

Mapas de flujos con Qgis

Movement data in GIS.
by Anita Graser



Maestría en Ciencias de la Información Geoespacial

Geoinformática

Mtro. Alberto Porras Velázquez

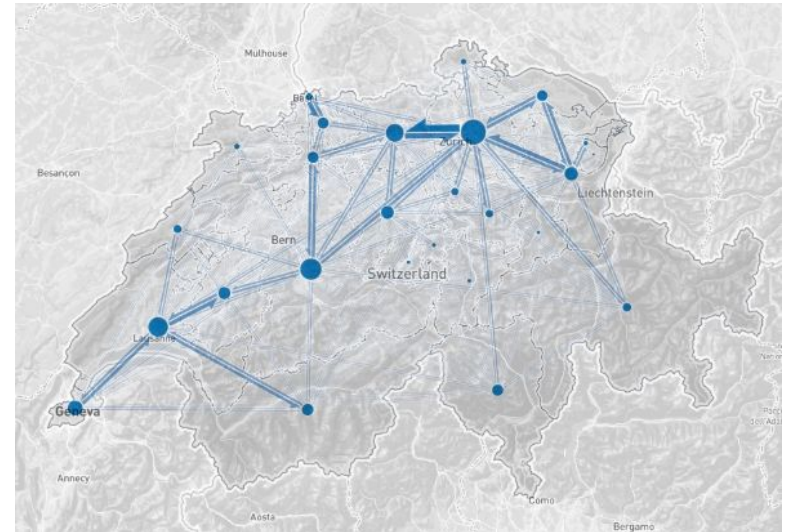
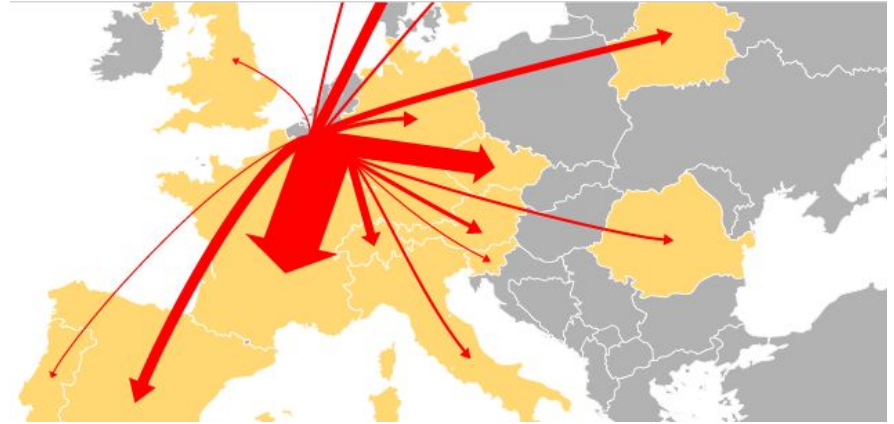
Cerón Alonso Jocelin

Cortés Solís Alonso

Los mapas de flujo muestran geográficamente el movimiento de información u objetos de un lugar a otro y su cantidad

Este tipo de mapas pueden representar características de tipo cualitativo y cuantitativo; mediante líneas podemos representar:

- a.- Qué tipo de movimiento es el que se da.
- b.- Qué cantidad de movimiento se está dando.

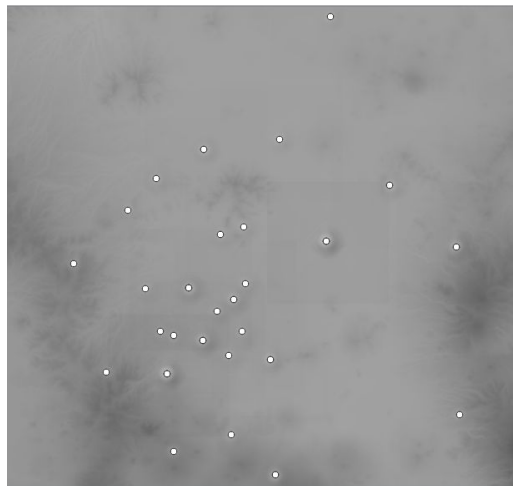


¿Qué necesitamos?

Para realizar este tipo de mapas necesitamos dos insumos:

1. Una capa geoespacial de puntos, con un ID
2. Una tabla que registre los flujos entre los puntos a partir de su ID, esto con el punto de origen y el punto de destino.

Adicionalmente nuestra tabla podría tener más columnas, en este ejemplo se tiene una columna que indica el número de viajes entre puntos.



	orig_id	dest_id	viajes
1	067	300	847
2	047	010	226
3	077	006	2500
4	063	045	1371
5	173	030	4967
6	102	016	4284
7	063	015	1470
8	063	068	1027
9	111	063	2098
10	054	060	3807

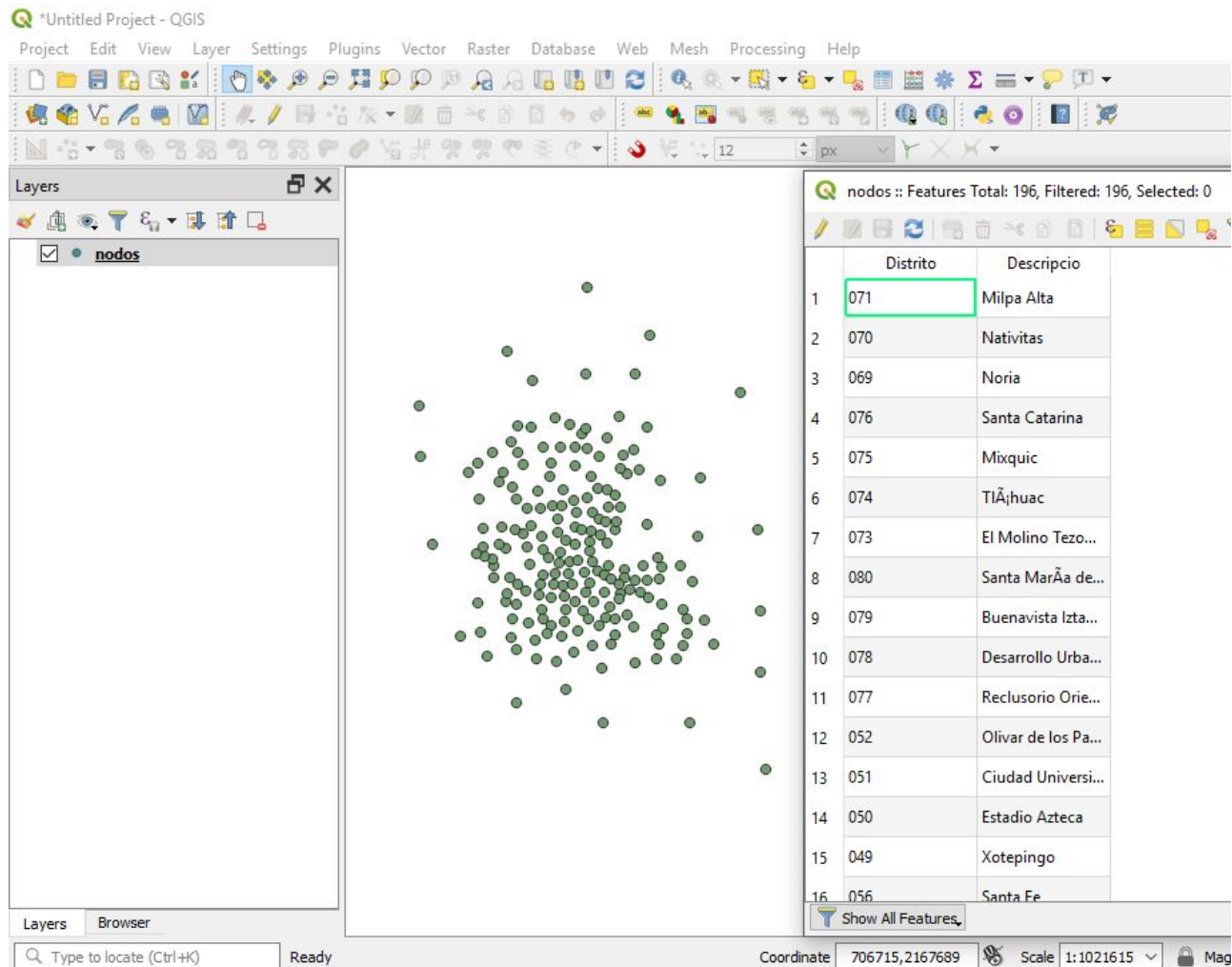
Para esta práctica se proporcionan dos archivos:

1. El archivo de “nodos”, construido a partir de los centroides de los distritos de la encuesta de origen y destino 2017 de la Zona Metropolitana del valle de México.
2. El archivo “viajes”, un archivo csv que contiene la relación entre los nodos en el que se inicia y terminan una serie de viajes entre los distritos de la encuesta origen destino.

