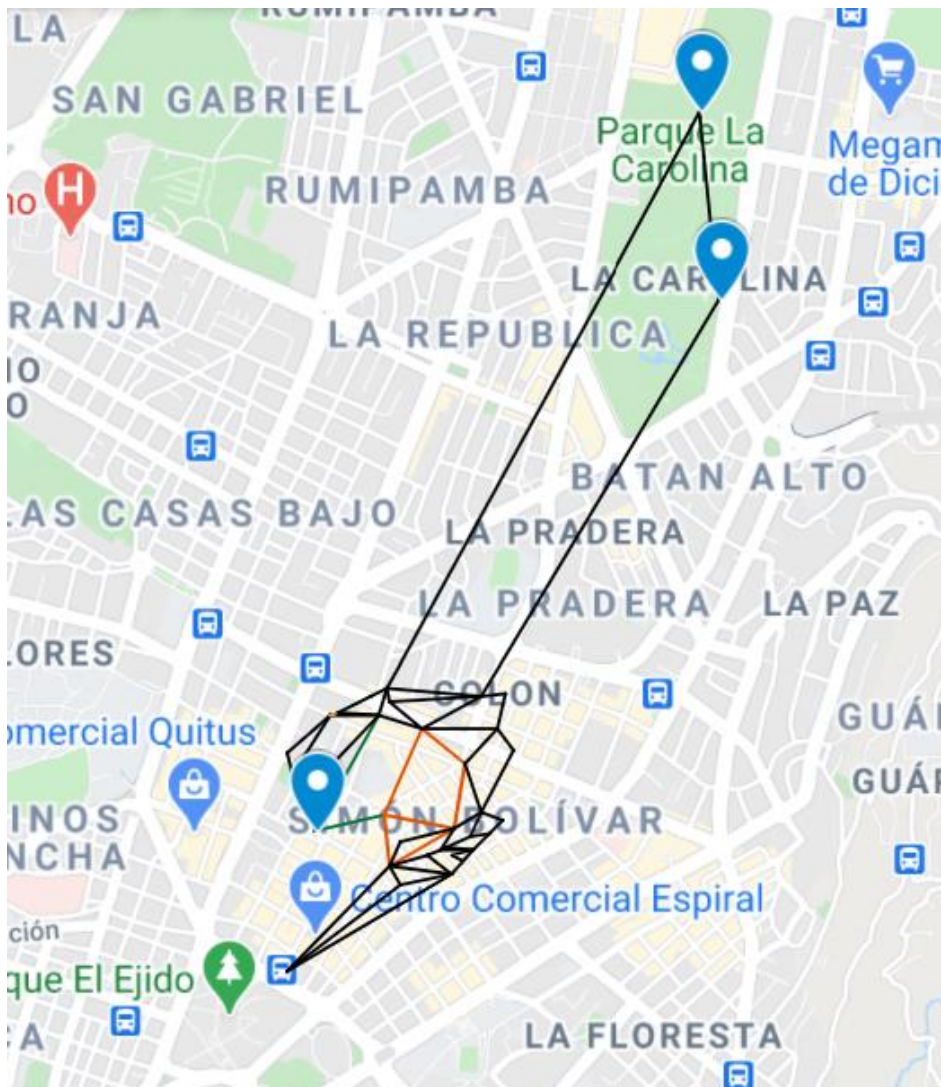
- Consolidar los conocimientos adquiridos en clase para los métodos de búsqueda.

Enunciado:

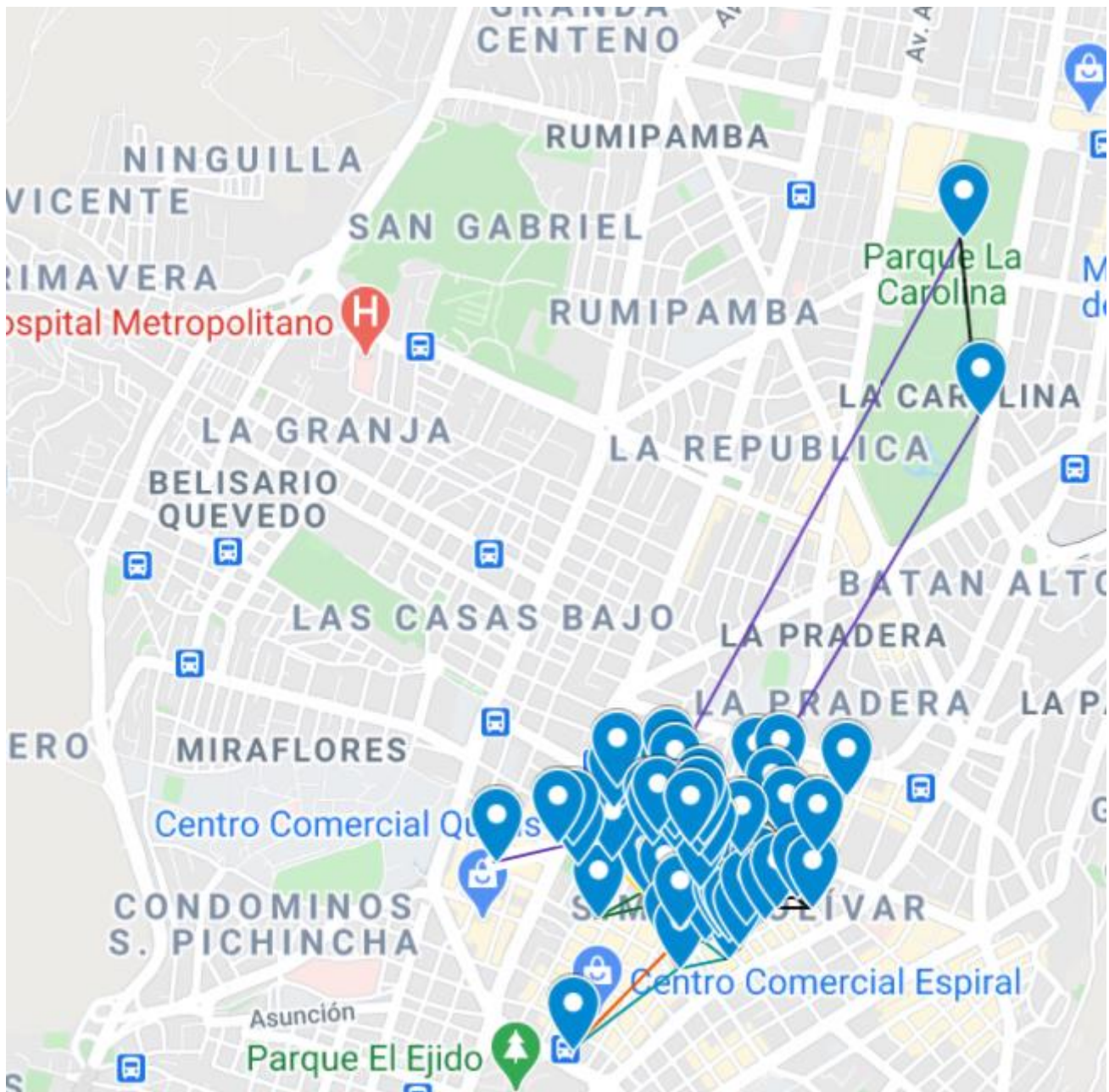
- Diseñe y desarrolle un mapa de nodos para encontrar la ruta mas corta de la iglesia central de cada ciudad a los hoteles y lugares turísticos al menos 50 para cada ciudad, para ello se debe seguir los siguientes pasos :
 - Se tiene los datos dentro de Google Maps (<https://www.google.com/maps/search/iglesias/@-2.891806,-79.013548,14.13z>),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

