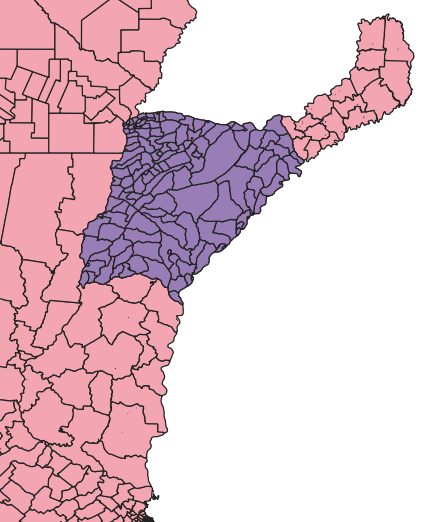
**PROVINCIA: CORRIENTES**

En las elecciones generales del pasado 14 de noviembre de 2021, los ciudadanos de la provincia de Corrientes, eligieron diputados y senadores nacionales.



**QGIS - CARGA DE INPUTS**

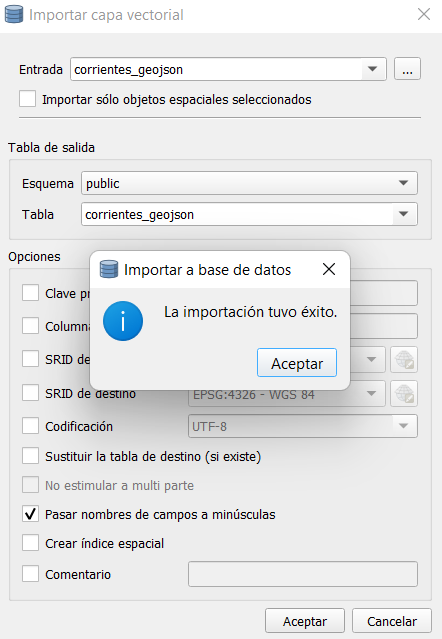
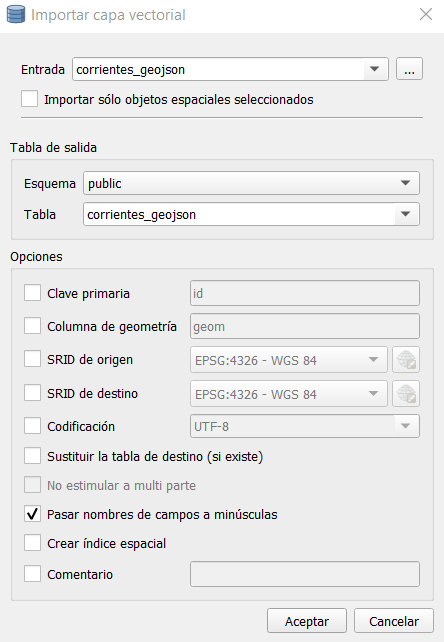
Resuelta la conexión PostGIS / PostgreSQL al database **geodb** (conexión geodb\_13), desde **QGIS** se agregarán las capas que se detallan a continuación:

# **Datos Geográficos**

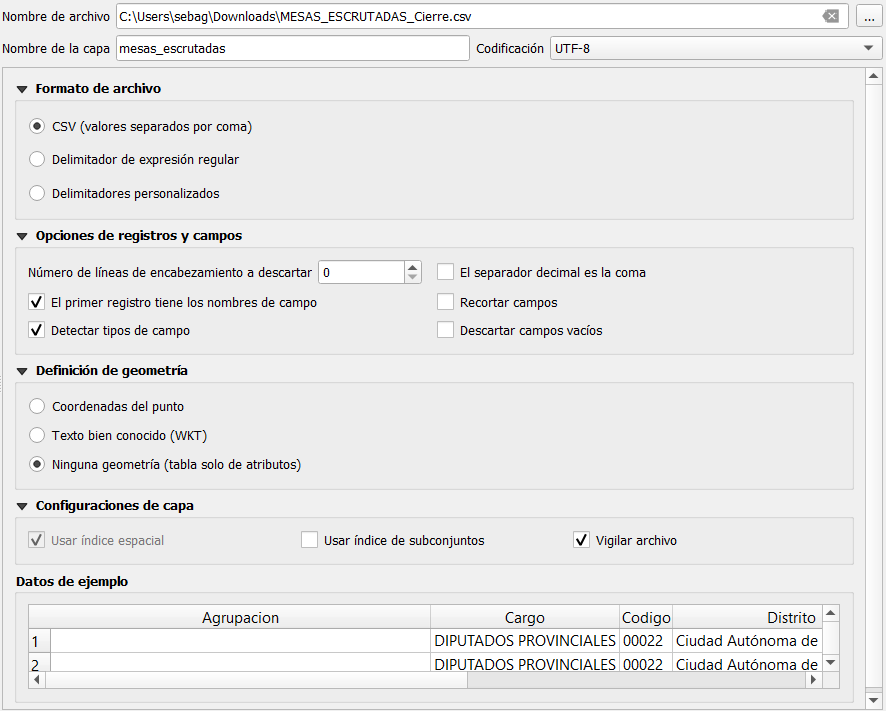
* + Añadir **capa vectorial** en QGIS;
  + Tipo de fuente: Protocolo HTTP(s), cloud, etc;

URI:<https://mapa2.electoral.gov.ar/geoserver/wfs?service=WFS&version=1.0.0&request=GetFeature&authkey=edb25b5f006fa780f3b9fcae20026b1d&typeName=descargas:circuito_05&maxFeatures=2000&outputFormat=application%2Fjson>;

* + Renombrar capa a **corrientes\_geojson**;
  + Desde **Administrador de bases de datos** (en Barra de Herramientas, Base de datos), seleccionar conexión geodb\_13, Importar capa / archivo, esquema público, seleccionar opción: Pasar nombres de campos a minúscula;

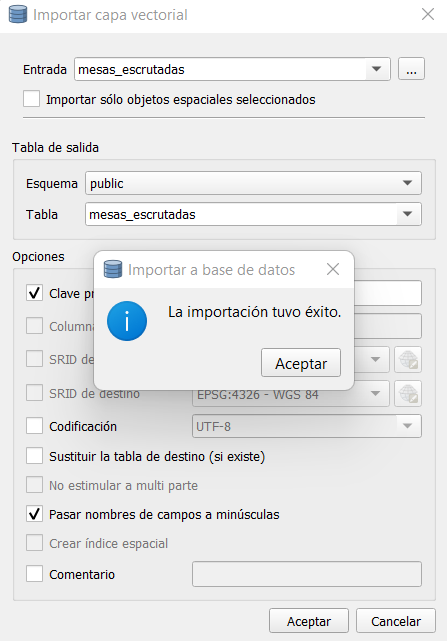
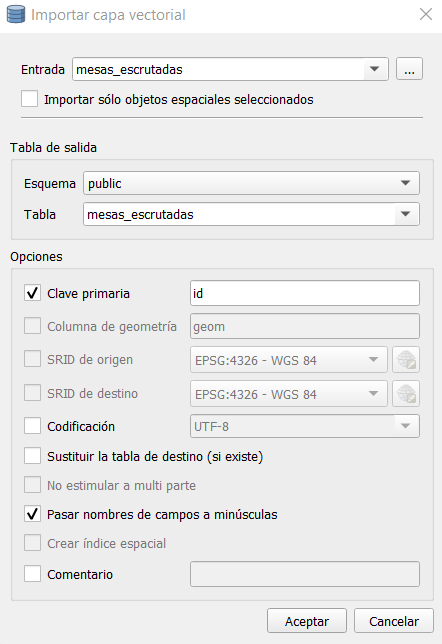


# **Datos de mesas escrutadas**

* + Se descarga el csv de mesas escrutadas, link: <https://telegramas.resultados.gob.ar/>;
  + Añadir **capa de texto delimitado** en QGIS;
  + Nombre de la capa: **mesas\_escrutadas**;
  + Codificación: **UTF-8** (evita inconvenientes para caracteres especiales, incluye acentos);
  + Opciones de registros y campos por default;
  + Definición de geometría: Ninguna geometría;

La demora en carga, se debe a que la fuente contiene 2.665.130 tuplas. A diferencia del apartado anterior que solo correspondía al distrito de Corrientes, acá se informa todas las jurisdicciones del país.

Luego, desde **Administrador de bases de datos** (en Barra de Herramientas, Base de datos), seleccionar conexión geodb\_13, Importar capa / archivo, esquema público, seleccionar opciones: Clave primaria, Pasar nombres de campos a minúscula;



**POSTGRESQL - PREPARACIÓN DE TABLAS**

# **Verificar** **ATRIBUTOS** para el **JOIN**



Se verifica el largo de los atributos diagramados.

* + **mesas\_escrutadas:** El atributo **idcircuito** de las 2.665.130 tuplas tienen un largo de **5 caracteres**;

*SELECT*

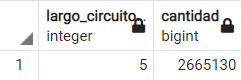
*LENGTH (idcircuito) as largo\_circuito,*

*COUNT (idcircuito) as Cantidad*

*FROM mesas\_escrutadas*

*GROUP BY 1*

*ORDER BY 1;*



* + **corrientes\_geojson:** El atributo **circuito** puede tomar **4, 5, 7 u 8 caracteres**.

*SELECT*

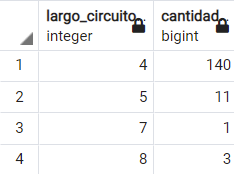
*LENGTH (circuito) as largo\_circuito,*

*COUNT (circuito) as Cantidad*

*FROM corrientes\_geojson*

*GROUP BY 1*

*ORDER BY 1;*



Se identifican con 7 caracteres el circuito 0036ver, y con 8 el circuito 0012/A/B. En cambio, los 140 circuitos de 4 caracteres no presentan mayores dificultades para realizar un update del atributo.

Se corrige las observaciones encontradas ejecutando el siguiente sript:

*UPDATE corrientes\_geojson*

*SET circuito = b.long\_circ*

*FROM (*

*SELECT*

*CASE WHEN LENGTH(circuito)= 4 THEN CONCAT('0', circuito)*

*WHEN circuito= '0036ver' THEN LEFT(CONCAT('0', circuito),5)*

*ELSE circuito END as long\_circ,*

*id as idb*

*FROM corrientes\_geojson) as b*

*WHERE id = b.idb;*

Se ejecuta nuevamente el script para control, y se verifica que solo el circuito 0012/A/B no cumple con el largo de **5 caracteres**, considerando que en mesas escrutadas se diferencian en 00012, 0012A y 0012B.

*SELECT*

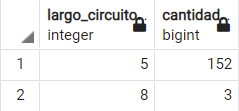
*LENGTH (circuito) as largo\_circuito,*

*COUNT (circuito) as Cantidad*

*FROM corrientes\_geojson*

*GROUP BY 1*

*ORDER BY 1;*



# **Crear** TABLA **corrientes\_mesas\_escrutadas**

Se realizó un JOIN de los atributos **idcircuito** (tabla mesas\_escrutadas) y **circuito** (tabla corrientes\_geojson). Se agregaron los votos por agrupación, tipo de voto (‘positivo’ / ‘resto’) y total votos, en los niveles establecimiento, circuito, sección y distrito. Para simplificar los tiempos de consulta, se filtró por el **iddistrito = 5** que corresponde a **Corrientes**.

*CREATE TABLE corrientes\_mesas\_escrutadas AS*

*SELECT*

*T0.id,*

*T1.geom,*

*T1.gid,*

*T1.provincia,*

*T1.departamen as departamento,*

*T1.cabecera,*

*T1.indec\_p,*

*T1.indec\_d,*

*T0.agrupacion,*

*T0.cargo,*

*T0.codigo,*

*T0.distrito,*

*T0.establecimiento,*

*T0.fecha,*

*T0.idcargo,*

*T0.idcircuito,*

*T0.iddistrito,*

*T0.idseccion,*

*T0.mesa,*

*T0.seccion,*

*T0.electores,*

*T0.envio,*

*T0.idagrupacion,*

*T0.idagrupacionint,*

*T0.tipovoto,*

*T0.votos,*

*T0.votosagrup\_x\_establecimiento,*

*T0.votosagrup\_x\_circuito,*

*T0.votosagrup\_x\_seccion,*

*T0.votosagrup\_x\_distrito,*

*T0.tipovoto\_x\_establecimiento,*

*T0.tipovoto\_x\_circuito,*

*T0.tipovoto\_x\_seccion,*

*T0.tipovoto\_x\_distrito,*

*T0.totalvotos\_x\_establecimiento,*

*T0.totalvotos\_x\_circuito,*

*T0.totalvotos\_x\_seccion,*

*T0.totalvotos\_x\_distrito*

*FROM*

*(SELECT*

*id,*

*agrupacion,*

*cargo,*

*codigo,*

*distrito,*

*establecimiento,*

*fecha,*

*idcargo,*

*idcircuito,*

*iddistrito,*

*idseccion,*

*mesa,*

*seccion,*

*electores,*

*envio,*

*idagrupacion,*

*idagrupacionint,*

*tipovoto,*

*votos,*

*-- Total votos POR AGRUPACIÓN en los distintos niveles*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idseccion, idcircuito, codigo, idagrupacion) as votosagrup\_x\_establecimiento,*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idseccion, idcircuito, idagrupacion) as votosagrup\_x\_circuito,*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idseccion, idagrupacion) as votosagrup\_x\_seccion,*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idagrupacion) as votosagrup\_x\_distrito,*

*-- Total votos POR TIPO en los distintos niveles*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idseccion, idcircuito, codigo, (CASE WHEN tipovoto = 'positivo' THEN 'positivo' ELSE 'resto' END)) as tipovoto\_x\_establecimiento,*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idseccion, idcircuito, (CASE WHEN tipovoto = 'positivo' THEN 'positivo' ELSE 'resto' END)) as tipovoto\_x\_circuito,*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idseccion, (CASE WHEN tipovoto = 'positivo' THEN 'positivo' ELSE 'resto' END)) as tipovoto\_x\_seccion,*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, (CASE WHEN tipovoto = 'positivo' THEN 'positivo' ELSE 'resto' END)) as tipovoto\_x\_distrito,*

*-- TOTAL VOTOS en los distintos niveles*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idseccion, idcircuito, codigo) as totalvotos\_x\_establecimiento,*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idseccion, idcircuito) as totalvotos\_x\_circuito,*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito, idseccion) as totalvotos\_x\_seccion,*

*SUM (votos) OVER (PARTITION BY idcargo, iddistrito) as totalvotos\_x\_distrito*

*FROM mesas\_escrutadas*

*WHERE iddistrito = '05') AS T0*

*INNER JOIN corrientes\_geojson AS T1*

*ON T0.idcircuito = T1.circuito;*

# **Calcular** RATIOS en tabla **corrientes\_mesas\_escrutadas**

Se agregaron columnas en la tabla **corrientes\_mesas\_escrutadas** para el cálculo de ratios.

*ALTER TABLE corrientes\_mesas\_escrutadas*

*ADD COLUMN ratio\_votos\_tipovoto\_x\_establecimiento decimal(5,2)*

*CHECK (ratio\_votos\_tipovoto\_x\_establecimiento>=0 and ratio\_votos\_tipovoto\_x\_establecimiento<=100),*

*ADD COLUMN ratio\_votos\_tipovoto\_x\_circuito decimal(5,2)*

*CHECK (ratio\_votos\_tipovoto\_x\_circuito>=0 and ratio\_votos\_tipovoto\_x\_circuito<=100),*

*ADD COLUMN ratio\_votos\_tipovoto\_x\_seccion decimal(5,2)*

*CHECK (ratio\_votos\_tipovoto\_x\_seccion>=0 and ratio\_votos\_tipovoto\_x\_seccion<=100),*

