

SIG. Práctica 4. Tema 3.  
Transformar e integrar información geográfica

José Samos Jiménez

Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Granada

2020 jsamos (LSI-UGR)

Curso 2020-2021

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Transformaciones a realizar sobre los datos</b>	<b>3</b>
2.1. Recorte de los datos . . . . .	3
2.2. Reproyección de los datos . . . . .	4
<b>3. Transformaciones básicas con <i>QGIS</i></b>	<b>4</b>
3.1. Reproyectar una capa . . . . .	4
3.2. Componer un ráster virtual a partir de varios ráster . . . . .	4
3.3. Recortar una capa ráster . . . . .	8
3.3.1. Mediante un rectángulo . . . . .	9
3.3.2. Mediante cualquier polígono . . . . .	10
3.4. Recortar una capa vectorial . . . . .	10
3.4.1. Recortar cualquier capa vectorial . . . . .	12
3.4.2. Copiar el estilo de una capa vectorial . . . . .	12
<b>4. Transformaciones básicas con <i>R</i></b>	<b>15</b>
4.1. Lectura y escritura de capas . . . . .	15
4.1.1. Cargar una capa vectorial y problemas con el SRC . . . . .	15
4.1.2. Guardar una capa vectorial y problemas con el SRC . . . . .	18
4.1.3. Cargar una capa ráster . . . . .	18
4.1.4. Guardar una capa ráster . . . . .	22
4.2. Reproyectar una capa . . . . .	22
4.2.1. Reproyectar una capa vectorial . . . . .	22
4.2.2. Reproyectar una capa ráster . . . . .	22
4.3. Componer un ráster virtual a partir de varios ráster . . . . .	22
4.4. Recortar una capa ráster . . . . .	24
4.4.1. Mediante un rectángulo . . . . .	24
4.5. Recortar una capa vectorial . . . . .	24
<b>Bibliografía</b>	<b>24</b>

Los objetivos de esta actividad son:

- el principal, es repasar los conceptos básicos sobre transformación e integración de información geográfica explicados en clase de teoría materializados en herramientas específicas, de uso generalizado;
- como objetivo secundario, se va a introducir *GDAL/OGR* usado desde *QGIS* y desde *R* mediante ejemplos;
- como resultado adicional, transformaremos los datos obtenidos en la actividad anterior:
  - tanto para datos ráster como para datos vectoriales, nos focalizaremos en la zona de trabajo;
  - para datos vectoriales, los transformaremos para que todos tengan el mismo SRC.

A continuación, después de una introducción, se definen las transformaciones que deberemos hacer sobre nuestros datos. En el resto del documento, se presenta cómo realizar las operaciones básicas para llevar a cabo esas transformaciones, en primer lugar mediante *QGIS* (basado en [Men19]), después en *R* (basado en [BC19]). Es decir, en esta actividad se indica qué hay que hacer y se facilitan posibles herramientas con la ayuda necesaria para desarrollarla. No se indica cómo hay que hacerla: **se puede usar una u otra herramienta, o una combinación de ambas**.

## 1. Introducción

En la actividad anterior, hemos obtenido la información geográfica para nuestro municipio. En algunos casos, hemos seleccionado la zona específica, en otros, hemos obtenido información para toda Andalucía o, incluso, España.

En esta actividad, vamos a filtrar la información disponible para considerar exclusivamente la del área que se ha considerado adecuada para nuestro municipio. Por otro lado, para cada caso, se seleccionará el SRC que se considere más adecuado.

Es decir, no se presenta un catálogo de posibles transformaciones, sino las transformaciones que se consideran convenientes para, posteriormente, almacenar los datos de forma adecuada.

## 2. Transformaciones a realizar sobre los datos

Partiendo de los datos obtenidos en la actividad anterior, vamos a transformarlos e integrarlos de manera que:

- todas las capas abarquen exclusivamente la zona definida para nuestro municipio,
- todas las capas vectoriales usen el mismo SRC y
- cada capa ráster use su SRC original<sup>1</sup>.

Trataremos todos los datos obtenidos excepto los datos de satélite, que solo consideraremos las bandas 2 a 5 de *Landsat 8*.

**Como resultado para entregar, genera un documento PDF con una captura de pantalla para cada una de las capas obtenidas donde se muestre el contenido de la capa y el entorno donde lo has realizado (la pantalla completa de *QGIS* o *RStudio*).**

Es suficiente desarrollarla con UNA herramienta.

### 2.1. Recorte de los datos

1. Recorta los datos ráster mediante la capa `municipio-box`.
2. Recorta los datos vectoriales mediante la capa `municipio`.

<sup>1</sup>No reproyectamos las capas ráster porque al reproyectar generalmente se pierde información. Si necesitamos trabajar conjuntamente con una capa ráster y otra vectorial, es preferible reproyectar la capa vectorial. Solo reproyectaremos una capa ráster si necesitamos trabajar conjuntamente con otra capa ráster con distinta proyección.

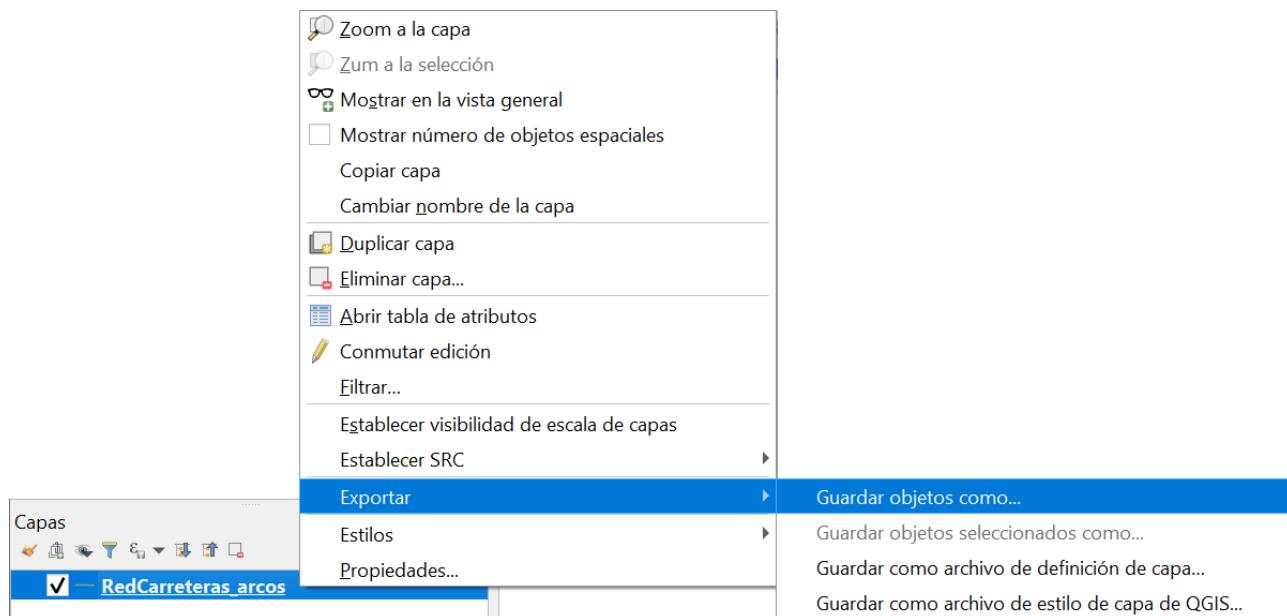


Figura 1: Reproyectar una capa.

## 2.2. Reproyección de los datos

3. Reproyecta los datos vectoriales de manera que todos tengan el SRC de código EPSG 25830.

Aunque en los ejemplos siguientes se muestra cómo los datos vectoriales se almacenan en formato *shapefile*, recomiendo almacenarlos también como *GeoPackage*.

## 3. Transformaciones básicas con *QGIS*

### 3.1. Reproyectar una capa

Para cambiar el SRC de una capa (reproyectarla), lo que hacemos es generar una nueva capa a partir de la original.

En el menú contextual de la capa, pulsamos sobre [«Exportar», «Guardar objetos como...»] (figura 1). En la ventana que se abre (figura 2), definimos el formato de la nueva capa, su nombre y ubicación, y, en el campo «SRC», pulsando sobre el icono «Seleccionar SRC» que hay a su derecha, seleccionamos el nuevo SRC. En el resto de campos, podemos dejar su valor por defecto.

La selección del SRC se lleva a cabo en la ventana de la figura 3, si conocemos el código EPSG de la nueva capa, lo podemos introducir en el campo «Filtrar» y seleccionamos el SRC correspondiente.

Esta operación se puede aplicar tanto a capas vectoriales como a capas ráster.

### 3.2. Componer un ráster virtual a partir de varios ráster

Si trabajamos con una zona relativamente amplia, seguramente hayamos obtenido varias capas ráster que, conjuntamente, cubren la zona de trabajo, probablemente cubran incluso una zona más amplia que la que necesitamos. Hay varias posibilidades para operar conjuntamente con todas las capas, por ejemplo, podríamos fusionarlos en un solo ráster (se genera una nueva capa con todos los datos). Otra posibilidad más sencilla es generar un ráster virtual a partir de los componentes (ocupa poco espacio nuevo porque se basa en las otras capas). Esta última posibilidad es adecuada sobre todo si después vamos a recortar el ráster resultante, por tanto, es la que vamos a considerar.

Para definir un ráster virtual a partir de varias capas, cargamos todas las capas en el visor para poder apreciar el ráster que forman conjuntamente. A continuación, pulsamos sobre [«Ráster», «Miscelanea», «Construir ráster virtual...»] (figura 4).