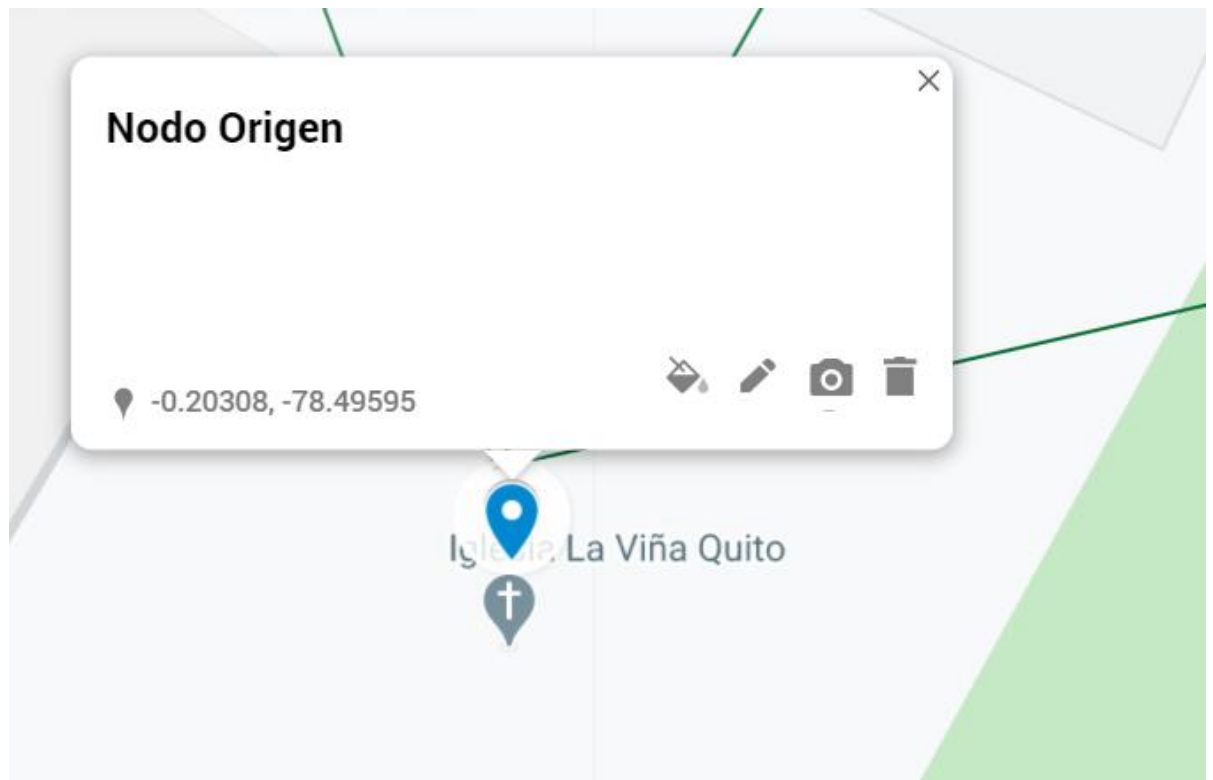


Simulación

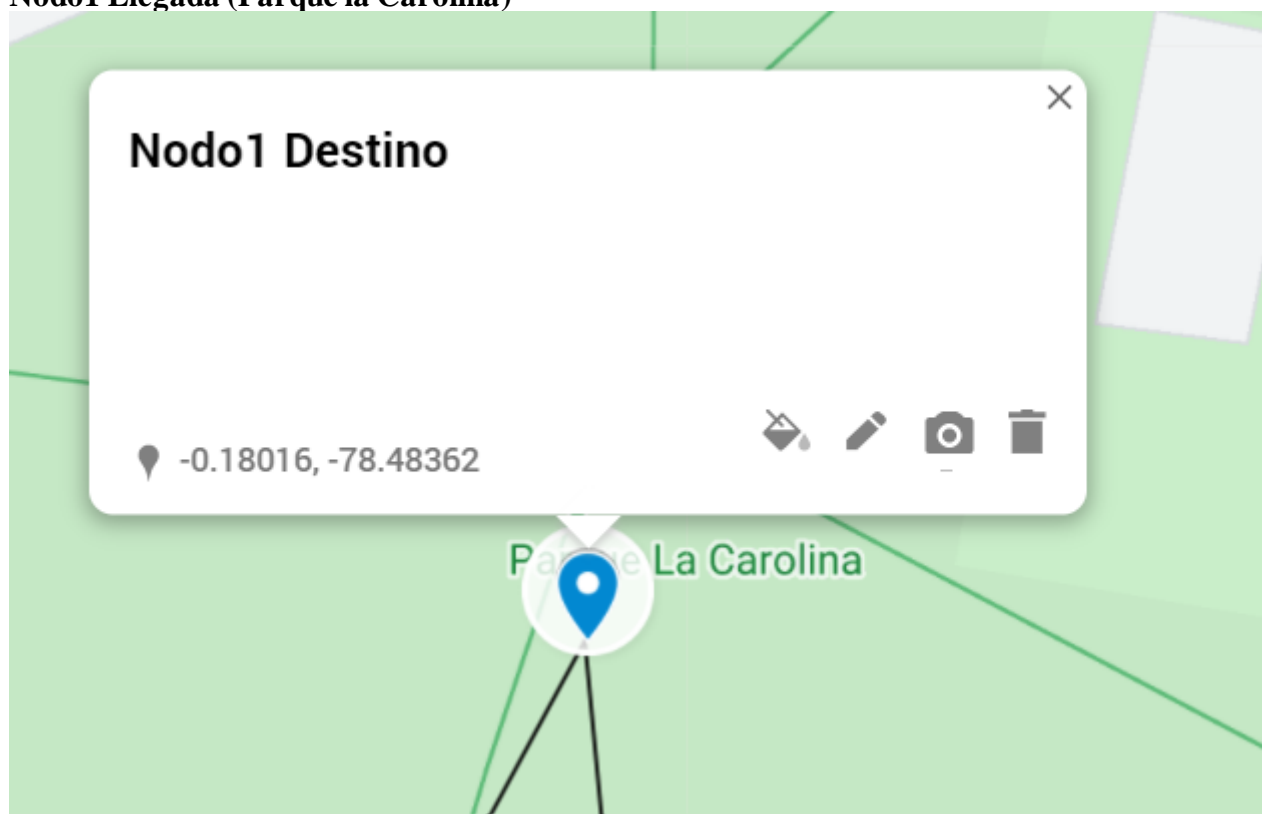
Tema: Inteligencia Artificial 1.

