

# UNIVERSIDAD POLITÉCNICA SALESIANA

## EL VECINO - CUENCA

**Estudiante:** Gustavo Gualpa

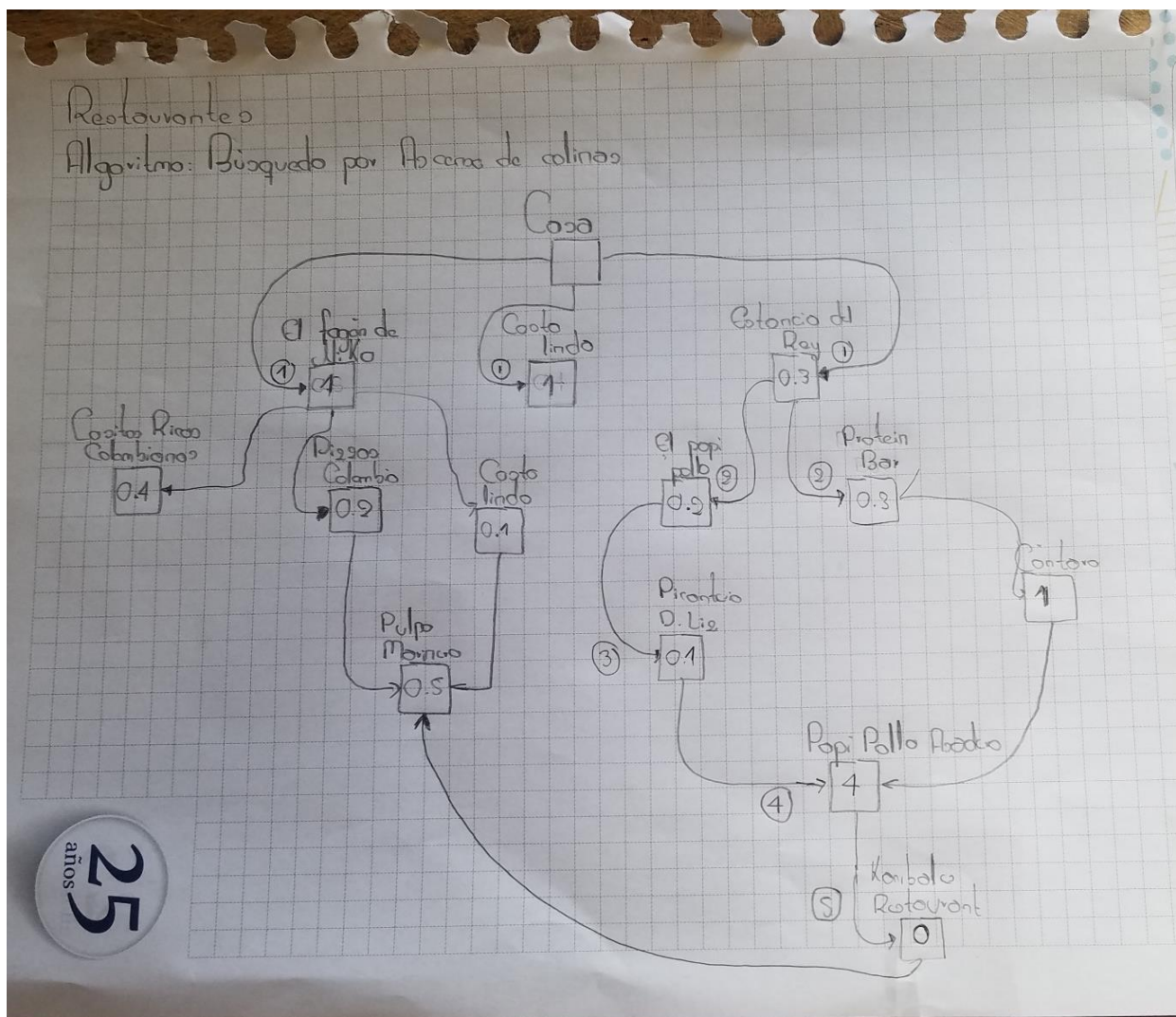
**Profesor:** Ing. Diego Quisi

**Asignatura:** Inteligencia Artificial

**Tema:** Algoritmo Ascenso de Colinas.

### SOLUCION A MANO

#### GRAFICO



## RESOLUCION A MANO

UPS

- Inicio = Casa
- Destino = Kanibales Restaurant.

① Empezamos con los nodos cercanos.  $\rightarrow$  Nivel 1  
Elegimos el menor.

lista Nodos = { casa, Estanco del Rey }

② Ahora desplegamos Estanco del Rey  $\rightarrow$  nivel 2  
Elegimos el menor.

lista Nodos = { casa, Estanco del Rey, El popi pollo }

③ Ahora desplegamos El popi pollo.

lista Nodos = { casa, Estanco del Rey, El popi pollo, Picanteiro D. Liz }

④ Ahora desplegamos Picanteiro D. Liz.

lista Nodos = { Casa, Estanco del Rey, El popi pollo, Picanteiro D. Liz, Popi pollo asadero }

⑤ Al desplegar Popi pollo asadero  $\rightarrow$  encontramos la solución

lista Nodos = { Casa, Estanco del Rey, El popi pollo, Picanteiro D. Liz, Popi pollo asadero }

Ruta = { Casa, Estanco del Rey, El popi pollo, Picanteiro D. Liz, Popi pollo asadero, Kanibales Restaurant }

# SOLUCION EN NEO4J

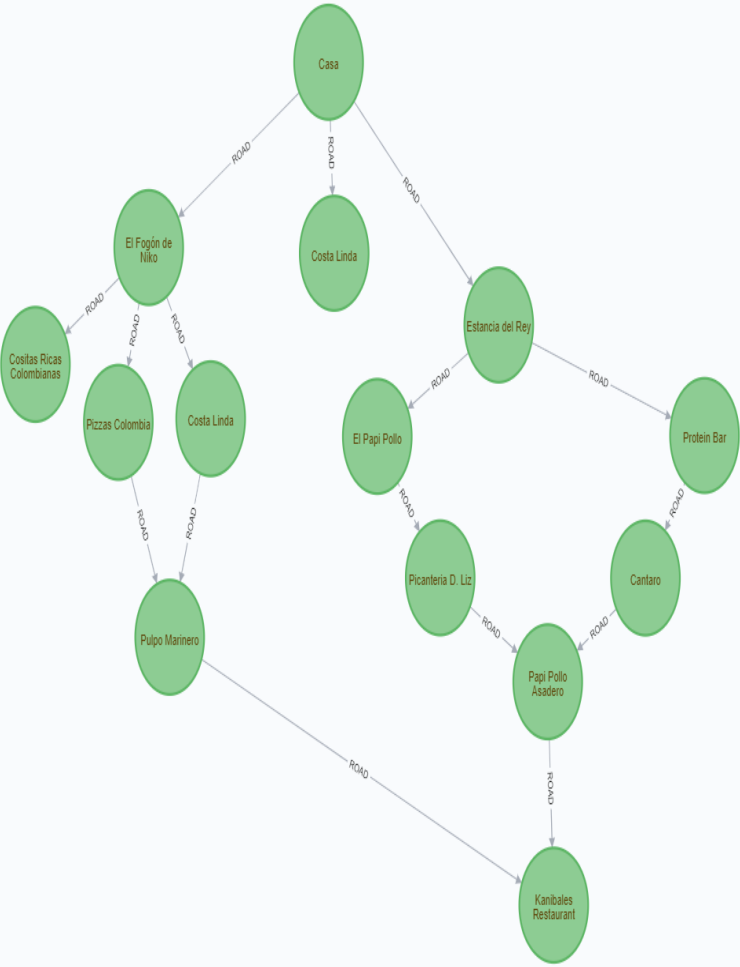
## GRAFICO

neo4j\$ match(n) return n

Download Refresh Zoom In Out Close

Graph  
\*(14) Loc(14)  
\*(16) ROAD(16)

Table  
Text  
Code



RESOLUCION CON EL ALGORITMO DE LA RUTA MAS CORTA

neo4j\$ MATCH (start:Loc {name: 'Casa'}), (end:Loc {name: 'Kanibales Restaurant'}) CALL gds.alpha.shortestPath.stream({ nodeProjection: 'Loc', relationshipProjec...

name	cost
"Casa"	0.0
"Estancia del Rey"	0.3
"El Papi Pollo"	0.5
"Picanteria D. Liz"	0.6
"Papi Pollo Asadero"	4.6
"Kanibales Restaurant"	4.6

Started streaming 6 records after 1 ms and completed after 2806 ms.

neo4j\$ MATCH (start:Loc {name: 'Casa'}), (end:Loc {name: 'Kanibales Restaurant'}) CALL gds.alpha.shortestPath.write({ nodeProjection: 'Loc', relationshipProject...

nodeCount	totalCost
6	4.6

Started streaming 1 records after 1 ms and completed after 29 ms.