En la ventana que se abre (figura 5), en el campo «Input layers», pulsando sobre el botón con puntos suspensivos, situado a la derecha, seleccionamos las capas que compondrán el ráster: en este

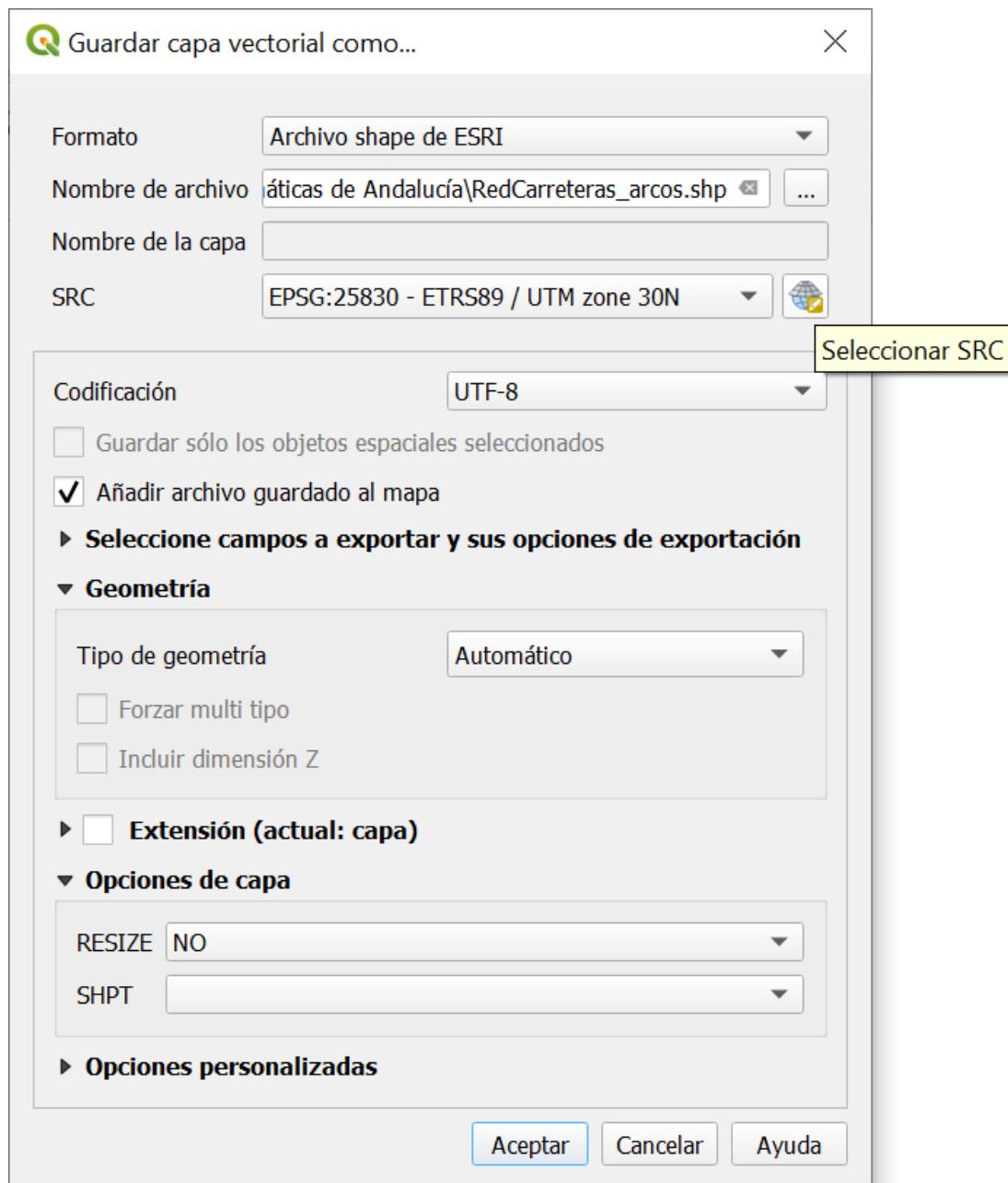


Figura 2: Definición de la nueva capa.

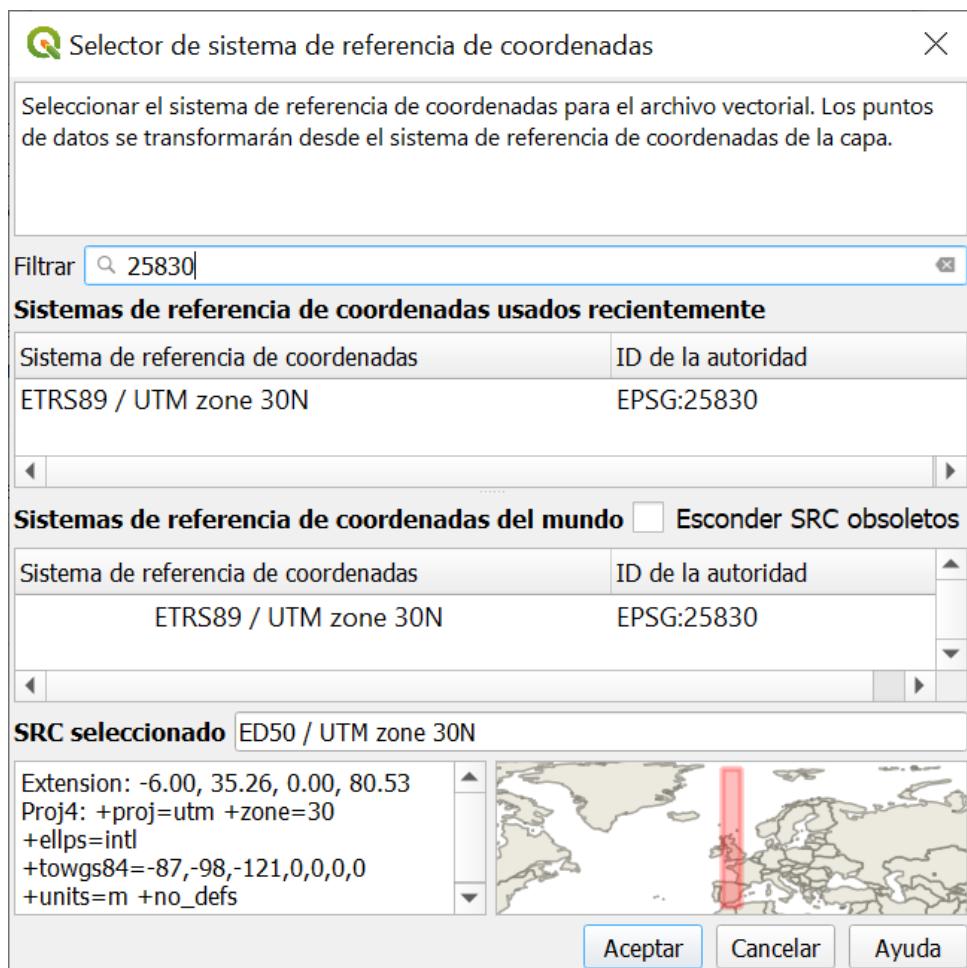


Figura 3: Selección del SRC de la nueva capa.

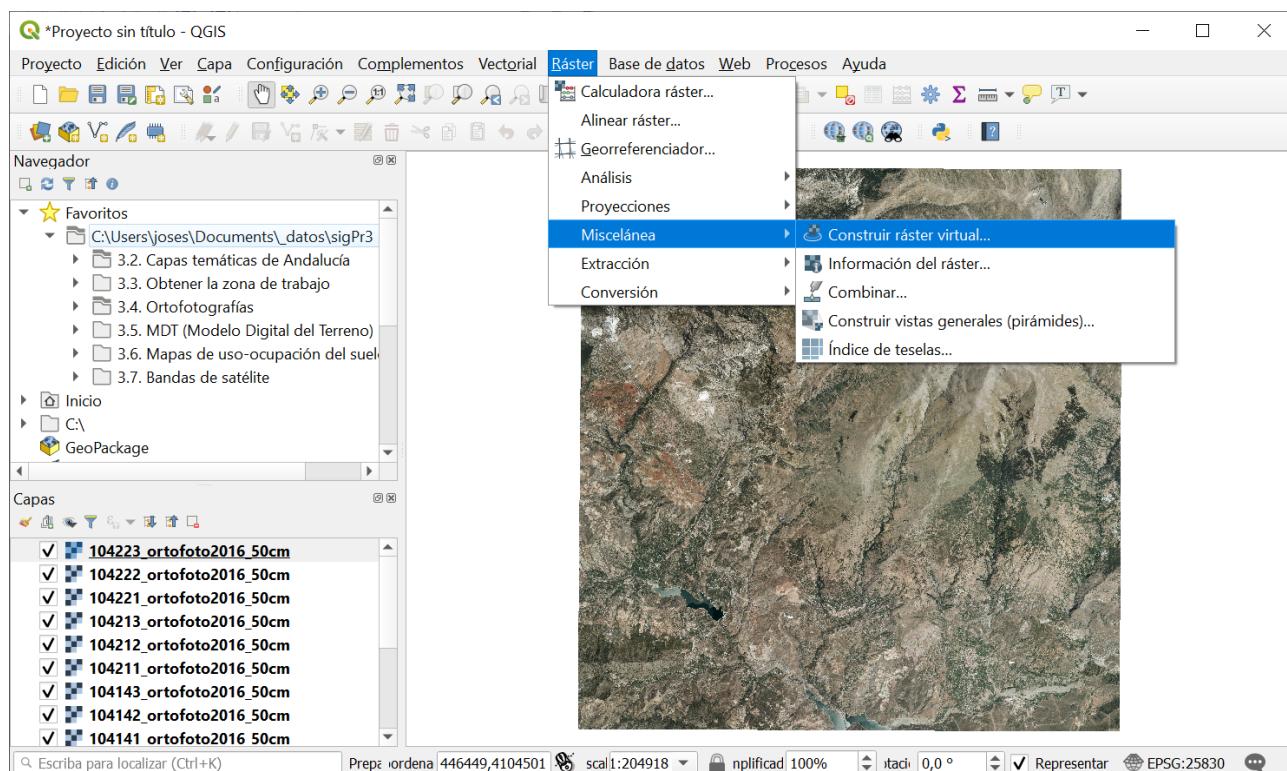


Figura 4: Construir un ráster virtual.

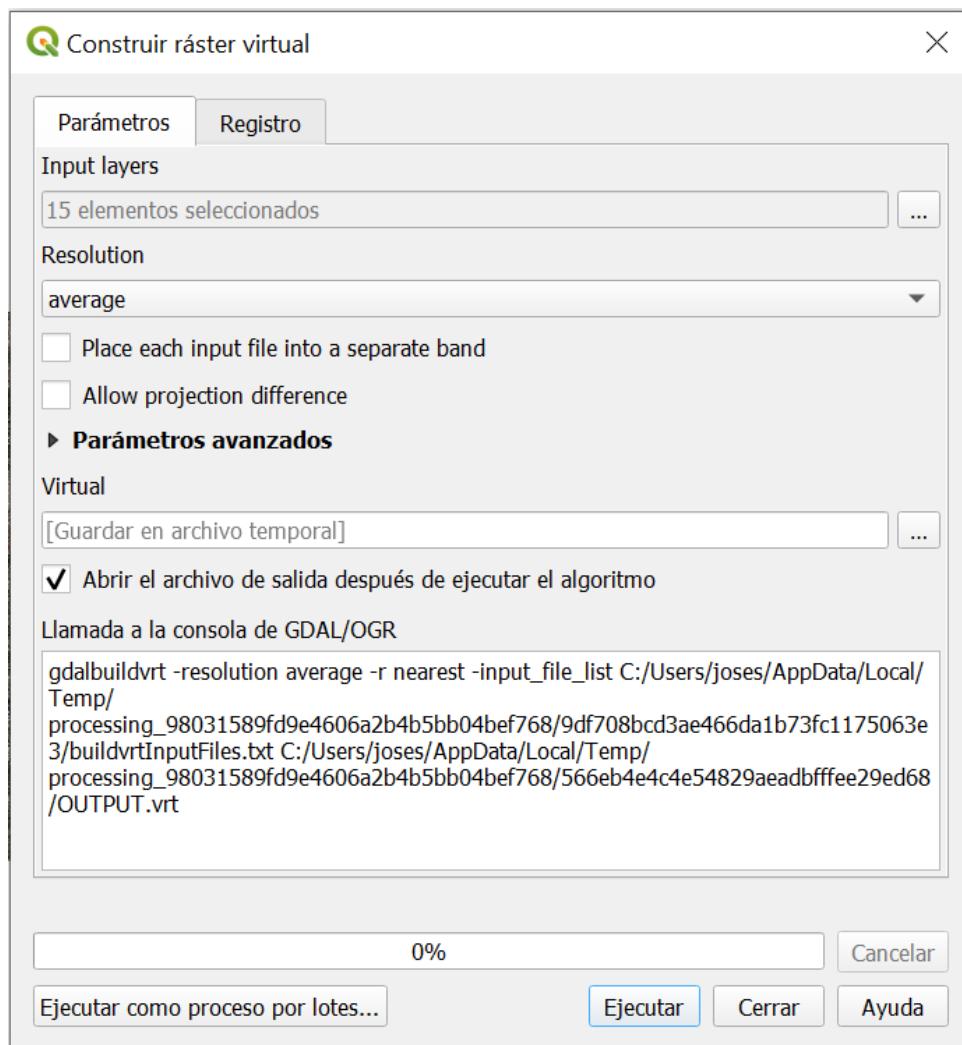


Figura 5: Elementos del ráster virtual.