Prueba 2

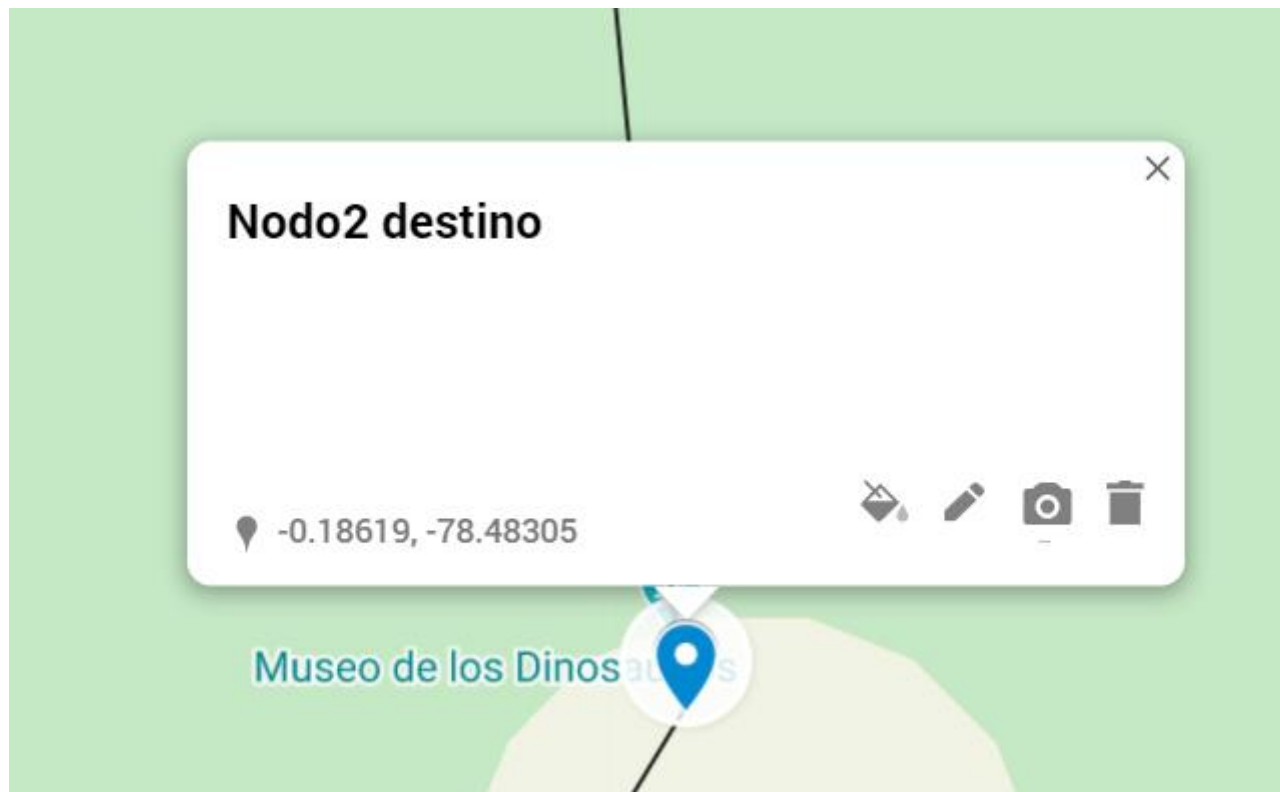
1/26/2021



Nodo1 Llegada (Parque la Carolina)

