

# SIG. Práctica 6. Técnicas de análisis espacial desde *QGIS* y *R*

José Samos Jiménez

Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Granada

2019 jsamos (LSI-UGR)

Curso 2019-2020

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Consultas espaciales sobre BD desde <i>QGIS</i></b>	<b>3</b>
2.1. Consulta sobre una capa . . . . .	3
2.1.1. Consulta sobre <i>PostGIS</i> . . . . .	3
2.1.2. Consulta sobre <i>SpatiaLite</i> . . . . .	5
2.1.3. Consulta sobre <i>GeoPackage</i> . . . . .	7
2.2. Consulta sobre dos capas . . . . .	7
2.3. Conclusiones . . . . .	9
<b>3. Análisis mediante operaciones del menú de <i>QGIS</i></b>	<b>9</b>
<b>4. Análisis mediante el modelador de procesos de <i>QGIS</i></b>	<b>12</b>
<b>5. <i>Consola de Python</i> en <i>QGIS</i></b>	<b>15</b>
<b>6. Consultas sobre <i>PostGIS</i> desde <i>R</i></b>	<b>19</b>
<b>7. Análisis mediante funciones de paquetes de <i>R</i></b>	<b>19</b>
<b>Bibliografía</b>	<b>21</b>

Los objetivos de esta actividad son:

- Presentar diversas técnicas para realizar análisis espacial desde *QGIS* y *R*, en concreto:
  - Realizar consultas espaciales básicas mediante SQL sobre *SpatiaLite*, *GeoPackage* y *PostGIS* desde *QGIS* para tratar de determinar las prestaciones o limitaciones de estos sistemas.
  - Usar las opciones del menú de *QGIS* para realizar las operaciones de análisis.
  - Usar el modelador de procesos de *QGIS* para definir procesos de análisis de datos.
  - Utilizar la *Consola de Python* para acceder a las operaciones de análisis.
  - Realizar consultas espaciales básicas mediante SQL sobre *PostGIS* desde *R*.
  - Realizar análisis espacial desde *R* mediante funciones de paquetes.

A continuación, después de una introducción, se irán presentando los pasos a realizar y cómo llevarlos a cabo con las distintas opciones consideradas.

**Realiza los pasos que se indican y entrega un documento PDF que contenga el título de cada uno de los apartados y, asociado a cada uno, capturas de las pantallas donde se muestre la operación realizada y los resultados obtenidos.**

## 1. Introducción

*QGIS* ofrece varias posibilidades para analizar información geográfica. En esta actividad se realiza una misma operación básica de análisis por los distintos métodos considerados.

En el primer caso, el análisis se realiza en la BD (*SpatiaLite*, *GeoPackage* o *PostGIS*), mediante la consulta que ejecuta: *QGIS* actúa como medio de acceso a la BD y de representación gráfica del resultado, si así se desea. Comprobaremos con un ejemplo las posibilidades y limitaciones de cada sistema.

En el resto de casos, el análisis se realiza mediante los medios que ofrece *QGIS* para acceder a las funciones de análisis disponibles: directamente desde las opciones del menú, mediante el modelador de procesos y mediante la consola de *Python*.

## 2. Consultas espaciales sobre BD desde *QGIS*

Las operaciones a realizar sobre BD están basadas en [Mar15]. Para formular una consulta en *SQL* desde *QGIS* sobre una BD espacial, en primer lugar, debemos acceder al administrador de BD de *QGIS* (pulsando sobre [«Base de datos», «Administrador de bases de datos...»]) y conectarnos a la BD.

Una vez con acceso a la BD, debemos abrir la pestaña para formular consultas: pulsando sobre la opción [«Base de datos», «Ventana SQL»] (figura 1).

El resultado es que se abre una pestaña de formulación de consultas asociada a la BD seleccionada en el momento de abrir la ventana (figura 1). Se pueden tener varias pestañas de consulta abiertas al mismo tiempo, una por cada BD distinta.

### 2.1. Consulta sobre una capa

Una vez abierta la ventana de consulta sobre la BD correspondiente, vamos a formular la consulta, en la que interviene una sola tabla (una capa), sobre los distintos sistemas, para comparar el resultado que obtenemos.

#### 2.1.1. Consulta sobre *PostGIS*

En la figura 2, se muestra la consulta en la que se obtiene un buffer (*ST\_Buffer*) a partir de las líneas de una capa. Adicionalmente, en la misma consulta, los componentes del resultado se fusionan (*ST\_Union*) si, al construir el buffer, estos se tocan. La consulta se ha definido para *PostGIS*, que

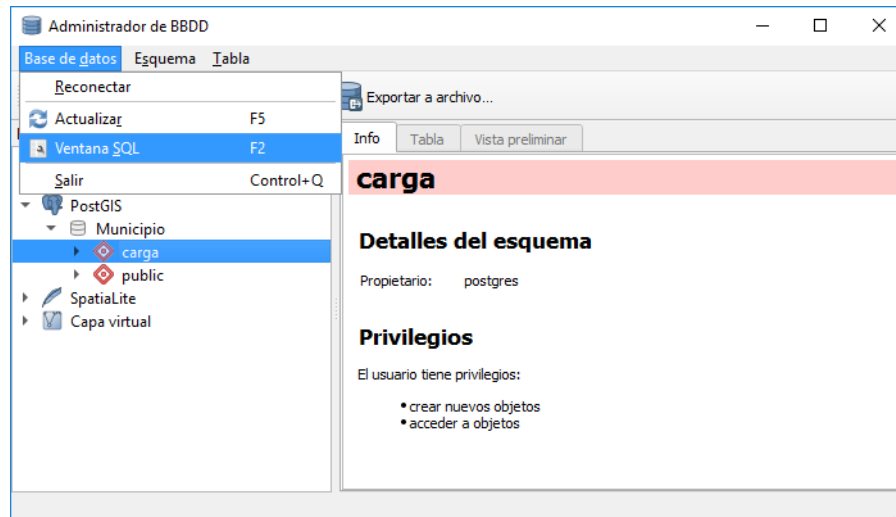
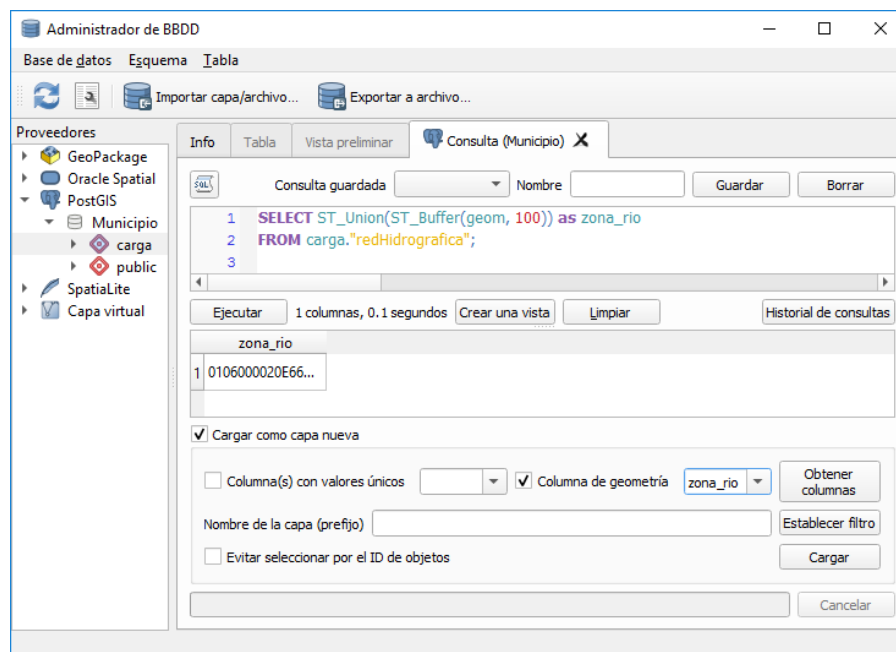


Figura 1: Acceso a la ventana SQL de la BD.

```
SELECT ST_Union(ST_Buffer(geom, 100)) as zona_rio  
FROM carga."redHidrografica";
```

Figura 2: Definición de la consulta sobre una capa para *PostGIS*.Figura 3: Consulta y resultado sobre *PostGIS*.

