



Mis notas con el QGIS

Consultas y aspectos básicos relacionados con el QGIS

Jose Ignacio González Gómez

Departamento de Economía Financiera y Contabilidad - Universidad de La Laguna

www.jggomez.eu

INDICE

1	Cuestiones generales del QGIS.....	3
1.1	Como añadir capas vectoriales y fuentes de información de datos (dbf, csv, xlsx, etc)	3
1.2	Fuentes	3
2	Adjuntar a un fichero vectorial (SVG) información contenida en otra fuente de datos (dbf, xlsx, etc..).....	3
2.1	Objetivo	3
2.2	Cómo unir tablas con QGIS (join)	3
2.2.1	Datos de partida	3
2.2.2	Identificar los campos de unión para la relación.....	4
2.2.3	Creación de la unión (join).....	4
2.2.4	Opción Custom Field Name Prefix	5
2.2.5	Comprobando el resultado de la unión.....	5
2.3	Convirtiendo la unión virtual en una nueva capa vectorial shapefile.....	6
2.4	Resultado final.....	7
2.5	Sobre las capas virtuales	8
2.5.1	Origen del problema.....	8
2.5.2	Que son las capas virtuales.....	8
2.5.3	Como generar una capa virtual, base SQL	8
3	Mi caso con las capas virtuales	10
3.1	Pasos básico.....	10
3.1.1	Cargo el shp de municipios canarios	10
3.1.2	Cargo fichero de datos población y genero la unión	10
3.1.3	Me exige convertir en capa virtual sino no me lo permite.....	12
3.2	Paso alternativo, crear capa virtual con los comandos Sql	12
3.3	Editar las propiedades de la capa virtual, SQL de la consulta.	13
3.4	SQLite básico para consultas QGIS y creación de capas virtuales	15

3.4.1	Consideraciones generales	15
3.4.2	Aspectos básicos.....	15
3.4.3	Particularidades del uso del lenguaje en QGIS	15
3.4.4	Comandos básicos de consultas (query) con SQLite, SELECT.....	15
3.5	<i>Fuentes</i>	17
4	Sobre los ficheros gpkg y QGIS	18
4.1	<i>Características generales y básicas</i>	18
4.2	<i>Cómo crear un GeoPackage desde cero</i>	18
4.3	<i>Importar datos a un GeoPackage</i>	19
4.4	<i>Cómo conectar un GeoPackage con QGIS</i>	20
4.5	<i>Crear un GeoPackage basado en shp y una nueva consulta</i>	20
4.6	<i>Fuentes</i>	21
5	Otros	21
5.1	<i>Complementos para QGIS</i>	21
5.2	<i>20 Trucos de QGIS que quizá no conocías</i>	21
5.3	<i>10 opciones de QGIS 3 imprescindibles, revisar importante</i>	21

1 Cuestiones generales del QGIS

1.1 Como añadir capas vectoriales y fuentes de información de datos (dbf, csv, xlsm.. etc)



La forma más directa para añadir datos a nuestro proyecto Qgis es a través del botón "**Añadir capa vectorial**".

En **QGIS no hay un botón exclusivamente para incorporar tablas al proyecto**, para ello se utiliza también el botón "Añadir capa vectorial" y en el selector de tipo de archivo seleccionar Todos los archivos (*), de esta forma QGIS reconocerá el formato .dbf

1.2 Fuentes

2 Adjuntar a un fichero vectorial (SVG) información contenida en otra fuente de datos (dbf, xls, etc..)

2.1 Objetivo

A menudo queremos representar en un mapa algún archivo vectorial pero nos encontramos con que su tabla de atributos es escasa en datos o no contiene aquellos que necesitamos. Y también a menudo sucede que esa información que necesitamos la tenemos en una base de datos aparte. Es entonces cuando debemos fusionar ambos archivos de alguna forma que nos permita disponer de toda la información en aquel archivo con el que deseamos trabajar. En este caso práctico vamos a ver lo que para mí es uno de los procesos más básicos e imprescindibles a la hora de crear nuestros mapas: enriquecer la información de un archivo vectorial con una tabla de datos externa, lo que nos ampliará las posibilidades de representación.

El caso frecuente es que contamos un fichero vectorial (SVG) y al cual le queremos vincular o añadir información disponible contenida en otro fichero con formato estándar del tipo dbf, Excel, texto, csv, etc.

2.2 Cómo unir tablas con QGIS (join)

2.2.1 Datos de partida

Como señalamos en la introducción este es el objetivo central de este apartado, unir tablas de información por ejemplo el dbf asociado al fichero vectorial con otra fuente de datos contenida en un fichero tipo dbf, xls, csv, etc., así esta operación sirve para explotar la información realizando una unión de tablas con la tabla de atributos de un shapefile. También es posible enlazar con datos de PostGIS.

Para una mejor comprensión planteamos el siguiente caso, disponemos de una fuente de datos externa en formato dbf (da igual, también podría ser txt o csv) con datos municipales y procederemos a realizar una unión ("Join") de ésta con la tabla de atributos original de una capa con la información geométrica de los municipios fichero shapefile y su asociado dbf.

Partimos por tanto que ya hemos añadido a nuestro proyecto el fichero shapefile y el de datos complementario.

2.2.2 Identificar los campos de unión para la relación

Vista la necesidad y seleccionando los ficheros a relacionar, antes de nada, necesitamos analizar el contenido de los mismos, campos disponibles lo que nos permitirá elegir aquellos campos de ambas tablas que permita relacionar adecuadamente ambas fuentes de datos.

Es decir, esta operación de unión de dos tablas de atributos requiere conocer las características de las tablas que se desean unir, y la dirección de la unión, es decir:

- Cuál es el campo que posee valores similares en cada tabla; es decir, el campo clave común que permitirá asociar los registros de una y otra tabla garantizando su integridad.
- Cuál es la tabla que va a aumentar el número de campos gracias a la anexión de otra tabla.

Ambas cuestiones son claves porque si no existe un campo con valores comunes en cada tabla, estas no pueden unirse.

