

QGIS

Guía de ejercicios de QGIS Nivel Avanzado

Versión 202412

SERVICIO
GEOLÓGICO
COLOMBIANO



Guía de ejercicios de QGIS

Nivel Avanzado

Autores:

Samuel Fernando Mesa
sfmesa@sgc.gov.co
Jaime Alberto Garzón
jgarzon@sgc.gov.co

Revisor:

Luis Eduardo Salas
lesalas@sgc.gov.co



Esta obra es distribuida bajo licencia internacional *Creative Commons* Atribución/Reconocimiento igual 4.0.

Versión 202412
Bogotá D.C., Colombia

Historial de versiones

Versión	Fecha	Cambios
202412	12 de diciembre de 2024	Versión inicial de la guía

Contenido

Introducción	1
Ejercicio 1. Entorno de procesamiento, edición en capas activas y procesamiento por lotes	2
1. Introducción	3
2. Datos, software y recursos necesarios	3
3. Configuración del entorno de procesamiento de QGIS	4
4. Edición objetos de la capa activa	7
5. Procesamiento por lotes	9
6. Entregable del ejercicio	13
Ejercicio 2. Modelador gráfico de flujos de procesos	14
1. Introducción	15
2. Datos, software y recursos necesarios	15
3. El modelador de flujo de procesos	16
4. Ejecutar el modelo y comprobar las salidas	18
5. Entregable del ejercicio	28
Ejercicio 3. Desarrollo de Scripts con Python en el entorno de QGIS	29
1. Introducción	30
2. Datos, software y recursos necesarios	30
3. Conociendo la sintaxis de Python	31
4. Procesamiento desde la consola de Python	39
5. Entregable del ejercicio	44
Ejercicio 4. Análisis espacial vectorial, scripts con R	45
1. Introducción	46
2. Datos, software y recursos necesarios	46
3. Análisis espacial vectorial	47
4. Generación de estadísticas y gráficos	49
5. Entregable del ejercicio	57
Ejercicio 5. Análisis espacial raster	58
1. Introducción	59
2. Datos, software y recursos necesarios	59
3. Análisis espacial raster	60
4. Entregable del ejercicio	73
Ejercicio 6. Publicación de Geoservicios WMS, WMTS, WFS y WCS	74
1. Introducción	75
2. Datos, software y recursos necesarios	75
3. Geoservicios OGC	76
4. Entregable del ejercicio	86
Créditos y Atribuciones	87

Introducción

El Servicio Geológico Colombiano (**SGC**) lidera la investigación científica básica y aplicada sobre el potencial de los recursos del subsuelo, monitorea amenazas de origen geológico, administra información geocientífica y garantiza la gestión segura de materiales nucleares y radiactivos en Colombia. Además, coordina proyectos de investigación nuclear y opera el Reactor Nuclear de la Nación, todo bajo el marco de las disposiciones constitucionales.

En consonancia con el Plan Nacional de Desarrollo 2022-2026 “Colombia Potencia Mundial de la Vida”, el **SGC** se ha comprometido a promover el uso de software libre, código abierto y tecnologías digitales emergentes, como se establece en el artículo 144 de dicho plan. En este contexto, hemos impulsado la implementación de software libre mediante capacitaciones en Sistemas de Información Geográfica (**SIG**), utilizando **QGIS** en niveles básico, intermedio y avanzado.

Como parte de nuestra contribución a la comunidad internacional de **QGIS**, hemos desarrollado una serie de ejercicios aplicados al área de las geociencias. Estos ejercicios están diseñados para compartir conocimientos, fomentar la colaboración y resaltar el potencial del software libre en la investigación y gestión geológica. Este esfuerzo refleja nuestro compromiso con la comunidad de software libre y con el fortalecimiento del acceso a tecnologías que promuevan el desarrollo sostenible y la innovación.

Esta guía de nivel avanzado está diseñada para usuarios que desean profundizar en las capacidades de **QGIS**. A través de seis ejercicios prácticos, exploraremos desde el geoprocesamiento avanzado de datos hasta la publicación de servicios geospaciales. El contenido abarca aspectos fundamentales para el análisis geoespacial profesional, comenzando con técnicas de procesamiento por lotes y edición avanzada, continuando con el modelado de flujos de trabajo y la programación mediante **Python** y **R** en el entorno **QGIS**. Profundizaremos en el análisis espacial tanto vectorial como raster, finalizando con la publicación de geoservicios OGC.

La estructura de cada ejercicio sigue un formato consistente que incluye una introducción conceptual, seguida por la descripción de los recursos necesarios, un desarrollo detallado paso a paso y los entregables esperados. Esta organización facilita un aprendizaje progresivo y permite verificar los conocimientos adquiridos durante el proceso.

Ejercicio 1

Entorno de procesamiento, edición en capas activas y procesamiento por lotes

1. Introducción

Una de las funcionalidades más destacables de los sistemas de información geográfica es su entorno de procesamiento, diseñado para ofrecer una serie de algoritmos organizados y debidamente documentados, así como la disponibilidad para integrar de manera flexible y desarrollo gráfico en flujos de proceso, en líneas de comandos y soporte para el desarrollo de *scripts* flexibles, lo cual permite personalizar y extender procesamientos a gran volumen de datos.

QGIS incluye un complemento de procesamiento preinstalado, que ofrece acceso a cientos de algoritmos nativos y a la integración de herramientas de terceros como **GRASS GIS**, **GNU R**, **SAGA GIS**, **Orfeo Toolbox**, **Whitebox Tools**, **PDAL**, entre otros (ver más información en **GeoTux**). Este complemento se puede integrar en diferentes interfaces gráficas, como el modelador de flujos de trabajo, o utilizarse en línea de comandos o como biblioteca en diversos lenguajes de programación, como **Python**.

2. Datos, software y recursos necesarios

Para el desarrollo correcto del presente ejercicio práctico se requiere la revisión de la sesión de conceptos teóricos de la semana 1 del Nivel Avanzado de **QGIS**:

- Revisión de los contenidos teóricos de la primera semana (grabación disponible en la **Plataforma de Aulas Virtuales** del **SGC**)([Enlace web](#)).

- **QGIS Desktop** versión 3.34.x o superior.

Instrucciones de instalación con **OSGeo4W**, incluye la mayor parte de software como **GRASS**, **PCRaster** y **SAGA GIS**: <https://link.aulageo.xyz/osgeo4w>

- **GNU R** versión 4.1.3:

<https://cran.r-project.org/bin/windows/base/old/4.1.3/R-4.1.3-win.exe>

- **Datos del ejercicio**: Corresponde a los datos del Mapa Geológico de Colombia del **SGC** y del Marco Geoestadístico Nacional del DANE: Descargar **datos**.

3. Configuración del entorno de procesamiento de QGIS

Se recomienda crear un perfil de usuario dedicado para el desarrollo de las prácticas y ejercicios del entrenamiento avanzado. Esto permite mantener un entorno de trabajo nuevo, y facilita la gestión de los datos y proyectos, para mayor información consultar la [documentación oficial de QGIS](#).

El entorno de procesamiento se ofrece como un complemento nativo de **QGIS**. Para extender e integrar herramientas de terceros o proveedores de algoritmos se requiere la instalación de los respectivos complementos que facilitan la configuración adecuada.

3.1. Configuración del complemento de procesamiento

Paso 1. Una vez creado el perfil de usuario, verificar el complemento de nombre **Processing** se encuentre habilitado. Posteriormente, ingresar a las opciones desde el menú de configuración y seleccionar la pestaña de **Procesos**.

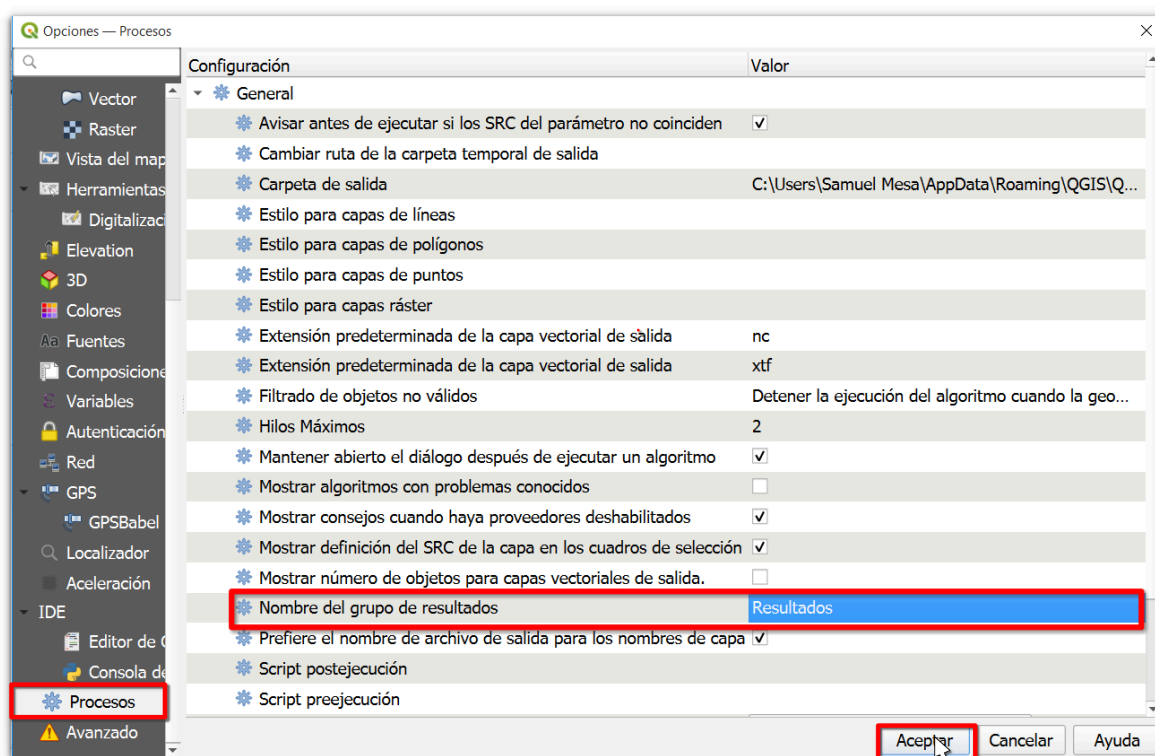


Imagen 1. Configuración del entorno de procesado

Revisar las opciones disponibles para el marco de procesamiento de **QGIS**, configurar como **Nombre del grupo de resultados** a **Resultados** en la rama de nombre **General**.

3.2. Instalar y configurar proveedores de algoritmos de terceros

Para lograr una integración correcta de software de terceros y ampliar los algoritmos en el entorno de procesamiento, es fundamental contar con un proveedor que facilite la configuración de dicha extensión. Las siguientes instrucciones proporcionan las herramientas necesarias para el desarrollo de este nivel del entrenamiento.

Configuración de GRASS, SAGA GIS y PCRaster

Por defecto en las instrucciones de **OSGeo4W** se instalan las herramientas de software como **GRASS GIS**, **SAGA GIS** y **PCRaster**. Es importante contar con estas versiones de software para garantizar un correcto funcionamiento, por lo que se recomienda realizar la configuración con esta utilidad de gestión de paquetes.

Paso 2. Realizar la instalación de los proveedores de algoritmos desde el menú de complementos, el listado de los proveedores a instalar son los siguientes:

- **GRASS GIS provider**
- **Processing Saga NextGen Provider**
- **PCRaster Tools**
- **Processing R Provider**
- **WhiteboxTools for QGIS**

Paso 3. Una vez instalados los proveedores, ingresar al menú de configuraciones, opciones y finalmente seleccionar la pestaña de **Procesos**. Desde aquí, configurar en Windows las rutas de las herramientas a usar y otras opciones:

- **GRASS:** habilitar **r.external** y **v.external** para hacer más rápido la importación de archivos a los formatos nativos de GRASS.
- **R:** Habilitar la versión de 64 bits. Por defecto **QGIS** habilita las rutas por defecto. De requerirlo, cambiar a la carpeta a la versión **4.1.3**.
- **SAGA GIS:** habilitar la optimización para importar y exportar, así como especificar la ruta de la carpeta a **C:/OSGeo4W/apps/saga9**.

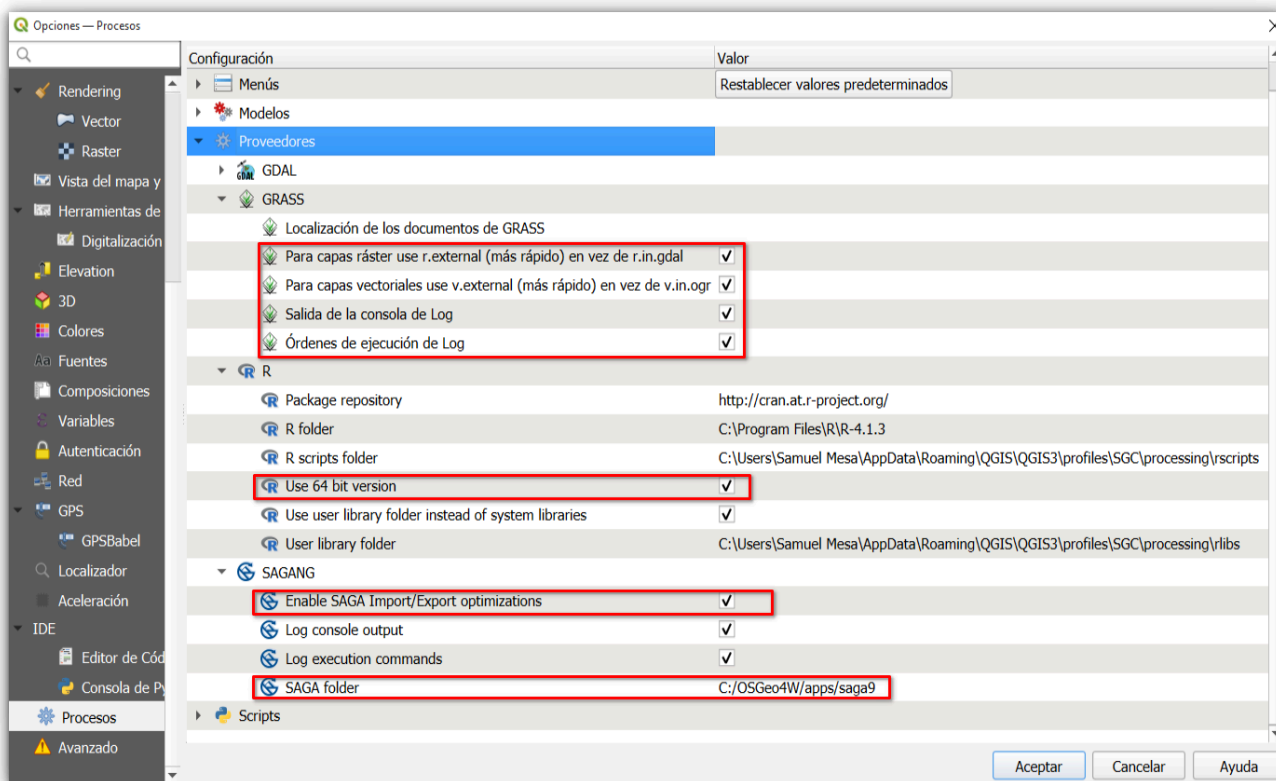


Imagen 2. Configuración de proveedores de GRASS GIS, R, SAGA GIS y PCRaster

Configuración de Whitebox Tools

Whitebox Tools es una herramienta altamente funcional que cuenta con una amplia variedad de algoritmos disponibles, diseñada para garantizar un procesamiento estable y eficiente en todo momento.

- **Whitebox Tools:** software ofrecido en una versión portable disponible en el siguiente enlace Web.

<https://www.whiteboxgeo.com/download-direct/>

Paso 4. Para comenzar, es necesario extraer el archivo `WhiteboxTools_win_amd64.zip` y ubicar la carpeta de nombre **WBT** dentro del directorio de **OSGeo4W** en la siguiente ruta: **C:\OSGeo4W\apps**. Posteriormente, se debe configurar el proveedor de algoritmo siguiendo las instrucciones que se muestran en la Imagen 3. La ruta específica para esta configuración es **C:/OSGeo4W/apps/WBT/whitebox_tools.exe**.

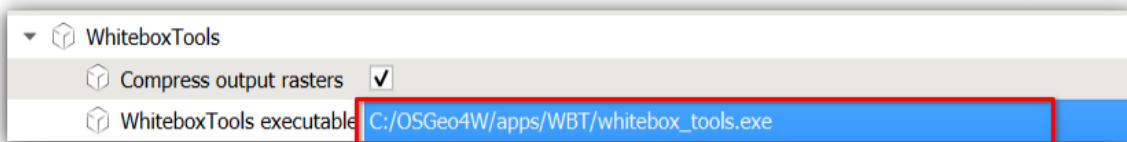


Imagen 3. Configuración del proveedor **Whitebox Tools**

Por último, verificar la configuración de los proveedores de algoritmos externos desde el menú **Procesos** » **Caja de herramientas**.

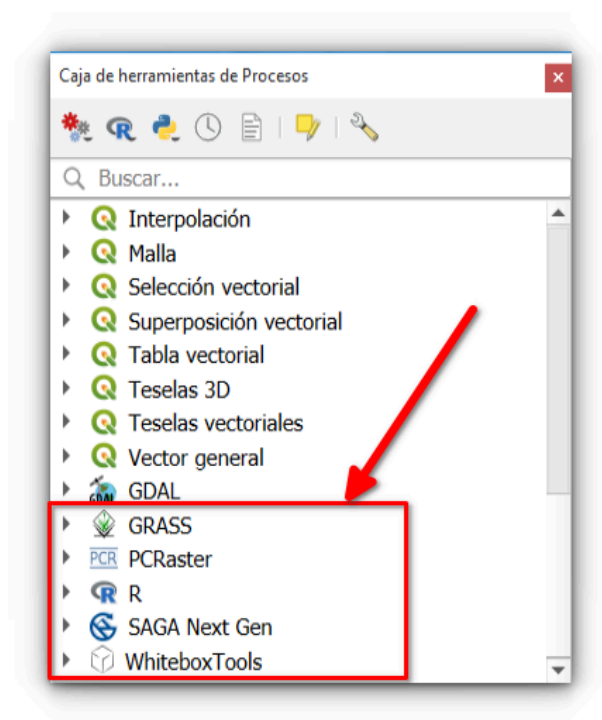


Imagen 4. Despliegue de proveedores de algoritmos

4. Edición objetos de la capa activa

En la mayoría de los casos, las salidas de la ejecución de las herramientas de los algoritmos generan un nuevo conjunto de datos, lo que a menudo implica un paso adicional de sobrescribir los datos de entrada. Para evitar esto, **QGIS** ofrece la funcionalidad de edición de objetos en la capa activa, permitiendo así realizar el cambio directo sobre la entrada y mantener la estructura original de los datos.

Paso 5. Para comenzar, es necesario conectar la carpeta que contiene los datos y cargar la capa de **departamento** de la base de datos `DANE-MGN-2018.gpkg`. Luego, se debe cargar la **Geodatabase del Mapa Geológico de Colombia 2023** del **SGC** con el nombre `mgc2023.gdb`. Es importante arrastrar el nombre de la base de datos a la vista de mapa, deshabilitar la opción de **Show empty vector layers** y habilitar la opción

de **Añadir capas a un grupo**. Por último, se recomienda desactivar la visualización para las capas que comiencen por **mbi_**, **ml_** y la capa de **Transparencia**.

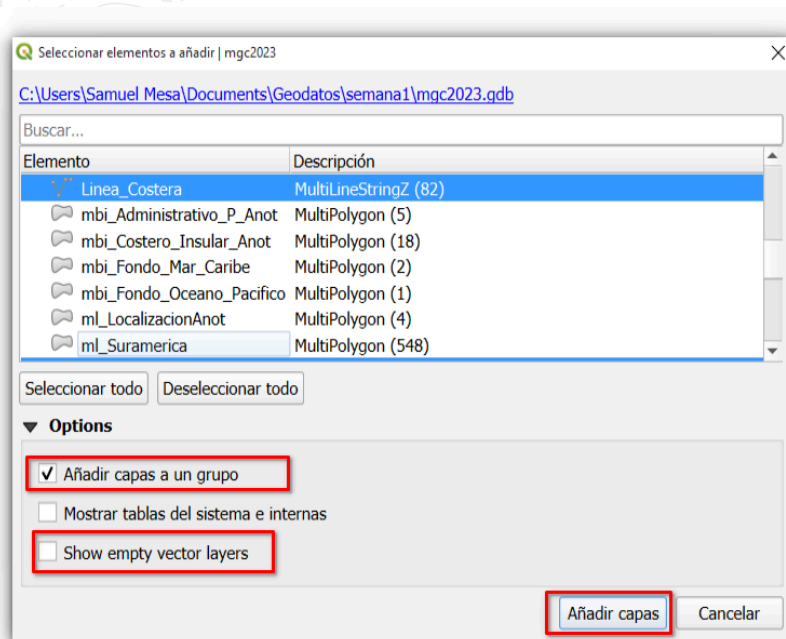


Imagen 5. