Alumno:

Juan Cañar

Docente:

Ing. Diego Quisi.

Materia:

IA

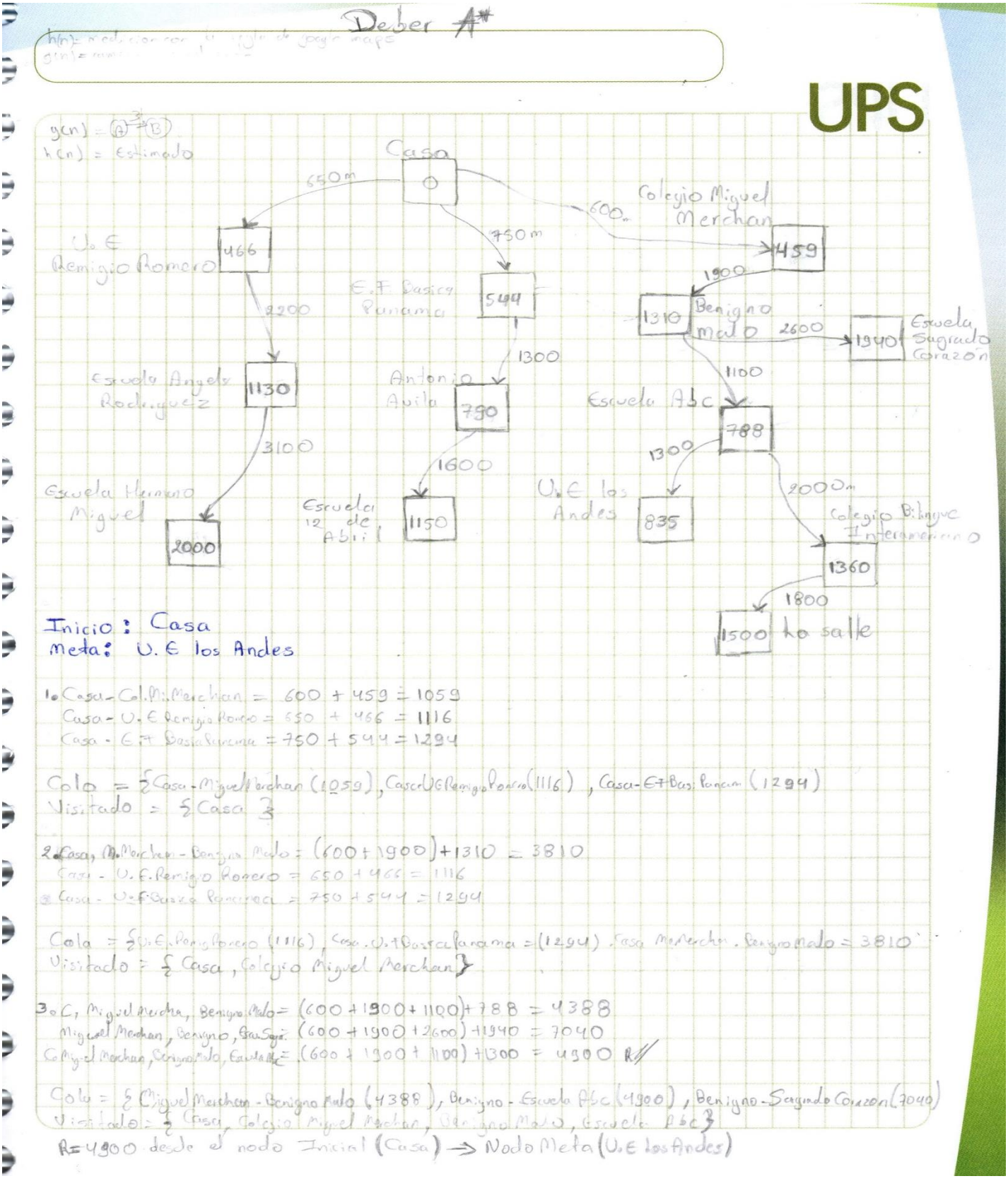
Ciclo:

9no

Fecha:

15/05/2020

Algoritmo A* Resolución a mano:



Algoritmo A*

Para el cálculo de caminos mínimos en una red. Se va a tratar de un algoritmo heurístico, ya que una de sus principales características es que hará uso de una función de evaluación heurística, mediante la cual etiquetará los diferentes nodos de la red y que servirá para determinar la probabilidad de dichos nodos de pertenecer al camino óptimo

Donde:
-g(n) indica la distancia del camino desde el nodo origen s al n.
-h(n) expresa la distancia estimada desde el nodo n hasta el nodo destino t.

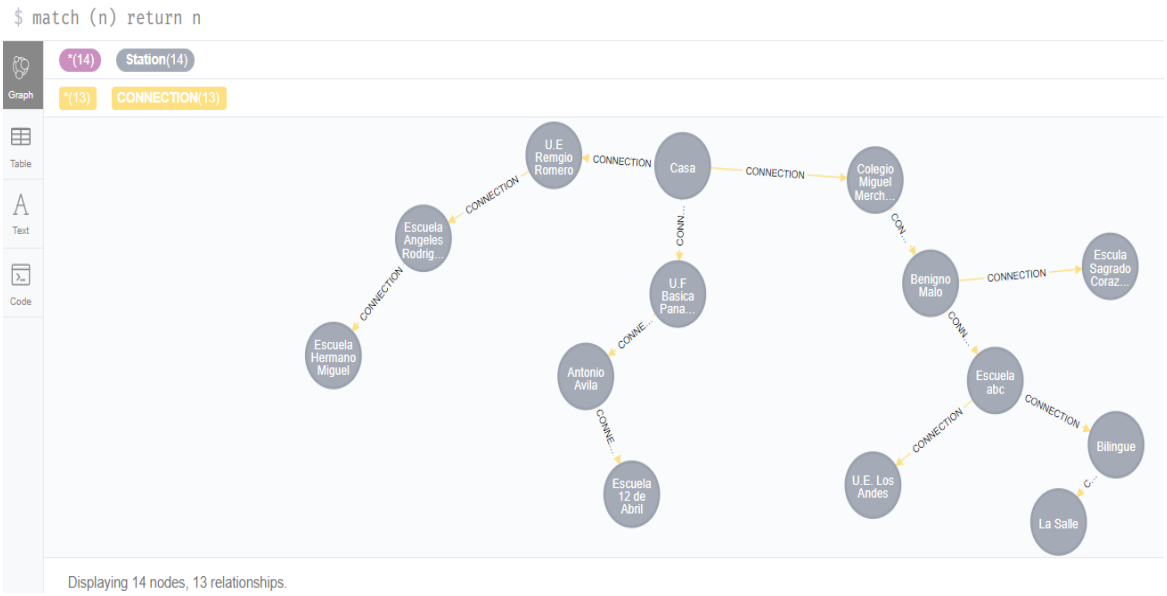
Codigo en NEO4J:

```
CREATE (a:Station {name: 'Casa',                                latitude:
51.5308, longitude: -0.1238}),
       (b:Station {name: 'U.E Remgio Romero',
latitude: 51.5282, longitude: -0.1337}),
       (c:Station {name: 'U.F Basica Panama',               latitude:
51.5392, longitude: -0.1426}),
       (d:Station {name: 'Colegio Miguel Merchan',       latitude:
51.5342, longitude: -0.1387}),
       (e:Station {name: 'Escuela Angeles Rodriguez', latitude: 51.5507,
longitude: -0.1402}),
       (f:Station {name: 'Antonio Avila', latitude: 51.5308, longitude: -
0.1238}),
       (g:Station {name: 'Benigno Malo',                    latitude:
51.5282, longitude: -0.1337}),
         (h:Station {name: 'Escula Sagrado Corazon',
latitude: 51.5282, longitude: -0.1337}),
       (i:Station {name: 'Escuela abc',               latitude: 51.5392,
longitude: -0.1426}),
       (j:Station {name: 'Escuela Hermano Miguel',       latitude:
51.5342, longitude: -0.1387}),
       (k:Station {name: 'Escuela 12 de Abril',             latitude:
51.5507, longitude: -0.1402}),
         (l:Station {name: 'U.E. Los Andes',              latitude:
51.5507, longitude: -0.1402}),
         (m:Station {name: 'Bilingue',              latitude: 51.5507,
longitude: -0.1402}),
         (n:Station {name: 'La Salle',              latitude: 51.5507,
longitude: -0.1402}),

         (a)-[:CONNECTION {time: 650}]->(b),
       (a)-[:CONNECTION {time: 750}]->(c),
       (a)-[:CONNECTION {time: 600}]->(d),

       (b)-[:CONNECTION {time: 2200}]->(e),
         (c)-[:CONNECTION {time: 1300}]->(f),
         (d)-[:CONNECTION {time: 1900}]->(g),

         (e)-[:CONNECTION {time: 3100}]->(j),
         (f)-[:CONNECTION {time: 1600}]->(k),

         (g)-[:CONNECTION {time: 1100}]->(i),
```