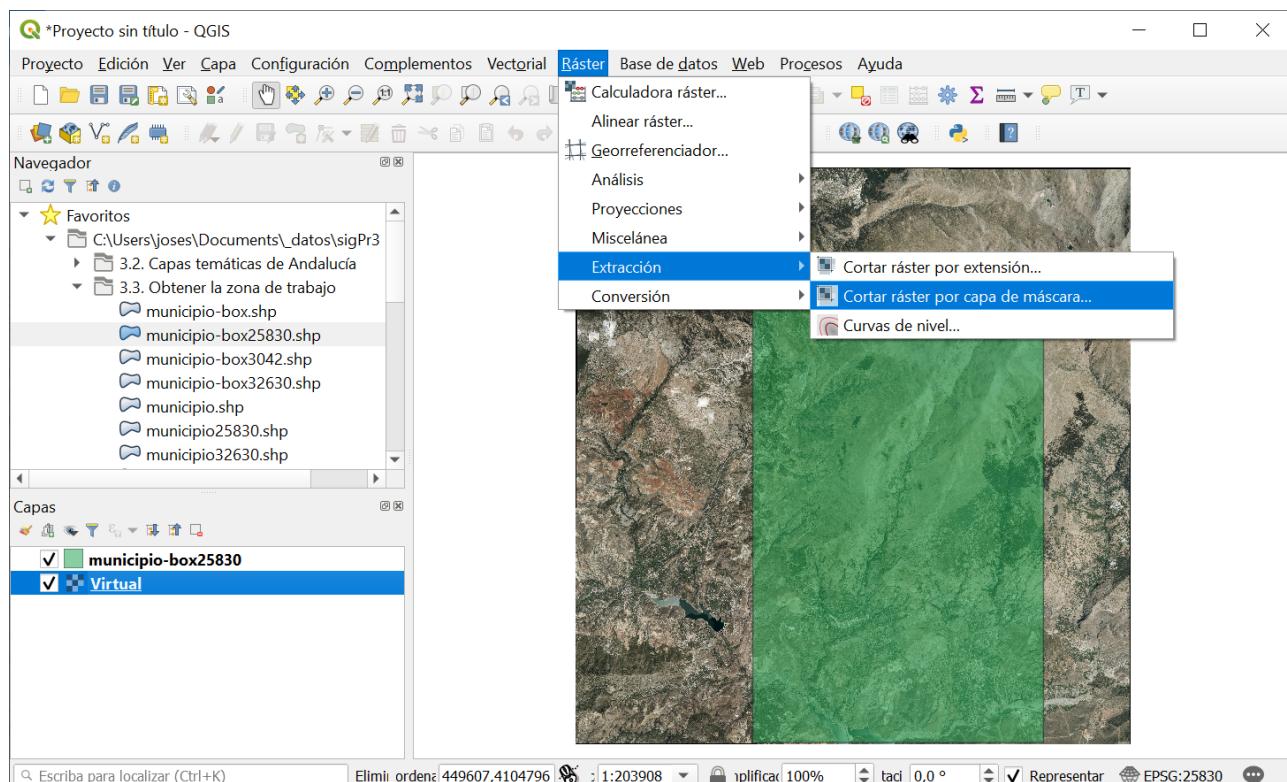


Figura 6: Recortar un ráster usando una capa de máscara.

caso, como solo hemos cargado las capas del ráster, podemos seleccionar todas las capas pulsando sobre el botón «Seleccionar todo». Dado que todas las capas tendrán la misma resolución, no afecta el criterio que elijamos en el campo «Resolution» (puede obtener la resolución media, la más fina o la más gruesa). Es muy importante deseleccionar la opción «Place each input file into a separate band» porque, en caso de estar seleccionada, el resultado sería un ráster multibanda y no es lo que necesitamos. En la parte inferior de la ventana, en «Llamada a la consola de GDAL/OGR», se muestra el código que se genera en función de los parámetros que vamos definiendo. Para generar esta nueva capa, *QGIS* se basa en la funcionalidad que ofrecen las librerías *GDAL/OGR*. Si vamos a usar la capa virtual en otros proyectos, podemos salvarla indicando el nombre del archivo en el campo «Virtual»; como solo queremos recortarla y lo vamos a hacer a continuación, en este caso podemos dejar que la guarde en un archivo temporal. Una vez definidos todos los parámetros, pulsamos sobre el botón «Ejecutar» para llevar a cabo el proceso de definición de la nueva capa. En la pestaña «Registro» de la ventana nos indica que lo va haciendo y también avisa cuando haya terminado.

Una vez generada la nueva capa (le asigna el nombre *Virtual*), para simplificar, podemos eliminar las capas componentes del visor de capas y trabajar solo con la nueva capa virtual.

### 3.3. Recortar una capa ráster

Para recortar una capa ráster, vamos a contemplar dos posibilidades: recortarla mediante un rectángulo orientado de Norte a Sur o hacerlo mediante cualquier polígono. En ambos casos, vamos a usar una capa vectorial como máscara de la operación de recorte.

Cuando vayamos a aplicar operaciones en las que intervengan varias capas, es importante que todas las capas tengan el mismo SRC<sup>2</sup>. Si la capa vectorial no tiene el mismo SRC que la capa ráster, generaremos una nueva capa vectorial reproyectando la capa original.

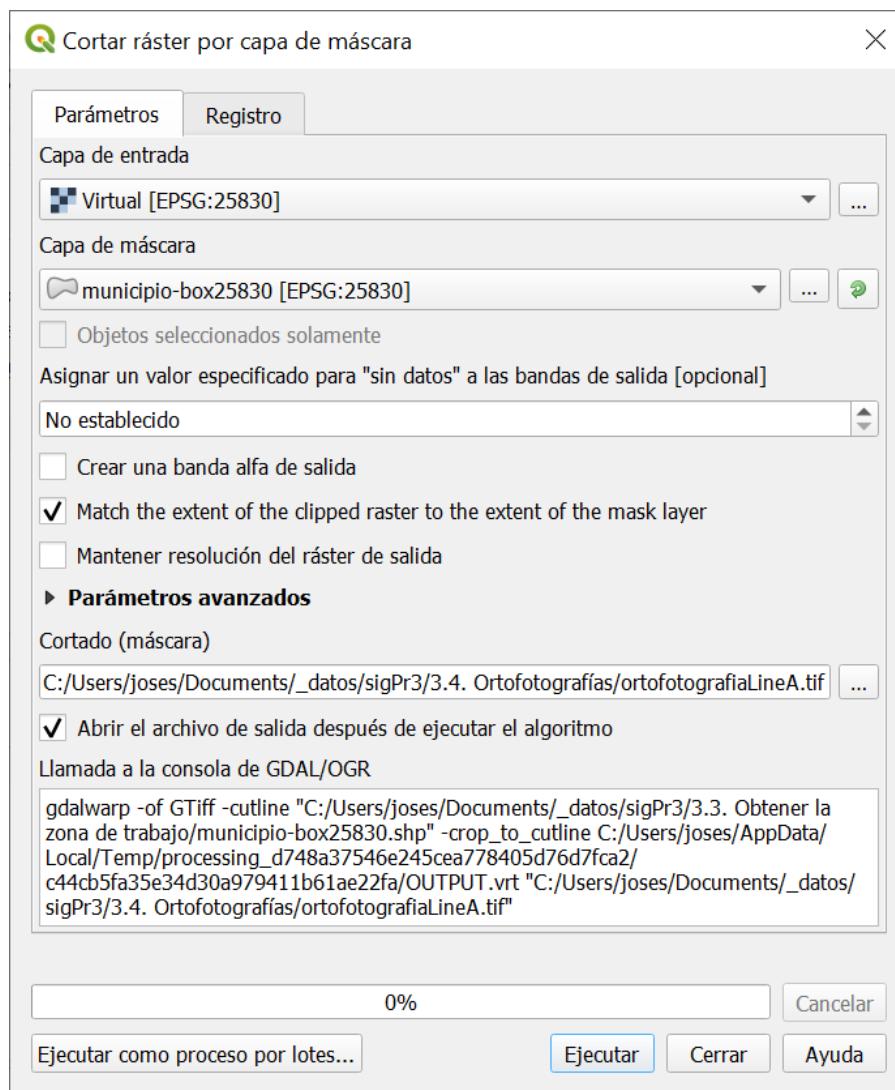


Figura 7: Parámetros para recortar un ráster por un rectángulo.

### 3.3.1. Mediante un rectángulo

Visualizamos la capa ráster y la capa vectorial (rectángulo orientado de Norte a Sur) conjuntamente, y pulsamos sobre [«Ráster», «Extracción», «Cortar ráster por capa de máscara...»] (figura 6).

Se abre una ventana para definir los parámetros de la operación (figura 7). Seleccionamos en el campo «Capa de entrada» la capa ráster y en el campo «Capa de máscara» la capa vectorial (aquí podemos comprobar que las capas tengan el mismo SRC). Comprobamos que la única opción seleccionada sea «Match the extent of the clipped raster to the extent of the mask layer». En el campo «Cortado (máscara)», pulsando sobre el botón con «...» situado a su derecha, elegimos la opción [«Guardar a archivo...»] e indicamos la ubicación, el nombre y el tipo: el tipo debe ser «TIF files (\*.tif)». También marcamos la opción «Abrir el archivo de salida después de ejecutar el algoritmo». El comando que ha generado con la configuración que hemos indicado, se puede ver en el apartado «Llamada a la consola GDAL/OGR». Para ejecutar la operación, pulsamos sobre el botón «Ejecutar». El resultado nos lo muestra en la pestaña «Registro» de la misma ventana de definición de parámetros. Podemos cerrar la ventana y desactivar las otras capas para ver el resultado en la capa que llama **Cortado (máscara)**.

<sup>2</sup>En principio deberían funcionar igual con distinto SRC, la experiencia es que se evitan errores y las operaciones se ejecutan más rápido si en cada una solo se realiza una transformación.

