

Universidad Politécnica Salesiana
Rayner Steven Palta Tenecela
Inteligencia Artificial, prueba 2.

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,-79.0135548,14.13z>), generar y agregar un captura de pantalla de la búsqueda y generación de los mapas:
 - Seleccionar como nodo de partido la iglesia central y al menos dos nodos lejanos como la llegada.
 - Agregar un grafico con los nodos conformados al menos cada nodo debe tener tres o mas hijos.
 - Generar un árbol de nodos que represente los datos del mapa para realizar la búsqueda.
 - Agregar el tipo de medida, ademas de tomar los datos $h(n)$ = Medición con la herramienta de regla Google, $g(n)$ = Costo de llegar con vehículo.
 - Realizar la búsqueda por Amplitud, Profundidad, Costo Menor, A^* y Ascenso por colinas $h(n)$ (Manualmente).

Programar y presentar los resultados mediante los algoritmos de búsqueda en Neo4j (Amplitud, Profundidad, A^* , CostoMenor).

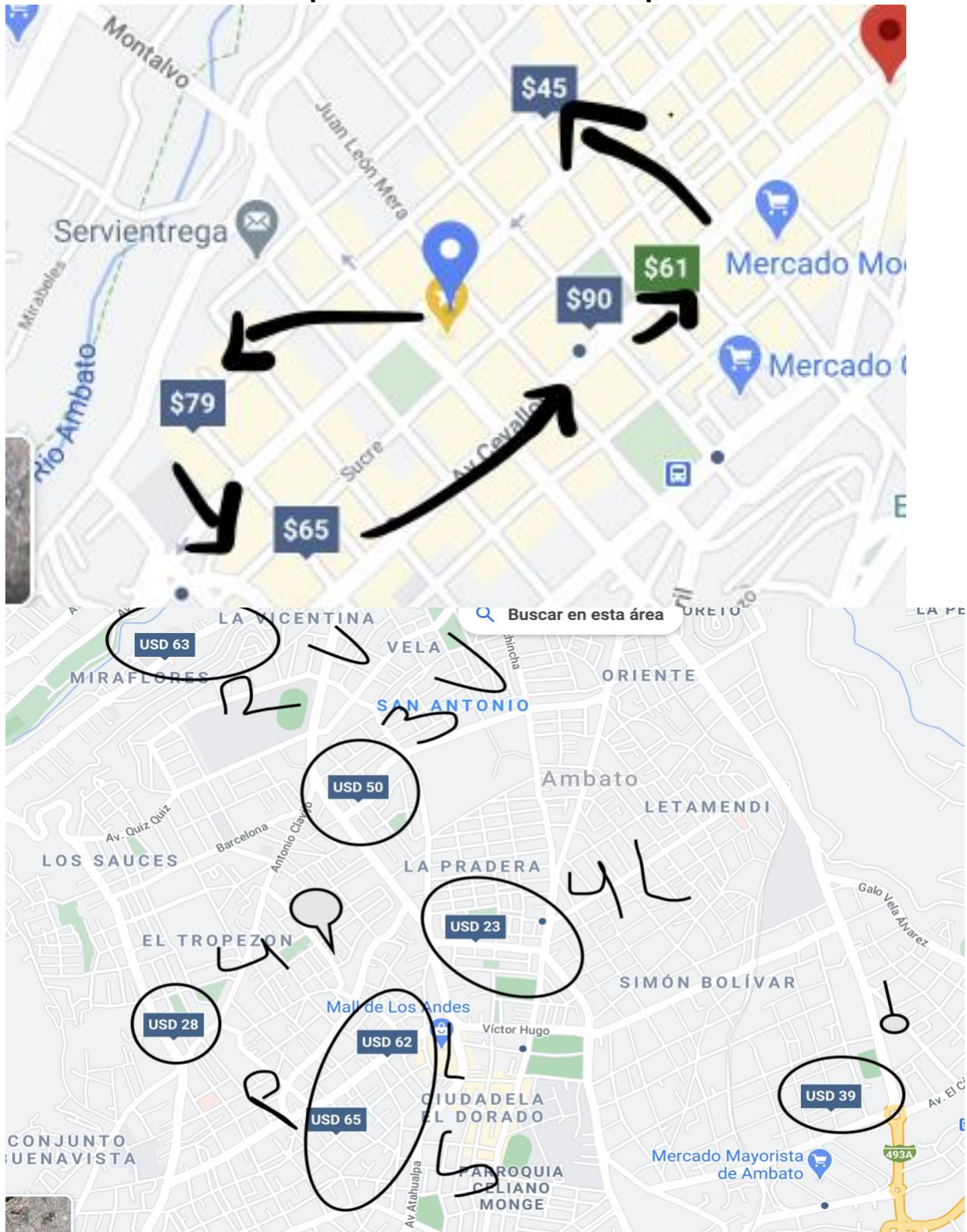
- El proceso de programación desarrollado deberá considerar los siguientes aspectos:
 - Se deberá tener un archivo que tenga todos los procesos o códigos de búsqueda y datos de Neo4j (<https://neo4j.com/docs/labs/apoc/current/export/cypher/>).
 - Los datos de entrada serán los mismo solo cambia el llamar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

Proceso:

Buscar todos los hoteles dentro del rango del punto de inicio: Basílica.



