



NOMBRE:

ALEX BENAVIDEZ

CARRERA:

INGENIERIA EN SISTEMAS

MATERIA:

INTELIGENCIA ARTIFICIAL

PROFESOR:

DIEGO QUISI

FECHA:

15/05/2020

- **INTRODUCCION**

Se procederá a realizar una búsqueda heurística, en donde utilizaremos el método A*. Para ello utilizaremos los centros educativos de la ciudad de Cuenca para realizar esta búsqueda.

- **DESARROLLO DE CONTENIDOS**

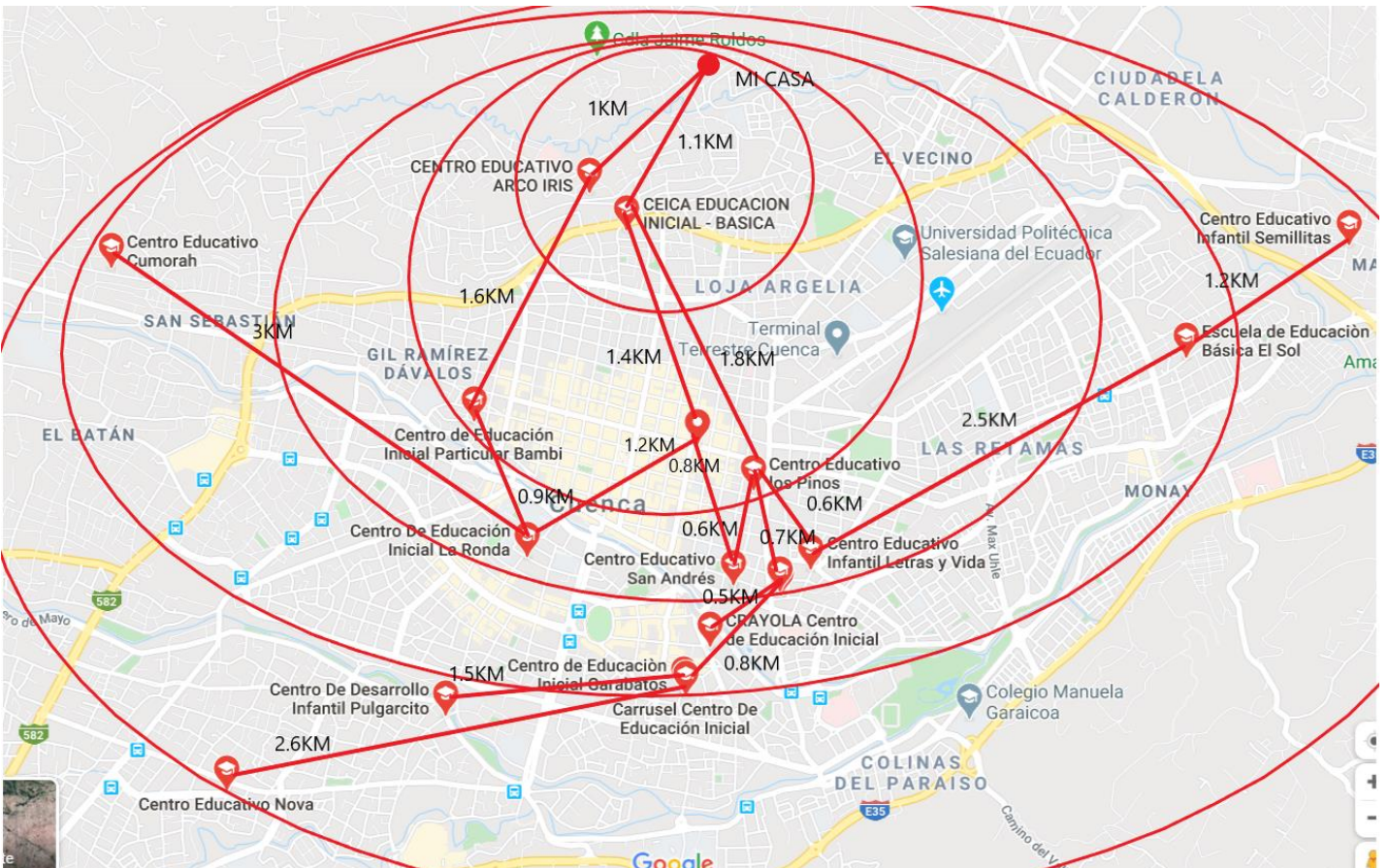
El algoritmo A* fue descrito por primera vez en 1968 por Peter Hart, Nils Nilsson y Bertram Raphael.