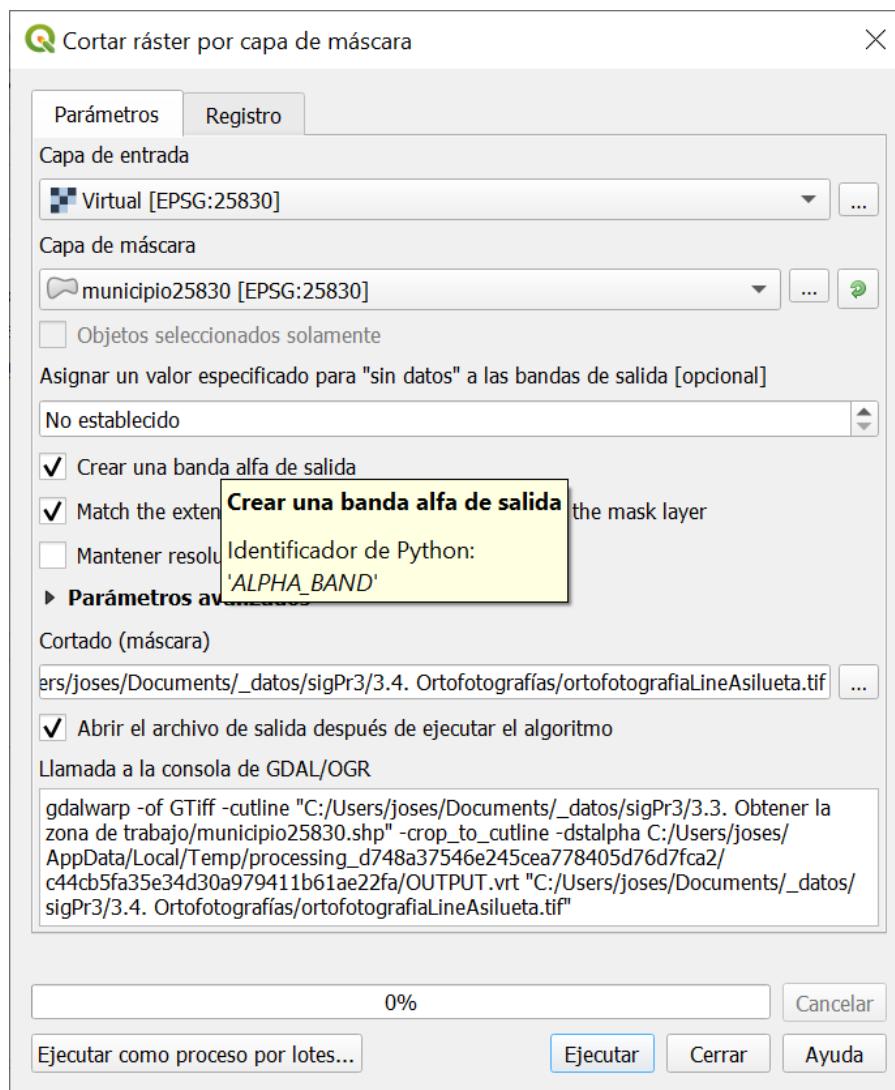


Figura 8: Parámetros para recortar un ráster por cualquier polígono.

### 3.3.2. Mediante cualquier polígono

Para recortar un ráster por cualquier polígono, al igual que para un rectángulo, visualizamos la capa ráster y la capa vectorial conjuntamente, y pulsamos sobre [«Ráster», «Extracción», «Cortar ráster por capa de máscara...»] (figura 6).

Para mantener los bordes de la capa vectorial, en la definición de los parámetros de la operación de recorte, hemos de seleccionar la opción «Crear una banda alfa de salida» (figura 8). El resto de parámetros se definen igual que para el recorte mediante un rectángulo.

El resultado se puede observar en la figura 9. Si no seleccionamos la opción «Crear una banda alfa de salida», recorta el ráster por el polígono pero, adicionalmente, lo incluye en un rectángulo orientado de Norte a Sur y al resto de celdas de este le asigna el valor 0 (figura 10).

## 3.4. Recortar una capa vectorial

Para recortar una capa vectorial, vamos a presentar la operación básica de recorte y, en casos donde la capa original tenga un estilo predefinido, adicionalmente vamos a copiar el estilo de la capa.

Al igual que para las capas ráster, cuando vayamos a aplicar operaciones en las que intervengan varias capas, es importante que todas las capas tengan el mismo SRC. En el caso de trabajar con capas vectoriales, se podría reproyectar cualquiera de ellas. Si vamos a recortar una capa, generalmente será de mayor tamaño que la capa que vamos a recortar, entonces reproyectaremos la capa de máscara para la operación de recorte.

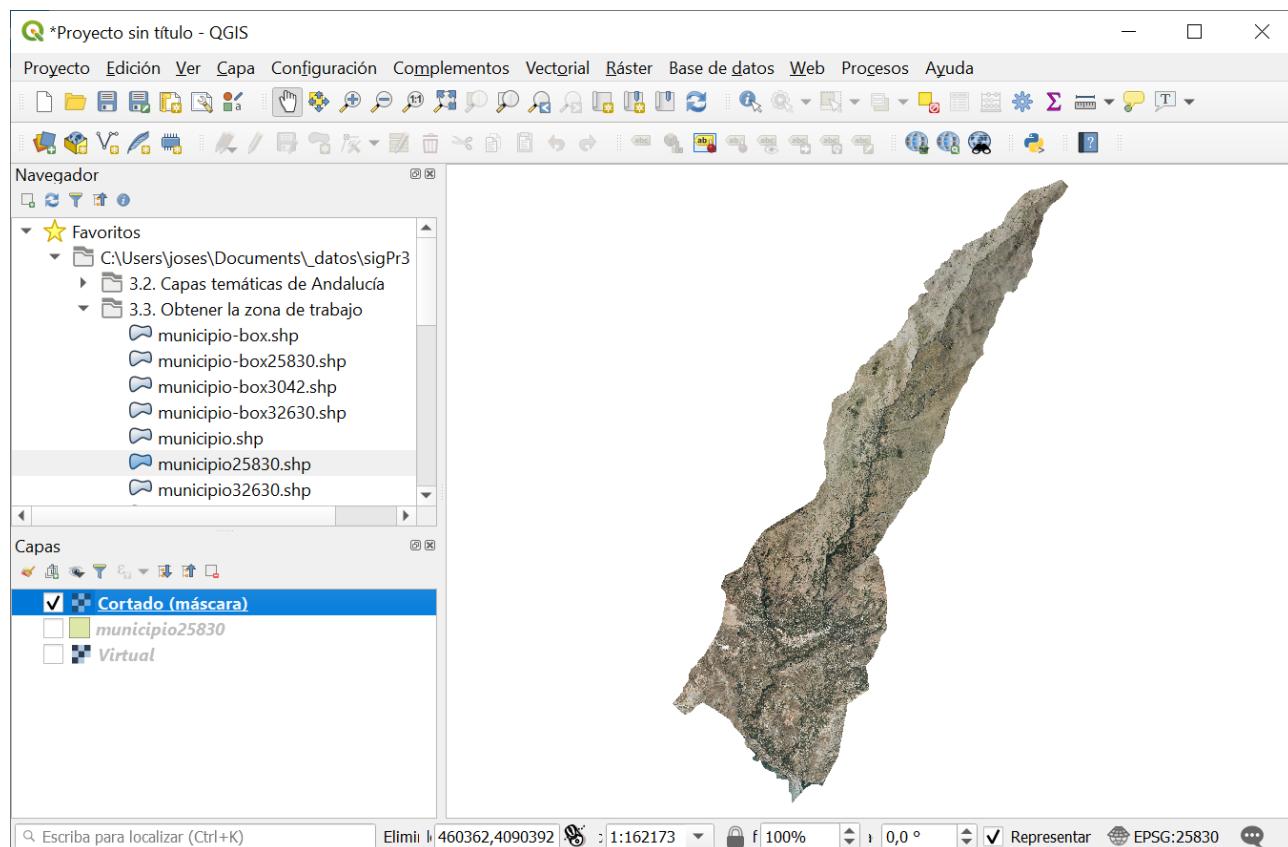


Figura 9: Resultado de recortar un ráster por cualquier polígono.

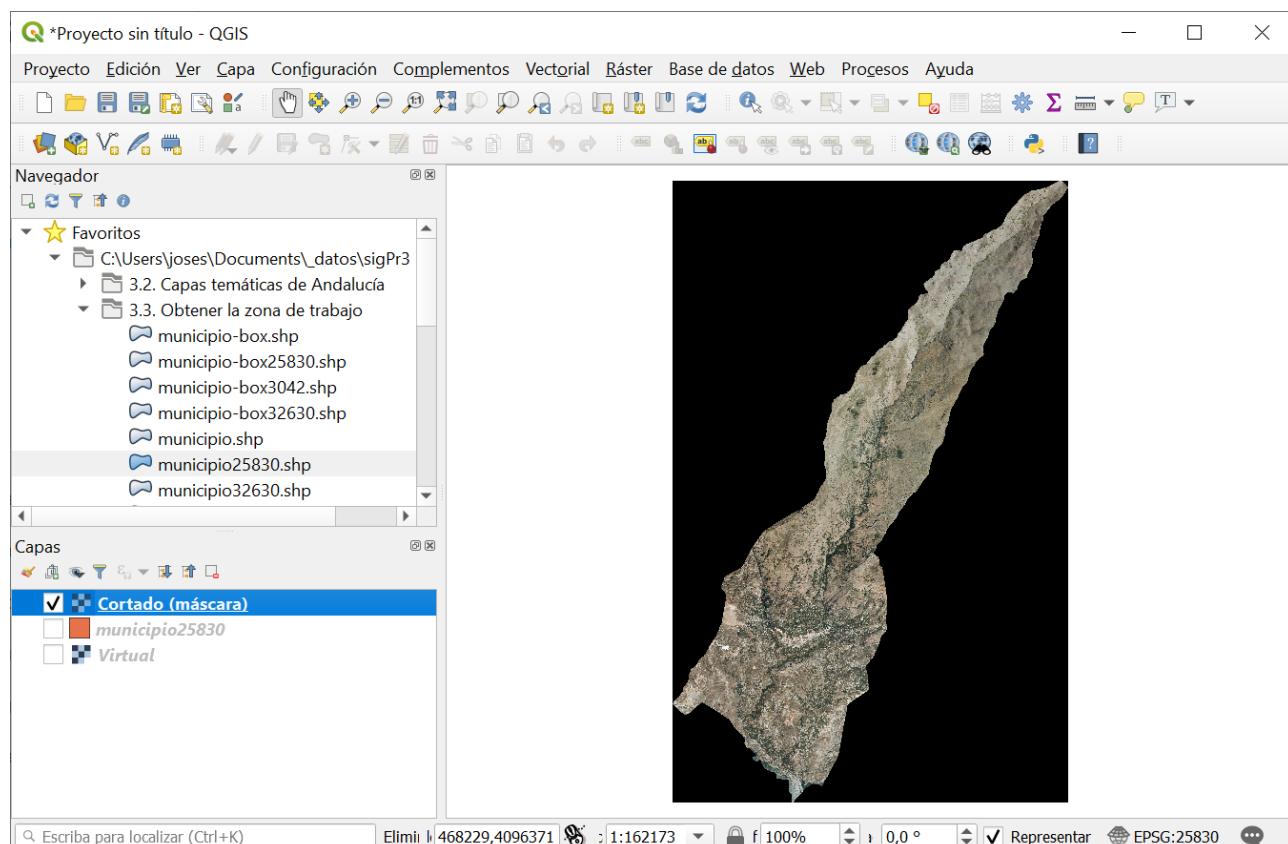


Figura 10: Resultado de recortar un ráster por cualquier polígono sin crear banda alfa.