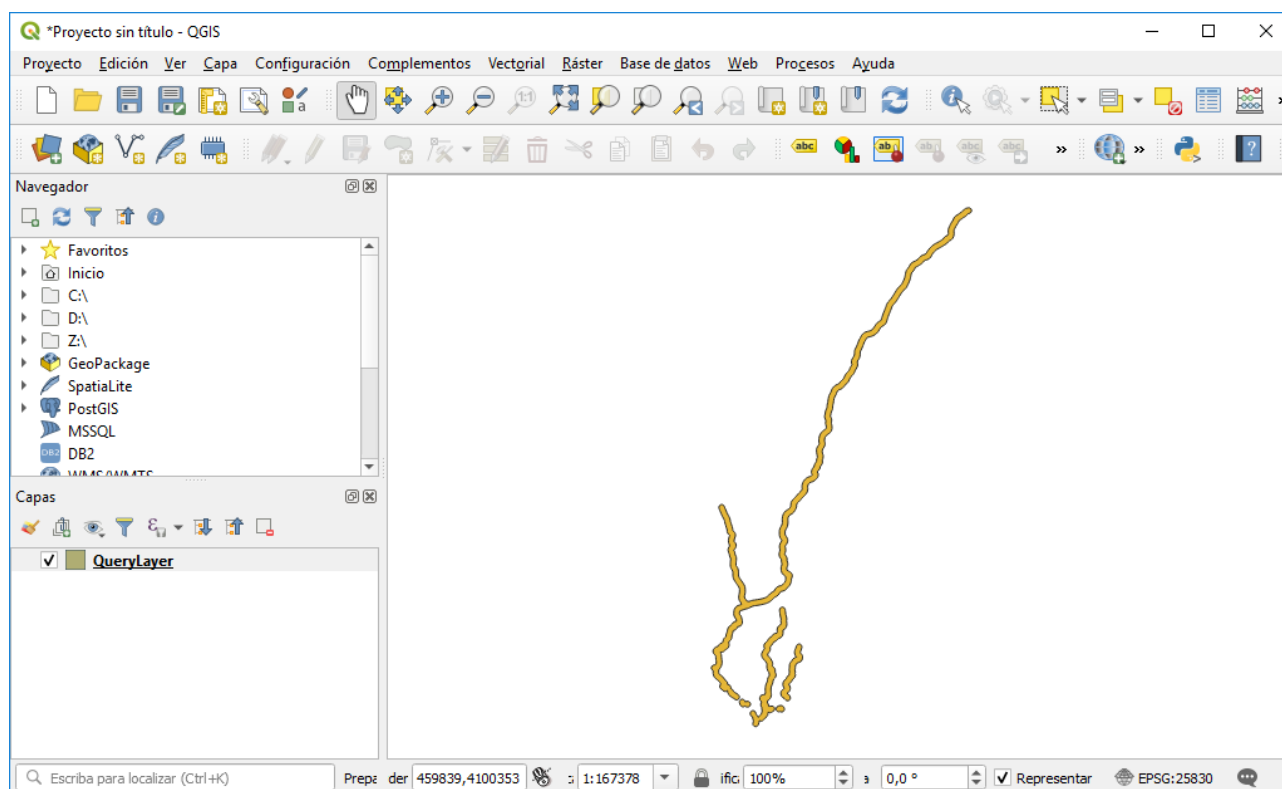


Figura 4: Resultado de la consulta de *PostGIS* como capa en *QGIS*.

```
SELECT ST_Union(ST_Buffer(geom, 100)) as zona_rio
FROM redHidrografica;
```

Figura 5: Definición de la consulta sobre una capa para *SpatiaLite*.

contempla el concepto de esquema, por este motivo se indica el esquema donde está la tabla a la que accedemos (*carga*).

En la figura 3, se puede ver la formulación de la consulta de la figura 2 en la ventana de consultas *PostGIS*. Al pulsar sobre el botón «Ejecutar», *PostGIS* ejecuta la consulta y devuelve el resultado en forma de tabla. Si seleccionamos la opción «Cargar como capa nueva», podemos indicar la columna que identifica los registros (si hay alguna) y la columna del resultado que contiene los datos geométricos (en la figura, la columna *zona\_rio*); si pulsamos sobre el botón «Cargar», carga los resultados como una capa más de *QGIS*, sobre la que podemos operar normalmente.

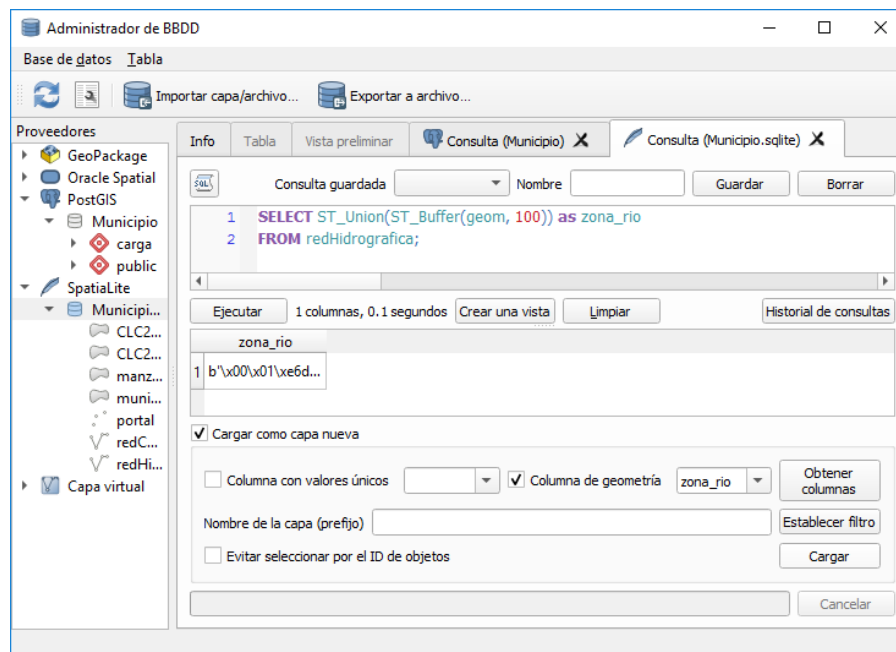
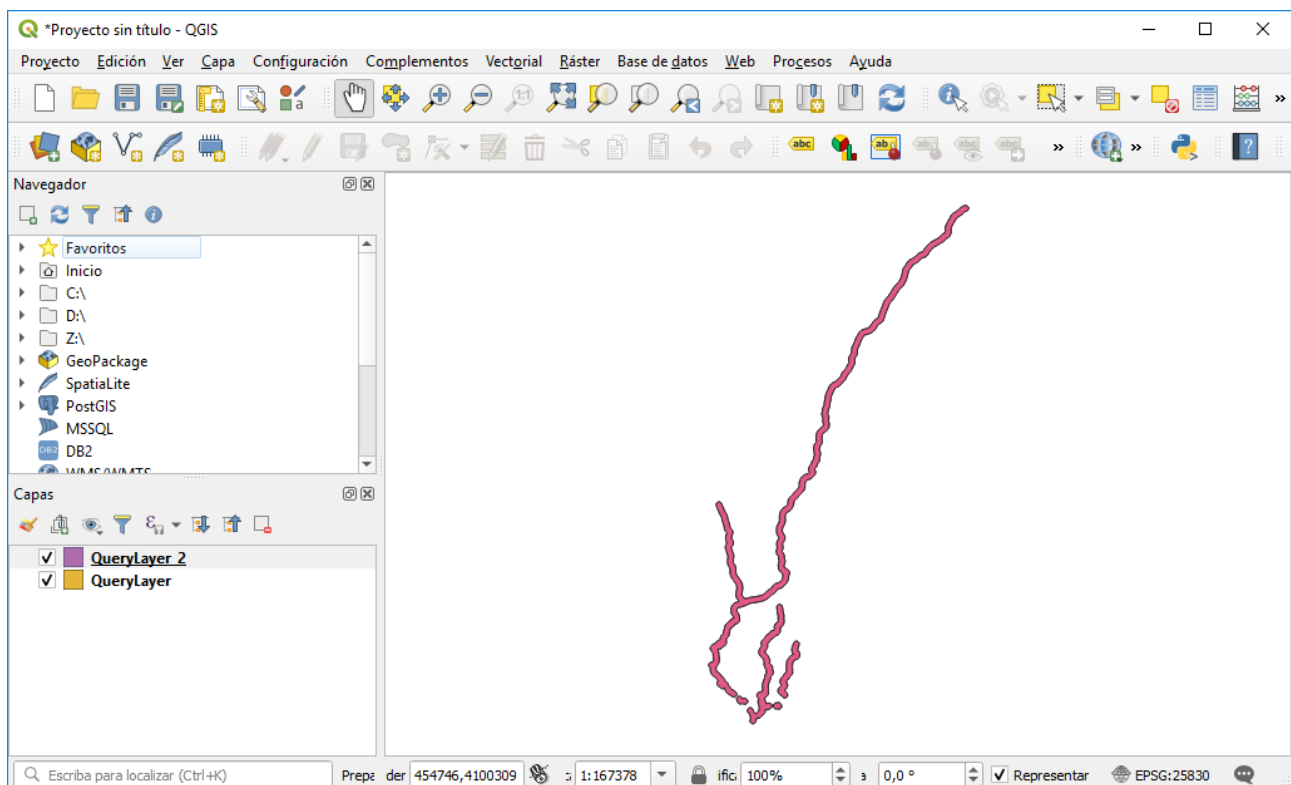
La capa resultado de la consulta cargada en *QGIS* se muestra en la figura 4. Partiendo de una capa vectorial compuesta por líneas, hemos obtenido una capa compuesta por polígonos mediante las operaciones de transformación que hemos utilizado.

1. Realiza una consulta similar a la de este apartado sobre los datos de tu municipio.

### 2.1.2. Consulta sobre *SpatiaLite*

La consulta equivalente a la de la figura 2 para *SpatiaLite* se muestra en la figura 5. La única diferencia es que en este caso no se indica el esquema en el que se encuentra la tabla.

La consulta y su resultado en la ventana de consultas del administrador de bases de datos se muestra en la figura 6; es similar al obtenido con *PostGIS*, aunque la representación de los valores para el campo resultado en este caso son distintos a los de la figura 3. El resultado de añadir como

Figura 6: Consulta y resultado sobre *SpatiaLite*.Figura 7: Resultado de la consulta de *SpatiaLite* como capa en *QGIS*.

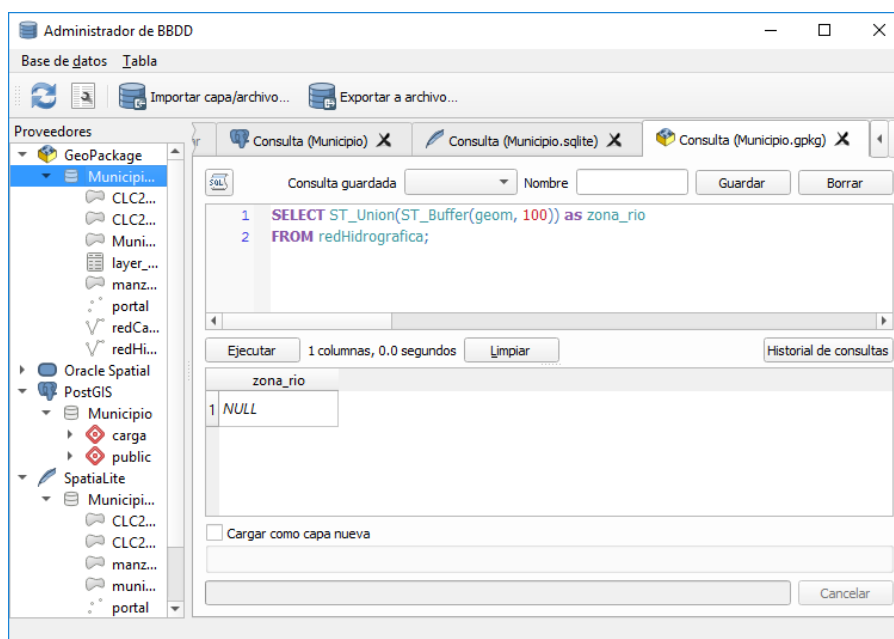


Figura 8: Consulta y resultado sobre *GeoPackage*.

```
SELECT p.*
FROM carga.portal p
WHERE EXISTS (
  SELECT id
  FROM carga."redHidrografica"
  WHERE ST_DWithin(p.geom, geom, 100)
);
```

Figura 9: Definición de la consulta sobre dos capas para *PostGIS*.

una capa de *QGIS* el resultado de la consulta, se muestra en la figura 7; en este caso es idéntico al obtenido con *PostGIS*.

**2. Realiza una consulta similar a la de este apartado sobre los datos de tu municipio.**

### 2.1.3. Consulta sobre *GeoPackage*

La consulta a formular sobre *GeoPackage* es la misma que sobre *Spatialite*, la de la figura 5. En la figura 8, se muestra la definición de la consulta y su resultado en la ventana del administrador de bases de datos. Se puede apreciar que, en este caso, el resultado que devuelve es el valor NULL para el campo con datos geográficos. Es decir, la consulta no funciona en *GeoPackage*. En las pruebas que he hecho, en *GeoPackage* funcionan las consultas en las que no intervienen datos geográficos, para datos geográficos devuelve el valor NULL.

**3. Realiza una consulta similar a la de este apartado sobre los datos de tu municipio.**

## 2.2. Consulta sobre dos capas

Para seguir con la prueba de las funcionalidades de las BD, a continuación, vamos a realizar una consulta sobre datos geográficos en la que intervienen dos capas. En la figura 9, se muestra la definición de la consulta considerada para *PostGIS*.

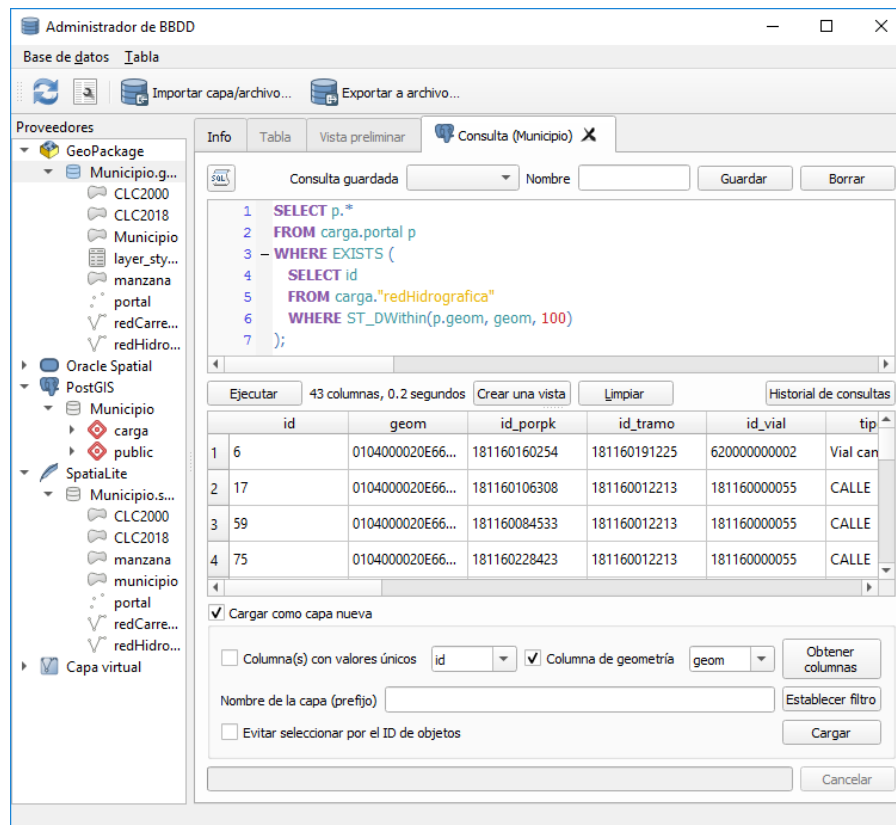


Figura 10: Consulta sobre dos capas y resultado sobre *PostGIS*.

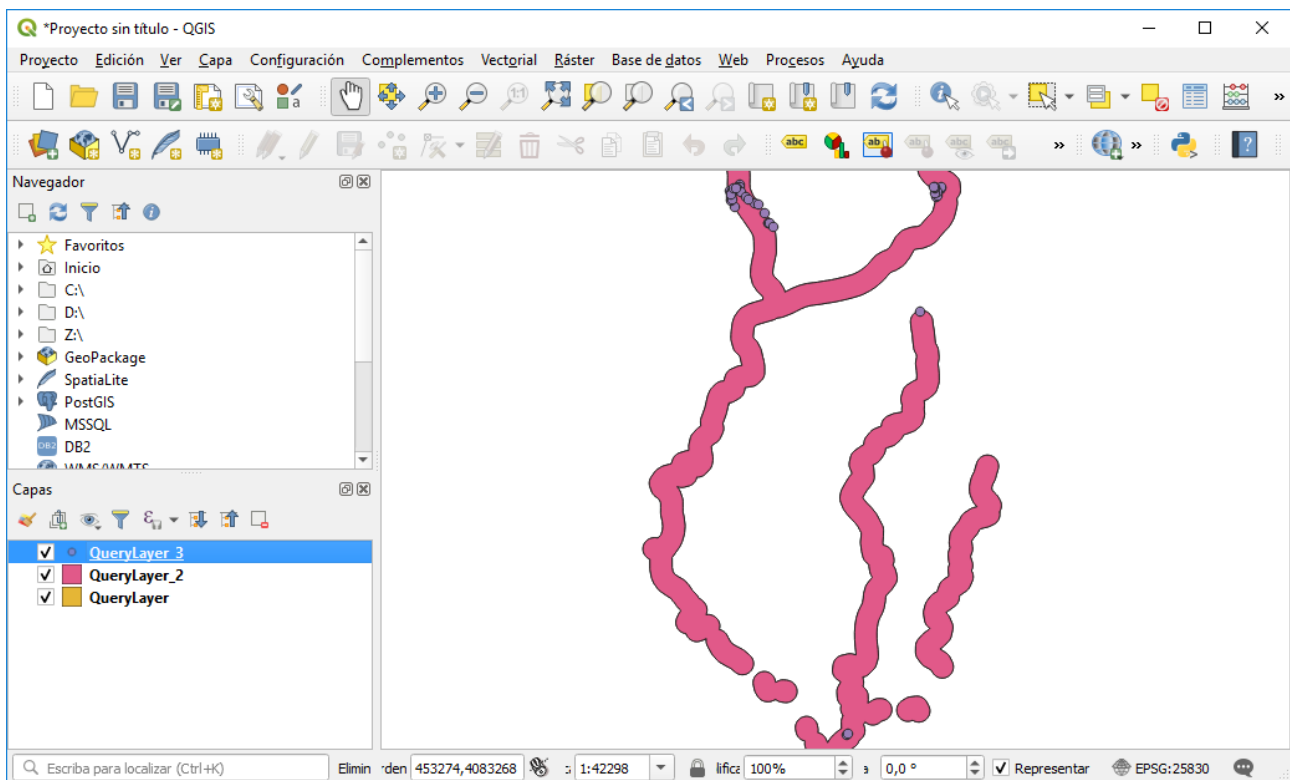


Figura 11: Resultado de la consulta sobre dos capas de *PostGIS* como capa en *QGIS*.



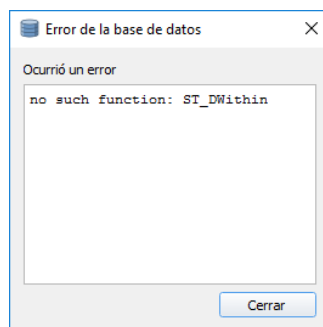


Figura 12: Aviso de error al formular la consulta sobre dos capas en *SpatiaLite*.

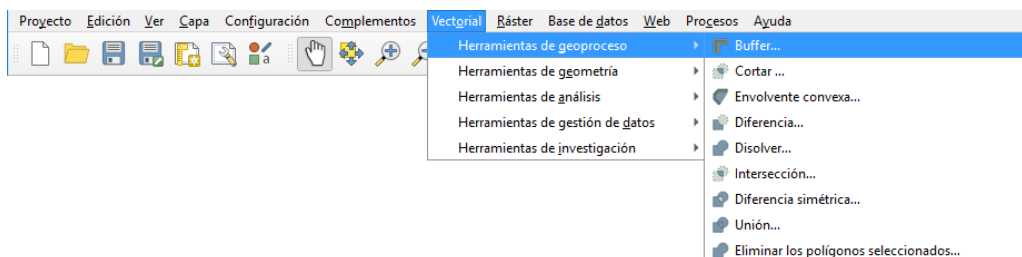


Figura 13: Operación *Buffer*.

El resultado de la definición y la presentación en forma de capa en QGIS se muestra en las figuras 10 y 11, respectivamente.

Si ejecutamos la consulta adaptada adecuadamente en *SpatiaLite*, aparece la ventana de error de la figura 12, indicando que la función utilizada no está definida. En <http://www.gaia-gis.it/spatialite-2.4.0-4/spatialite-sql-2.4-4.html> se puede comprobar que la función `ST_DWithin` no está definida para *SpatiaLite*.

Al formular esta consulta sobre *GeoPackage*, obtenemos el mismo tipo de respuesta que obtuvimos para la consulta sobre una sola tabla.

4. Realiza una consulta similar a la de este apartado sobre los datos de tu municipio para cada uno de los sistemas considerados.

## 2.3. Conclusiones

Como conclusión, a la vista de las pruebas realizadas, podemos afirmar que:

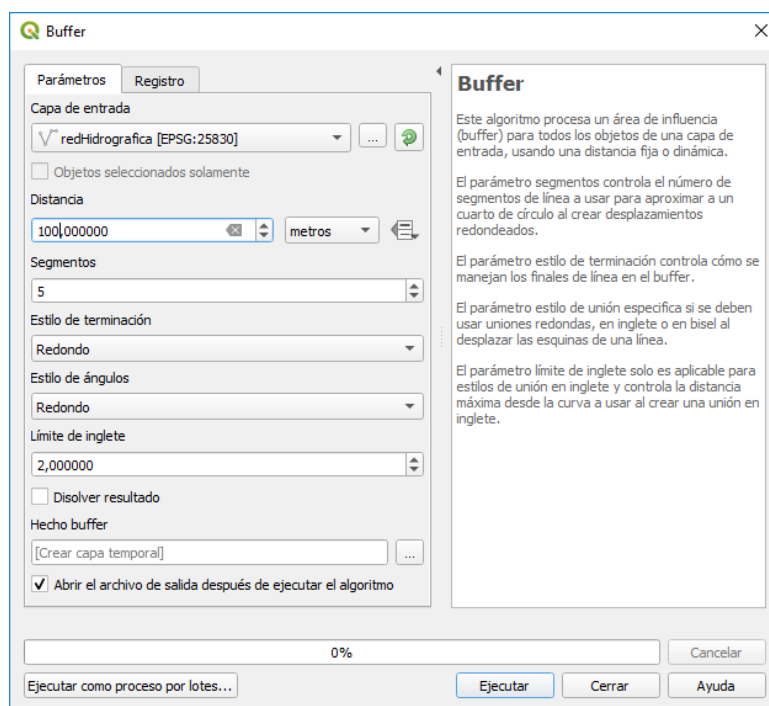
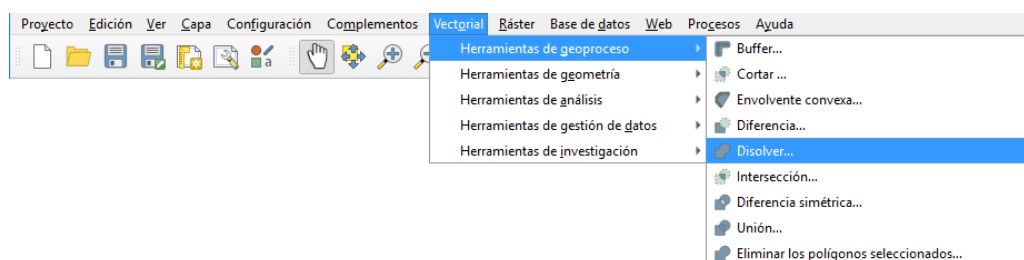
- *GeoPackage* es un buen contenedor de capas ráster y vectoriales, además soporta consultas SQL no espaciales sobre las tablas de atributos de las capas.
- *SpatiaLite*, de forma inmediata, soporta solo capas vectoriales y algunas consultas espaciales sobre ellas.
- *PostGIS* soporta capas ráster y vectoriales, y todo tipo de consultas sobre ellas.

Por tanto, en adelante, nos basaremos exclusivamente en *PostGIS* para realizar consultas espaciales sobre BD.

## 3. Análisis mediante operaciones del menú de *QGIS*

En lugar de utilizar las operaciones de análisis de la BD, podemos realizar el análisis mediante las operaciones del menú de *QGIS*.

Vamos a realizar la misma consulta de la figura 2: en primer lugar, obtenemos un buffer a partir de las líneas de una capa. La operación buffer se selecciona pulsando sobre [«Vectorial», «Herramientas de geoproc.»] (figura 13).

Figura 14: Parámetros de la operación *Buffer*.Figura 15: Operación *Disolver*.

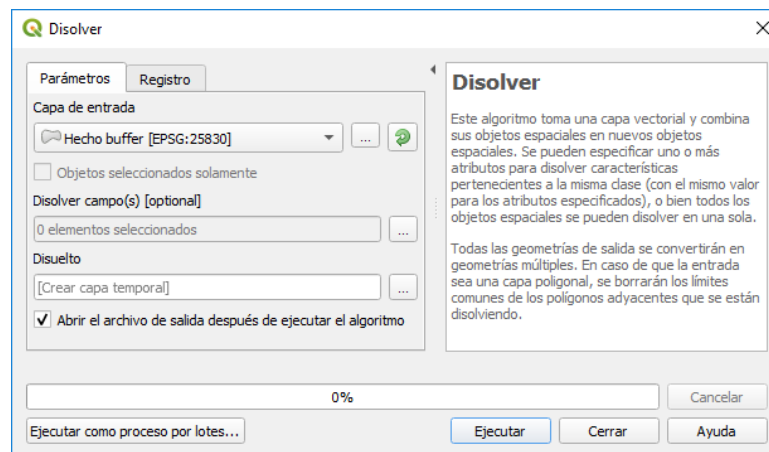
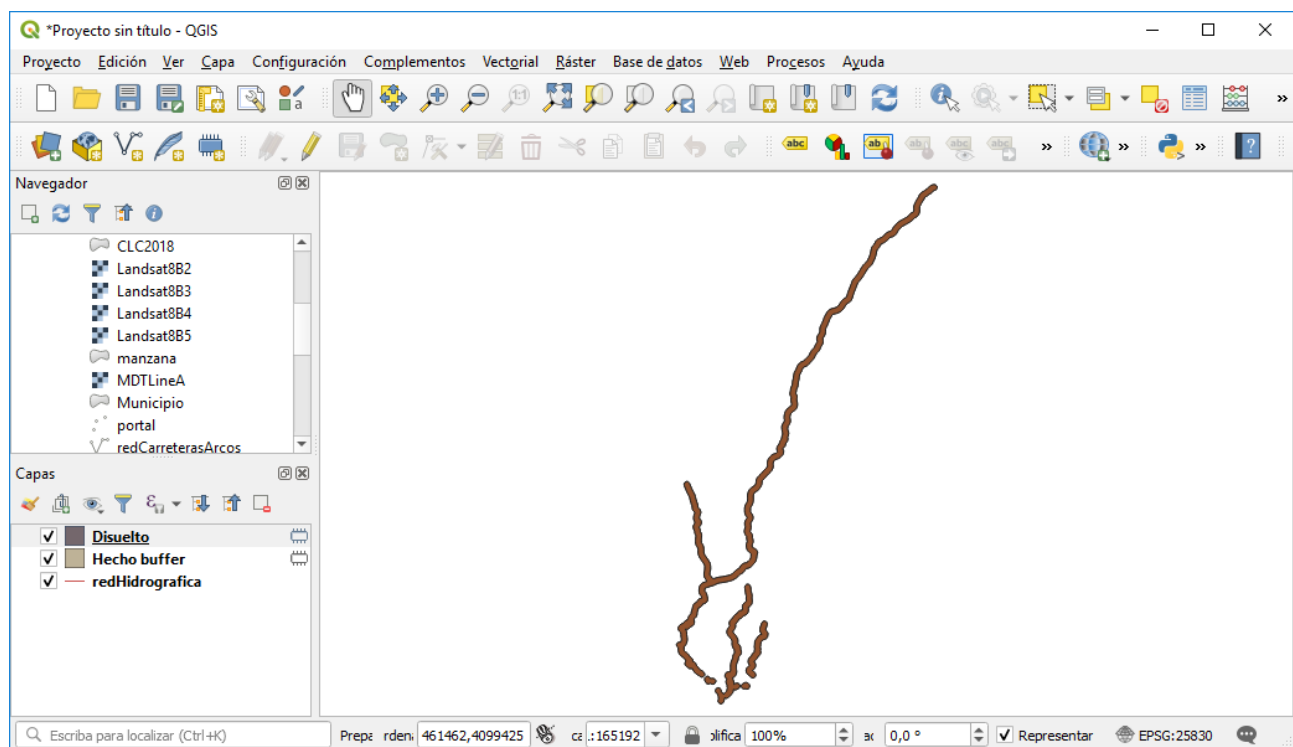
En la figura 14, se muestran los parámetros de la operación buffer: seleccionamos la capa y la distancia, para el resto de parámetros podemos dejar las opciones por defecto. Con esta operación obtenemos como resultado una capa que, si no hemos indicado un archivo donde guardarla, será temporal (podemos salvarla posteriormente si nos interesa).

Para llevar a cabo las mismas operaciones que se realizan en la figura 2, adicionalmente, debemos fusionar los componentes del resultado si, al construir el buffer, estos se tocan. La operación que necesitamos es [«Vectorial», «Herramientas de geoprocso», «Disolver...»] (figura 15).

Como parámetro de esta operación, seleccionamos la capa resultado de la operación anterior (figura 16). El resultado final es el mismo obtenido a partir de las operaciones en la BD, como se puede observar en la figura 17.

Trabajar directamente con las opciones del menú es adecuado para hacer pruebas o también para realizar análisis puntuales. Si necesitamos repetir el mismo análisis varias veces utilizando distintos datos, es más adecuado utilizar alguna solución que nos ayude a gestionar el proceso de análisis: en esta línea se enmarcan el *modelador de procesos* y la *consola de Python*, que se presentan a continuación.

5. Realiza una consulta similar a la de este apartado sobre los datos de tu municipio.
6. Realiza la consulta del apartado 2.2 mediante operaciones del menú de *QGIS*.

Figura 16: Parámetros de la operación *Disolver*.Figura 17: Resultado de las operaciones *Buffer* y *Disolver* en *QGIS*.

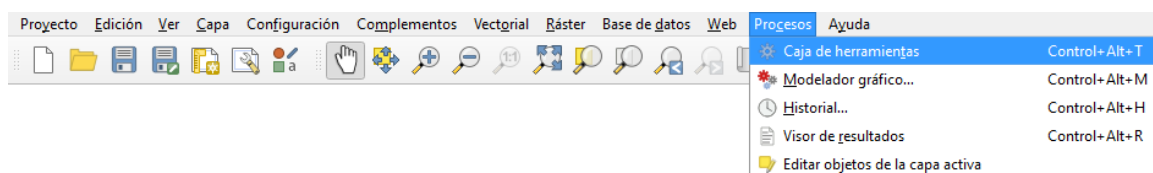


Figura 18: Caja de heramientas.

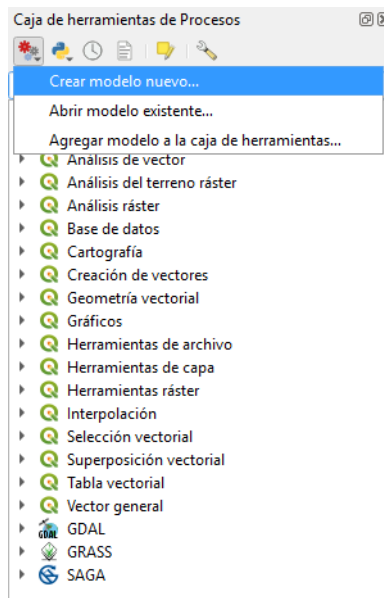


Figura 19: Crear nuevo modelo.

## 4. Análisis mediante el modelador de procesos de *QGIS*

El modelador de procesos de *QGIS* permite modelar gráficamente un proceso de análisis, definiendo parámetros y transformaciones a realizar. Los parámetros se pueden instanciar mediante las capas que tengamos disponibles.

Accedemos al modelador de procesos pulsando sobre [«Procesos», «Caja de herramientas»] (figura 18). Se abre la ventana «Caja de herramientas de Procesos» donde se muestran los temas de las funciones de *QGIS* accesibles, así como también las de otras herramientas como *GDAL*, *GRASS* y *SAGA* (figura 19).

Para crear un nuevo modelo, en la ventana «Caja de herramientas de Procesos», pulsamos sobre la opción [«Crear modelo nuevo...»] (figura 19). Indicamos un nombre para el modelo y un grupo: cuando tengamos varios modelos, se mostrarán agrupados por el nombre del grupo.

En un modelo se pueden definir entradas y algoritmos (operaciones). Entre otras posibilidades, podemos definir que una entrada sea una capa vectorial. No se define la capa vectorial concreta, sino que se define como un parámetro del que se indica el nombre y el tipo (figura 20). La definición se realiza en la pestaña «Entradas». En este caso, solo necesitamos una capa de entrada: solo tendremos que definir un parámetro en el modelo.

A partir de las capas de entrada, definimos operaciones en la pestaña «Algoritmos». Para realizar las operaciones de la consulta de la figura 2, seleccionamos la operación [«Geometría vectorial», «Buffer»] (figura 21). En la ventana que se abre, definimos la capa de los parámetros sobre la que se aplica y los valores necesarios para esta operación, en particular, el parámetro «Distancia».

De igual forma, seleccionamos la operación [«Geometría vectorial», «Disolver»] (figura 22). En la ventana de definición de la operación, seleccionamos que esta se aplique a la salida de la operación buffer. Si el resultado es una capa final, podemos indicar que, por ejemplo, se guarde en un archivo.

Conforme vamos definiendo el modelo, vemos una representación gráfica de los parámetros, las operaciones que realizamos sobre ellos y los resultados obtenidos (figura 23). Una vez definido el

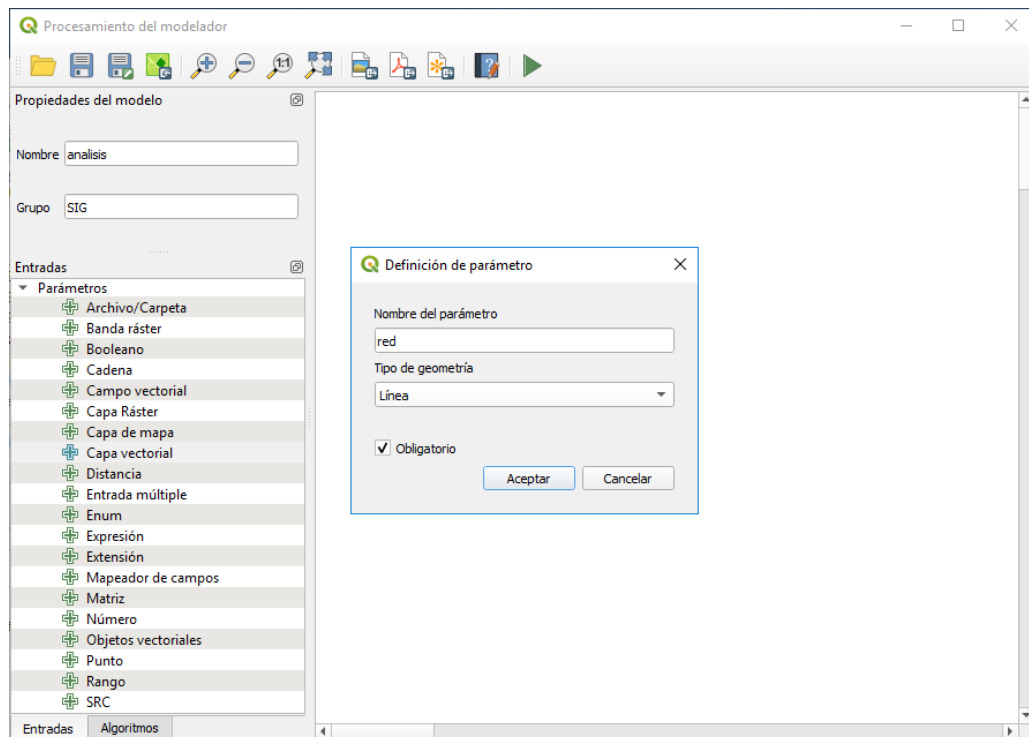
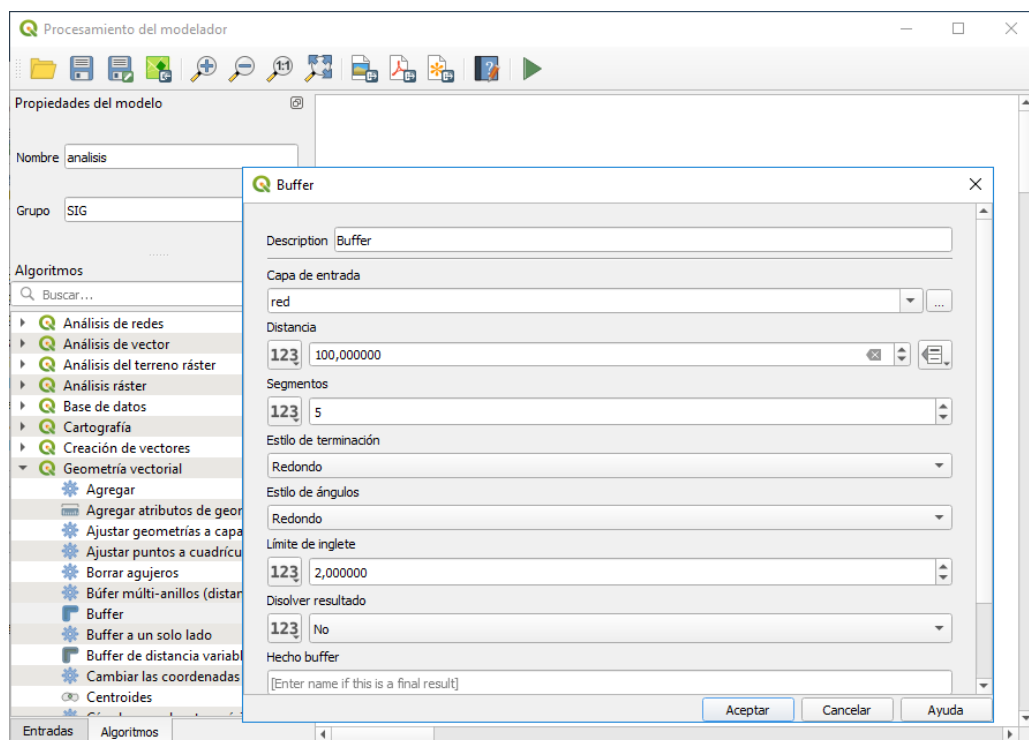


Figura 20: Definición de parámetro.

Figura 21: Definición de la operación *Buffer*.

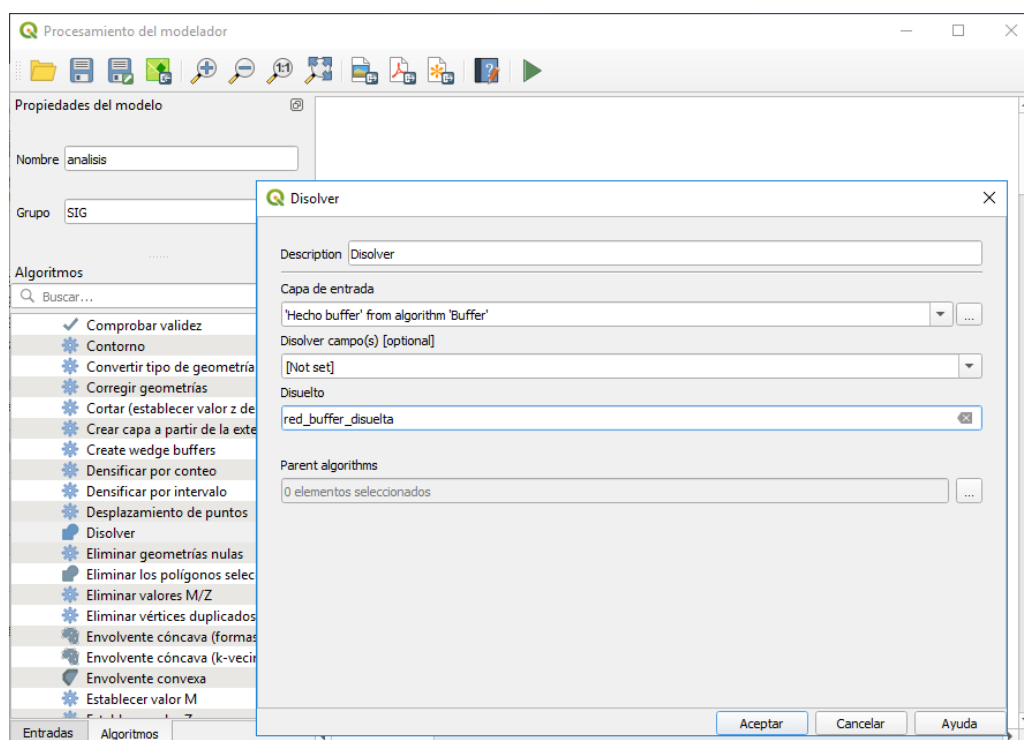
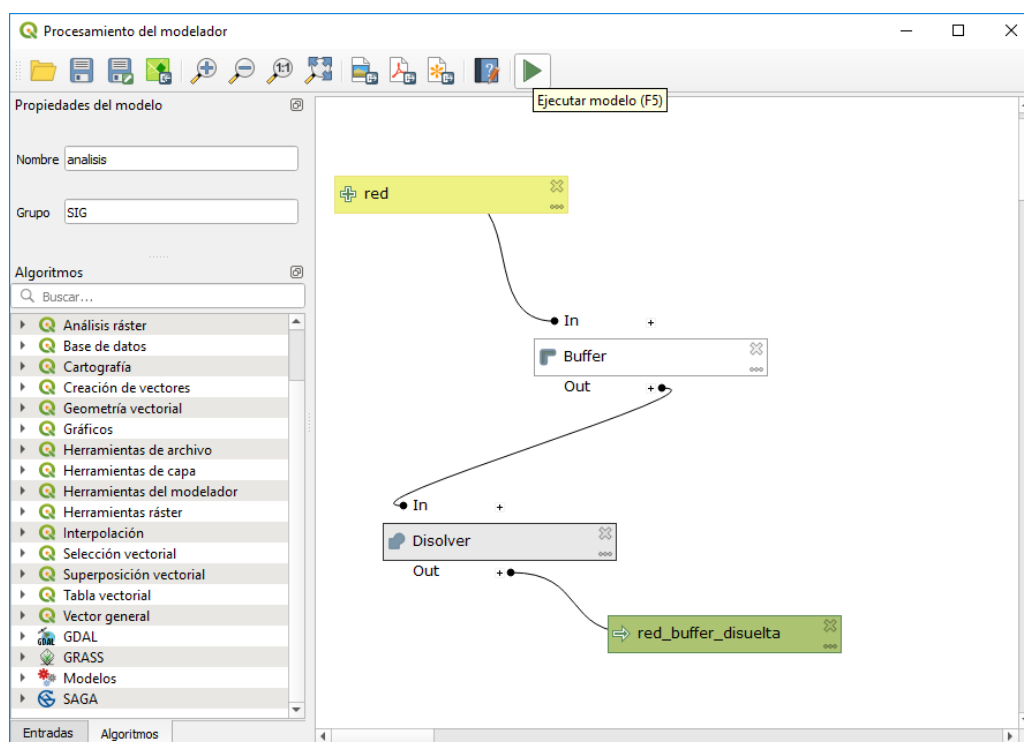
Figura 22: Definición de la operación *Disolver*.

Figura 23: Representación gráfica de las operaciones.

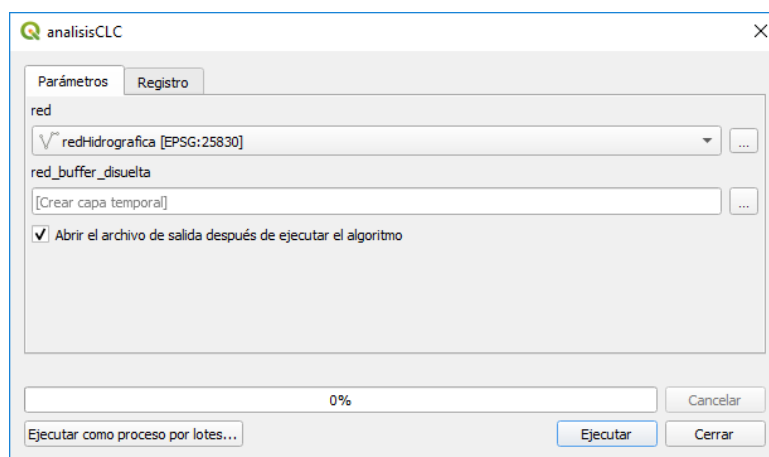


Figura 24: Paso de parámetros para la ejecución del modelo.

modelo, pulsando sobre el botón [«Ejecutar modelo»] (figura 23), podemos ejecutar los algoritmos definidos sobre los parámetros y datos considerados.

Al ejecutar un modelo, se abre una ventana en la que se pueden instanciar los parámetros definidos y también acabar de concretar el formato de la salida de las operaciones (figura 24).

El resultado de la ejecución de este modelo se muestra en la figura 25, idéntico a los obtenidos anteriormente mediante otros medios. En particular, en la parte inferior derecha de la ventana «Caja de herramientas de Procesos» de la figura 25, se puede apreciar que el modelo aparece con el nombre que le asignamos, bajo el grupo indicado en el momento de su creación. Podemos utilizarlo cada vez que lo necesitemos, aplicando las mismas operaciones a los datos que pasemos como parámetro en el momento de su ejecución.

7. Realiza una consulta similar a la de este apartado sobre los datos de tu municipio.

8. Realiza la consulta del apartado 2.2 mediante el modelador de procesos de *QGIS*.

## 5. Consola de *Python* en *QGIS*

Se puede usar *Python* desde *QGIS* para realizar operaciones de análisis sobre capas de información geográfica. En [https://docs.qgis.org/testing/en/docs/user\\_manual/processing/console.html](https://docs.qgis.org/testing/en/docs/user_manual/processing/console.html) se puede encontrar la información en la que se basa esta parte de la actividad.

Accedemos a la consola pulsando sobre [«Complementos», «Consola de Python»] (figura 26)). Se abre una ventana donde podemos introducir instrucciones y observar el resultado. En la versión 3 de *QGIS* se trabaja sobre la versión 3 de *Python*.

En primer lugar, debemos importar las funciones de proceso disponibles. Esta operación se realiza mediante la instrucción de la figura 27. Al ejecutarla en la consola no muestra ningún resultado visible (figura 28).

En la figura 29, se muestran las instrucciones necesarias para listar las funciones de proceso disponibles<sup>1</sup>. El resultado de la ejecución es una lista de más de novecientas líneas; parte de este resultado se puede ver en la figura 30.

En la misma consola, podemos obtener ayuda sobre cualquier algoritmo disponible. En la figura 31 se muestra un ejemplo sobre cómo obtener ayuda sobre el algoritmo `native:buffer`. En la figura 32, se muestra parte del resultado obtenido.

Por último, en la figura 33 se muestra la instrucción para ejecutar el algoritmo `native:buffer` y cargar el resultado en *QGIS*; por usar el parámetro `'DISSOLVE': True` combina el resultado obtenido, como se ha hecho en los apartados anteriores.

El resultado de la ejecución se muestra en la figura 34. Adicionalmente, el resultado se ha guardado como una nueva capa en un archivo. El resultado se muestra en *QGIS* porque se ha usado la función

<sup>1</sup>Una vez introducidas en la consola, para ejecutarlas es necesario pulsar «Enter» dos veces.

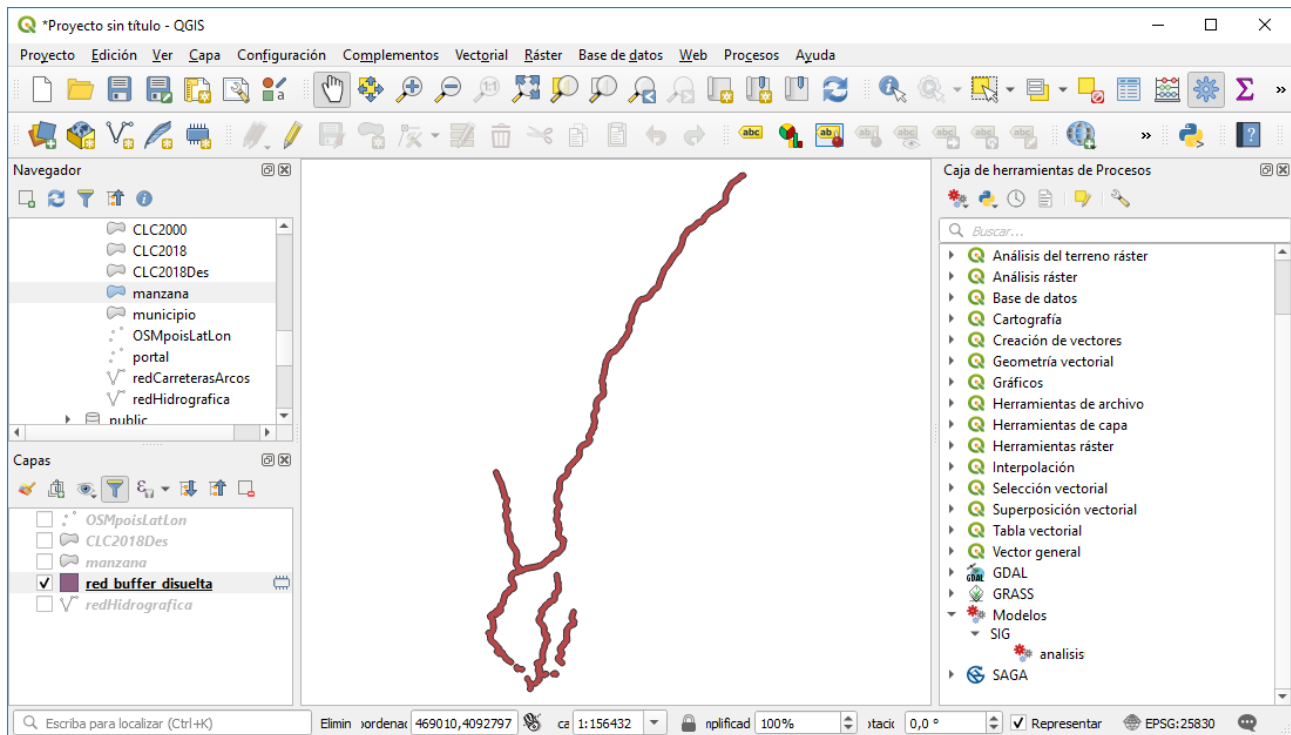


Figura 25: Resultado de la ejecución.

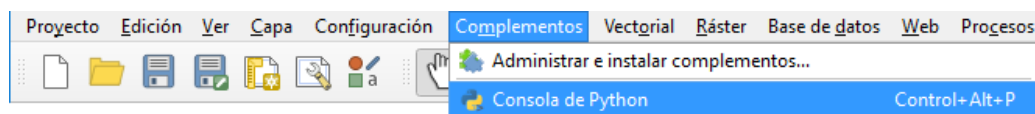


Figura 26: Acceso a la Consola de Python.

```
import processing
```

Figura 27: Instrucción para importar las funciones de proceso.

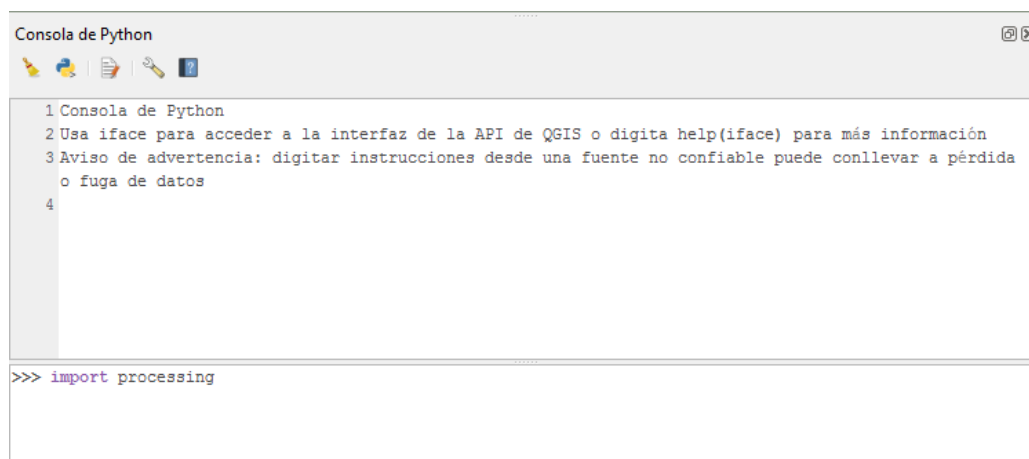
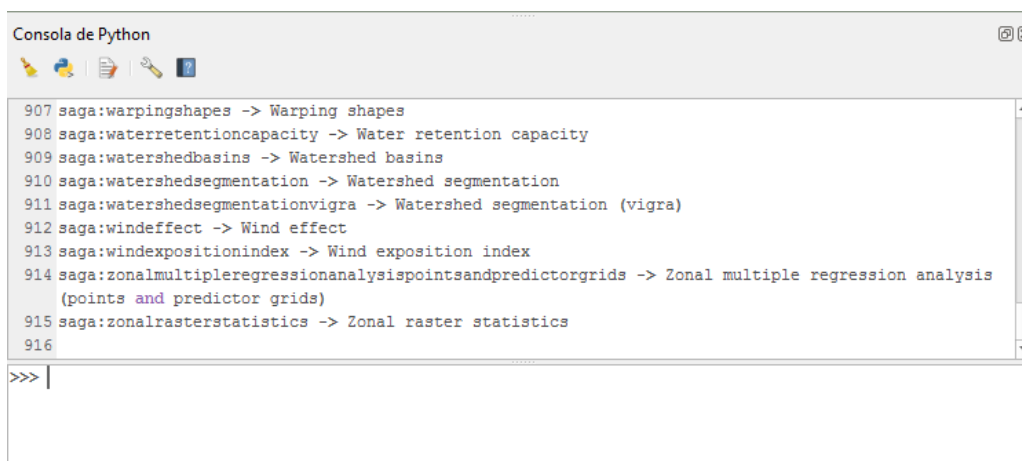


Figura 28: Ejecución de la instrucción para importar las funciones de proceso.



```
for alg in QgsApplication.processingRegistry().algorithms():  
    print(alg.id(), "->", alg.displayName())
```

Figura 29: Instrucciones para listar las funciones de proceso disponibles.

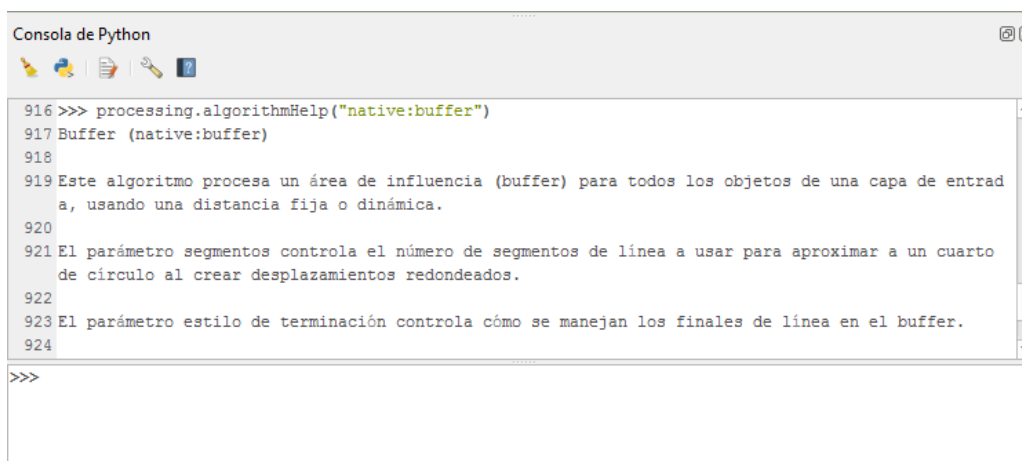


```
Consola de Python  
907 saga:warpingshapes -> Warping shapes  
908 saga:waterretentioncapacity -> Water retention capacity  
909 saga:watershedbasins -> Watershed basins  
910 saga:watershedsegmentation -> Watershed segmentation  
911 saga:watershedsegmentationvibra -> Watershed segmentation (vibra)  
912 saga:windeffect -> Wind effect  
913 saga:windexpositionindex -> Wind exposition index  
914 saga:zonalmultipleregressionanalysispointsandpredictorgrids -> Zonal multiple regression analysis  
    (points and predictor grids)  
915 saga:zonalrasterstatistics -> Zonal raster statistics  
916  
>>> |
```

Figura 30: Resultado de la ejecución de las instrucciones para listar las funciones de proceso disponibles.

```
processing.algorithmHelp("native:buffer")
```

Figura 31: Instrucción para obtener ayuda sobre un algoritmo.



```
Consola de Python  
916 >>> processing.algorithmHelp("native:buffer")  
917 Buffer (native:buffer)  
918  
919 Este algoritmo procesa un área de influencia (buffer) para todos los objetos de una capa de entrada,  
    usando una distancia fija o dinámica.  
920  
921 El parámetro segmentos controla el número de segmentos de línea a usar para aproximar a un cuarto  
    de círculo al crear desplazamientos redondeados.  
922  
923 El parámetro estilo de terminación controla cómo se manejan los finales de línea en el buffer.  
924  
>>>
```

Figura 32: Obtener ayuda sobre un algoritmo.

```
processing.runAndLoadResults("native:buffer",  
    'INPUT': 'Z:/sigPr4/2capasTematicas/redHidrografica.shp',  
    'DISTANCE': 100.0,  
    'SEGMENTS': 10,  
    'DISSOLVE': True,  
    'END_CAP_STYLE': 0,  
    'JOIN_STYLE': 0,  
    'MITER_LIMIT': 10,  
    'OUTPUT': 'Z:/sigPr6/redHidrograficaBuffers.shp')
```

Figura 33: Ejecución y carga del resultado de un algoritmo sobre una capa.

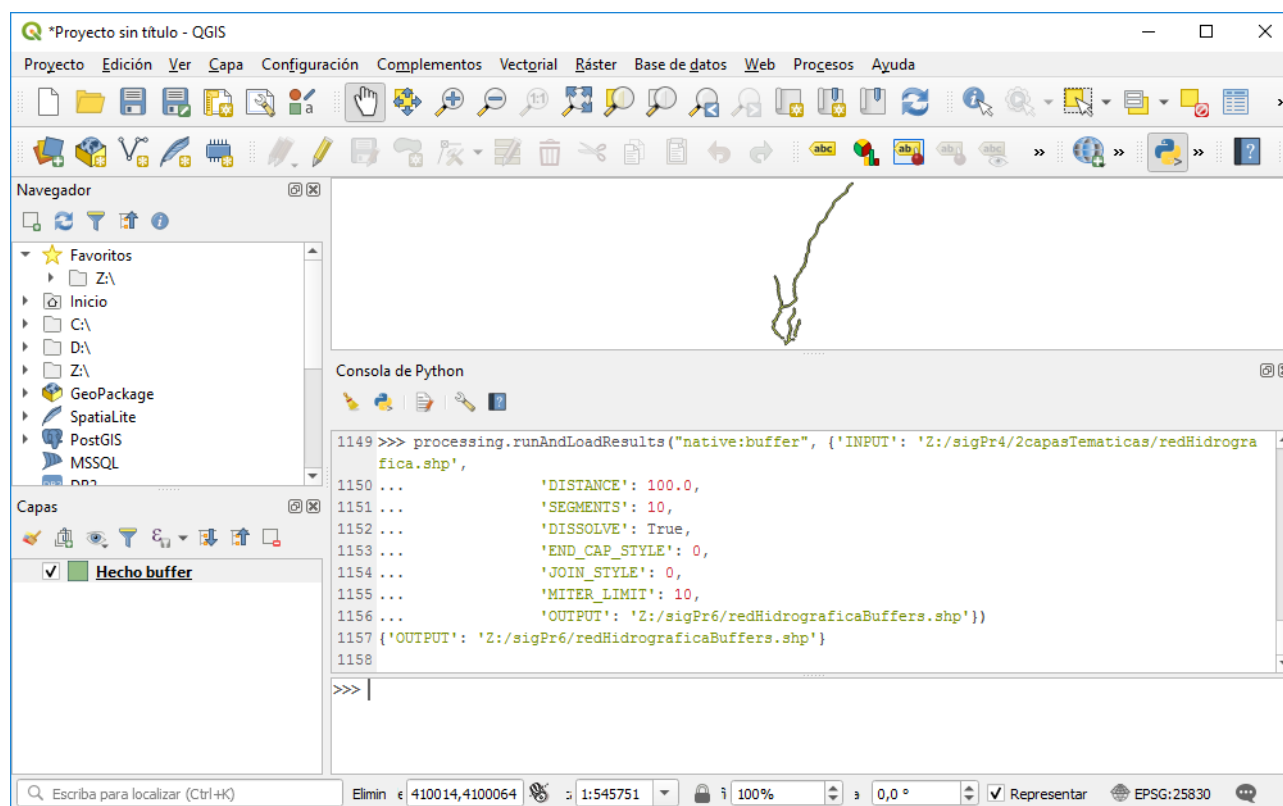


Figura 34: Resultado de la ejecución de un algoritmo.

```
library(sf)

dsn = paste(
  "PG:dbname='Municipio'",
  "host='localhost'",
  "port='5432'",
  "user='postgres'",
  "password='postgres'",
  sep = " "
)

query = paste(
  'SELECT ST_Union(ST_Buffer(geom, 100)) as zona_rio',
  'FROM carga."redHidrografica"',
  sep = " "
)

resultado = st_read(dsn, "nuevaCapa", query = query)
plot(st_geometry(resultado))
```

Figura 35: Consulta SQL desde *R*.