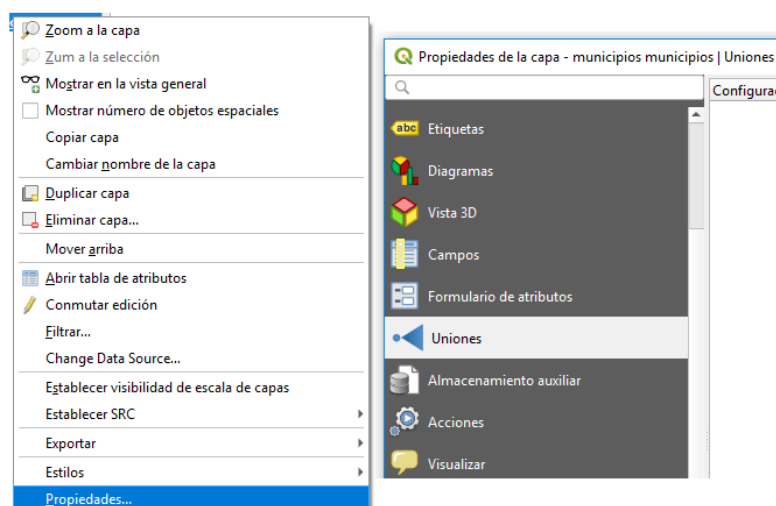
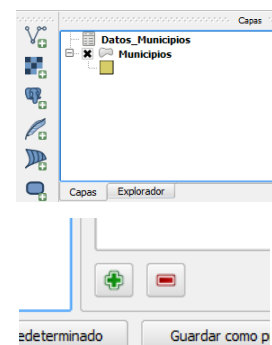
ES IMPRESCINDIBLE PARA LA UNION DE DOS TABLAS DE DATOS QUE AMBAS COMPARTAN UN CAMPO CON VALORES COMUNES

Para el caso que nos ocupa el campo común que nos permitirá realizar la unión de las dos tablas será el código INE

2.2.3 Creación de la unión (join)

Partiendo de los ficheros cargados en nuestro proyecto tanto el fichero shapefile y el de datos complementario, vamos a proceder a unir la información accediendo a las propiedades de la capa Municipios y, en Propiedades de la capa, seleccionamos el apartado Uniones ("Joins").

Para establecer una Unión de tablas ("Join"), presione el botón de adición (cruz verde) en la parte inferior de la ventana de Propiedades de la capa.



Aparecerá en pantalla la ventana Añadir unión vectorial (“Add vector join”).... en la que se deberán de introducir los parámetros para llevar a cabo dicha unión.

- Unir capa (“Join layer”): Datos_Municipios.dbf
- Unir campo (“Join field”): CODIGO_INE
- Campo objetivo (“Target field”): CODIGO_INE

x en Cachear capa de unión en memoria virtual (“Cache join layer in virtual memory”)

x en Crear índice de atributos en el campo unión (“Create attribute index on join field”), éste índice se utiliza para acelerar las búsquedas.

Para completar la unión de tablas aplicamos y aceptamos los cambios.

2.2.4 Opción Custom Field Name Prefix

Esta opción es importante porque nos permite establecer un prefijo a los campos anexados a la tabla original permitiendo identificar los mismos con el prefijo definido.

2.2.5 Comprobando el resultado de la unión

Abriendo la tabla de atributos de la capa se puede observar que la Unión (“Join”) se ha realizado satisfactoriamente (la denominación de los campos nuevos aparece en forma compuesta por el nombre de la tabla origen y el nombre que éstos poseían en ella), a no ser que usemos la opción **Custom Field Name Prefix** vista anteriormente.

	codigo_in...	datos_municipio...	datos_municipio...	datos_municipio...	datos_municipio...	datos_municipio...	datos_municipio...	datos_municipio...	datos_municipio...
1	111	Alcala de los Ga...	70	22915	0	0	5596	2710	288
2	112	Alcala del Valle	155	1724	0	0	5292	2283	300
3	113	Algar	87	108	0	0	1755	812	94
4	114	Algeciras	120	1389	0	0	105066	53238	5182
5	115	Algodonales	118	9453	0	0	5612	2746	286
6	116	Arco de la Fro...	67	9569	0	0	28104	14036	1406
7	117	Barbate	64	2381	0	0	21914	10928	1096
8	118	Barrios (Los)	131	18696	0	0	17283	8605	867
9	1123	Medina-Sidonia	46	10577	0	0	10775	5190	558
10	119	Benaocaz	120	4338	0	0	662	318	34
11	1110	Bornos	79	0	0	0	7979	4018	396
12	1111	Rosne (El)	101	1647	0	0	1868	912	95

Ahora nuestra capa vectorial de polígonos cuenta en su tabla de atributos, además de con sus datos originales, con los nuevos campos que le hemos añadido desde la tabla de datos.

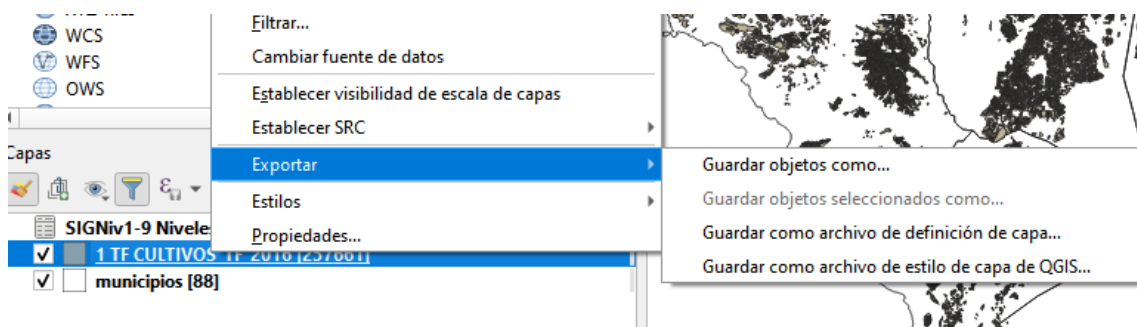
ID	NOMBRE	CODIGOINE	JURISDICC	PROVINCIA	ePoblacion P	sesDePoblacion	eDePoblab	sesDePoblab	blacion P	NombreCategor_Municipios_EstadosDePoblacion MUNICIPIOS_ayr_AUTEX	
878	A Arnoia	32003	32003	32	1040	2069	4436	20355	Outero Cruz	92	76
128	A Baña	15007	15007	15	3698	9818	9602	5324	San Vicente	281	271
889	A Boia	32014	32014	32	1339	3490	2222	25540	A Boia	15	515
5280	A Caliza	36009	36009	36	5230	10603	04	56015	A Caliza	1845	570
2140	A Capela	15018	15018	15	1334	5759	5996	39561	As Neves	204	429
139	A Coruña	15030	15030	15	243870	3783	0149	51300	A Coruña	213904	21
287	A Estrada	36017	36017	36	21025	28072	216	10326	A Estrada	8388	293
229	A Fonsagrada	27018	27018	27	3838	43842	18	121013	A Fonsagrada	1123	949
293	A Guarda	36023	36023	36	10193	2050	3865	24895	A Guarda	6067	57
909	A Gudixá	32034	32034	32	1392	17141	5879	62691	A Gudixá	737	974
332	A Illa de Arousa	36901	36901	36	4956	691	5039	45673	Arousa	2034	6

Columnas o campos nuevos añadidos (en amarillo) con el nombre de campo predefinido en el prefijo (verde).