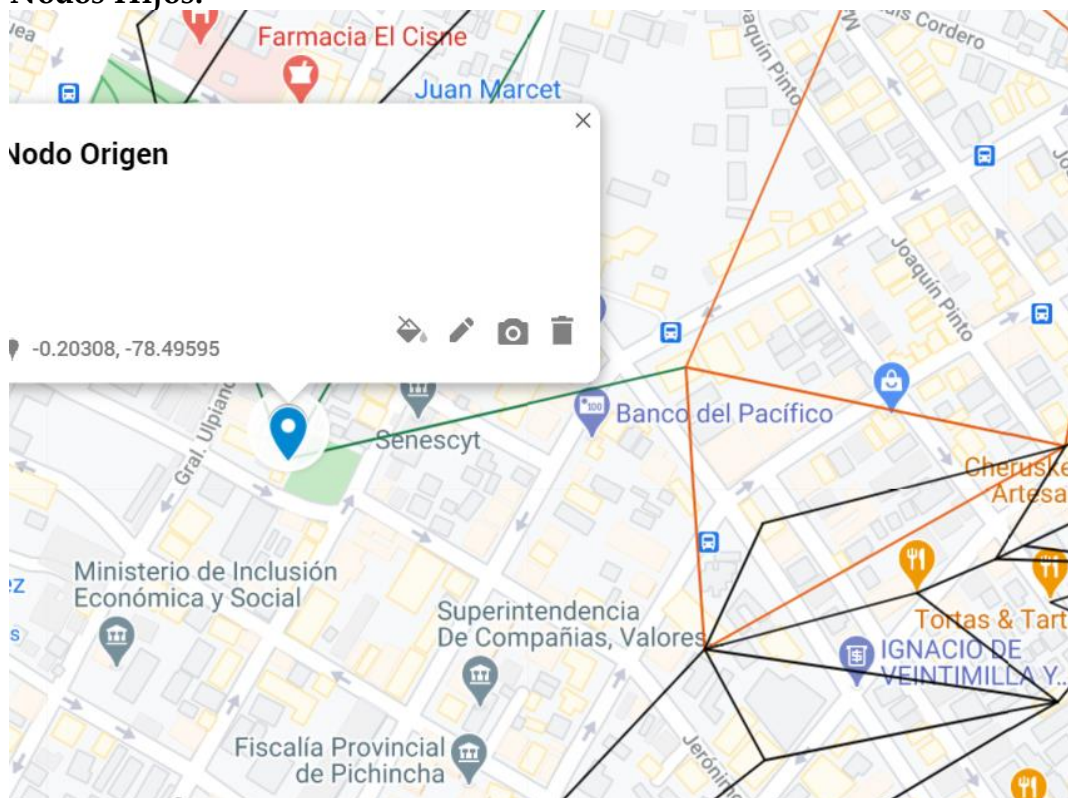


Nodo2 Llegada (Museo de los Dinosaurios)



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

Nodos Hijos.