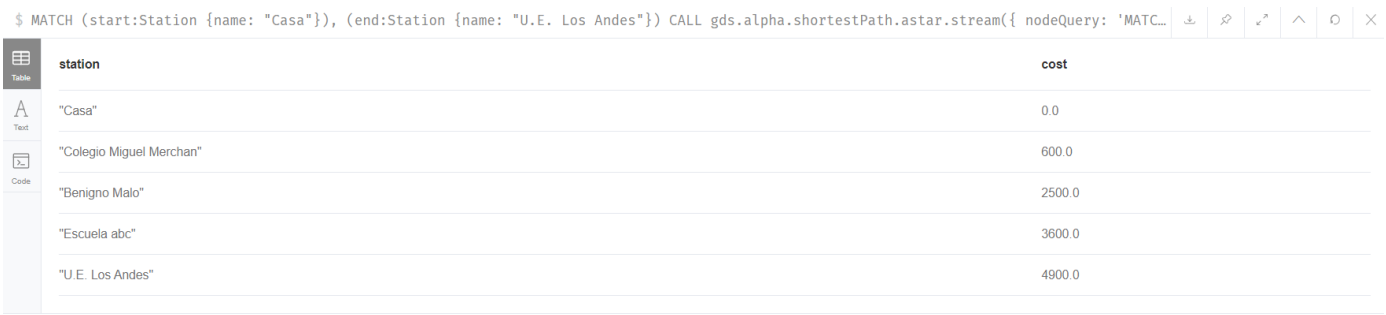
```
        (g)-[:CONNECTION {time: 2600}]->(h),

        (i)-[:CONNECTION {time: 1300}]->(l),
        (i)-[:CONNECTION {time: 2000}]->(m),

      (m)-[:CONNECTION {time: 1800}]->(n)
```

```
MATCH (start:Station {name: "Casa"}), (end:Station {name: "U.E. Los
Andes"})
CALL gds.alpha.shortestPath.astar.stream({
  nodeQuery: 'MATCH (p:Station) RETURN id(p) AS id',
  relationshipQuery: 'MATCH (p1:Station)-[r:CONNECTION]->(p2:Station)
RETURN id(p1) AS source, id(p2) AS target, r.time AS weight',
  startNode: start,
  endNode: end,
  relationshipWeightProperty: 'weight',
  propertyKeyLat: 'latitude',
  propertyKeyLat: 'longitude'
})
YIELD nodeId, cost
RETURN gds.util.asNode(nodeId).name AS station, cost
```

Grafo:



Costo:



$ MATCH (start:Station {name: "Casa"}), (end:Station {name: "U.E. Los Andes"}) CALL gds.alpha.shortestPath.astar.stream({ nodeQuery: 'MATC...

Table | Text | Code

station	cost
"Casa"	0.0
"Colegio Miguel Merchan"	600.0
"Benigno Malo"	2500.0
"Escuela abc"	3600.0
"U.E. Los Andes"	4900.0

Conclusión:

Mediante este algoritmo, se pude buscar las rutas de menor coste siempre y cuando se cumplan ciertas condiciones.