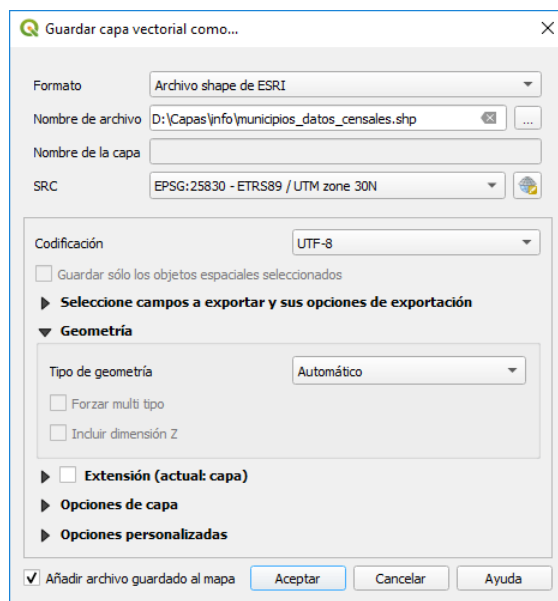
2.3 Convirtiendo la unión virtual en una nueva capa vectorial shapefile

La unión de datos hasta este momento es sólo virtual y se pierde si cerramos alguna tabla, por lo que conviene guardar la capa vectorial como un nuevo archivo shp que mantenga la nueva estructura de la tabla de datos de forma permanente. Simplemente botón derecho sobre la capa y Guardar como...

Hasta el momento se trata de una “unión virtual” entre tablas, y se pierde si cerramos alguna tabla, por lo que conviene guardar la capa vectorial como un nuevo archivo shp que mantenga la nueva estructura de la tabla de datos de forma permanente, es decir, para que la unión sea definitiva es necesario guardar la nueva capa generada. Para ello, con un clic del botón secundario del ratón sobre la capa desplegamos sus opciones y seleccionamos *Exportar/Guardar objetos como...* (“Save as”).



Por último, en Guardar capa vectorial como.... indicamos un directorio en el que ésta nueva capa quedará guardada, así como el Sistema de Coordenadas (opcional) y si deseamos aplicarle algún tipo de simbología.



Así en el directorio elegido obtendremos la relación de archivos correspondientes del documento vectorial integrando además en el dbf la información adjunta requerida

- El hierro con datos.cpg
- El hierro con datos.dbf
- El hierro con datos.prj
- El hierro con datos.qpj
- El hierro con datos.shp
- El hierro con datos.shx

2.4 Resultado final

¿Qué hemos conseguido con esto? Que donde había una tabla de datos con escasa información que apenas podíamos clasificar por ningún dato relevante, tenemos ahora algunos de los datos más importantes para cualquier simbología que queramos aplicar a los municipios: población, superficie, perímetro, altitud...

Jugando ahora con nuestra nueva tabla de datos integrada en el vectorial y el magnífico QGIS nos será muy fácil elaborar interesantes mapas sobre la base de los municipios españoles.

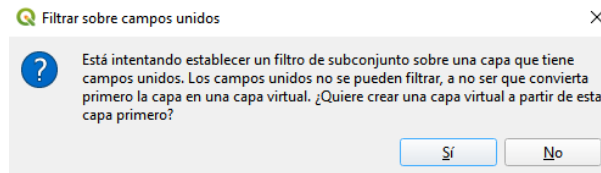
De esta forma podemos filtrar los datos y trabajar con ellos.

2.5 Sobre las capas virtuales

2.5.1 Origen del problema

Como vimos anteriormente cuando creamos una unión de dos ficheros se genera una conexión temporal que para poder filtrar con los campos nuevos combinados hemos tenido que genera un nuevo fichero shapefile.

Pero si quisiéramos aplicar filtros sobre esta conexión virtual la aplicación nos lo permite y nos exige crear primero



2.5.2 Que son las capas virtuales

Las capas virtuales son un tipo especial de capa vectorial que permite definirse como resultado de una consulta avanzada, utilizando lenguaje SQL, es decir son el resultado de una consulta avanzada realizada sobre otra capa vinculada o no a una fuente de datos

Las capas virtuales sirven para mostrar vistas de otras capas, como filtros de tabla y filtros espaciales, siempre utilizando lenguaje SQL; es decir, no tienen datos por sí mismas, sino que están calculadas sobre otras capas.

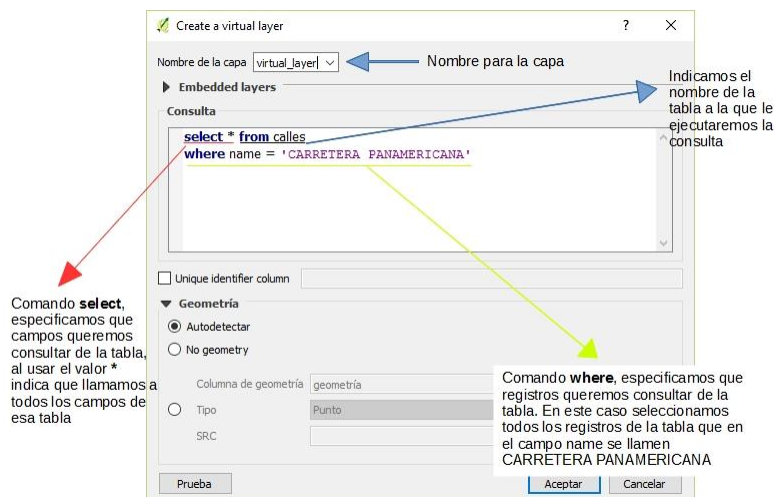
Por tanto, las capas virtuales nos permitirá crear tablas y capas a través de consultas SQL, incluso creando tablas dinámicas que se actualizan automáticamente al tiempo que se modifica la tabla a la que están vinculadas.

2.5.3 Como generar una capa virtual, base SQL

<http://sig-uyendo.blogspot.com/2016/03/agregar-layer-virtual-nueva-opcion-en.html>

También podemos probar si la consulta que estamos creando es válida con el botón Probar y nos mostrara la capa resultante igual a una de las que tenemos cargada en el panel de capas, pero que solo muestre ciertas entidades con un valor específico en alguno de sus campos.