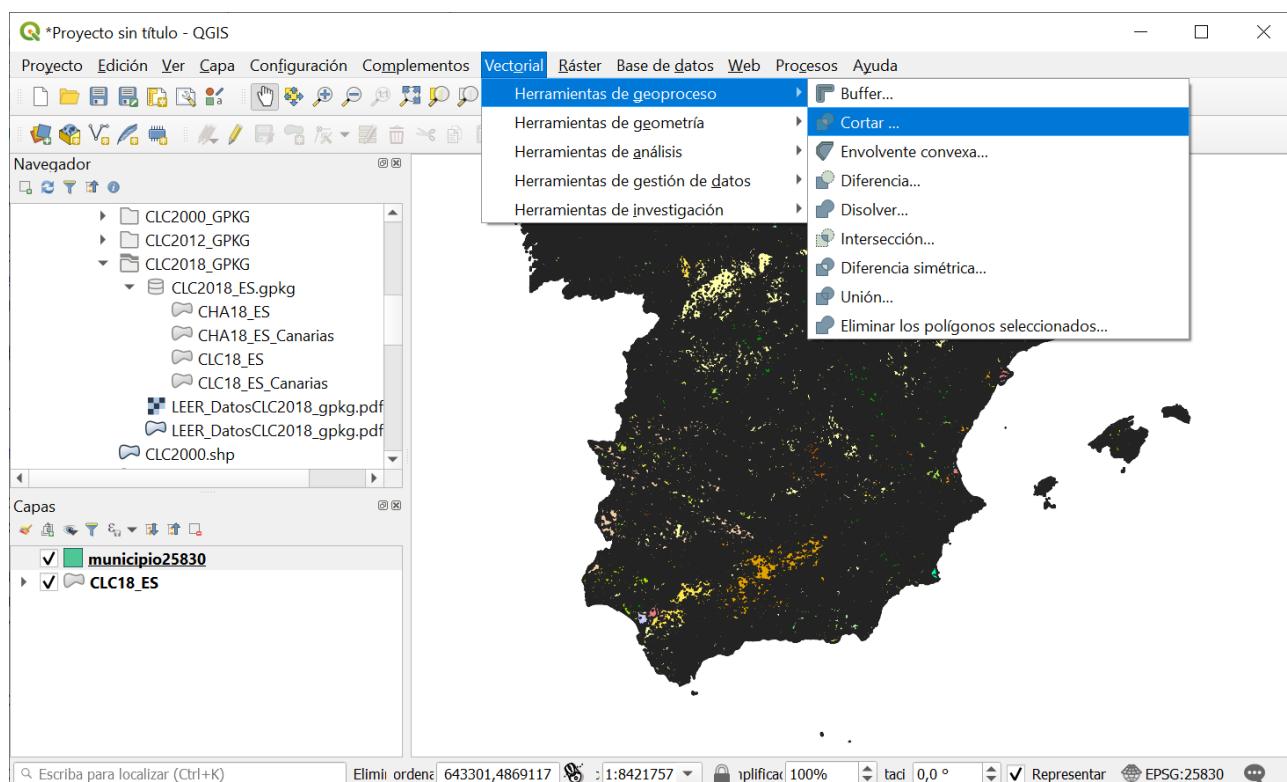


Figura 11: Recortar una capa vectorial mediante cualquier polígono.

### 3.4.1. Recortar cualquier capa vectorial

Visualizamos las dos capas vectoriales conjuntamente, y pulsamos sobre [«Vectorial», «Herramientas de geoproceso», «Cortar...»] (figura 11).

Se abre una ventana para definir los parámetros de la operación (figura 11). Seleccionamos en el campo «Capa de entrada» la capa vectorial a recortar y en el campo «Capa de superposición» la capa vectorial a usar como plantilla (aquí podemos comprobar que las capas tengan el mismo SRC). En el campo «Cortado», pulsando sobre el botón con «...» situado a su derecha, elegimos la opción [«Guardar a archivo...»] e indicamos la ubicación, el nombre y el tipo: el tipo debe ser «SHP files (\*.shp)». También marcamos la opción «Abrir el archivo de salida después de ejecutar el algoritmo». El comando que ha generado con la configuración que hemos indicado, se puede ver en el apartado «Llamada a la consola GDAL/OGR». Para ejecutar la operación, pulsamos sobre el botón «Ejecutar». El resultado nos lo muestra en la pestaña «Registro» de la misma ventana de definición de parámetros. Podemos cerrar la ventana y desactivar las otras capas para ver el resultado en la capa que llama **Cortado** (figura 13).

### 3.4.2. Copiar el estilo de una capa vectorial

Si comparamos el estilo de la capa resultado (figura 13) con el estilo de la capa original de la figura 14, observamos que la capa resultado lo ha perdido. Se puede copiar el estilo de una capa vectorial y aplicárselo a otra, en este caso, a la capa resultado.

Para copiar el estilo de una capa, en el menú contextual de la capa, pulsamos sobre [«Estilos», «Copiar estilo», «All style Categories»] (figura 15). De igual forma, en el menú contextual de la capa destino, pulsamos sobre [«Estilos», «Pegar estilo», «All style Categories»]. El resultado se puede observar en la figura 16.

Para guardar ese estilo asociado a la capa, lo hacemos desde el menú contextual de la capa, pulsando sobre la opción [«Exportar», «Guardar como archivo de definición de capa...»] e indicamos la carpeta y el nombre del archivo.

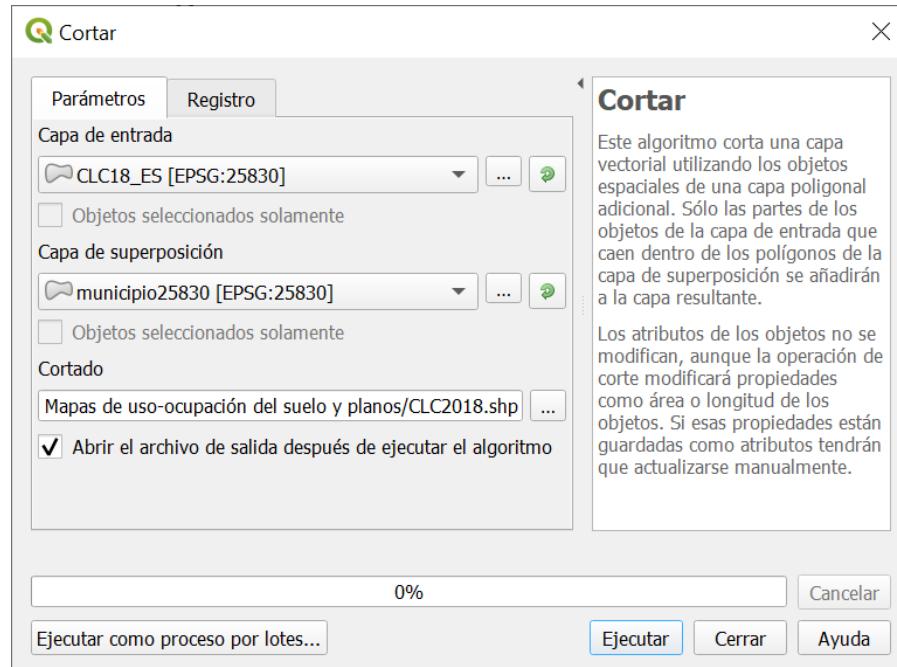


Figura 12: Parámetros para recortar una capa vectorial mediante cualquier polígono.

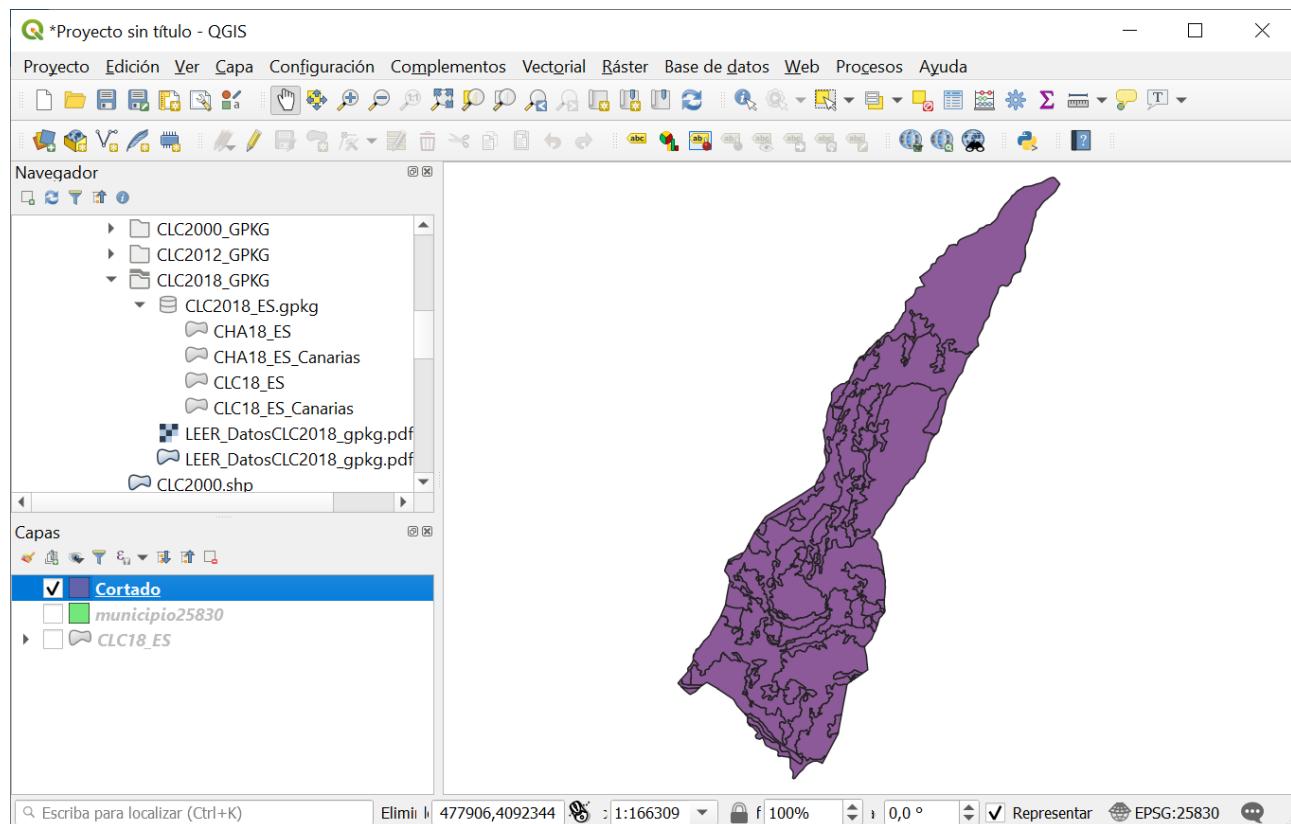


Figura 13: Resultado de recortar una capa vectorial mediante cualquier polígono.