<http://giitral.iingen.unam.mx/Estudios/EstudioOD-ZMVM-2017.html>

Como primer paso necesitamos cargar nuestra capa de puntos (nodos) en Qgis.

Cómo podemos apreciar, existen 196 puntos, y en su tabla de atributos tenemos la columna de “Distrito” que nos indica el identificador del punto; y “Descripcio” que nos indica el nombre del Distrito.



*Untitled Project - QGIS

Project Edit View Layer Settings Plugins Vector Raster Database Web Mesh Processing Help

Layers

Layers: ☒ nodos

nodos :: Features Total: 196, Filtered: 196, Selected: 0

	Distrito	Descripcio
1	071	Milpa Alta
2	070	Nativitas
3	069	Noria
4	076	Santa Catarina
5	075	Mixquic
6	074	TIÃ¿huac
7	073	El Molino Tezo...
8	080	Santa MarÃ¿a de...
9	079	Buenavista Izta...
10	078	Desarrollo Urba...
11	077	Reclusorio Orie...
12	052	Olivar de los Pa...
13	051	Ciudad Universi...
14	050	Estadio Azteca
15	049	Xotepingo
16	056	Santa Fe

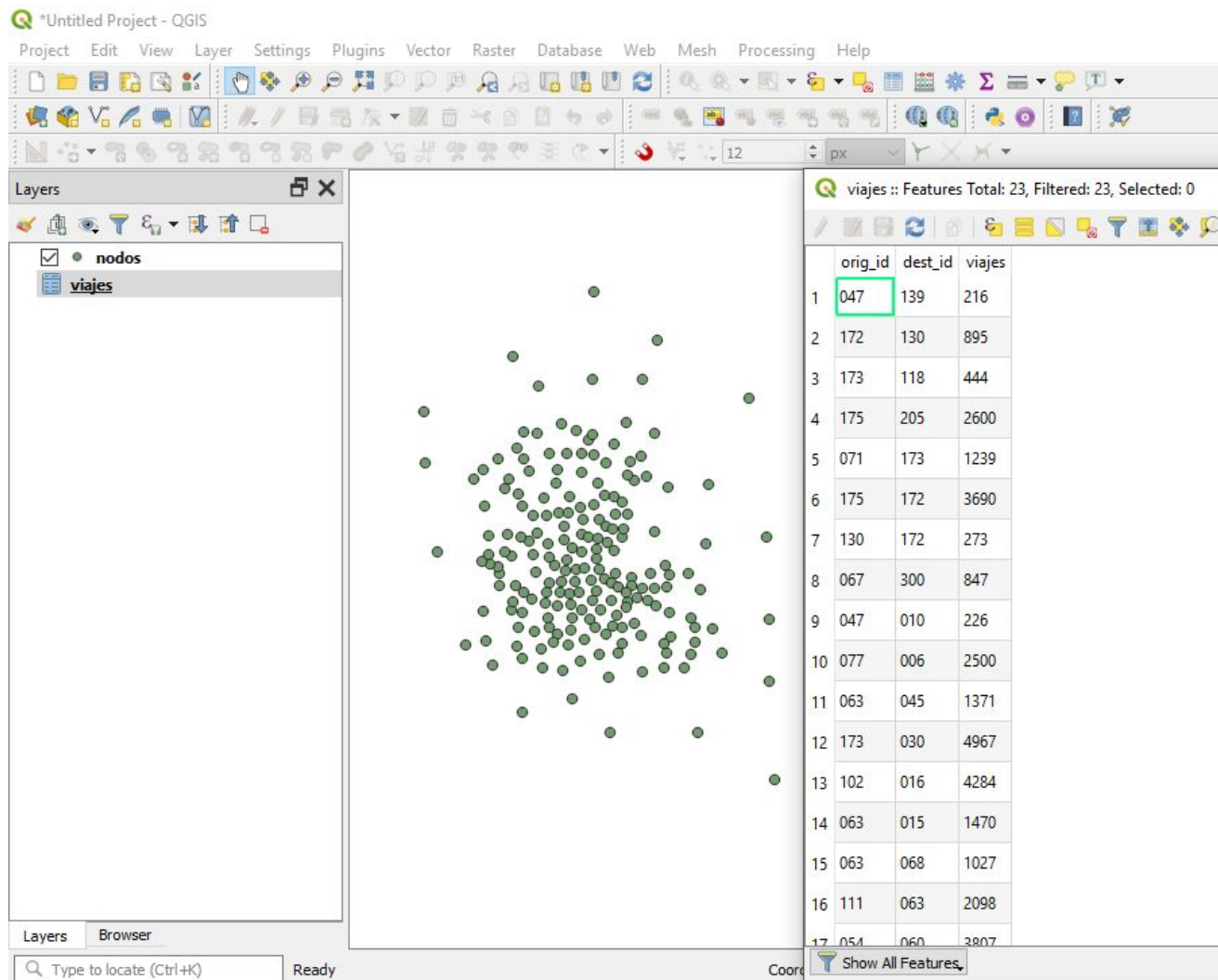
Layers Browser

Type to locate (Ctrl+K) Ready

Coordinate 706715,2167689 Scale 1:1021615

Ahora cargamos el archivo csv
“viajes”. Cada una de las filas de
esta tabla corresponde a la serie
de viajes que se dan entre dos
nodos en particular.