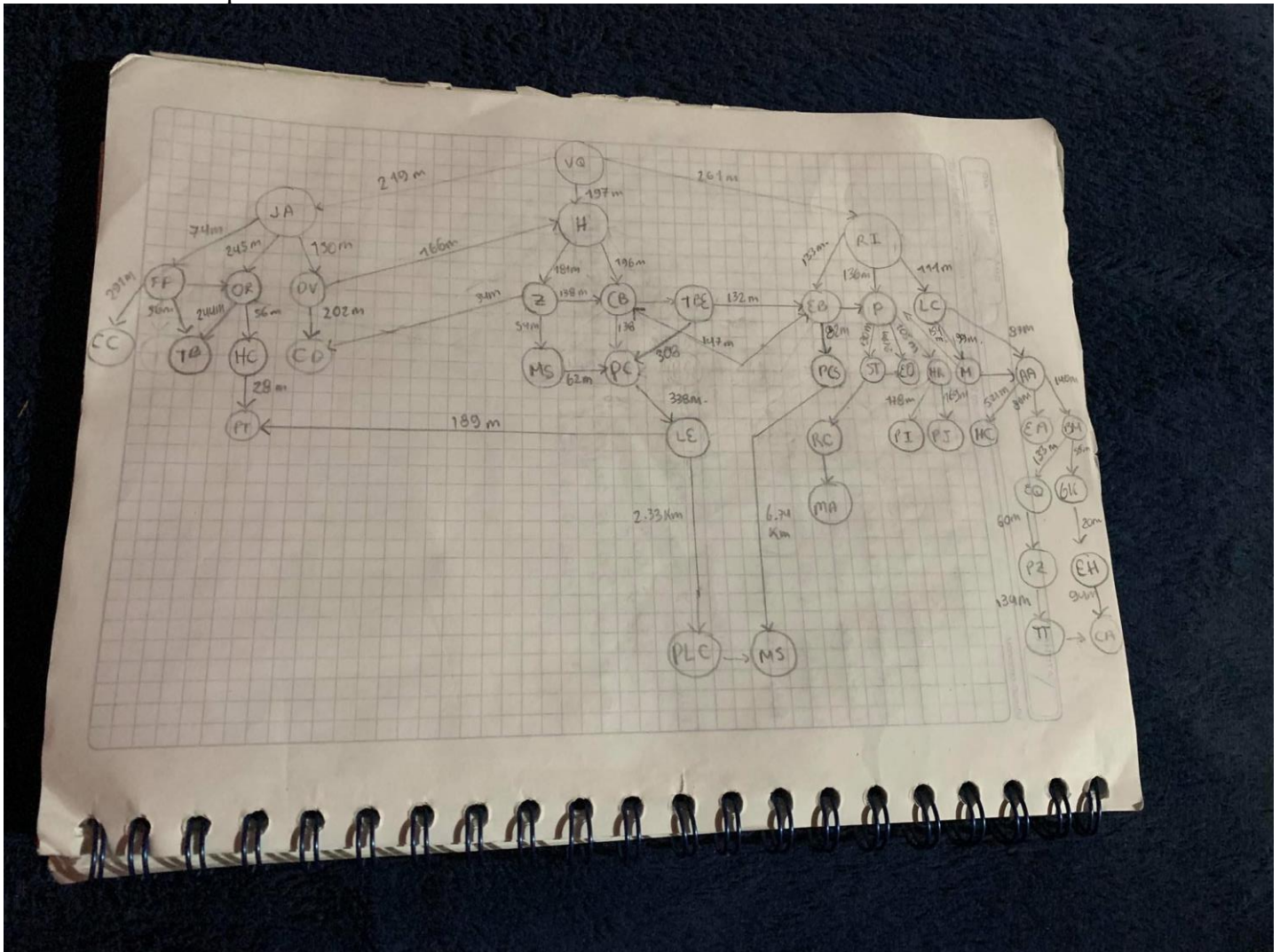
- Generar un árbol de nodos que represente los datos del mapa para realizar la



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. Julio Andrade	617 m	
22	Parque de la Mujer	1.65 km	
23	Vulkano Park	2.27 km	
24	La Vecindad de los Cuates	556 m.	
25	Mono Bolon Ecuatoriana	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	Coffee 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 a la corteza abre todas las puertas

Búsqueda por Amplitud

Lista de Nodos = {JA, H, RI}

Visitados = {VQ}

Lista de Nodos = { }

Visitados = {VQ, JA}

VQ = JA, H, RI
JA = FF, OR, DV
H = Z, CB, TBE
RI = EB, P, LC
FF = CC, TB, HC
OR = DV, H
DV = CC, TB
Z = MS, CB
CB = TBE, PC
TBE = PC, EB
EB = PCS, P
P = ST, EO, HR
LC = M, AA
HC = PT
PC = LE
PCS = MS
ST = BC, EO
HR = PI, PS
M = P, AA
AA = HC, EA, BM
LE = PLC
RC = MA
BM = EQ, GK
EQ = PZ, TT
GK = EH, CA

Diagrama de árbol de búsqueda:

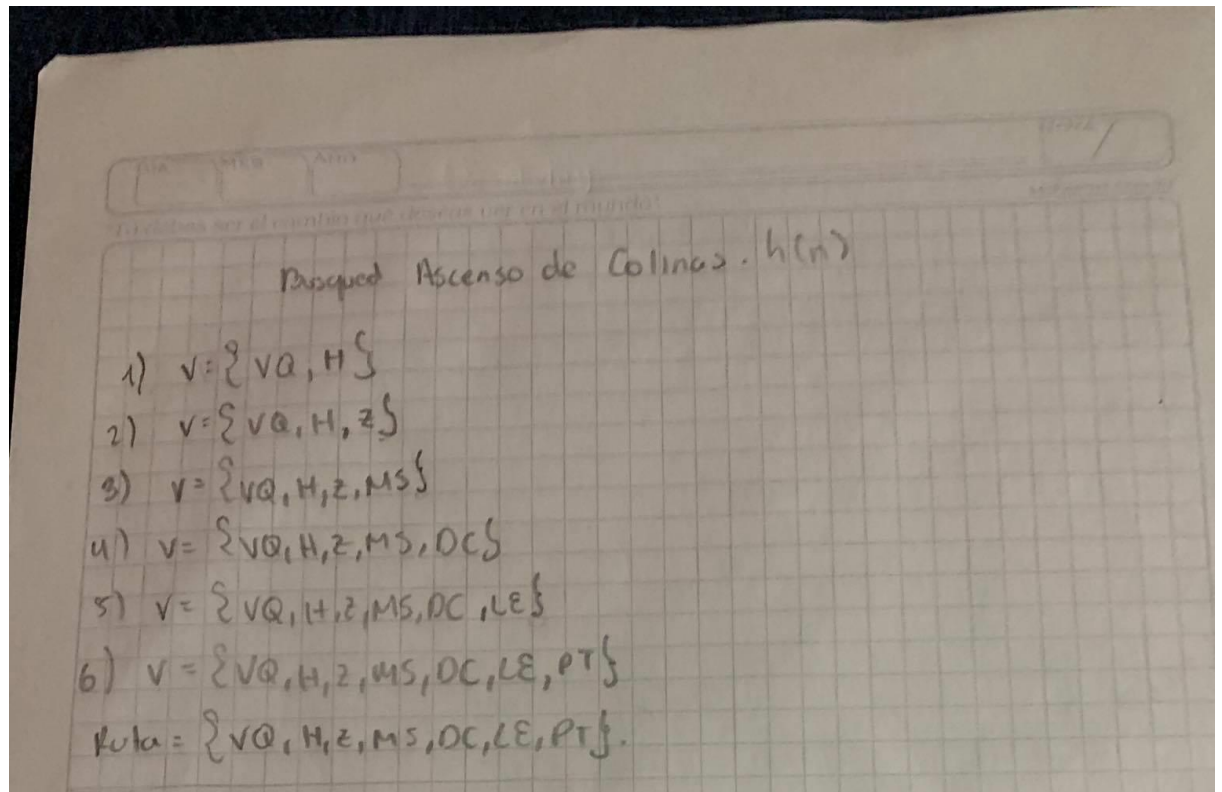
```
graph TD
    VQ((VQ)) --- JA((JA))
    VQ --- H((H))
    VQ --- RI((RI))
    JA --- FF((FF))
    JA --- OR((OR))
    JA --- DV((DV))
    H --- Z((Z))
    H --- CB((CB))
    H --- TBE((TBE))
    RI --- EB((EB))
    RI --- P((P))
    RI --- LC((LC))
    FF --- CC((CC))
    FF --- TB((TB))
    OR ---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

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

Búsqueda por Ascenso por colinas

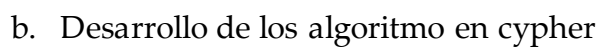
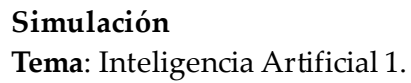


- Programar y presentar los resultados mediante los algoritmos de búsqueda en Neo4j (Amplitud, Profundidad, A*, CostoMenor).
- El proceso de programación desarrollado deberá considerar los siguientes aspectos:
 - Se deberá tener un archivo que tenga todos los procesos o códigos de búsqueda y datos de 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



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

	name	cost
1	"Viña Quito"	0.0
2	"Reina Isabel"	1.1
3	"Toronto Bufe"	2.2
4	"Los Pinchos de la Colon"	3.3000000000000003
5	"La Empanaderia"	4.4
6	"Parque la Carolina"	5.5

Started streaming 6 records after 2 ms and completed after 3815 ms

Fecha de Entrega: 28/01/21 – 14:00

Python: Cuaderno de Python.



Simulación

Tema: Inteligencia Artificial 1.



Prueba 2

1/26/2021

Subir en formato pdf, el proceso manual y capturas de pantalla del Neo4J, además de los scripts dentro del repositorio.