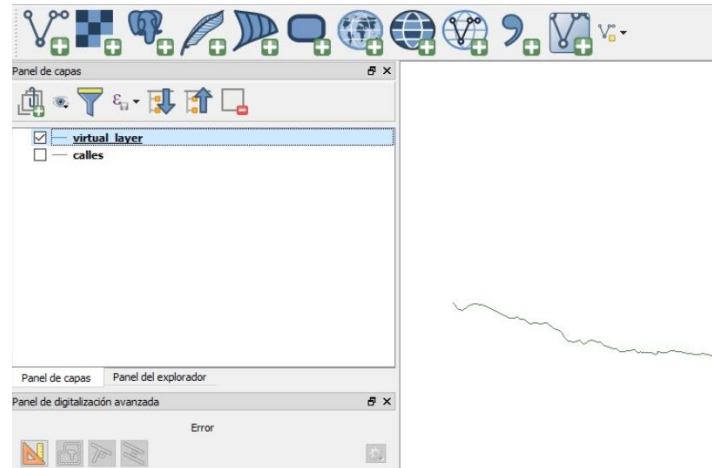
Para su creación es necesario realizar consultas básicas con el lenguaje SQL, para nuestro ejemplo hemos cargado una capa llamada calles. Y seleccionaremos solo las que formen parte de la carretera Panamericana.



En la ilustración anterior hemos desglosado la consulta en sus partes más relevantes
El query básico es:

- **select * from calles**
- **where name = 'CARRETERA PANAMERICANA';**

El resultado es el siguiente:



El cuadro de diálogo permite especificar una consulta SQL. La consulta puede usar el nombre (o id) de las capas vectoriales existentes, así como los nombres de campo de estas capas.

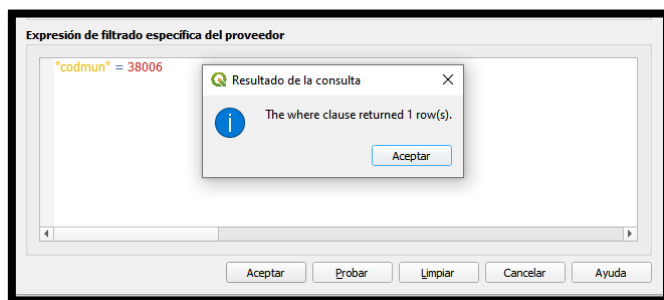
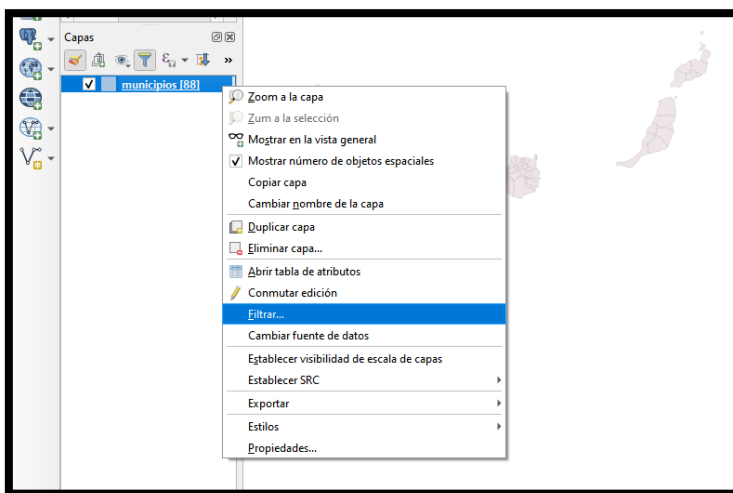
Por ejemplo, si tiene una capa llamada regions, puede crear una nueva capa virtual con una consulta SQL como `SELECT * FROM regions WHERE id < 100`. La consulta SQL se ejecutará, sea cual sea el proveedor subyacente de la regionscapa e incluso si este proveedor no admite directamente consultas SQL.

Las uniones y consultas complejas también se pueden crear simplemente usando directamente los nombres de las capas que se van a unir.

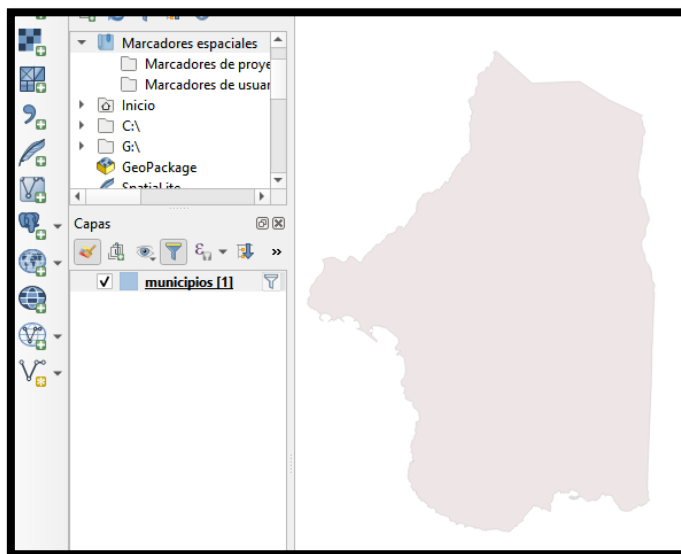
3 Mi caso con las capas virtuales

3.1 Pasos básico

3.1.1 Cargo el shp de municipios canarios



Como se puede ver me permite acceder al filtrado y seleccionar un solo municipio por ejemplo Arona.



3.1.2 Cargo fichero de datos población y genero la unión

A continuación cargo el fichero de datos relacionada con la población de los municipios de canarias periodo 2018 y 2019

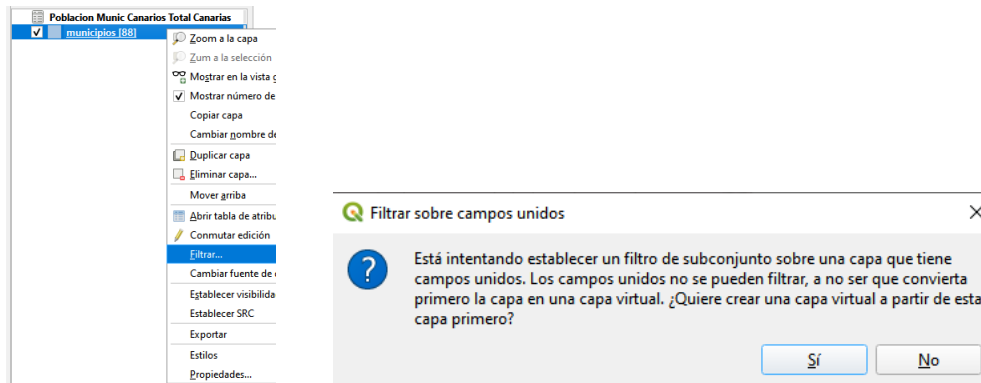
	Codigo INE MUN	Nombre	Año Poblacion	Poblacion
1	38012	Fasnia	2019	2786
2	38011	Candelaria	2019	27985
3	38014	Fuencaliente de...	2019	1722
4	38013	Frontera	2019	4093