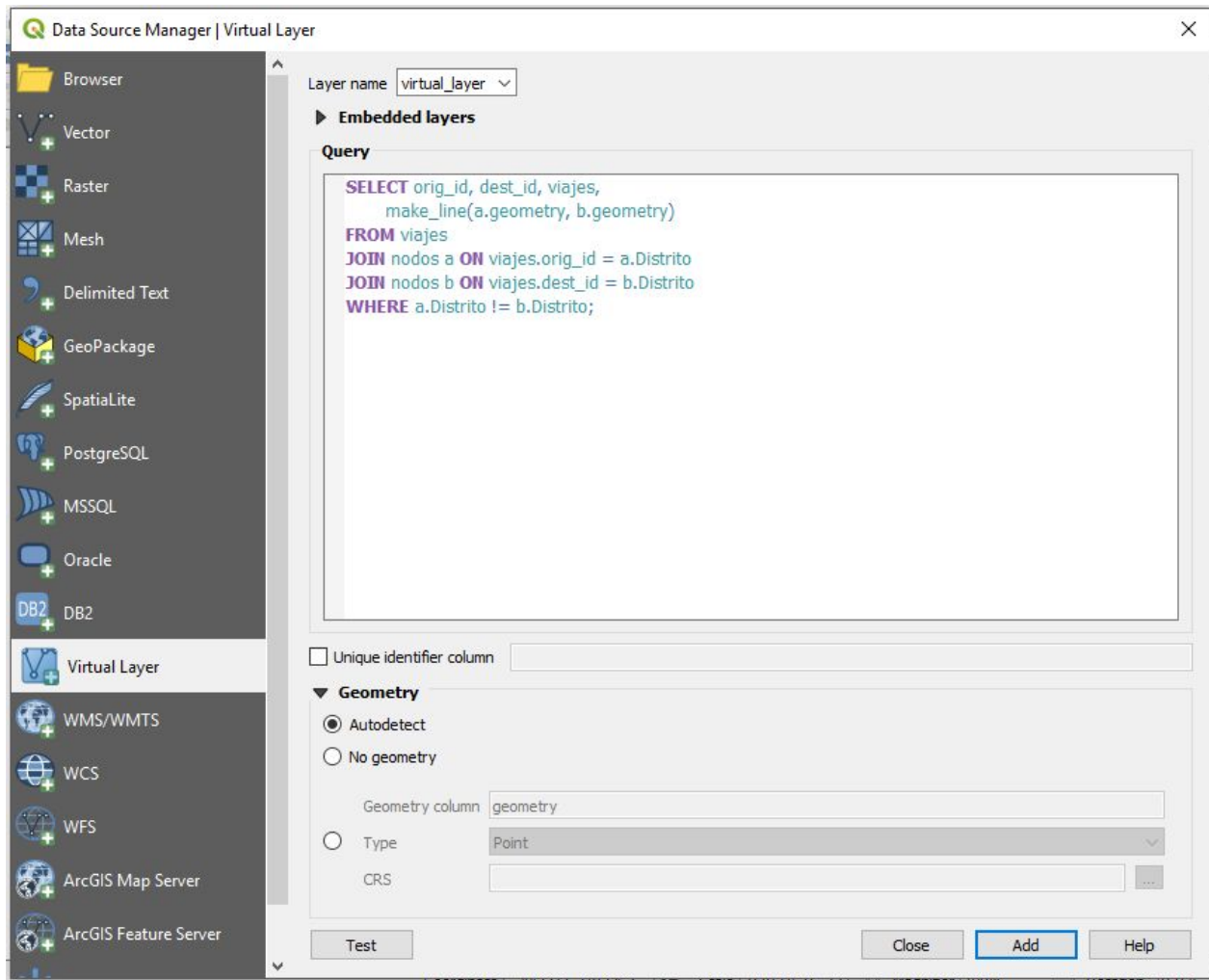
Este archivo se compone de 23
entidades, en las que se detalla el
nodo de origen “orig_id”, el nodo
destino “dest_id” y el número de
viajes que se realiza en ese
movimiento “viajes”



Una vez cargadas nuestros datos, es hora de realizar la capa de líneas que representara el movimiento entre los nodos: para esto haremos uso de los “virtual layers” de Qgis, el cual es accesible desde el Data Source Manager

En el que haremos uso de Structured Query Language SQL, para escribir un código que nos permita crear las líneas a partir de nuestros archivos “viajes” y “nodos”

Para esto es necesario que peguemos el siguiente código en la ventana de “Query” del virtual layer.



A grandes rasgos lo que realiza el siguiente código es un “join” entre nuestros dos archivos.

En la primera parte del código se seleccionan las columnas que sean parte de nuestro nuevo archivo (orig_id, dest_id, viajes) esto del archivo de viajes, y se construirá una línea con la geometría de “a” y “b” (make_line)

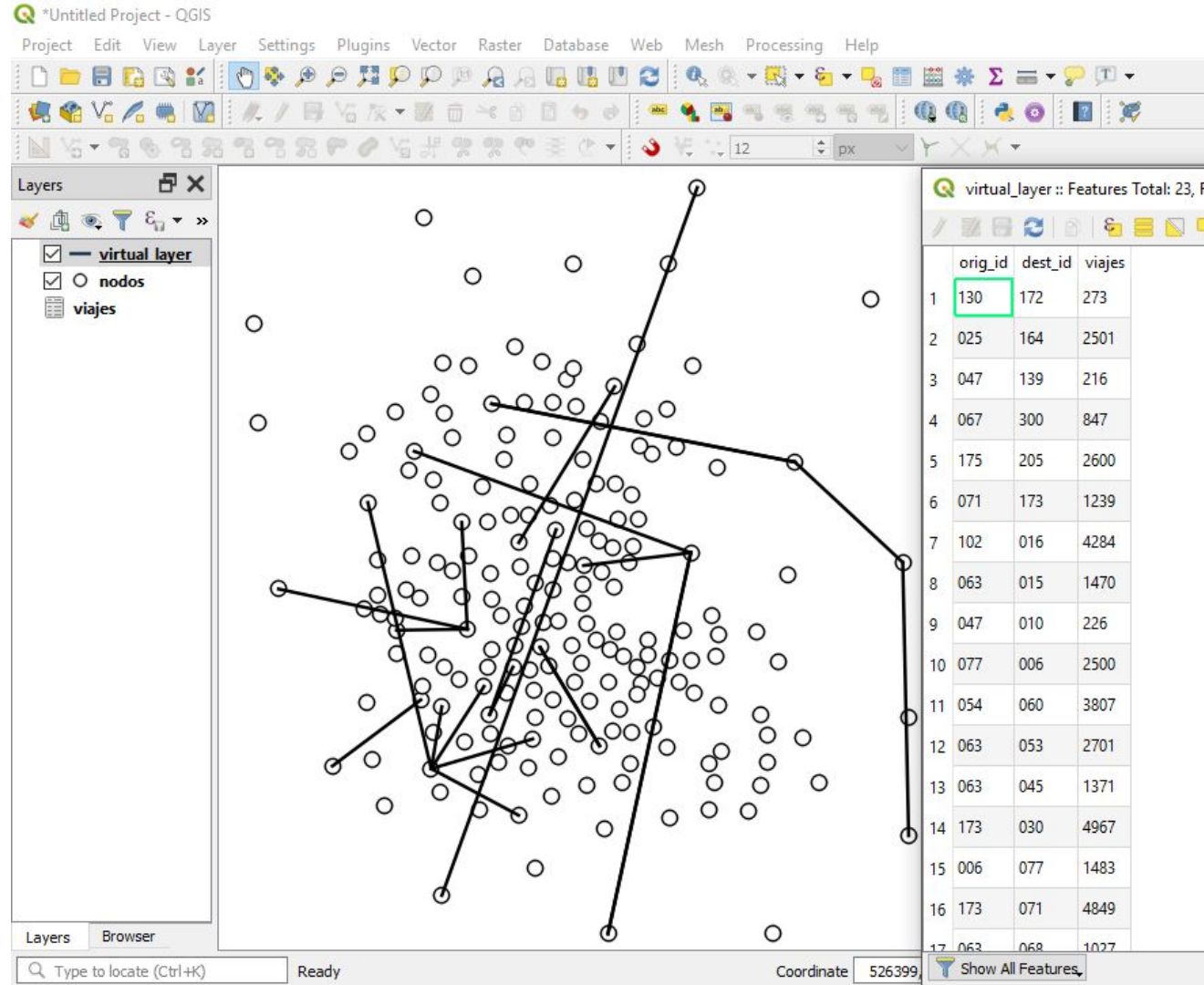
La siguiente parte hace el join entre los archivos y sirve para la construcción de la línea, esto se compone dos partes:

1. Une el archivo de “nodos” (a) con el archivo de “viajes” esto en donde el valor de la columna “orig_id”(viajes) sea igual al valor de la columna de “Distritos” (nodos) esto para crear el punto de inicio de las líneas.
2. Une el archivo de “nodos” (a) con el archivo de “viajes” esto en donde el valor de la columna “dest_id”(viajes) sea igual al valor de la columna de “Distritos” (nodos) esto para crear el punto final de las líneas.

Todo esto siempre y cuando el punto de origen sea distinto al punto de destino, esto para evitar la creación de líneas que lleguen d donde parten.