*ADD COLUMN ratio\_votos\_tipovoto\_x\_distrito decimal(5,2)*

*CHECK (ratio\_votos\_tipovoto\_x\_distrito>=0 and ratio\_votos\_tipovoto\_x\_distrito<=100),*

*ADD COLUMN ratio\_votos\_totalvotos\_x\_establecimiento decimal(5,2)*

*CHECK (ratio\_votos\_totalvotos\_x\_establecimiento>=0 and ratio\_votos\_totalvotos\_x\_establecimiento<=100),*

*ADD COLUMN ratio\_votos\_totalvotos\_x\_circuito decimal(5,2)*

*CHECK (ratio\_votos\_totalvotos\_x\_circuito>=0 and ratio\_votos\_totalvotos\_x\_circuito<=100),*

*ADD COLUMN ratio\_votos\_totalvotos\_x\_seccion decimal(5,2)*

*CHECK (ratio\_votos\_totalvotos\_x\_seccion>=0 and ratio\_votos\_totalvotos\_x\_seccion<=100),*

*ADD COLUMN ratio\_votos\_totalvotos\_x\_distrito decimal(5,2)*

*CHECK (ratio\_votos\_totalvotos\_x\_distrito>=0 and ratio\_votos\_totalvotos\_x\_distrito<=100);*

Se poblaron las columnas con los ratios calculados.

*UPDATE corrientes\_mesas\_escrutadas*

*SET*

*ratio\_votos\_tipovoto\_x\_establecimiento = b.ratio\_votos\_tipovoto\_x\_establecimiento,*

*ratio\_votos\_tipovoto\_x\_circuito = b.ratio\_votos\_tipovoto\_x\_circuito,*

*ratio\_votos\_tipovoto\_x\_seccion = b.ratio\_votos\_tipovoto\_x\_seccion,*

*ratio\_votos\_tipovoto\_x\_distrito = b.ratio\_votos\_tipovoto\_x\_distrito,*

*ratio\_votos\_totalvotos\_x\_establecimiento = b.ratio\_votos\_totalvotos\_x\_establecimiento,*

*ratio\_votos\_totalvotos\_x\_circuito = b.ratio\_votos\_totalvotos\_x\_circuito,*

*ratio\_votos\_totalvotos\_x\_seccion = b.ratio\_votos\_totalvotos\_x\_seccion,*

*ratio\_votos\_totalvotos\_x\_distrito = b.ratio\_votos\_totalvotos\_x\_distrito*

*FROM (*

*SELECT*

*id as idb,*

*votosagrup\_x\_establecimiento,*

*tipovoto\_x\_establecimiento,*

*CASE WHEN tipovoto\_x\_establecimiento <> 0 THEN CAST(votosagrup\_x\_establecimiento::numeric / tipovoto\_x\_establecimiento::numeric \*100 as decimal (5,2)) ELSE 0 END as ratio\_votos\_tipovoto\_x\_establecimiento,*

*CASE WHEN tipovoto\_x\_circuito <> 0 THEN CAST(votosagrup\_x\_circuito::numeric / tipovoto\_x\_circuito::numeric \*100 as decimal (5,2)) ELSE 0 END as ratio\_votos\_tipovoto\_x\_circuito,*

*CASE WHEN tipovoto\_x\_seccion <> 0 THEN CAST(votosagrup\_x\_seccion::numeric / tipovoto\_x\_seccion::numeric \*100 as decimal (5,2)) ELSE 0 END as ratio\_votos\_tipovoto\_x\_seccion,*

*CASE WHEN tipovoto\_x\_distrito <> 0 THEN CAST(votosagrup\_x\_distrito::numeric / tipovoto\_x\_distrito::numeric \*100 as decimal (5,2)) ELSE 0 END as ratio\_votos\_tipovoto\_x\_distrito,*

*CASE WHEN totalvotos\_x\_establecimiento <> 0 THEN CAST(votosagrup\_x\_establecimiento::numeric / totalvotos\_x\_establecimiento::numeric \*100 as decimal (5,2)) ELSE 0 END as ratio\_votos\_totalvotos\_x\_establecimiento,*