Pasamos a crear la conexión entre el shp y el fichero de datos Excel cargados

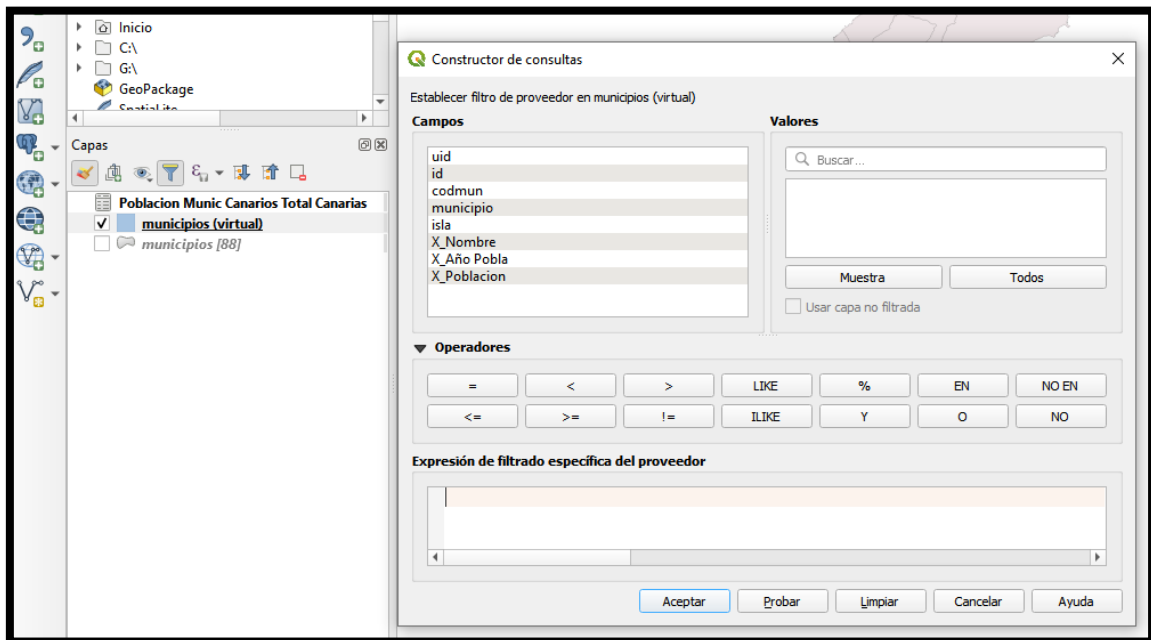
El resultado es que dispongo los datos de las dos tablas relacionadas y los nuevos campos disponibles aparecen con el X_ que corresponde al fichero Excel.

	id	codmun	municipio	isla	X_Nombre	X_Año Poblacion	X_Poblacion
57	8	38033	SAN ANDRÉS...	LA PALMA	San Andrés y Sa...	2018	4112
58	86	38034	SAN JUAN DE L...	TENERIFE	San Juan de la ...	2018	4799
59	15	38035	SAN MIGUEL D...	TENERIFE	San Miguel de ...	2018	19672

3.1.3 Me exige convertir en capa virtual sino no me lo permite
 Ahora si quisiera filtrar una vez creada la realcion entre las dos tablas y a pesar que tengo los campos unidos correctamente me sale el siguiente mensaje.

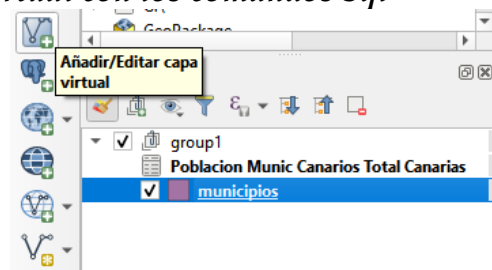


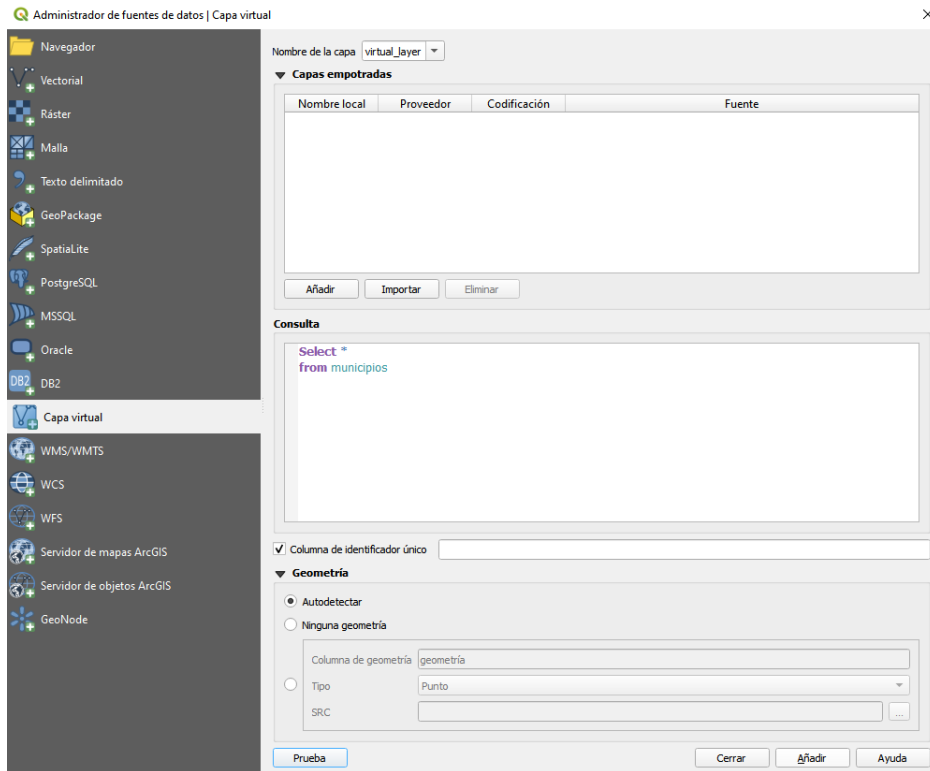
Pero a veces no me sale.....



3.2 Paso alternativo, crear capa virtual con los comandos Sql

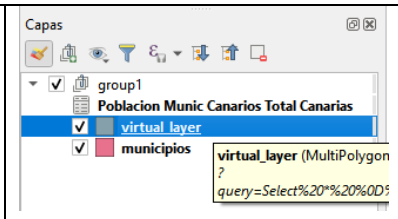
Una vez creada la unión entre la capa principal y los datos en Excel tal y como vimos al principio, podemos crear las capas virtuales con una simple instrucción SQL. Es decir, generada la unión accedemos a la opción añadir capa virtual, con lo que se nos abre el cuadro de dialogo correspondiente para crear con sql la capa.