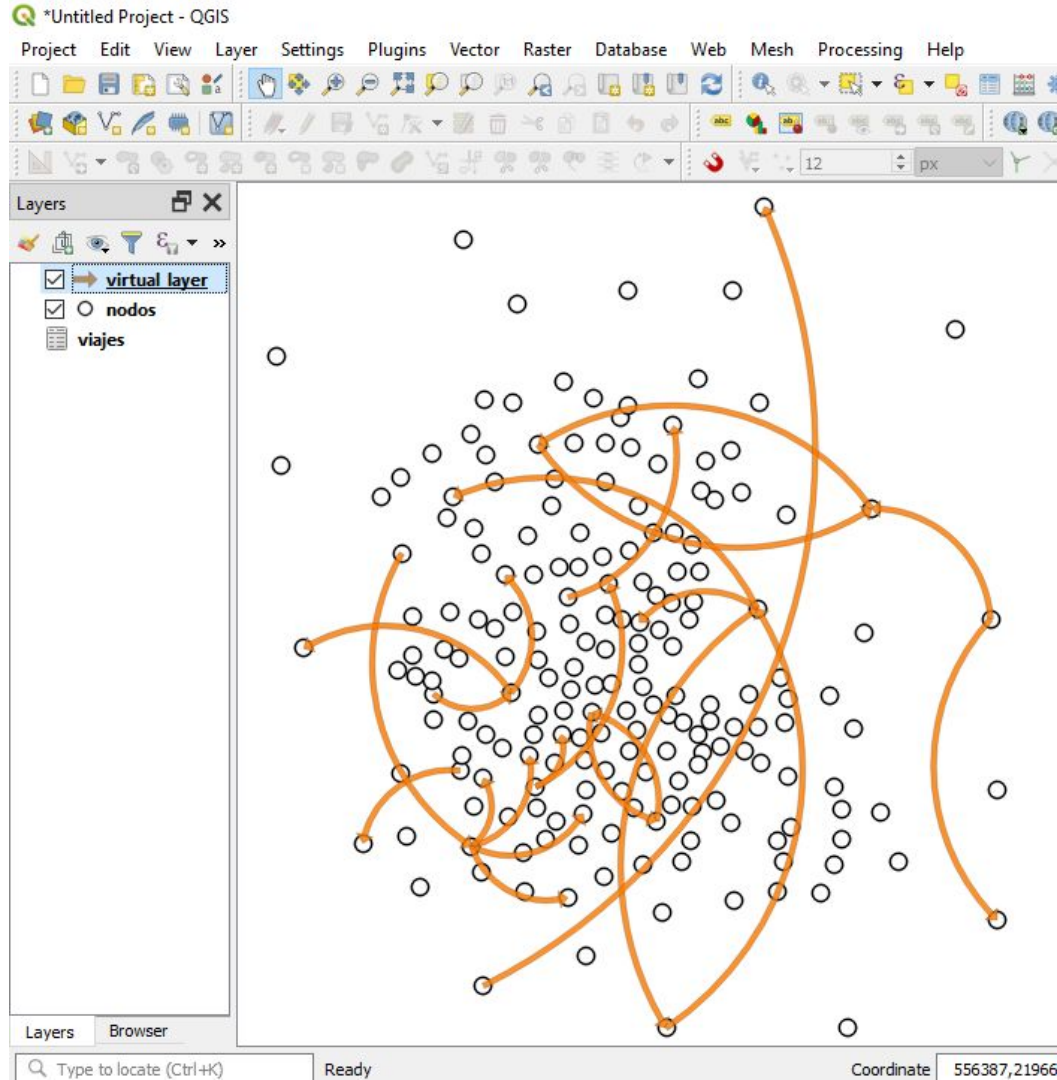
```
-- Mapa de Fijos con QGIS --  
  
--Creación de líneas apartir de csv  
SELECT orig_id, dest_id, viajes,  
       make_line(a.geometry, b.geometry)  
FROM viajes  
JOIN nodos a ON viajes.orig_id = a.Distrito  
JOIN nodos b ON viajes.dest_id = b.Distrito  
WHERE a.Distrito != b.Distrito;
```


Al aceptar la creación de este “virtual layer” obtendremos las líneas que unen los nodos de destino con los de viaje, y en su tabla podremos apreciar la misma información que en el archivo de “viajes”.

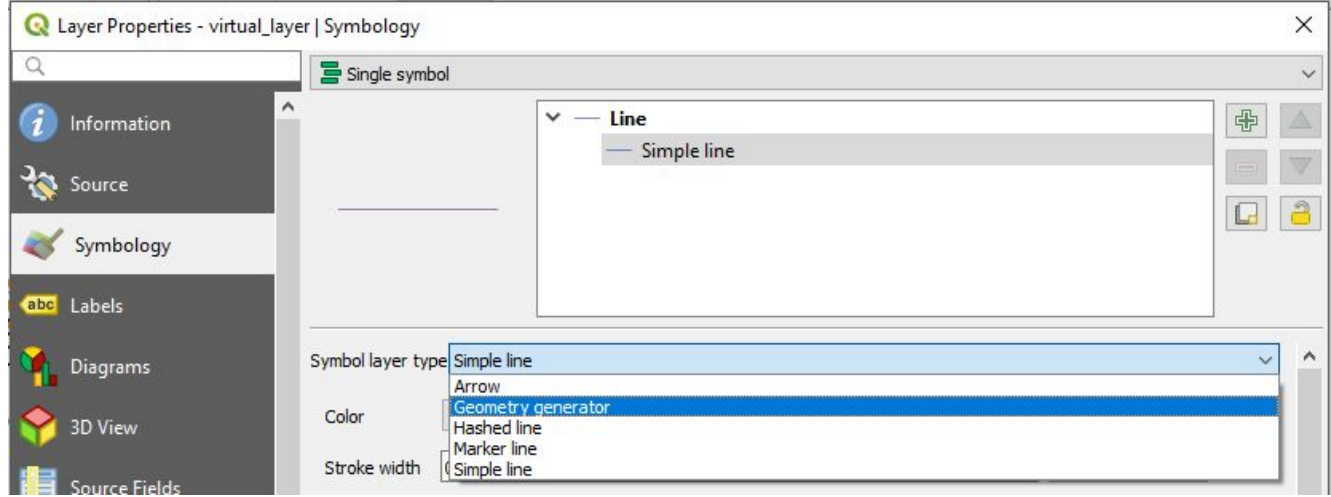


Por ejemplo, con el siguiente código podemos obtener un tercer punto de nuestras líneas que nos permite crear líneas curvas

```
make_line(  
  start_point($geometry),  
  centroid(  
    offset_curve(  
      $geometry,  
      length($geometry)/-10  
    ),  
  ),  
  end_point($geometry)  
)
```

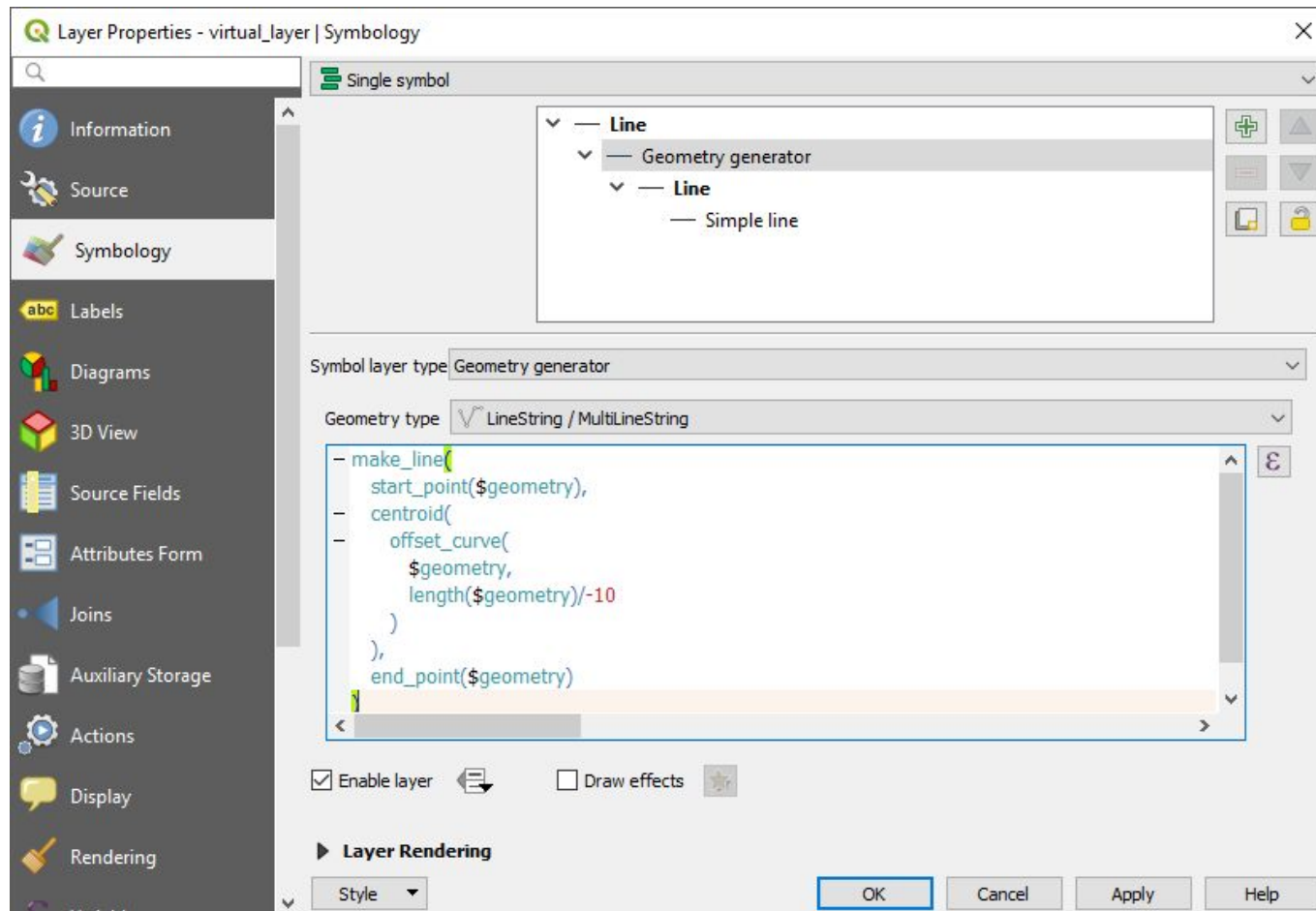


Para esto debemos acceder a la ventana de Symbology dentro de las propiedades de nuestra capa, una vez ahí en las opciones “Symbol layer Type” debemos escoger “Geometry generator”

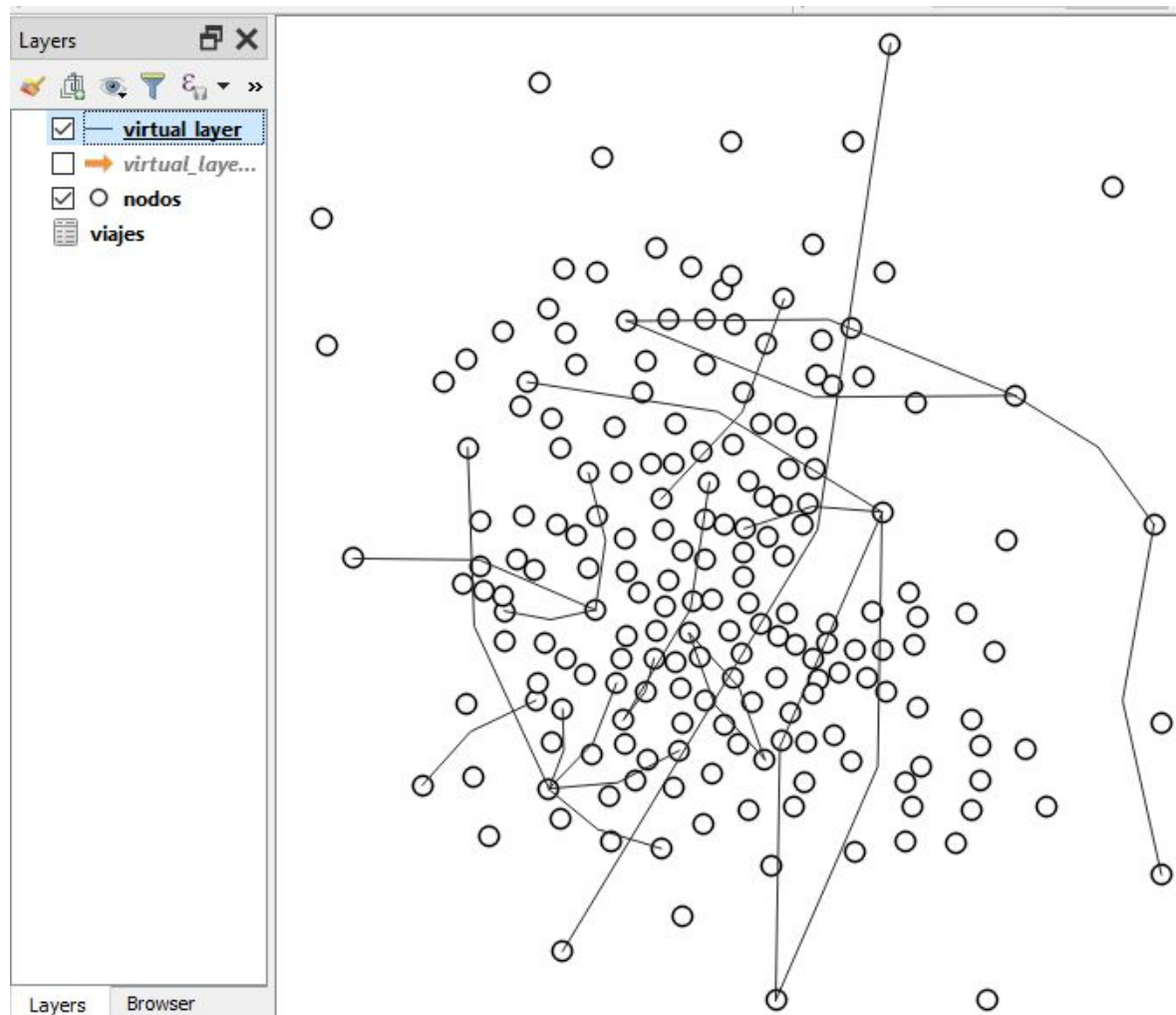
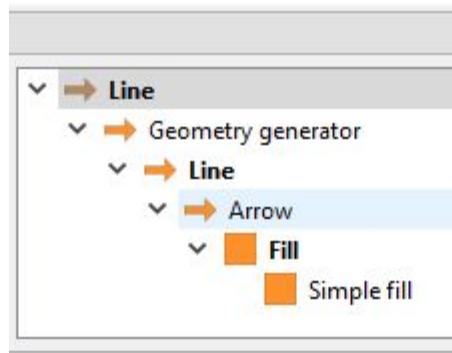


Una vez aquí solo es cuestión de definir el “Geometry Type” a “LineString/MultiLineString”, borrar los datos de la ventana y pegar el código correspondiente.

```
make_line(  
  start_point($geometry),  
  centroid( offset_curve(  
    $geometry, length($geometry)/-10  
  )  
),  
  end_point($geometry))
```

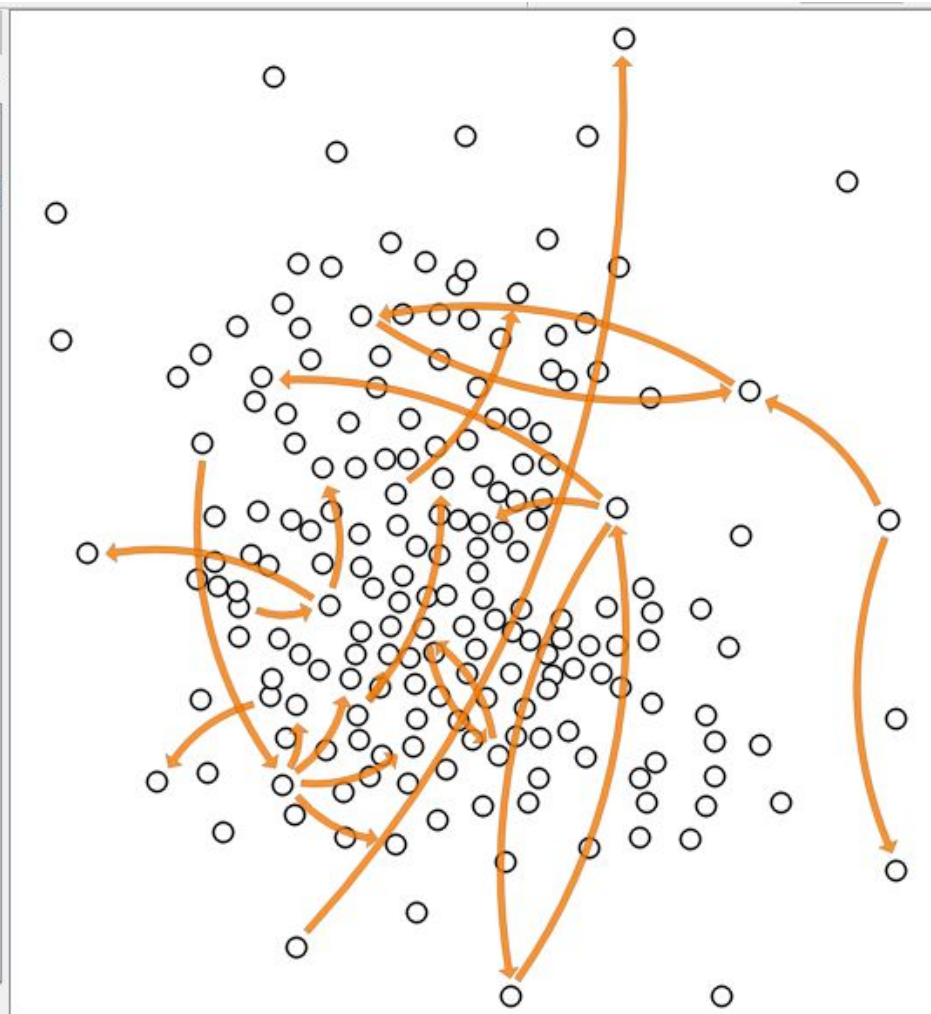
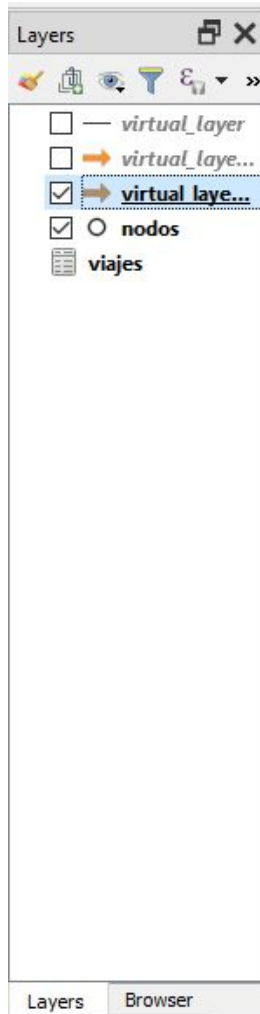


Con el código anterior, como podemos apreciar se obtiene un tercer punto con el cual, haciendo uso de las herramientas de simbología, se podrían obtener líneas curvas que mejoran la calidad visual de nuestro mapa.



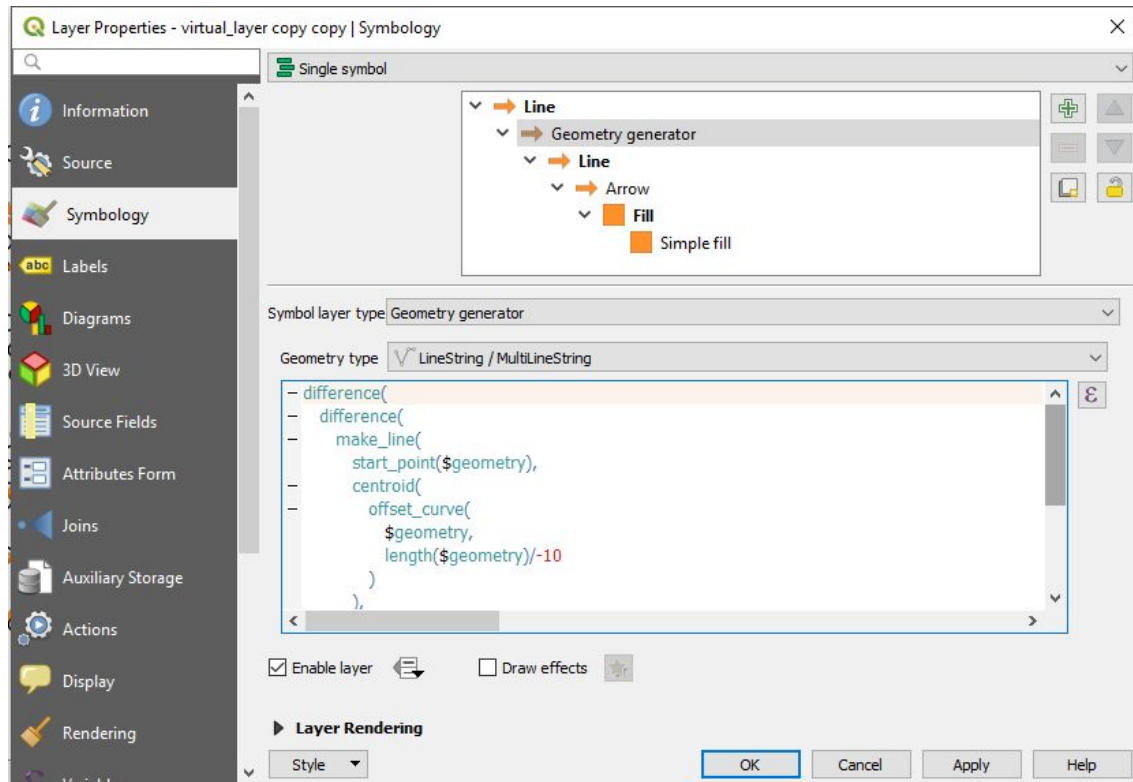
Otro ejemplo de las aplicaciones que podemos tener con SQL, es hacer que nuestras líneas se corten a cierta distancia del punto de origen y del punto de destino

```
difference(  
  difference(  
    make_line(  
      start_point($geometry),  
      centroid(  
        offset_curve(  
          $geometry,  
            length($geometry)/-10  
        )  
      ),  
      end_point($geometry)  
    ),  
    buffer(start_point($geometry),1500)  
  ),  
  buffer(end_point($geometry),1500)  
)
```



Para esto solo tendríamos que corregir el código anterior por el siguiente y definir una distancia a la queremos que se corten nuestras líneas.

```
difference(  
  difference(  
    make_line(  
      start_point($geometry),  
      centroid(offset_curve(  
        $geometry, length($geometry)/10 ) ),  
      end_point($geometry)),  
    buffer(start_point($geometry),1500)),  
    buffer(end_point( $geometry), 1500))
```

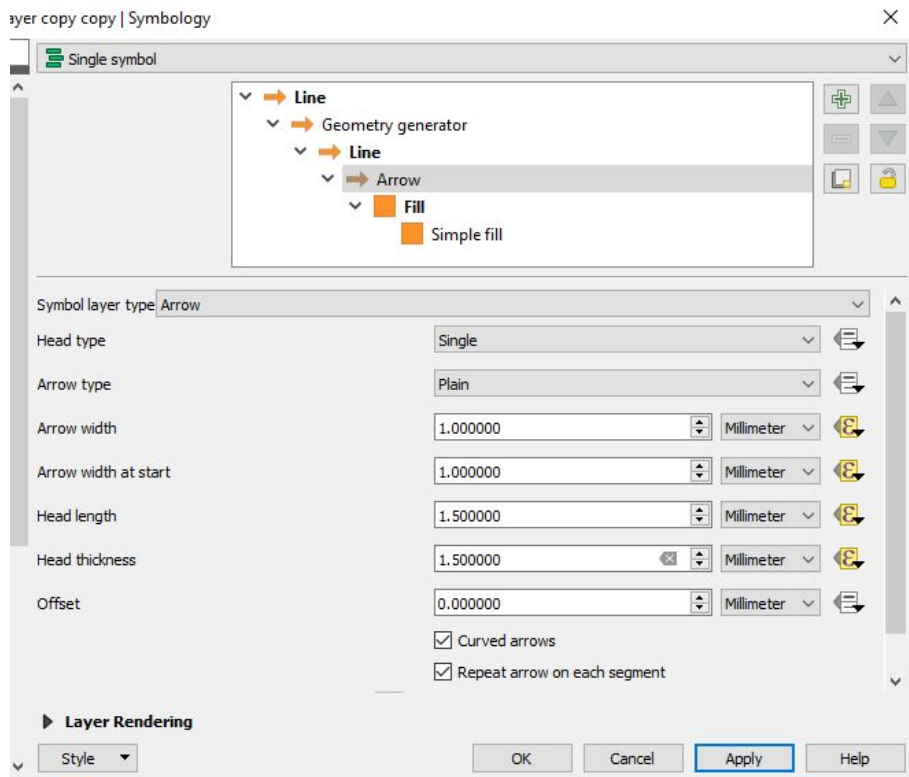
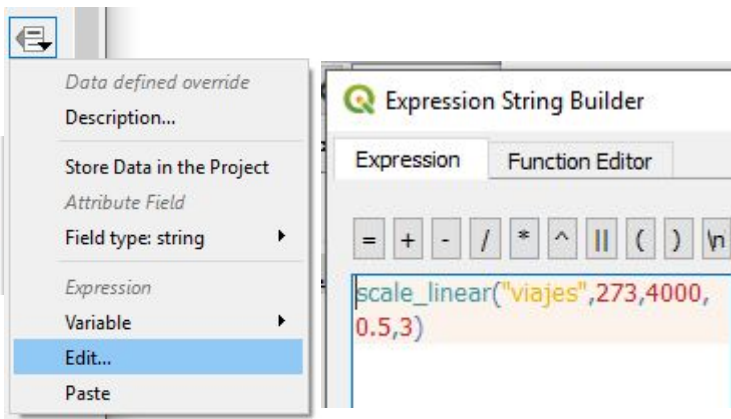


Nuestro mapa ya casi se encuentra terminado, solo restaría darle una ponderación a las flechas para así poder diferenciar cuales tienen un mayor número de viajes, para esto usaremos la columna de “viajes”.

Para esto a las opciones de arrow width, arrow width at start, arrow length y head thickness les aplicaremos la siguiente sentencia.

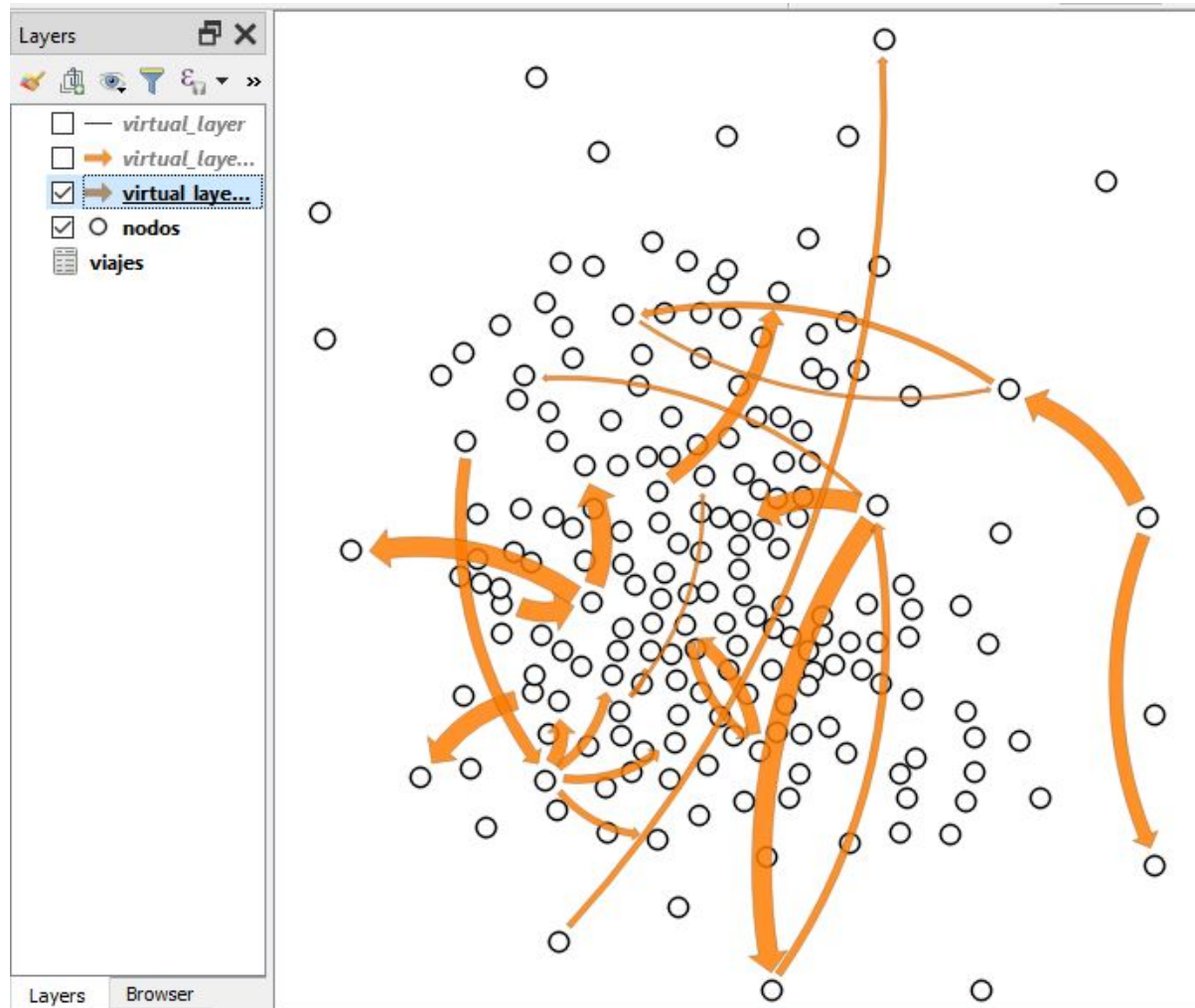
```
scale_linear("viajes",273,4000,0.5,3)
```

La cual hace un re escalamiento de los valores de la columna “viajes”, que tiene valores que van desde el 273 hasta el 4000, que serán rescalados a valores que van desde el 0.5 hasta el 3.



Al aceptar estos cambios, si las sentencias son correctamente aplicadas deberían tener una salida visual similar a la que aquí se muestra.