- En A*, dividimos el costo de la ruta en dos partes:
 - **g(n)**
Este es el costo de la ruta de acceso desde el punto de partida hasta algún nodo n.
 - **h(n)**
Esto representa el costo estimado de la ruta de acceso desde el nodo n hasta el nodo de destino, calculado por una heurística (una conjetura inteligente).

El algoritmo A* equilibra y, a medida que itera el gráfico, lo que garantiza que en cada iteración elija el nodo con el menor costo total. $g(n) + h(n) = f(n)$

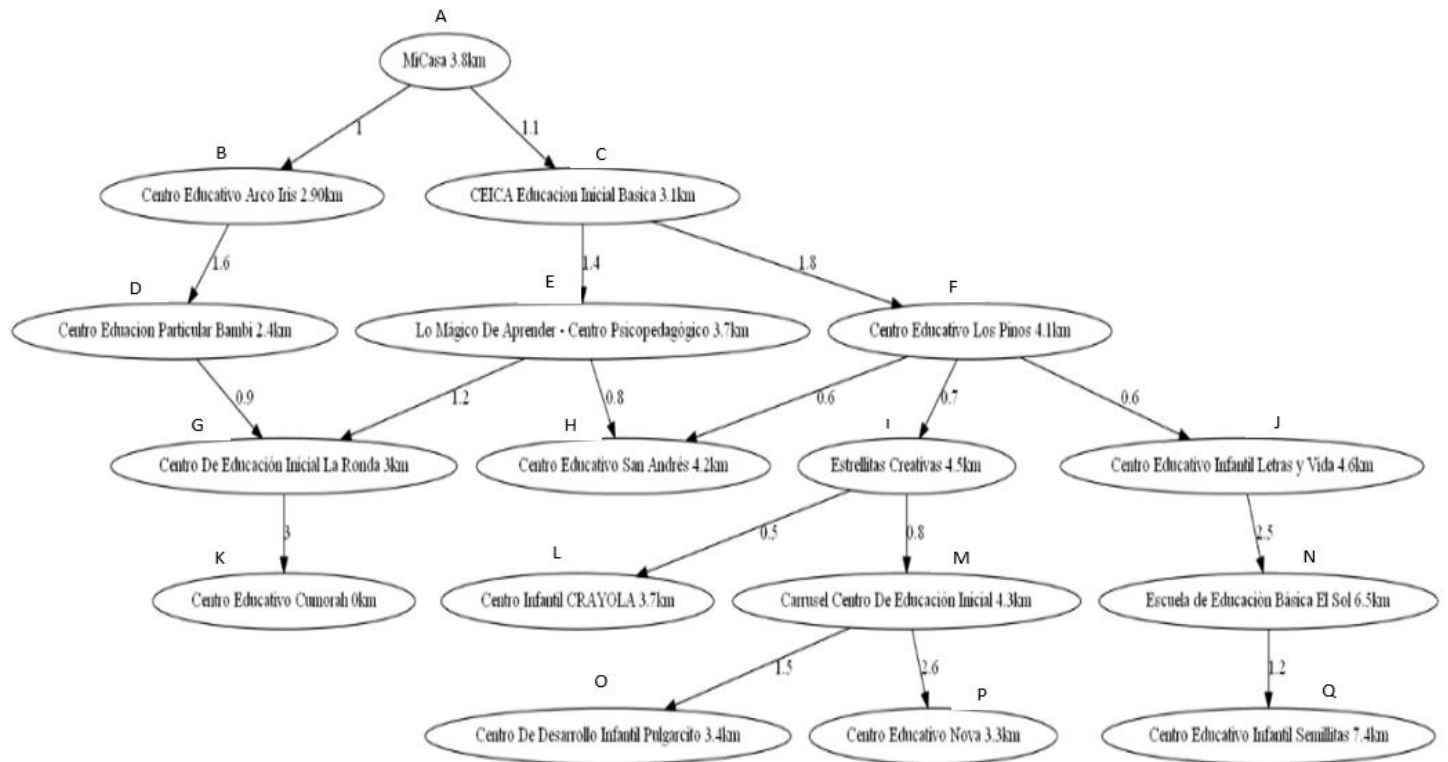
1. Creación de Nodos y relaciones utilizando la herramienta Google Maps.

Para la creación de los nodos y las relaciones hemos utilizado google maps, en donde he procedido a asignar los centros educativos que utilizaremos en esta búsqueda y medir los costos entre ellos. En donde procedemos a calcular tanto nuestro $h(n)$ y nuestro $g(n)$, en este caso nuestro nodo meta será el Centro Educativo Cumorah.



2. Creación del árbol en donde he utilizado el lenguaje de programación Python y algunas librerías.

Para proceder a realizar la búsqueda mediante el algoritmo A*, he procedido a realizar el árbol de búsqueda con el lenguaje de programación Python y unas librerías específicas que me facilitarían realizar esto de manera efectiva.



3. Resolución de la búsqueda utilizando el algoritmo A* de manera manual.

3.1. Primera Iteración

$C = \{A \rightarrow B(3.9), A \rightarrow C(4.2)\}$

$V = \{A(3.8)\}$

3.2. Segunda Iteración

$C = \{A \rightarrow C(4.2), A \rightarrow B \rightarrow D(5)\}$

$V = \{A(3.8), B(3.9)\}$

3.3. Tercera Iteración

$C = \{A \rightarrow B \rightarrow D(5), A \rightarrow C \rightarrow E(6.2), A \rightarrow C \rightarrow F(7)\}$

$V = \{A(3.8), B(3.9), C(4.2)\}$

3.4. Cuarta Iteración

$C = \{A \rightarrow C \rightarrow E(6.2), A \rightarrow B \rightarrow D \rightarrow G(6.5), A \rightarrow C \rightarrow F(7)\}$

$V = \{A(3.8), B(3.9), C(4.2), D(5)\}$

3.5. Quinta Iteración

$C = \{A \rightarrow B \rightarrow D \rightarrow G(6.5), A \rightarrow C \rightarrow E \rightarrow G(6.7), A \rightarrow C \rightarrow F(7), A \rightarrow C \rightarrow E \rightarrow H(7.5)\}$

$V = \{A(3.8), B(3.9), C(4.2), D(5), E(6.2)\}$

3.6. Sexta Iteración

$C = \{A \rightarrow B \rightarrow D \rightarrow G \rightarrow K(6.5), A \rightarrow C \rightarrow F(7), A \rightarrow C \rightarrow E \rightarrow H(7.5)\}$

$V = \{A(3.8), B(3.9), C(4.2), D(5), E(6.2), G(6.5)\}$

3.7. Solución

$RUTA = \{A \rightarrow B \rightarrow D \rightarrow G \rightarrow K\}$

$COSTO = 6.5$

4. Resolución de la búsqueda utilizando la base de datos orientada a Grafos NEO4J

Para la utilización de esta base de datos crearemos primeramente los nodos y las conexiones y seguidamente utilizaremos una sentencia que nos ayudara a resolver esta búsqueda mediante el algoritmo A*