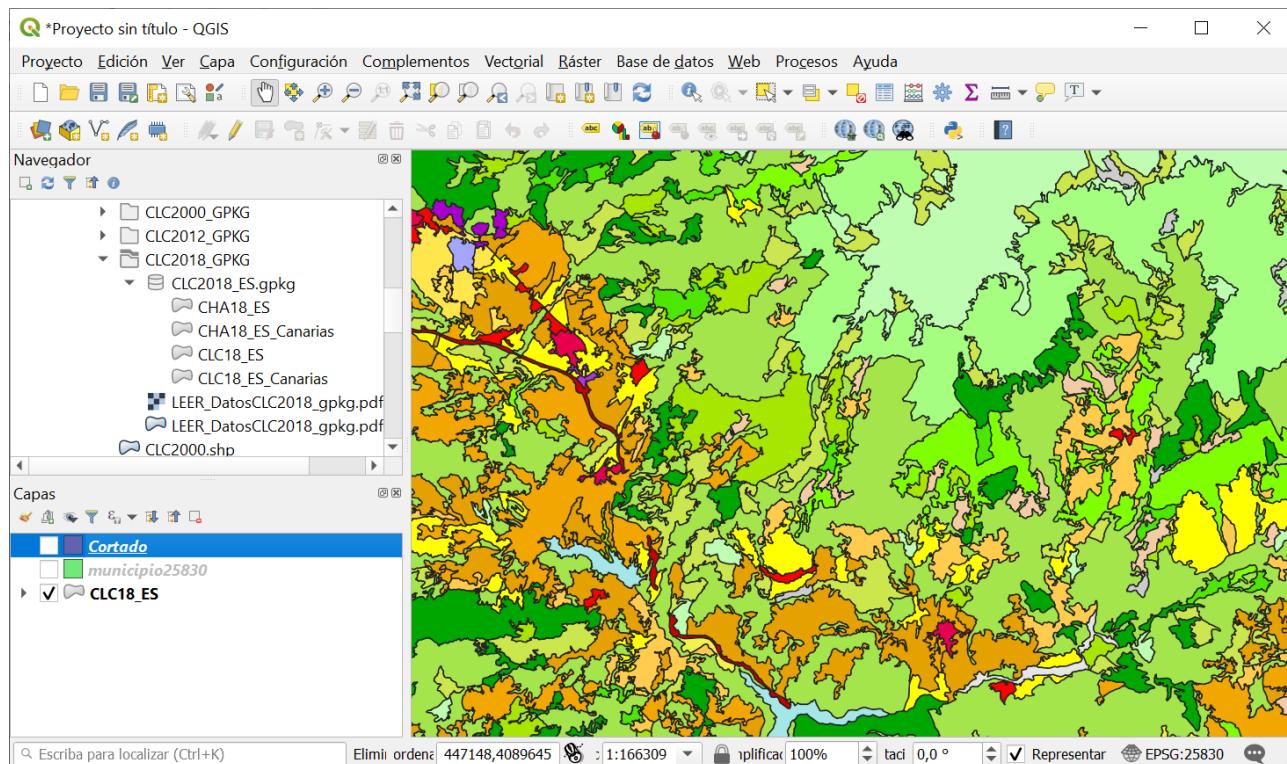
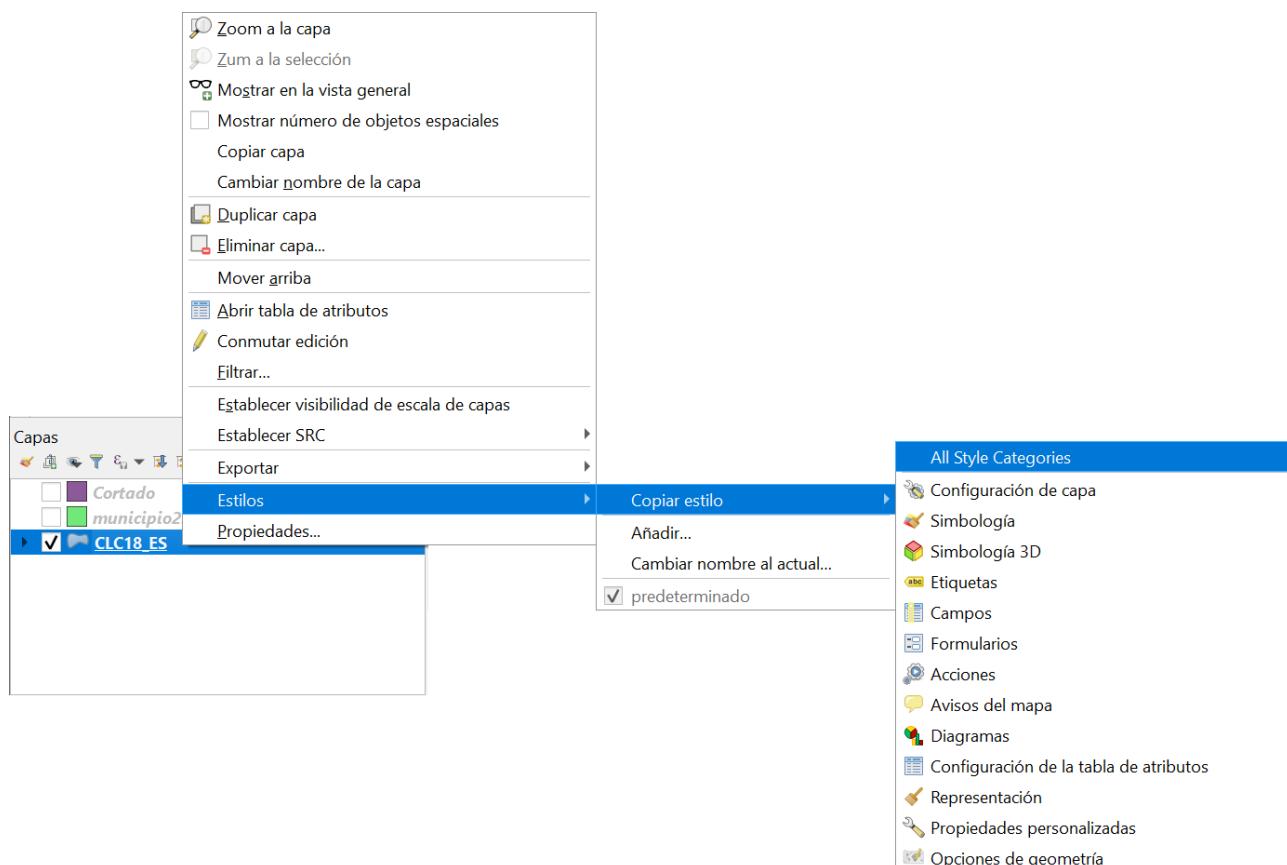
Figura 14: Estilo de la capa original (*CORINE Land Cover*).

Figura 15: Copiar el estilo de una capa.

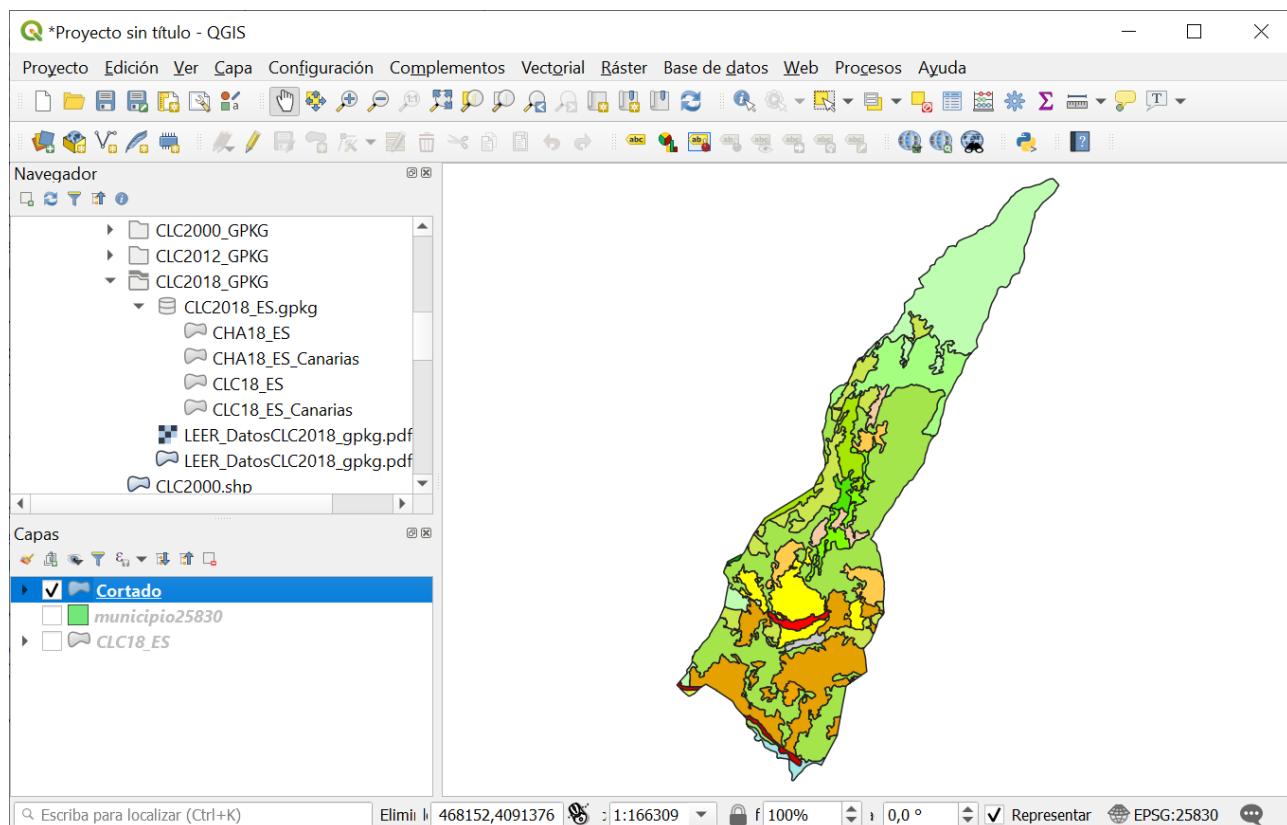


Figura 16: Estilo de la capa resultado.

## 4. Transformaciones básicas con *R*

### 4.1. Lectura y escritura de capas

#### 4.1.1. Cargar una capa vectorial y problemas con el SRC

Al cargar desde *R* un Shapefile, nos encontramos con diversos problemas, dependiendo del SRC de la capa.

En la figura 17, se puede observar que leemos la capa correctamente pero el código EPSG tiene el valor NA.

Por otra parte, en la figura 18, se puede apreciar que, leyendo exactamente igual la capa, en este caso no está definida la cadena de caracteres que del campo `proj4string`.