Ahora solo restaría obtener una paleta de colores que nos ayudará a diferenciar aún más nuestros flujos



Para hacer los cambios de color, dentro de la pestaña de symbology navegaremos hasta la opción de simple fill y después a fill color en donde utilizaremos el siguiente código.

CASE WHEN

"viajes" > 2000

THEN

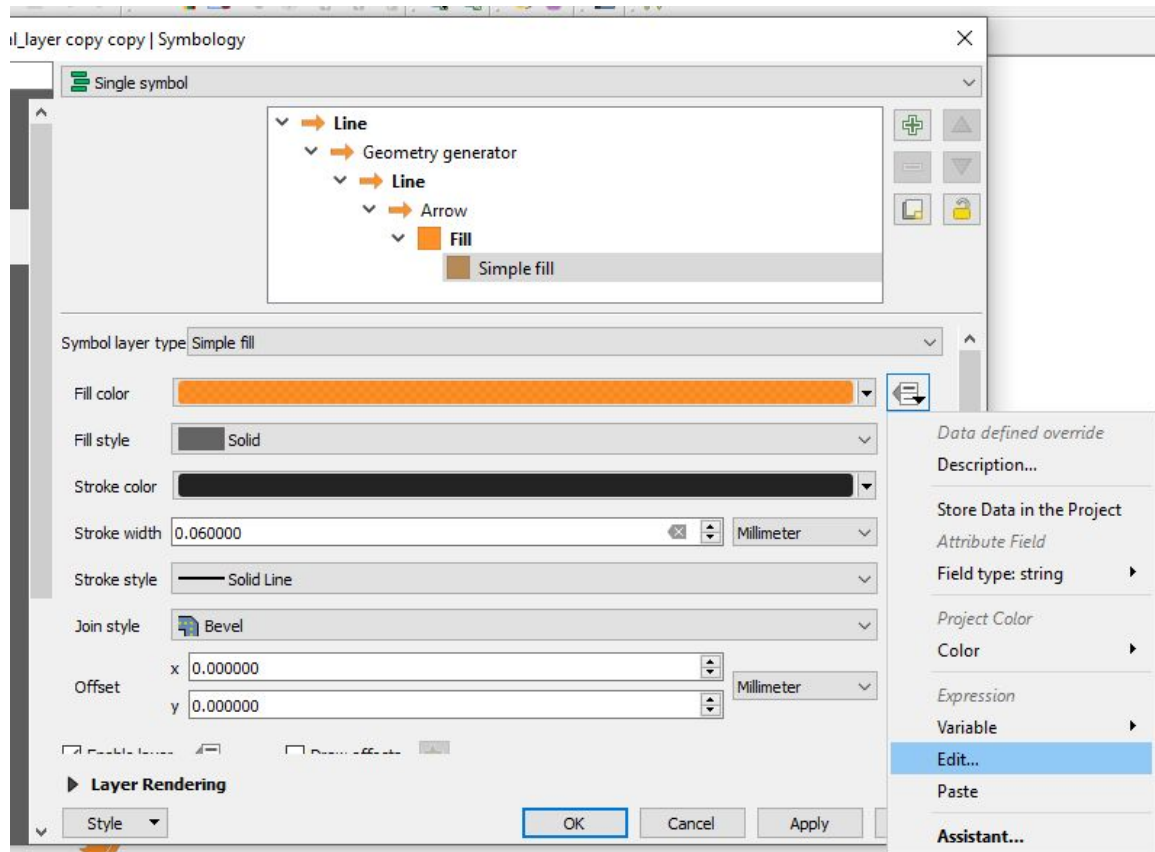
'#27A4A6'

ELSE

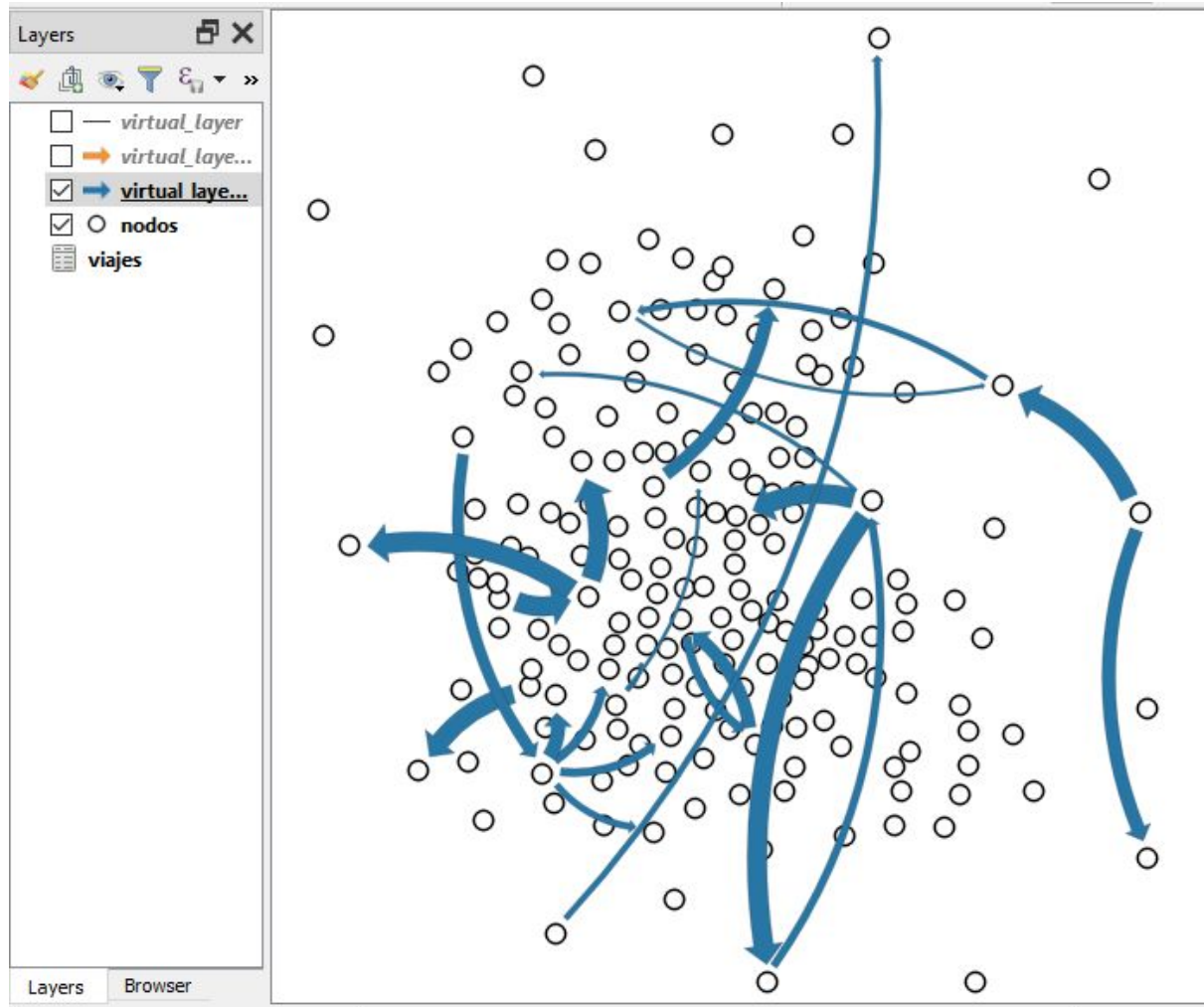
'#2776A6'

END

El cual en caso de que el valor de la columna "viajes" sea mayor les asignara el color '#27A4A6' y en caso de que no sea se asignara el color '#2776A6'.



Finalmente obtenemos la siguiente visualización, en la que podemos modificar los distintos códigos proporcionados así como añadir más elementos que ayuden a representar de mejor manera nuestros datos.



¡Gracias!



¡Gracias!