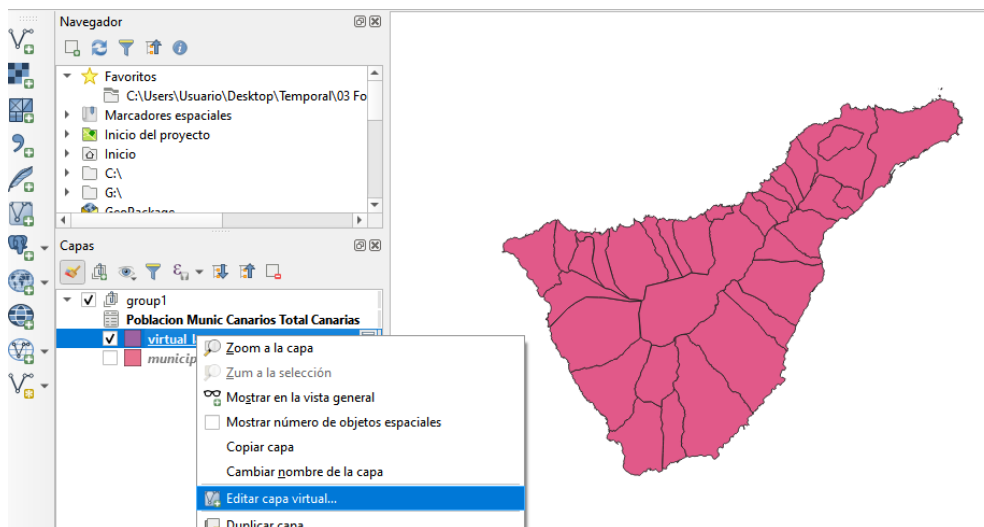
En consulta creamos la siguiente que selecciona todos los campos del shp deseado con la conexión y base de la capa virtual

Select *
from municipios (nombre de la capa shp que contiene la unión)



Y de esta forma se genera la nueva capa que podemos cambiarle el nombre

3.3 Editar las propiedades de la capa virtual, SQL de la consulta. Si quisiera editar las propiedades de la capa virtual basta con



Crear una capa virtual
✕

Nombre de la capa

▼ **Capas empotradas**

Nombre local	Proveedor	Codificación	Fuente

Consulta

```
Select *
from municipios
```

Columna de identificador único

▼ **Geometría**

Autodetectar

Ninguna geometría

Columna de geometría

Tipo

SRC

3.4 SQLite básico para consultas QGIS y creación de capas virtuales

3.4.1 Consideraciones generales

Vamos a dar presentar unas notas básicas relacionadas con el query para generar capas virtuales en Qgis.

Destacar en primer lugar que el lenguaje soportado es SQLite y Spatialite para operar, significa que se puede usar todo el SQL que entiende su instalación local de SQLite.

3.4.2 Aspectos básicos

El cuadro de diálogo permite especificar una consulta SQL. La consulta puede usar el nombre (o id) de las capas vectoriales existentes, así como los nombres de campo de estas capas.

Por ejemplo, si tiene una capa llamada regions, puede crear una nueva capa virtual con una consulta SQL como `SELECT * FROM regions WHERE id < 100`. La consulta SQL se ejecutará, sea cual sea el proveedor subyacente de la regionscapa e incluso si este proveedor no admite directamente consultas SQL.

Las uniones y consultas complejas también se pueden crear simplemente usando directamente los nombres de las capas que se van a unir.

3.4.3 Particularidades del uso del lenguaje en QGIS

Contrariamente a una consulta SQL pura, todos los campos de una consulta de capa virtual deben ser nombrados. No olvide usar la palabra clave para nombrar sus columnas si son el resultado de una llamada de cálculo o función.

El motor de capa virtual intenta determinar el tipo de cada columna de la consulta. Si falla, se busca la primera fila de la consulta para determinar los tipos de columna.

El tipo de una columna en particular se puede especificar directamente en la consulta mediante el uso de algunos comentarios especiales.

La sintaxis es la siguiente: `/*:type*/`. Tiene que colocarse justo después del nombre de una columna. `type` puede ser `int` para enteros, para números real de coma flotante o `text`.

Por ejemplo: `SELECT id+1 as nid /*:int*/ FROM table`

3.4.4 Comandos básicos de consultas (query) con SQLite, SELECT

El comando SELECT es lo básico que necesitamos para hacer consulta a las capas y bases de datos, la sintaxis básica es:

```

1 | SELECT [ ALL | DISTINCT [ ON ( expression [, ...] ) ] ]
2 | * | expression [ [ AS ] output_name ] [, ...]
3 | [ FROM from_item [, ...] ]
4 | [ WHERE condition ]
5 | [ ORDER BY expression [ ASC | DESC ] ]

```

- SELECT: con él indicamos que campos queremos mostrar separándolos por comas, para mostrar todos los campos de una tabla usamos el símbolo `*`.
- FROM: aquí indicamos las tabla o tablas que vamos a consultar, y por lo tanto, obtener los datos.

Ejemplo consultar una tabla

```

1 | SELECT *
2 | FROM cb_language;

```

Ejemplos de instrucciones con consultas:

Selección básica uso de where

- Consultar una tabla:

`select *from bus;` -- el asterisco indica que seleccione todos los campos de la tabla

- Otra consulta
`SELECT * FROM bus WHERE capacidad >80;`--todos los buses con capacidad mayor a 80
- Cantidad de registro de una tabla
`SELECT count(*) FROM bus ;`--total registros de la tabla bus
- Podemos especificar los campos que queremos ver separándolo por comas
`select titulo, autor from libros;`
`select nombre, sueldobasico from empleados;`

Estableciendo condiciones en la consulta Operadores relaciones

Los operadores relaciones son:

- = igual
- <> distinto
- > mayor
- < menor
- >= mayor o igual
- <= menor o igual

- seleccionar los registros cuyo autor sea diferente de 'Borges', para ello usamos la condición:
`select * from libros
where autor<>'Borges';`
- Queremos mostrar los títulos y precios de los libros cuyo precio sea mayor a 20 pesos:
`select titulo, precio
from libros
where precio>20;`
- Queremos seleccionar los libros cuyo precio sea menor o igual a 30:
`select * from libros
where precio<=30;`
- Seleccionamos los registros cuyo precio supere los 20 pesos, sólo el título y precio:
`select titulo, precio
from libros
where precio>20;`

Estableciendo condiciones en la consulta Operadores Lógicos (and - or - not)