4.1. Creación de nodos y relaciones.

```
CREATE (a:Centro {name: 'MiCasa', latitude: -2.876008,
longitude: -78.999377}),
(b:Centro {name: 'Centro Educativo Arco Iris', latitude: -
2.883164, longitude: -79.008018}),
(c:Centro {name: 'CEICA Educacion Inicial Basica', latitude:
-2.885121, longitude: -79.004263}),
(d:Centro {name: 'Centro Educacion Particular Bambi',
latitude: -2.895750, longitude: -79.012503}),
(e:Centro {name: 'Lo Mágico De Aprender - Centro
Psicopedagógico', latitude: -2.897013, longitude: -
79.000414}),
(f:Centro {name: 'Centro Educativo Los Pinos', latitude: -
2.899537, longitude: -78.997447}),
(g:Centro {name: 'Centro De Educacion Inicial La Ronda',
latitude:-2.903223, longitude: -79.009643 }),
(h:Centro {name: 'Centro Educativo San Andrés', latitude: -
2.904777, longitude: -78.998520}),
(i:Centro {name: 'Estrellitas Creativas', latitude:-
2.905079, longitude: -78.995992}),
```

```

(j:Centro {name: 'Centro Educativo Infantil Letras y Vida',
latitude: -2.903915, longitude: -78.994266}),
(k:Centro {name: 'Centro Educativo Cumorah', latitude: -
2.887367, longitude: -79.032232}),
(l:Centro {name: 'Centro Infantil CRAYOLA', latitude: -
2.908399, longitude: -79.006940}),
(m:Centro {name: 'Carrusel Centro De Educación Inicial',
latitude:-2.910838, longitude: -79.001074}),
(n:Centro {name: 'Escuela de Educaciòn Básica El Sol',
latitude: -2.892269, longitude: -78.973936}),
(o:Centro {name: 'Centro De Desarrollo Infantil Pulgarcito',
latitude: -2.911889, longitude: -79.014078}),
(p:Centro {name: 'Centro Educativo Nova', latitude: -
2.916041, longitude: -79.025959}),
(q:Centro {name: 'Centro Educativo Infantil Semillitas',
latitude: -2.886121, longitude: -78.965176}),
(a)-[:CONNECTION {distancia: 1}]->(b),
(a)-[:CONNECTION {distancia: 1.1}]->(c),
(b)-[:CONNECTION {distancia: 1.6}]->(d),
(c)-[:CONNECTION {distancia: 1.4}]->(e),
(c)-[:CONNECTION {distancia: 1.8}]->(f),
(d)-[:CONNECTION {distancia: 0.9}]->(g),
(e)-[:CONNECTION {distancia: 1.2}]->(g),
(e)-[:CONNECTION {distancia: 0.8}]->(h),
(f)-[:CONNECTION {distancia: 0.6}]->(h),
(f)-[:CONNECTION {distancia: 0.7}]->(i),
(f)-[:CONNECTION {distancia: 0.6}]->(j),
(g)-[:CONNECTION {distancia: 3}]->(k),
(i)-[:CONNECTION {distancia: 0.5}]->(l),
(i)-[:CONNECTION {distancia: 0.8}]->(m),
(j)-[:CONNECTION {distancia: 2.5}]->(n),
(m)-[:CONNECTION {distancia: 1.5}]->(o),
(m)-[:CONNECTION {distancia: 2.6}]->(p),
(n)-[:CONNECTION {distancia: 1.2}]->(q)

```

4.2. Creación de la sentencia

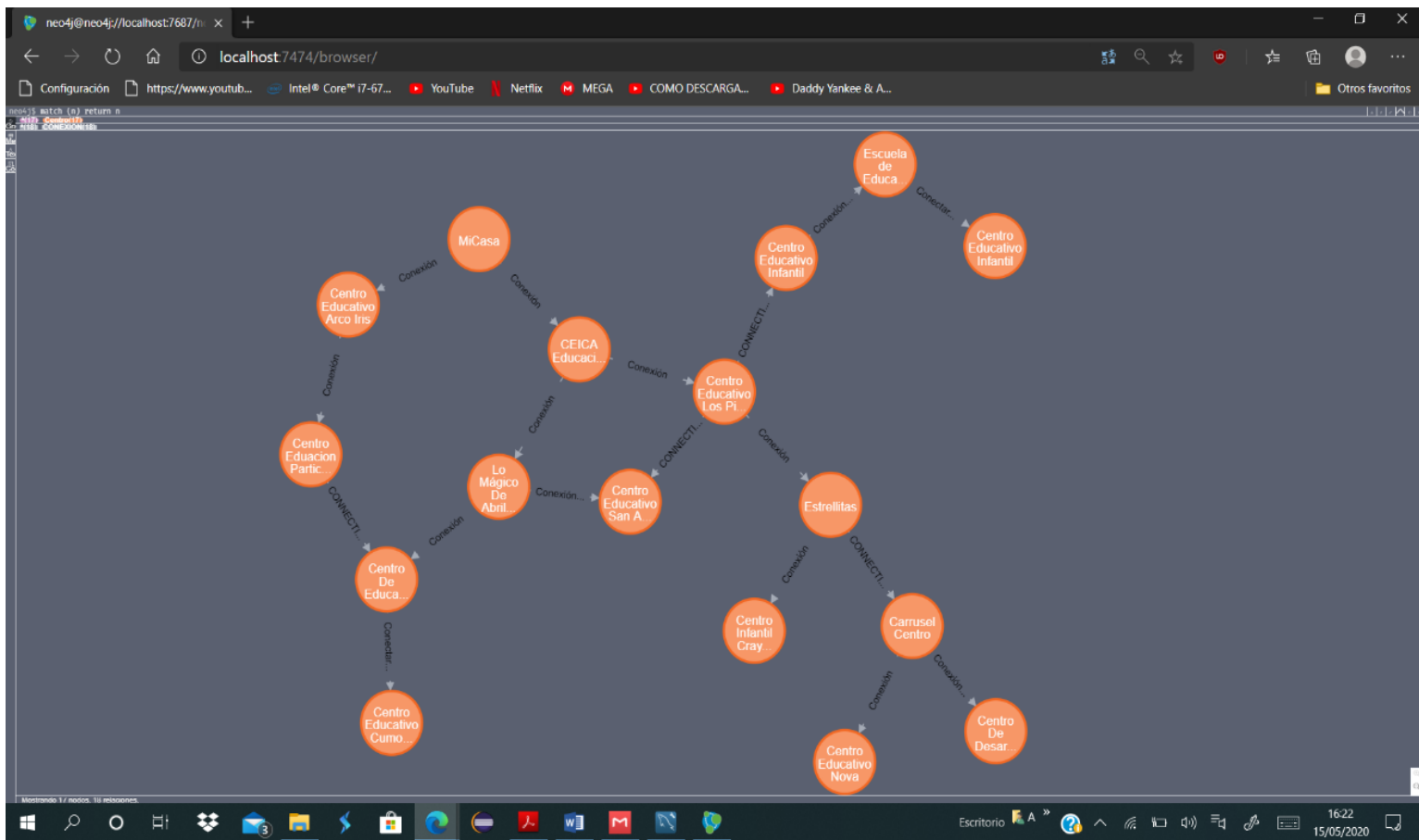
```

MATCH (start:Centro {name: "MiCasa"}), (end:Centro {name:
"Centro Educativo Cumorah"})
CALL gds.alpha.shortestPath.astar.stream({
  nodeQuery: 'MATCH (p:Centro) RETURN id(p) AS id',
  relationshipQuery: 'MATCH (p1:Centro)-[r:CONNECTION]-
>(p2:Centro) RETURN id(p1) AS source, id(p2) AS target,
r.distancia AS weight',
  startNode: start,
  endNode: end,
  relationshipWeightProperty: 'weight',
  propertyKeyLat: 'latitude',
  propertyKeyLon: 'longitude'
})
YIELD nodeId, cost
RETURN gds.util.asNode(nodeId).name AS station, cost