`processing.runAndLoadResults` en lugar de `processing.run` que se habría limitado a ejecutar el algoritmo de análisis indicado.

9. Realiza una consulta similar a la de este apartado sobre los datos de tu municipio.
10. Actividad opcional: Realiza la consulta del apartado 2.2 mediante la *Consola de Python* en *QGIS*.

## 6. Consultas sobre *PostGIS* desde *R*

En la figura 35, se muestran las instrucciones para realizar consultas SQL desde *R* sobre *PostGIS*. En una variable se define la cadena de conexión a la BD (para que sea más legible se puede usar la función `paste`, a la que se le pasan cadenas de caracteres y el separador usado para unirlos). La función `st_read` se puede usar para lanzar consultas sobre la BD. El resultado es una capa de información geográfica.

Hay varias formas para integrar *R* y *QGIS*, por ejemplo:

- El complemento *Processing R Provider* permite integrar *R* en *QGIS*.
- Se puede usar *QGIS* desde *R* con el paquete `RQGIS`<sup>2</sup>.

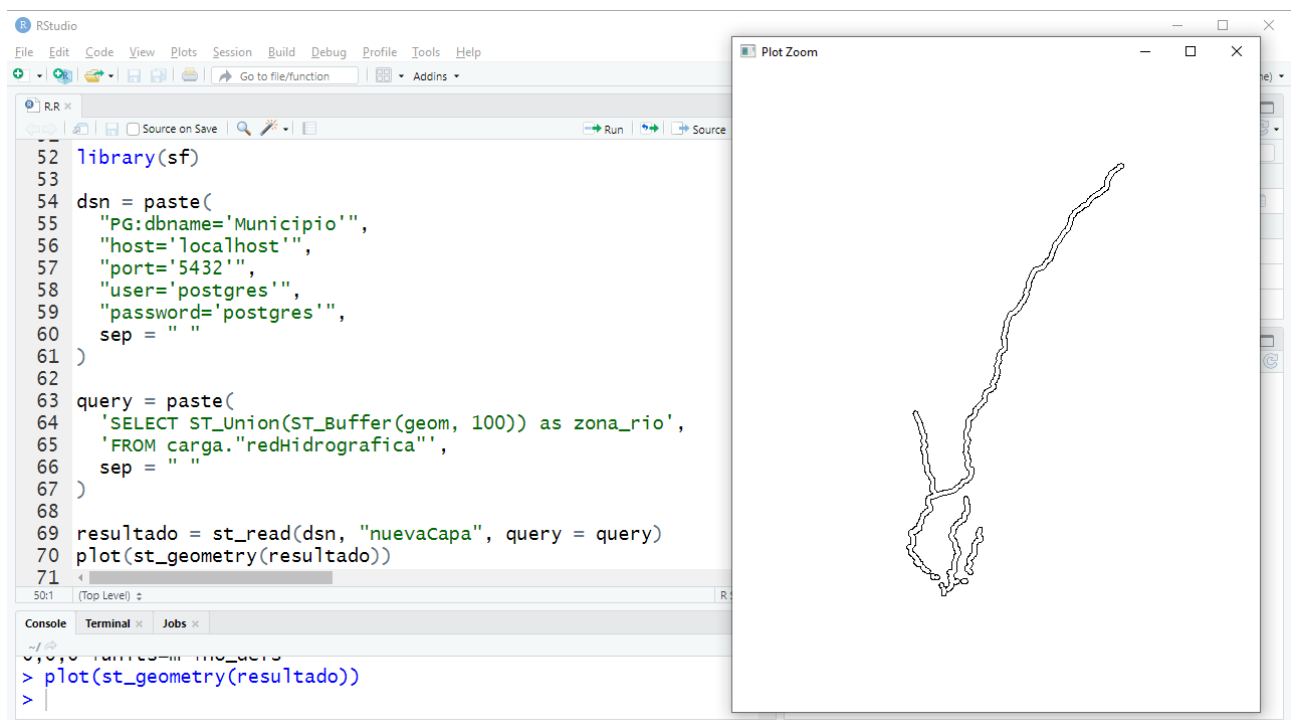
En este caso, no vamos a usar ninguna de esas posibilidades, vamos a usar *R* desde *RStudio* con el paquete `sf`. En la figura 36, se muestra el resultado de la ejecución de la consulta de la figura 35 en *RStudio*.

11. Realiza una consulta similar a la de este apartado sobre los datos de tu municipio.
12. Realiza la consulta del apartado 2.2 mediante SQL desde *R*.

## 7. Análisis mediante funciones de paquetes de *R*

Podemos usar la BD de *PostGIS* simplemente para leer los datos de las capas con las que vamos a trabajar y, a continuación, aplicar sobre ellas las funciones que necesitemos para realizar el análisis.

<sup>2</sup><https://cran.r-project.org/web/packages/RQGIS/index.html>

Figura 36: Resultado de la consulta SQL desde *R*.

```
library(sf)

dsn = paste(
  "PG:dbname='Municipio'",
  "host='localhost'",
  "port='5432'",
  "user='postgres'",
  "password='postgres'",
  sep = " "
)

redHidro <- st_read(dsn, "carga.redHidrografica")

resultado2 <- st_union(st_buffer(redHidro, dist = 100))

plot(st_geometry(resultado2))
```

Figura 37: Consulta mediante funciones en *R*.

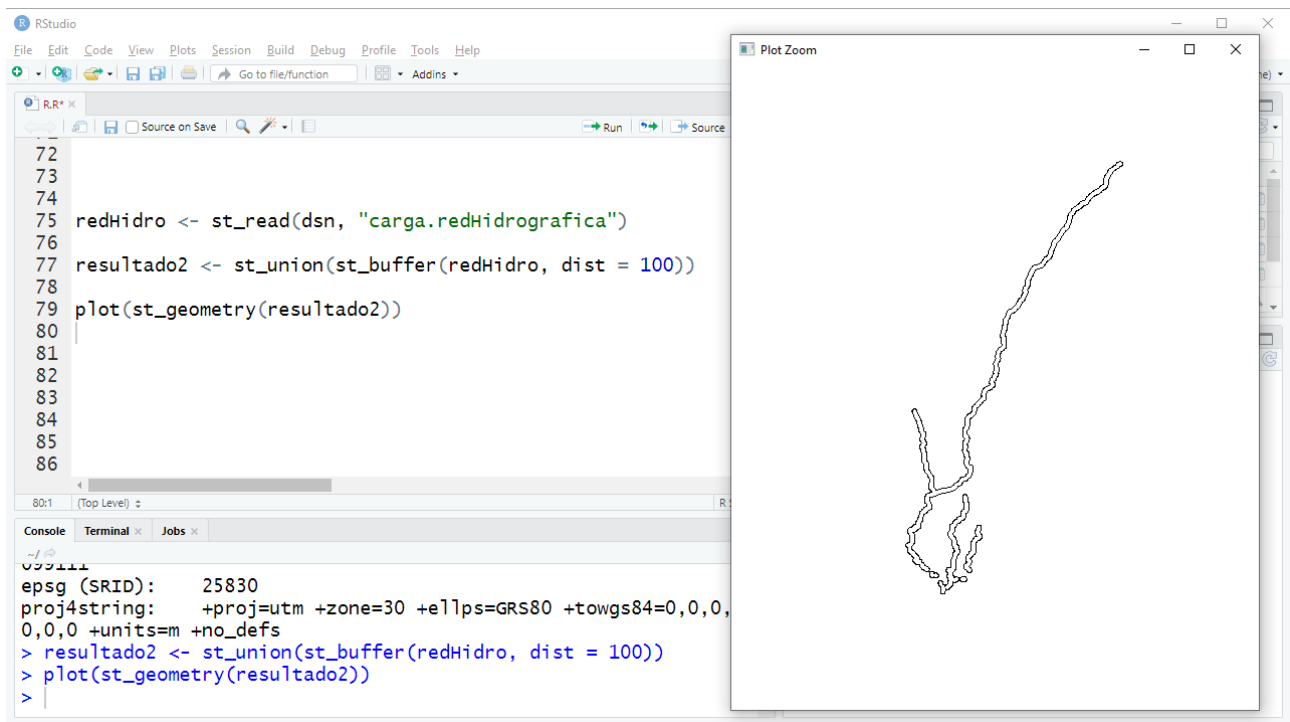


Figura 38: Resultado de la consulta mediante funciones en *R*.

En la figura 37, en este caso, se usa la función `st_read` para leer una capa de la BD. A continuación, aplicamos las funciones de análisis del paquete `sf`.

El resultado obtenido en *RStudio* se muestra en la figura 38. Podemos observar que coincide con el resultado de la figura 36, obtenido mediante SQL.

13. Realiza una consulta similar a la de este apartado sobre los datos de tu municipio.
14. Realiza la consulta del apartado 2.2 mediante funciones en *R*.

## Bibliografía

[Mar15] Angel Marquez. *PostGIS Essentials*. Packt Publishing, 2015.