*CASE WHEN totalvotos\_x\_circuito <> 0 THEN CAST(votosagrup\_x\_circuito::numeric / totalvotos\_x\_circuito::numeric \*100 as decimal (5,2)) ELSE 0 END as ratio\_votos\_totalvotos\_x\_circuito,*

*CASE WHEN totalvotos\_x\_seccion <> 0 THEN CAST(votosagrup\_x\_seccion::numeric / totalvotos\_x\_seccion::numeric \*100 as decimal (5,2)) ELSE 0 END as ratio\_votos\_totalvotos\_x\_seccion,*

*CASE WHEN totalvotos\_x\_distrito <> 0 THEN CAST(votosagrup\_x\_distrito::numeric / totalvotos\_x\_distrito::numeric \*100 as decimal (5,2)) ELSE 0 END as ratio\_votos\_totalvotos\_x\_distrito*

*FROM corrientes\_mesas\_escrutadas) AS b*

*WHERE id = b.idb;*

**QGIS - MAPAS GEOGRÁFICOS**

# Mapa graduado por agrupación

Para los mapas de cada agrupación con su correspondiente distribución de votos por circuito, se decidió crear la tabla **corrientes\_votos\_x\_circuito** con los datos necesarios, considerando solo los votos positivos.

*CREATE TABLE corrientes\_votos\_x\_circuito AS*

*SELECT*

*cargo,*

*iddistrito,*

*distrito,*

*idseccion,*

*seccion,*

*idcircuito,*

*geom,*

*agrupacion,*

*MAX(ratio\_votos\_tipovoto\_x\_circuito) as ratio\_votos\_tipovoto\_x\_circuito*

*FROM corrientes\_mesas\_escrutadas*

*WHERE tipovoto = 'positivo'*

*GROUP BY 1,2,3,4,5,6,7,8*

*ORDER BY 2,4,6,8;*

Desde QGIS se armó cada mapa (ver anexo), considerando:

* + Filtrar por:
    - cargo: ‘DIPUTADOS NACIONALES’ o ‘SENADORES NACIONALES’
    - agrupación: ‘ECO + VAMOS CORRIENTES’ o ‘FRENTE DE TODOS’ o ‘VAMOS CON VOS’ o ‘COMPROMISO FEDERAL’
  + Simbología: Graduado.
    - Se definieron 5 clases
    - Valor: ROUND( "ratio\_votos\_tipovoto\_x\_circuito",2)
    - Modo: conteo igual (cuantil)
  + Etiquetas: Single Labels.
    - Valor: ROUND( "ratio\_votos\_tipovoto\_x\_circuito",2)

# Mapa agrupación ganadora

Se agregaron nuevas columnas a la tabla **corrientes\_votos\_x\_circuito**.

*ALTER TABLE corrientes\_votos\_x\_circuito*

*ADD COLUMN agurpacion\_ganadora\_circuito varchar,*

*ADD COLUMN votos\_agurpacion\_ganadora\_circuito decimal(5,2);*

Se pobló con el ganador de cada circuito y la cantidad de votos.

*update corrientes\_votos\_x\_circuito*

*set*

*agurpacion\_ganadora\_circuito = b.bgrup,*

*votos\_agurpacion\_ganadora\_circuito = b.max\_votos\_circuito*

*from (*

*select idcircuito as idb, agrupacion as bgrup, cargo as cargob, max\_votos\_circuito*

*from(*

*select agrupacion, cargo, idcircuito, seccion, ratio\_votos\_tipovoto\_x\_circuito,*

*max(ratio\_votos\_tipovoto\_x\_circuito) over (partition by cargo, idcircuito, seccion) as max\_votos\_circuito*

*from corrientes\_votos\_x\_circuito) as sub1*

*where ratio\_votos\_tipovoto\_x\_circuito = max\_votos\_circuito*

*order by idcircuito) as b*

*where idcircuito = b.idb and cargo = b.cargob*

Finalmente, desde QGIS se armó el mapa con los ganadores de cada circuito.