En ambos casos, si cargamos esas capas en **QGIS**, el CRS está correctamente definido.

En *R*, cuando está definido correctamente el campo `proj4string` (como en la figura 17) podemos trabajar sin problemas. Sin embargo, cuando este no está definido, aunque lo esté el campo `epsg` (como en la figura 18) muchas funciones no funcionan, avisando de un error en los datos.

Por ejemplo, una de las funciones que presentan este error es la de reproyectar una capa (figura 19).

En la figura 20, se puede ver que no se presenta ese error en la capa con el campo `proj4string` correctamente definido.

Como conocemos el EPSG de la capa, podemos probar a asignárselo a ver si obtiene los datos para el campo `proj4string`. Esto es lo que se ha hecho en la figura 21: se puede observar que no lo permite, da un error.

La manera que he encontrado de resolver este problema es eliminar previamente el valor de los campo `epsg` y `proj4string`, y realizar de nuevo la asignación, como se muestra en la figura 22.

En la figura 23, se puede apreciar que, para la misma capa y operación que presentaba error en la figura 19, en este caso no se produce el problema. El único cambio es que le hemos vuelto a asignar el código EPSG (figura 22) y así el campo `proj4string` ya tiene un valor adecuado.

```

library(sf)

setwd("~/datos/sigPr3/3.3. Obtener la zona de trabajo/")

municipio <- st_read("municipio.shp")

## Reading layer `municipio' using driver `ESRI Shapefile'
## Simple feature collection with 1 feature and 6 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:            xmin: 453362 ymin: 4081219 xmax: 465192 ymax: 4099854
## epsg (SRID):   NA
## proj4string:    +proj=utm +zone=30 +ellps=intl +units=m +no_defs

```

Figura 17: Capa sin código EPSG.

```

setwd("~/datos/sigPr3/3.2. Capas tematicas de Andalucia/originales")

nucleosPobl <- st_read("NucleosPoblacion_contorno.shp")

## Reading layer `NucleosPoblacion_contorno' using driver `ESRI Shapefile'
## Simple feature collection with 7457 features and 8 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:            xmin: 102890.4 ymin: 3987415 xmax: 619269.5 ymax: 4274255
## epsg (SRID):   23030
## proj4string:

```

Figura 18: Capa sin proj4string.

```

nucleosPobl25830 <- st_transform(nucleosPobl, crs = "+init=epsg:25830")

## OGR: Corrupt data

## Error in CPL_crs_equivalent(e1$proj4string, e2$proj4string): OGR error

```

Figura 19: Error al tratar una capa sin proj4string.

```

municipio25830 <- st_transform(municipio, crs = "+init=epsg:25830")

```

Figura 20: Tratamiento correcto de una capa con proj4string.

```
(epsg <- st_crs(nucleosPobl)$epsg)
## [1] 23030
st_crs(nucleosPobl) <- st_crs(epsg)
## OGR: Corrupt data
## Error in CPL_crs_equivalent(e1$proj4string, e2$proj4string): OGR error
```

Figura 21: Error en la asignación del valor del campo epsg.

```
st_crs(nucleosPobl) <- NA_crs_
(st_crs(nucleosPobl) <- st_crs(epsg))

## Coordinate Reference System:
##   EPSG: 23030
##   proj4string: "+proj=utm +zone=30 +ellps=intl
##                 +towgs84=-87,-98,-121,0,0,0,0 +units=m +no_defs"
```

Figura 22: Asignación correcta del valor del campo epsg.

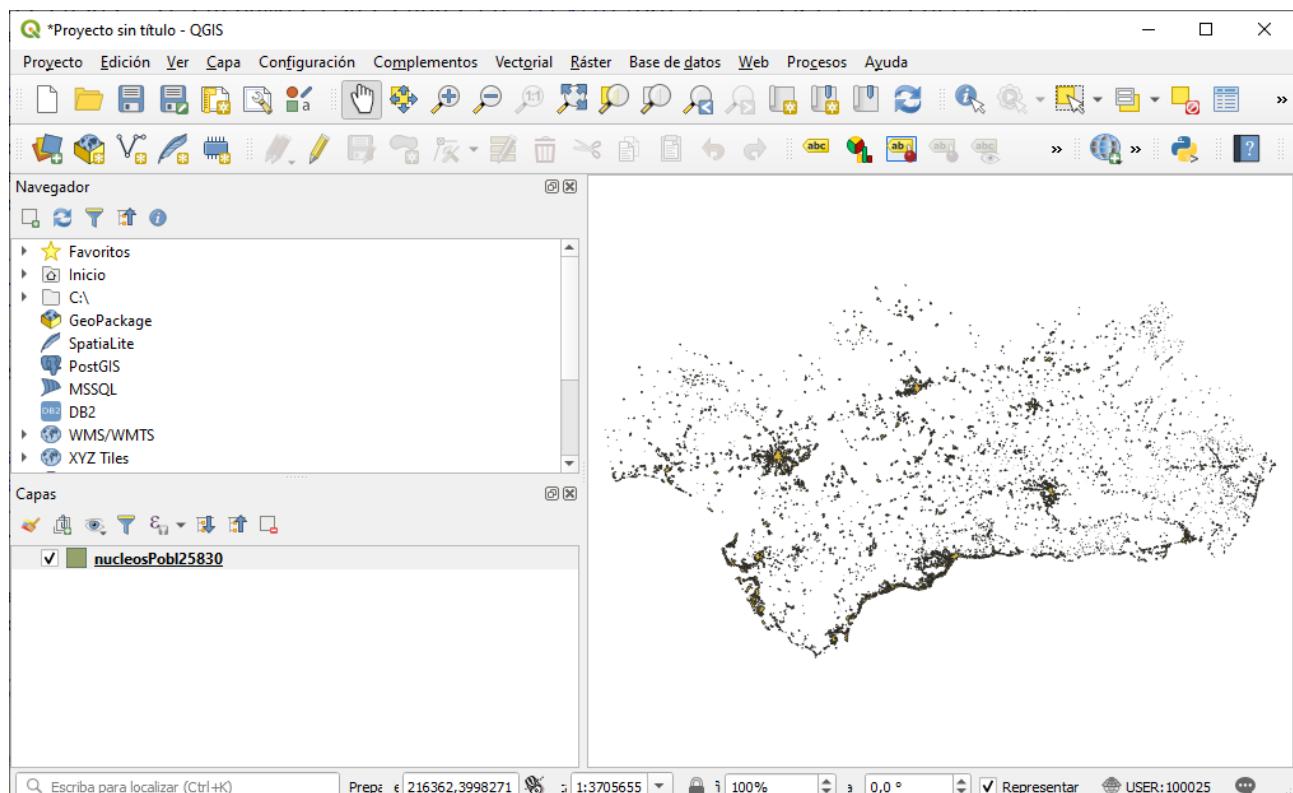
```
nucleosPobl25830 <- st_transform(nucleosPobl, crs = "+init=epsg:25830")
```

Figura 23: Tratamiento correcto de la capa después de la corrección.

```
st_write(nucleosPobl25830, "nucleosPobl25830.shp", delete_layer = T)

## Deleting layer `nucleosPobl25830' using driver `ESRI Shapefile'
## Writing layer `nucleosPobl25830' using driver `ESRI Shapefile'
## Writing 7457 features with 8 fields and geometry type Multi Polygon.
```

Figura 24: Guardar una capa vectorial.

Figura 25: CRS en *QGIS* de una capa guardada en *R*.

#### 4.1.2. Guardar una capa vectorial y problemas con el SRC

Si guardamos en un archivo una capa vectorial que tiene correctamente definido el CRS (figura 24) y abrimos esa capa en *QGIS* (figura 25), se puede observar (en la esquina inferior derecha) que el CRS que le asigna es **USER:100025**. Es decir, no ha conservado el EPSG definido.

Para que conserve correctamente el código EPSG, he definido la función de la figura 26<sup>3</sup>

En la figura 27, se guarda la misma capa vectorial mediante la nueva función definida. El resultado, al cargar la capa en *QGIS*, se puede ver en la figura 28: en este caso el CRS de la capa que se muestra es el que había definido en *R*, EPSG:25830.

#### 4.1.3. Cargar una capa ráster

Se pueden cargar todas las bandas de un ráster conjuntamente mediante la función **brick**. Las presentamos combinándolas con la función **plotRGB** (figura 29).

También podemos leer las bandas una a una mediante la función **raster** y, con la función **stack**, construir un ráster multibanda partiendo de las bandas leídas (figura 30). Aunque la estructura de datos **stack** es distinta a **brick**, el resultado obtenido en ambos casos es el mismo.

<sup>3</sup>Basada en <https://gis.stackexchange.com/questions/281083/epsg-code-lost-when-writing-shapefile-with-sf?noredirect=1&lq=1>.

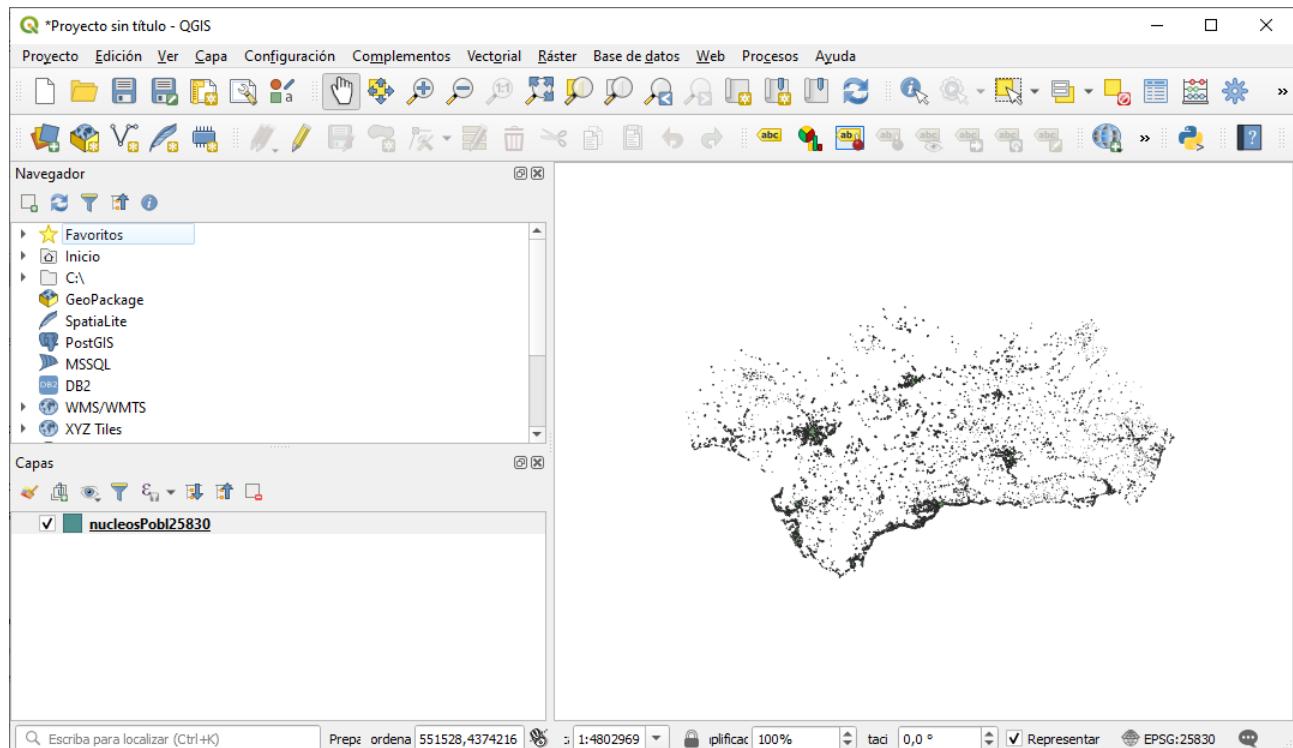
```
st_write_shp_epsg <- function(obj, name) {
  st_write(obj, paste(name, ".shp", sep = ""), delete_layer = T)
  fileprj <- paste(name, ".prj", sep = "")
  prj <- readLines(fileprj, warn = FALSE)
  epsg <- st_crs(obj)$epsg
  authority <- paste(']', AUTHORITY["EPSG", "", epsg, "]]]', sep = ""))
  new_prj <- gsub("]]$]", authority, prj)
  writeLines(new_prj, fileprj)
}
```