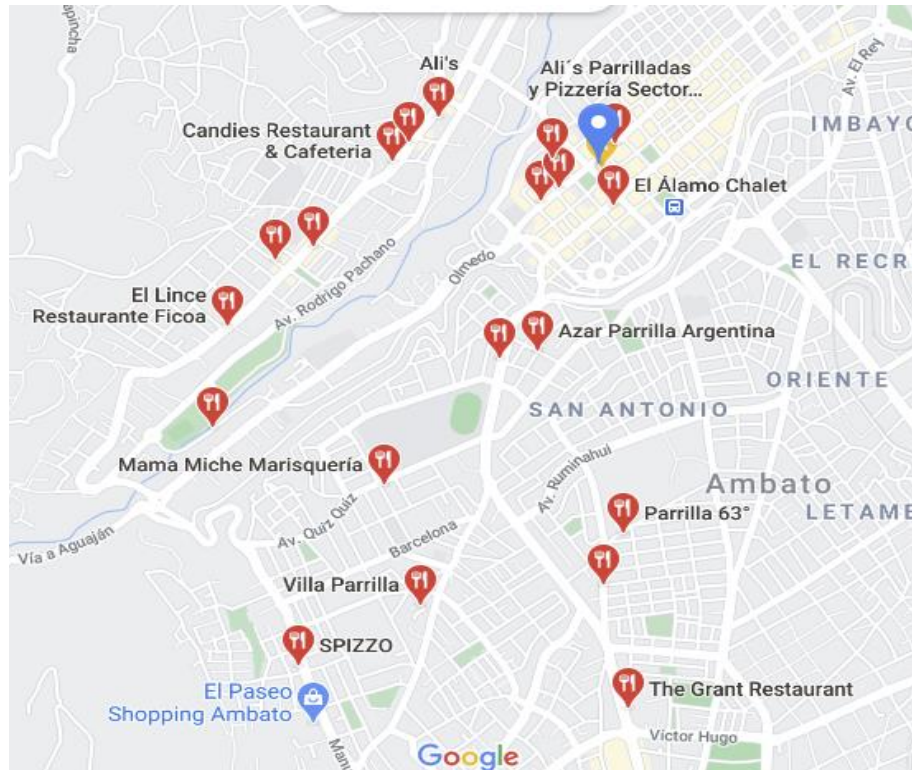
Definir el orden en la que será realizado la búsqueda.





Se hará lo mismo con los restaurant y sitios turísticos.





Una vez definido todos los puntos se procede a crear el árbol de jerarquía para una mejor comprensión del diseño



Algoritmo de profundidad.

eo4j\$ CALL gds.graph.create('profundidad', 'Sitios', 'Cercano', {re...

	nodeProjection	relationshipProjection	graphName	nodeCount	relationshipCount	createMillis
1	<pre>{ "Sitios": { "properties": { }, "label": "Sitios" } }</pre>	<pre>{ "Cercano": { "orientation": "NATURAL", "aggregation": "DEFAULT", "type": "Cercano", "properties": { "gn": { "property": "gn", "aggregation": "DEFAULT", "defaultValue": null } } }</pre>	"profundidad"	90	45	196

eo4j\$ MATCH (a:Sitios{nombre:'Basilica Seniora de Ambato'}), (d:Sit...

nombres	
1	"Azar Parrilla Argentina"
2	"Basilica Seniora de Ambato"
3	"Gran Hotel Napoleon"
4	"Gran Hotel Napoles Emperador 2"
5	"Hotel Colony"
6	"Hotel Cosmopolita Ambato"
7	"... .."

Algoritmo A*

```
MATCH (start:Sitios {nombre: 'Basilica Seniora de Ambato'}), (end:Sitios {nombre: 'Hotel Internacional Prestige'})
```

```
CALL gds.alpha.shortestPath.astar.stream({
```

```
  nodeProjection: {
```

```
    Sitios: {
```

```
      properties: ['hn', 'gn']
```

```
    }
```

```
  },
```

```
  relationshipProjection: {
```

```
    Cercano: {
```

```
      type: 'Cercano',
```

```
      orientation: 'UNDIRECTED'
```

```
    }
```

```
  },
```

```
  startNode: start,
```

```
  endNode: end,
```

```
  propertyKeyLat: 'hn',
```

```
  propertyKeyLon: 'gn'
```

```
})
```

```
YIELD nodeId, cost
```

```
RETURN gds.util.asNode(nodeId).nombre AS station, cost as costo
```

	station	costo
1	"Basilica Seniora de Ambato"	0.0
2	"Hotel Emperador"	1.0
3	"Gran Hotel Napoles Emperador 2"	2.0
4	"Hotel Imperial Ambato"	3.0
5	"Hotel Internacional Prestige"	4.0