

Maestría en Ciencias de Información Geoespacial

Sistemas de Información Geográfica



Proyecto Final

Repartición de garrafones de agua Bonafont
“Travelling Salesperson Problem with SQL”

Estudiantes:

Jorge Eduardo Cárdenas Arroyo
Leonardo Coronado Arvayo
Jeziret Sahadi González Gallardo

Profesora:

Karime González

Fecha de entrega: 6 de mayo 2020

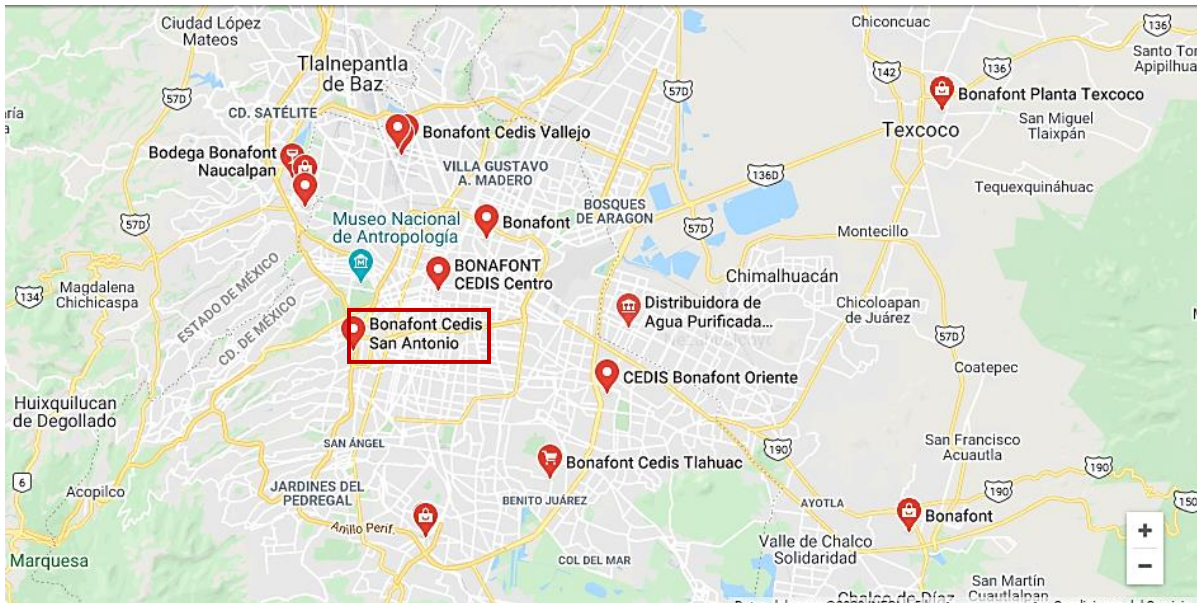
Repartición de garrafones de agua Bonafont mediante el “Travelling Salesperson Problem with SQL”

Problema

Bonafont es una empresa que vende agua embotellada, entre sus clientes están tiendas, oficinas y hogares y es la misma empresa quien se encarga de repartir los garrafones de agua. Los camiones salen de diversos centros de distribución (CEDIS) que tienen en la Ciudad de México; el problema es que la empresa organiza sus rutas basándose en lo que los choferes saben sobre sus clientes y, por tanto, las rutas no llevan una planeación cuidadosa lo que se significa un desperdicio de recursos traducidos en tiempo y dinero.

Para este proyecto se eligió únicamente el CEDIS San Antonio (ver Mapa 1) y a las alcaldías Álvaro Obregón y Benito Juárez como sitios de repartición. Por otra parte, se tomaron en cuenta únicamente tiendas de abarrotes como clientes para entrega de garrafones.

El problema del agente viajero nos permite que dada una lista de nodos que se desean visitar (tienditas), se defina la ruta más corta para visitar todos los nodos y regresar al punto o nodo de origen sin pasar dos veces por el mismo sitio. Dado que la base de datos nos muestra que los puntos de tienditas son demasiados, el ejercicio se hará tomando únicamente una cantidad limitada de puntos, pero el diseño de rutas y el resto de los pasos se realizaron en su totalidad.



Mapa 1. Ubicación de los CEDIS de Bonafont, en recuadro rojo CEDIS San Antonio

Los datos usados

- INEGI: shapes de CDMX (geometría de alcaldías)
- DENU: abarrotes de Álvaro Obregón y Benito Juárez
- Google Maps: ubicación del centro de distribución de Bonafont “San Antonio”
- OSM: red vial

Tutorial para SQL

1. Crear base de datos

En pgAdmin se creará una base de datos llamada “proyecto” a la cual se le añadirá la extensión de postgis y de pgrouting.

```
CREATE EXTENSION POSTGIS;
```

```
CREATE EXTENSION PGROUTING;
```

2. Importar capas

Se importarán todas las tablas/bases de datos con las que trabajaremos, es decir, alcaldías, red vial OSM, CEDIS San Antonio y se agregarán nuevas cuando se hayan creado las rutas.

3. Reproyectar las capas

Para evitar errores futuros, todas las capas deben estar en la misma proyección, en este caso todas estarán en 32614. Esto se hará con toda nueva capa que se agregue en caso de que no esté en la misma proyección.

//Consultar la proyección de las capas

```
select * from geometry_columns
```

//Reproyección de capas

```
ALTER TABLE "vialidades_BJAO"
```

```
ALTER COLUMN geom TYPE Geometry(MultiLinestring,32614)
```

```
USING ST_Transform(geom, 32614);
```

```
ALTER TABLE "alcadia_AO"
```

```
ALTER COLUMN geom TYPE Geometry(MultiPolygon,32614)
```

```
USING ST_Transform(geom, 32614);
```

```
ALTER TABLE "alcadia_BJ"
```

```
ALTER COLUMN geom TYPE Geometry(MultiPolygon,32614)
```

```
USING ST_Transform(geom, 32614);
```

4. Crear la topología para la capa de “vialidades_BJAO” se pondrá entre comillas por la diferencia entre mayúsculas y minúsculas.

```
ALTER TABLE "vialidades_BJAO" ADD COLUMN source INTEGER;  
ALTER TABLE "vialidades_BJAO" ADD COLUMN target INTEGER;  
SELECT pgr_createTopology('vialidades_BJAO',10, 'geom', 'id');
```

5. Crear una tabla con los datos filtrados del DENUE.

Los códigos relevantes son el 462112 y el 461110 correspondientes a tiendas de abarrotes. Se incluirán también datos que puedan ser útiles más adelante como la alcaldía donde se ubica, código postal, colonia y nombre. Esas columnas se renombrarán para que sean más fáciles de identificar.

```
CREATE TABLE denue as (  
  
select a.codigo_act as codigo, a.geom as geom, a.id as id, a.nom_estab as nombre,  
a.cve_mun as alcaldia, a.tipo_vial as vialidad, a.nomb_asent as colonia, a.cod_postal  
as cp  
  
FROM "denue_aobj" a  
  
WHERE a.codigo_act = '462112' or a.codigo_act = '461110');
```

//Revisar si se hizo correctamente

```
SELECT * from denue limit 5;
```

NOTA: este es un query que nos sirvió para seleccionar por string un nombre de alguna columna

```
select*FROM denue where SUBSTRING(nombre, 1 , 5)='EXTRA' or  
SUBSTRING(nombre, 1 , 4)= 'OXXO' or SUBSTRING(nombre, 1 , 8)= '7 ELEVEN';
```

6. Reproyectar la capa de Distribuidora y hacer buffer de 6.5Km que será la distancia máxima a la que repartirán camiones que salgan de CEDIS San Antonio. Este buffer abarca todo Benito Juárez y una gran parte de Álvaro Obregón.

```
ALTER TABLE "Distribuidora"  
  
ALTER COLUMN geom TYPE Geometry(Point,32614)  
  
USING ST_Transform(geom, 32614);  
  
SELECT name, st_buffer((geom), 6500.0) AS buffer_cedis  
  
FROM "Distribuidora"
```

```
CREATE TABLE buffer_cedis (
id serial primary key,
geom geometry (Polygon, 32614)
);
INSERT INTO buffer_cedis (geom)
SELECT ST_Buffer(geom, 6500.0)
FROM "Distribuidora";
```

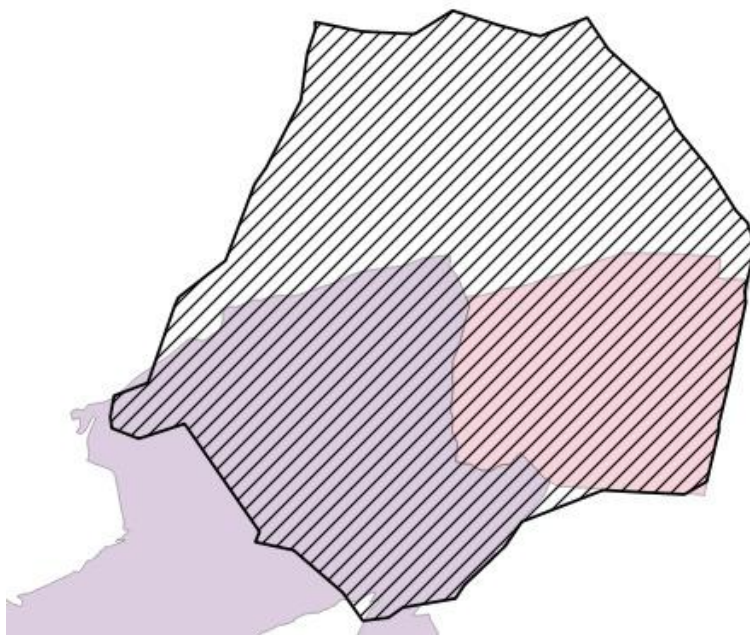
6.1 Alternativamente se puede hacer un área de servicio calculando un tiempo de 10 minutos para saber hasta dónde se harán entregas, además se toma en cuenta el sentido de las calles ("true"). *NOTA: al final del archivo se muestra el mapa con la ruta dentro del área de servicio.*

```
CREATE TABLE area_servicio as (
WITH DD AS (
SELECT * FROM pgr_drivingDistance(
'SELECT id, source, target, indice as cost FROM osm_recorte',
23152, 10, true))
SELECT id::integer, the_geom as geom, d.node as node
FROM osm_recorte_vertices_pgr w, DD d
WHERE w.id = d.node);
```



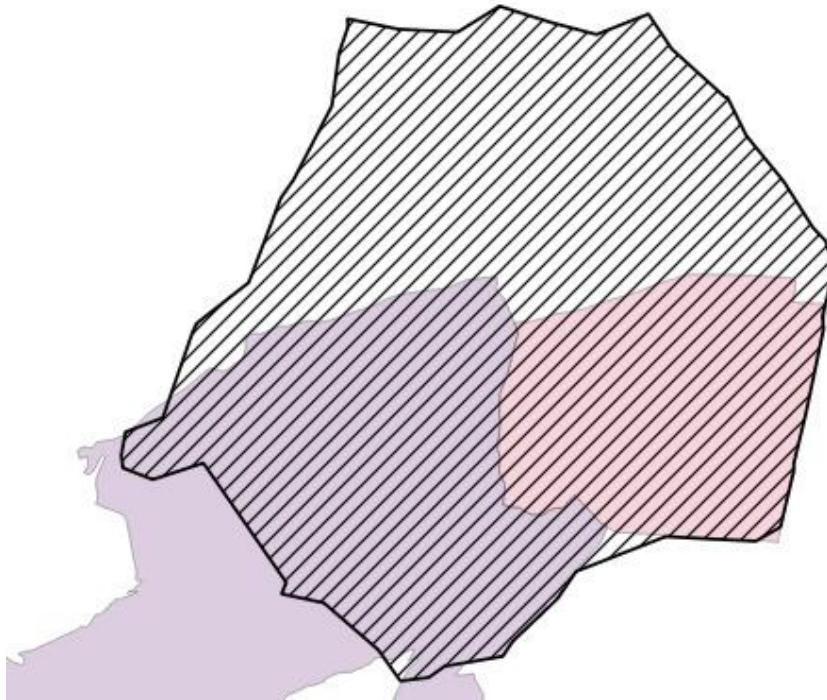
// Se hace la interpolación de puntos y se crea un polígono

```
CREATE TABLE area_ser as  
SELECT (st_concavehull(st_collect(geom),0.99)) as geom  
FROM area_servicio;
```



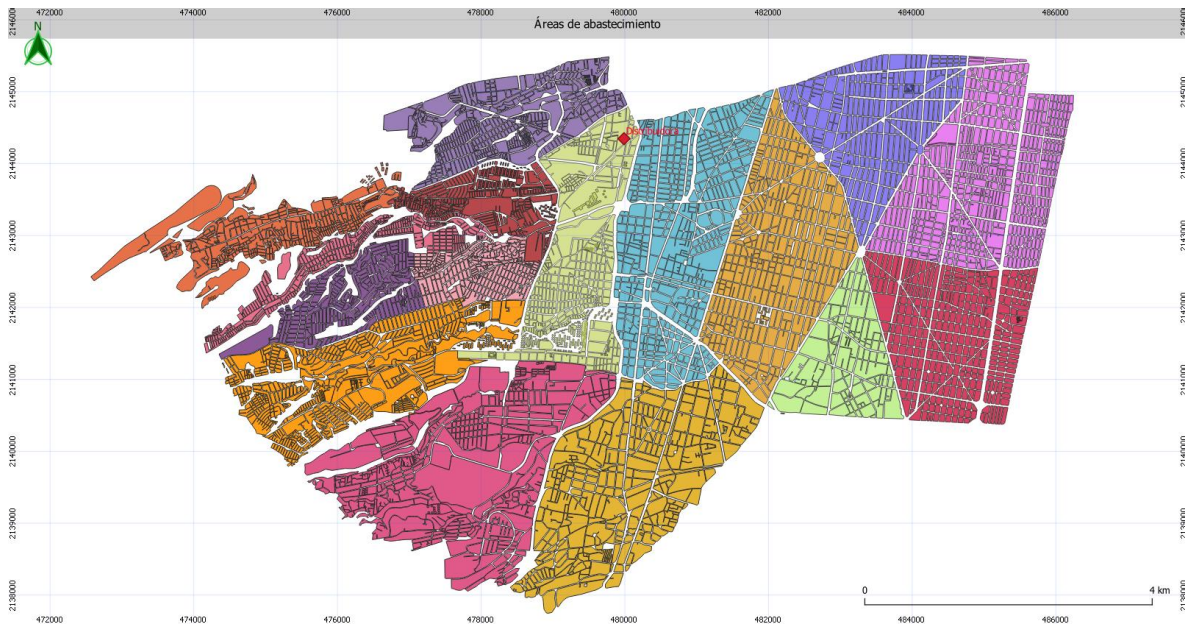
// Se divide en las dos alcaldías

```
CREATE TABLE alcaldias as  
select*from "alcadia_AO"  
union  
select*from "alcadia_BJ";  
  
CREATE TABLE a_s as  
SELECT layer2.id,  
       st_union(ST_INTERSECTION(layer1.geom, layer2.geom))  
FROM area_ser layer1, alcaldias layer2  
WHERE ST_Intersects(layer1.geom, layer2.geom)  
GROUP BY layer2.id;
```

7. Crear polígonos de Benito Juárez y Álvaro Obregón. Esta parte se hizo de forma manual directamente en QGIS usando como criterio las avenidas principales.

Salieron 6 rutas para Benito Juárez y 10 para Álvaro Obregón



8. Agregar la topología al “denue” y a “Distribuidora” por closest node.

8.1 Se Crear topología para “denue”

```
ALTER TABLE denue add column closest_node bigint;  
UPDATE denue set closest_node = c.closest_node from  
(select b.id as id_denue, (  
SELECT a.id  
FROM "vialidades_BJAO_vertices_pgr" As a  
ORDER BY b.geom <-> a.the_geom LIMIT 1)as closest_node  
FROM denue b) as c  
WHERE c.id_denue = denue.id;
```

//Se verifica que haya quedado bien

```
select * from denue limit 3;
```

8.2 Se hace la topología para “Distribuidora”

```
ALTER TABLE "Distribuidora" add column closest_node bigint;  
UPDATE "Distribuidora" set closest_node = c.closest_node from  
(select b.id as id_Distribuidora, (  
SELECT a.id  
FROM "vialidades_BJAO_vertices_pgr" As a  
ORDER BY b.geom <-> a.the_geom LIMIT 1)as closest_node  
FROM "Distribuidora" b) as c  
WHERE c.id_Distribuidora = "Distribuidora".id;
```

//Se verifica que haya quedado bien

```
select * from "Distribuidora";
```


9. Aplicar agente viajero

Una vez con los datos preparados, se aplicará el agente viajero. Para este ejercicio se tomaron únicamente 5 puntos de una de la ruta 9 de Álvaro Obregón, ya que los puntos a visitar, así como el de salida y llegada deben ponerse de forma manual y hasta ahora no sabemos la forma de ingresar tantos puntos de manera automatizada.

//Seleccionar el nodo de inicio y el de término:

En este caso es el **23152** correspondiente a la “Distribuidora” que es de donde salen los camiones y a donde regresan completada su ruta de entrega.

```
SELECT seq, node, the_geom FROM pgr_TSP($$  
  
SELECT * FROM pgr_dijkstraCostMatrix('SELECT id, source, target, indice as cost, -  
indice as reverse_cost FROM osm_recorte',  
  
(SELECT ARRAY[23152, 21464, 20581, 20596, 19613] node_array), directed := false)  
$$, start_id := 23152, randomize := false) as res  
  
JOIN osm_recorte_vertices_pgr ON osm_recorte_vertices_pgr.id = res.node;
```

//Crear la tabla con la secuencia de nodos a visitar y unir las rutas

```
CREATE TABLE ruta_reparte as  
  
(WITH path_1 AS (SELECT b.geom, a.* from (select node, edge as id, cost FROM  
pgr_bdDijkstra('SELECT id::int4, source::int4 ,target::int4 as target, indice::float8  
AS cost FROM osm_recorte', 23152, 21464 ,FALSE))  
AS a join osm_recorte b on a.id = b.id), path_2 AS (  
  
SELECT b.geom, a.* from (select node, edge as id, cost  
  
from pgr_bdDijkstra('SELECT id::int4, source::int4 ,target::int4 as target,  
indice::float8 AS cost FROM osm_recorte', 21464,20581,FALSE))  
AS a join osm_recorte b on a.id = b.id), path_3 AS (  
  
SELECT b.geom, a.* from (select node, edge as id, cost  
  
from pgr_bdDijkstra('SELECT id::int4, source::int4 ,target::int4 as target,  
indice::float8 AS cost FROM osm_recorte', 20581, 20596,FALSE))  
AS a join osm_recorte b on a.id = b.id),path_4 AS (
```

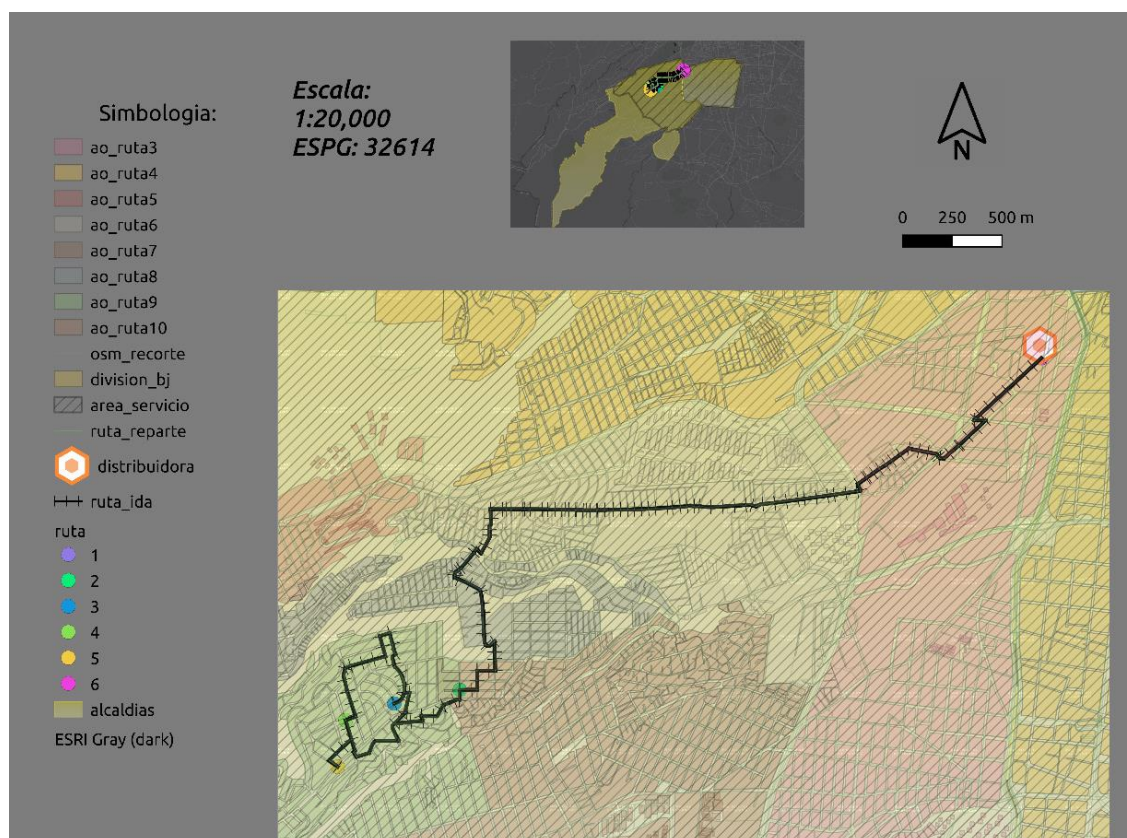
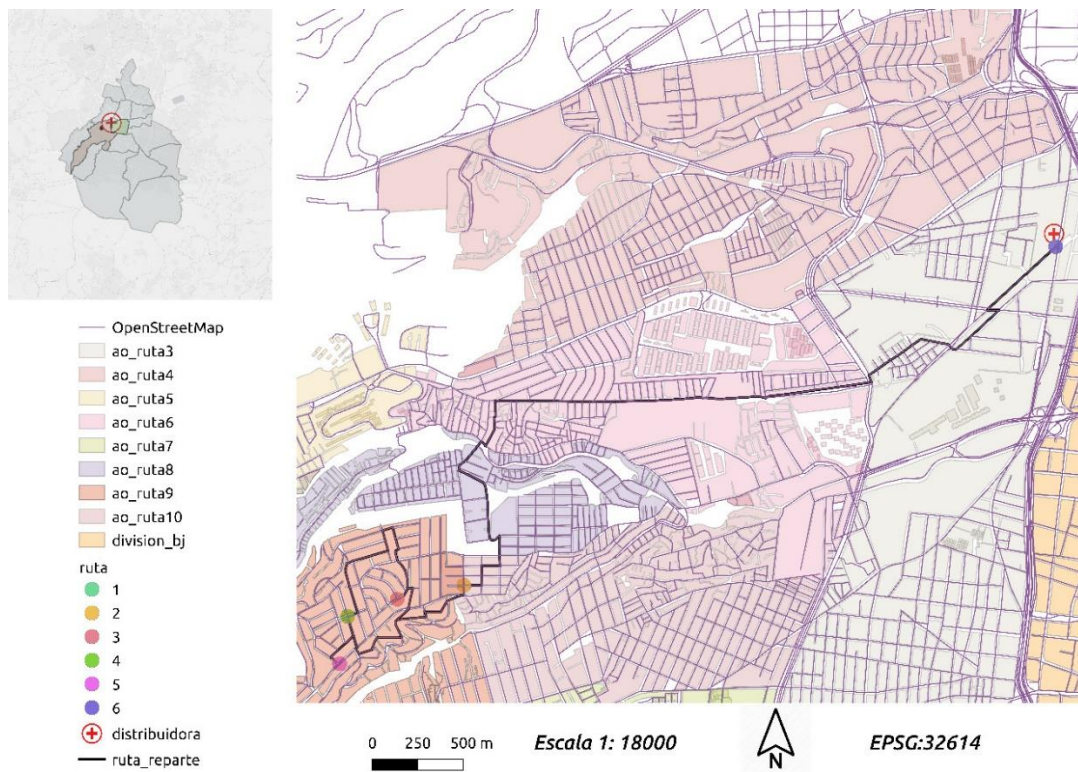
```

SELECT b.geom, a.* from (select node, edge as id, cost
from pgr_bdDijkstra('SELECT id::int4, source::int4 ,target::int4 as target,
indice::float8 AS cost FROM osm_recorte', 20596,19613, FALSE))
AS a join osm_recorte b on a.id = b.id),path_5 AS (

SELECT b.geom, a.* from (select node, edge as id, cost
from pgr_bdDijkstra('SELECT id::int4, source::int4 ,target::int4 as target,
indice::float8 AS cost FROM osm_recorte', 19613,23152, FALSE)) as a join
osm_recorte b on a.id = b.id)

SELECT * FROM path_1
UNION
SELECT * FROM path_2
UNION
SELECT * FROM path_3
UNION
SELECT * FROM path_4
UNION
SELECT * FROM path_5 )

```



Fuentes consultadas

- Agente viajero. https://github.com/eurekastein/agente/blob/master/agente_viajero.sql
- Creating map of travelling salesperson problem solution with pgrouting?
<https://gis.stackexchange.com/questions/323747/creating-map-of-travelling-salesperson-problem-solution-with-pgrouting>