Figura 26: Nueva función para guardar una capa vectorial en Shapefile.

```
st_write_shp_epsg(nucleosPobl25830, "nucleosPobl25830")

## Deleting layer `nucleosPobl25830' using driver `ESRI Shapefile'
## Writing layer `nucleosPobl25830' using driver `ESRI Shapefile'
## Writing 7457 features with 8 fields and geometry type Multi Polygon.
```

Figura 27: Guardar una capa vectorial usando la nueva función.

Figura 28: CRS en *QGIS* de una capa guardada en *R* con la nueva función definida.

```
library(raster)

setwd("~/datos/sigPr3/3.4. Ortofotografias")

ortofoto <- brick("ortofotoLineA.tif")
plotRGB(ortofoto, r = 1, g = 2, b = 3)
```



Figura 29: Cargar todas las bandas de un ráster.

```
setwd("~/datos/sigPr3/3.4. Ortofotografias")

ortofotoB1 <- raster("ortofotoLineA.tif")
ortofotoB2 <- raster("ortofotoLineA.tif", band = 2)
ortofotoB3 <- raster("ortofotoLineA.tif", band = 3)

ortofotoStack <- stack(ortofotoB1, ortofotoB2, ortofotoB3)
plotRGB(ortofotoStack, r = 1, g = 2, b = 3)
```



Figura 30: Cargar las bandas de un ráster una a una y definir un ráster multibanda.

```
writeRaster(ortofotoB1, "ortofotoB1k.tif", datatype = "INT1U",
            overwrite = TRUE)

writeRaster(ortofotoStack, "ortofotoStack.tif", datatype = "INT1U",
            overwrite = TRUE)
```

Figura 31: Guardar un ráster.

```
ortofoto25830 <- projectRaster(ortofoto, crs = "+init=epsg:25830")
```

Figura 32: Reproyectar un ráster.

#### 4.1.4. Guardar una capa ráster

Independientemente del número de bandas de ráster y de la estructura de datos de este, lo podemos guardar en un archivo mediante la función `writeRaster` (figura 31). Mediante el parámetro `datatype` se indica el tipo de los datos de las bandas del ráster<sup>4</sup>.

### 4.2. Reproyectar una capa

#### 4.2.1. Reproyectar una capa vectorial

Para una capa vectorial, una vez solventados los posibles problemas descritos en el apartado 4.1.1, reproyectarla se reduce a llamar a la función correspondiente (figura 23).

#### 4.2.2. Reproyectar una capa ráster

En la figura 32, se muestra cómo reproyectar un ráster. Hay que tener presente que en la operación se pierde información ya que la correspondencia no suele ser directa entre las celdas del ráster inicial y del resultado.

### 4.3. Componer un ráster virtual a partir de varios ráster

Mediante la función `gdalbuildvrt` (figura 33) podemos componer un ráster virtual a partir de los archivos de una carpeta (también permite indicarlos explícitamente).

El resultado lo almacena en un archivo, se puede obtener su información con la función `gdalinfo`. Podemos trabajar con este ráster como cualquier otro, mediante la función `brick`, y mostrarlo con `plotRGB` (figura 34).

---

<sup>4</sup>Los posibles tipos de datos se pueden consultar en <https://www.rdocumentation.org/packages/raster/versions/3.0-7/topics/dataType>.

```
library(gdalUtils)

setwd("~/datos/sigPr3/3.4. Ortofotografias/LineA")

gdalbuildvrt(gdalfile = "*.jp2", output.vrt = "raster.vrt")
```

Figura 33: Componer un ráster virtual a partir de los archivos de una carpeta.

```
setwd("~/datos/sigPr3/3.4. Ortofotografias/LineA")  
  
rb <- brick("raster.vrt")  
plotRGB(rb, r = 1, g = 2, b = 3)
```



Figura 34: Trabajar con un ráster virtual.

```
cbm <- crop(rb, municipio25830)

plotRGB(cbm, r = 1, g = 2, b = 3)
plot(st_geometry(municipio25830), add = TRUE)
```



Figura 35: Recortar una capa ráster.

## 4.4. Recortar una capa ráster

### 4.4.1. Mediante un rectángulo

En la figura 35, se puede observar que, aunque la capa vectorial no sea un rectángulo, recorta el ráster por el rectángulo envolvente, orientado de Norte a Sur.

## 4.5. Recortar una capa vectorial

Si recortamos una capa vectorial mediante otro vector mediante la función `st_crop`, en lugar del segundo vector, considera el mínimo rectángulo envolvente, orientado de Norte a Sur (figura 36).

Para considerar la capa vectorial propiamente dicha en el recorte, debemos realizar la intersección entre las dos capas mediante la función `st_intersection`, como se muestra en la figura 37.

## Bibliografía

- [BC19] Chris Brundsdon and Lex Comber. *An Introduction to R for Spatial Analysis & Mapping (Second Edition)*. SAGE, 2019.
- [Men19] Kurt Menke. *Discover QGIS 3.x*. Locate Press LLC, 2019.

```
rcm <- st_crop(nucleosPobl25830, municipio25830)

plot(st_geometry(municipio25830), axes = TRUE)
plot(st_geometry(rcm), col = "lightgreen", lwd = 1.5, add = TRUE)
```

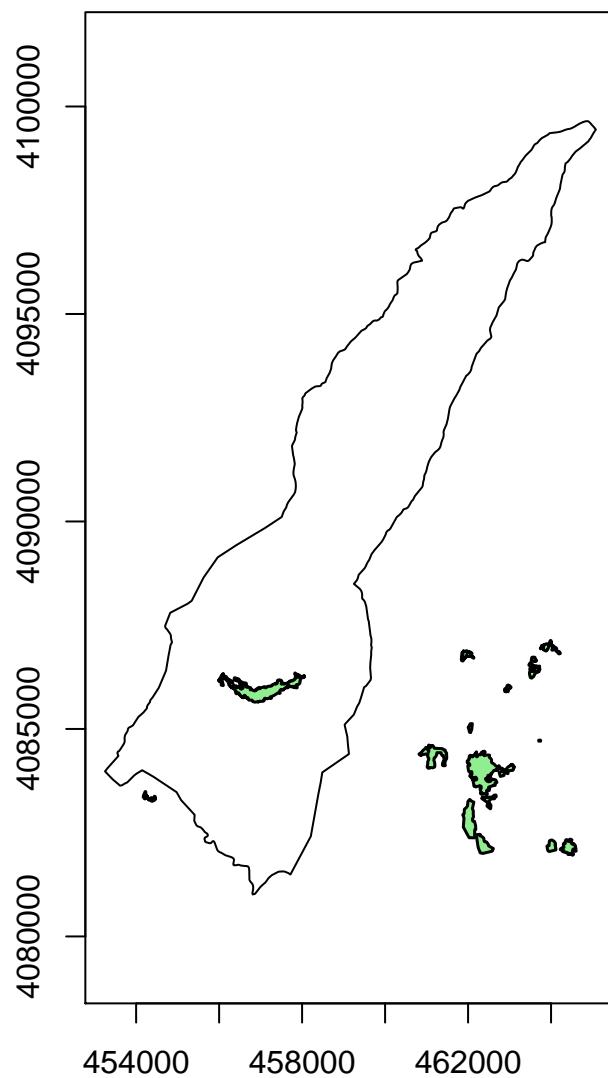


Figura 36: Recortar una capa vectorial mediante la función `st_crop`.

```
rcm_inter <- st_intersection(nucleosPobl25830, municipio25830)

plot(st_geometry(municipio25830), axes = TRUE)
plot(st_geometry(rcm_inter), col = "lightgreen", lwd = 1.5, add = TRUE)
```

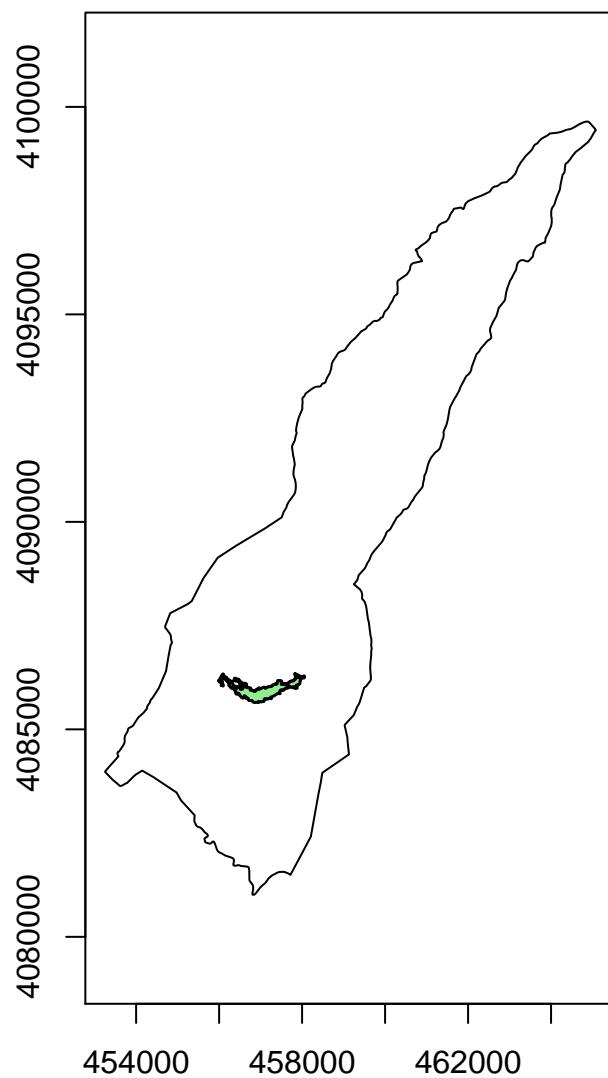


Figura 37: Recortar una capa vectorial mediante la función `st_intersection`.