- Queremos recuperar todos los registros cuyo autor sea igual a "Borges" y cuyo precio no supere los 20 pesos, para ello necesitamos 2 condiciones:
`select * from libros
where autor='Borges' and
precio<=20;`
- Queremos ver los libros cuyo autor sea "Borges" y/o cuya editorial sea "Planeta":
`select * from libros
where autor='Borges' or
editorial='Planeta';`
- Queremos recuperar los libros que no cumplan la condición dada, por ejemplo, aquellos cuya editorial NO sea "Planeta":
`select * from libros
where not editorial='Planeta';`

- Los paréntesis se usan para encerrar condiciones, para que se evalúen como una sola expresión.

```
select * from libros
where (autor='Borges') or
      (editorial='Paidós' and precio<20);
```

Otro ejemplo similar

```
select*from libros
where (autor='Borges' or editorial='Paidós') and
      (precio<20);
```

Usando el operador Operador: "is null", "is not null"

Se emplea el operador "is null" para recuperar los registros en los cuales esté almacenado el valor "null" en un campo específico:

```
select * from libros
where editorial is null;
```

Para obtener los registros que no contiene "null", se puede emplear "is not null", esto mostrará los registros con valores conocidos:

```
select * from libros where precio is not null;
Seleccionamos los libros que no contiene "null" en "precio":
select * from libros where precio is not null;
```

Operador: "between"

Otro operador relacional es "between", trabaja con intervalos de valores.

Hasta ahora, para recuperar de la tabla "libros" los libros con precio mayor o igual a 20 y menor o igual a 40, usamos 2 condiciones unidas por el operador lógico "and":

```
select * from libros
where precio>=20 and
      precio<=40;
```

Podemos usar "between" y así simplificar la consulta:

```
select * from libros
where precio between 20 and 40;
```

3.5 Fuentes

<https://geomatalibre.wordpress.com/2019/11/21/capas-virtuales-en-qgis/>
<https://mappinggis.com/2013/12/union-de-tablas-con-qgis-2-0-dufour/>
<https://digimapas.blogspot.com/2016/05/qgis-uniendo-tablas-de-datos.html>
<http://sig-uniendo.blogspot.com/2016/03/agregar-layer-virtual-nueva-opcion-en.html>
<https://www.tutorialesprogramacionya.com/sqliteya/detalleconcepto.php?punto=18&codigo=18&inicio=0>

Otras cuestiones

Trucos de Qgis

Fuente:

4 Sobre los ficheros gpkg y QGIS

4.1 Características generales y básicas

El *shapefile* de *ESRI* es un formato de archivo para almacenar datos espaciales vectoriales. Ha existido desde principios de la década de 1990 y aún sigue siendo el formato de intercambio de datos vectorial más comúnmente utilizado, pese al elevado número de desventajas y limitaciones que tiene.

En este apartado, queremos aprovechar y mostrar la alternativa más moderna y recomendable al formato **.shp**, el **GeoPackage**.



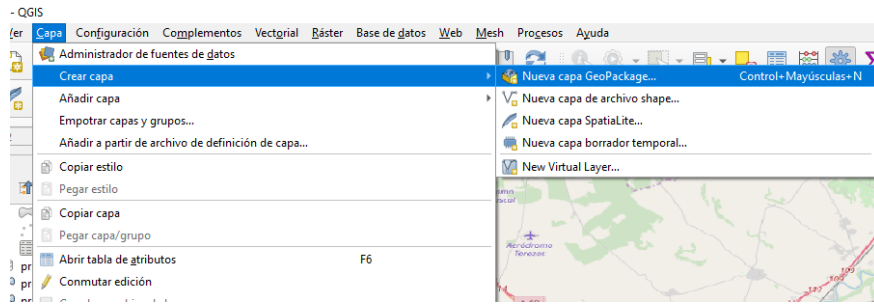
El GeoPackage, es uno de los formatos SIG más prometedores, diseñado para las aplicaciones modernas de hoy en día que se construye sobre una base de datos *SQLite*. Es el estándar de facto en QGIS 3.x

Los principales motivos por lo que es recomendable trabajar con un **GeoPackage** son:

- Es un formato *compacto*, abierto, basado en estándares de la *OGC*, e independiente de plataformas o aplicaciones.
- Se trata de un único archivo **.gpkg**, por lo que es ideal para transferir información geoespacial.
- Soporta datos espaciales *vectoriales* y *ráster*.
- Soporta el **uso directo**.
- Se ha diseñado para almacenar **datos complejos y voluminosos** (hasta 140 TB). Además, los atributos de las geometrías pueden contener nombres muy largos.
- Casi todos los *software GIS* de referencia, y por supuesto *QGIS*, son capaces de **leer** los archivos **GeoPackage**.
- Los **GeoPackages** son particularmente útiles en dispositivos móviles y en entornos de comunicación en los que hay conectividad y ancho de banda limitados.

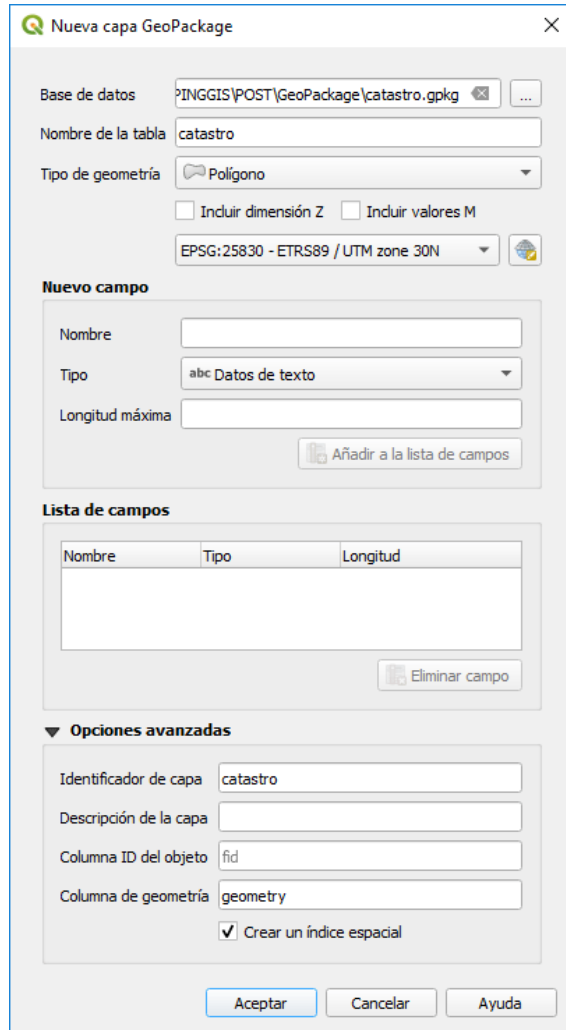
4.2 Cómo crear un GeoPackage desde cero

Abre un proyecto en blanco en *QGIS* y accede al menú *Capa/Crear capa* y selecciona **Nueva capa GeoPackage**.