```

4.3. Verificación del algoritmo en Neo4j.

- Para verificar que las sentencias antes descritas han funcionado procedemos a visualizar el grafico generado, mediante la consulta match (n) return n.



- Ahora verificamos si la sentencias creada anteriormente para que me haga la búsqueda funcione correctamente, y me imprima el resultado correcto.

The screenshot shows the Neo4j Browser interface at localhost:7474. The main area displays a Cypher query and its results. The query is: `MATCH (start:Centro ?name: 'MiCasa'), (end:Centro ?name: 'Centro Educativo Cumorah')-> CALL gds.alpha.shortestPath.astar.stream(' nodeQuery: 'MATCH (p:Centro) RETURN p')`. The results are displayed in a table with two columns: Estación and Costo.

Estación	Costo
"MiCasa"	0.0
"Centro Educativo Arco Iris"	1.0
"Centro Educativo Particular Bambi"	2.6
"Centro de Educación Inicial La Ronda"	3.5
"Centro Educativo Cumorah"	6.5

- En este caso podemos apreciar que el algoritmo ha servido de manera correcta, y nos ha imprimido el resultado correcto, el cual hemos sacado primeramente de manera manual.

• CONCLUSIONES

- Se ha realizado la práctica de manera correcta, en donde se ha realizado de manera manual y utilizando la base de datos orientada a grafos neo4j.
- Se ha implementado el método heurístico A* de manera correcta, en donde se ha podido fortalecer los conocimientos del uso del mismo.
- Se ha podido desarrollar el método A* en la base de datos orientada a grafos neo4j correctamente.

BIBLIOGRAFÍA

Francisco Escolano Ruiz, M. A. (23). *Inteligencia artificial: modelos, técnicas y áreas de aplicación*. Alicante: Thomsom.

Malagón, C. (s.f.). *www.nebrija.es*. Obtenido de https://www.nebrija.es/~cmalagon/ia/transparencias/busqueda_heuristica.pdf

neo4j.com. (2020). Obtenido de https://neo4j.com/docs/graph-data-science/current/alpha-algorithms/a_star/