En la ventana de la herramienta deberemos introducir los siguientes parámetros:

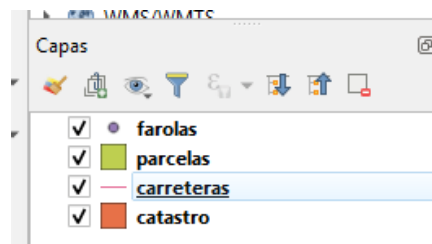
- **Base de datos:** Donde daremos un nombre al nuevo **GeoPackage** que vamos a crear.
- **Nombre de la tabla:** Deja el nombre que ha aparecido por defecto.
- **Tipo de geometría:** Aquí debemos de incluir las propiedades de la primera capa que contendrá el **GeoPackage**, por ejemplo, indica el tipo *polígono*.
- En el siguiente parámetro debes de indicar el *Sistema de Coordenadas* para el archivo **.gpkg**



4.3 Importar datos a un GeoPackage

El siguiente paso, es importar nuestros datos dentro del GeoPackage que acabamos de crear. Esto puede hacerse de una forma sencilla en QGIS, con la técnica de arrastrar y soltar o drag and drop.

Carga en QGIS las capas que desees añadir al GeoPackage.

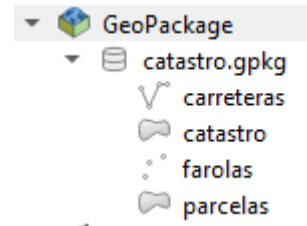


Haz clic sostenido de ratón sobre una de las capas y arrástrala hasta el GeoPackage que hemos creado, cuando te encuentres sobre él aparecerá un símbolo + y, entonces, podrás soltar el clic de ratón.

Aparecerá un mensaje confirmando que la importación se realizó correctamente.

Realiza el mismo proceso con el resto de capas que has cargado a la interfaz de QGIS.

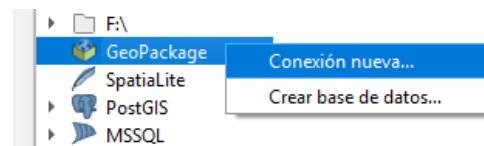
Una vez importada toda la información comprueba que esta se refleja en tu conexión al GeoPackage



4.4 Cómo conectar un GeoPackage con QGIS

El siguiente paso, es crear una conexión al **GeoPackage** que acabamos de crear, para ello, primero, vamos a añadir a la interfaz de *QGIS*, si no lo está ya, el panel del *Navegador*.

Una vez en el *Navegador*, haz clic derecho de ratón sobre el proveedor **GeoPackage** y selecciona *Conexión nueva...*

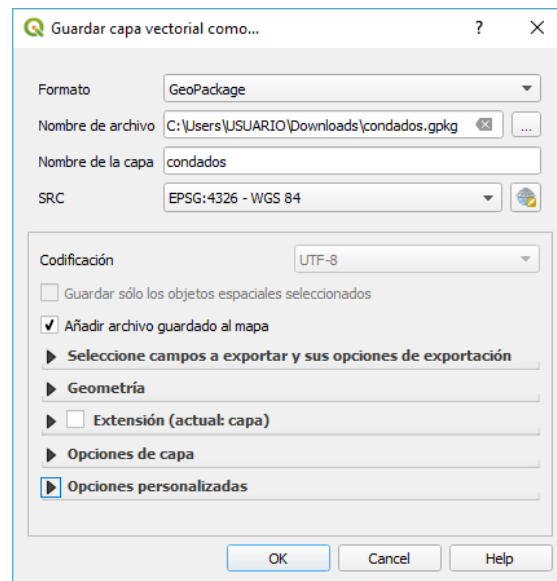


4.5 Crear un GeoPackage basado en shp y una nueva consulta

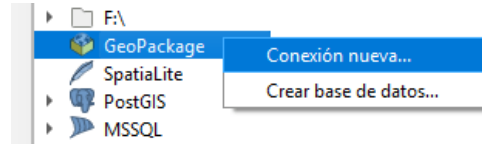
En QGIS 2.x, el shapefile ha sido el formato de archivo predeterminado para la exportación de datos. Ahora, en QGIS 3.x, el formato de salida estándar es el GeoPackage.

Para exportar una capa, en primer lugar debemos añadir **cualquier archivo vectorial admitido por OGR** a QGIS e ir al menú superior **Capa > Guardar como...**

En la ventana de dialogo que se abre seleccionaremos como formato **GeoPackage** y le asignamos un nombre y un directorio de salida.



El siguiente paso, es crear una conexión al **GeoPackage** que acabamos de crear, para ello, primero, vamos a añadir a la interfaz de *QGIS*, si no lo está ya, el panel del *Navegador*.



Una vez en el *Navegador*, haz clic derecho de ratón sobre el proveedor **GeoPackage** y selecciona *Conexión nueva...*

Selecciona en el explorador de archivos el **GeoPackage** recién creado y pulsa en *Abrir*.

4.6 Fuentes

<https://mappinggis.com/2019/06/como-crear-e-importar-datos-a-un-geopackage-con-qg>

Porceos de trabajo en autoabastecimiento, crear los geopacde de cada isla

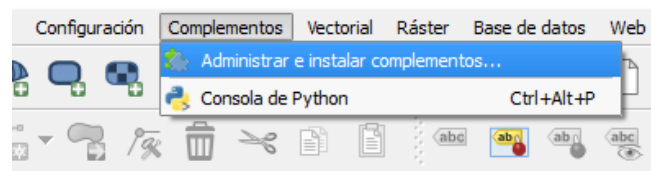
5 Otros

5.1 Complementos para QGIS

https://docs.qgis.org/2.18/es/docs/user_manual/plugins/core_plugins.html

Destacar en general como

1. Abrimos QGIS.
2. En el menú superior de QGIS vamos a Complementos > Administrar e instalar complementos...



3. Se recopilarán todos los complementos y se listarán todos plugins a los que tenemos acceso de los repositorios que tenemos configurados.
4. Introducimos el nombre del plugin en el buscador y veremos el icono de nuestro plugin.

5.2 20 Trucos de QGIS que quizá no conocías

<https://mappinggis.com/2015/01/trucos-de-qgis-que-quiza-no-conocias/>

5.3 10 opciones de QGIS 3 imprescindibles, revisar importante

<https://mappinggis.com/2019/01/10-opciones-de-qgis-que-seguramente-desconocias/>

Contiene:

- Aceleración de procesado
- Anular etiquetado parcial

- Creación de GeoPackage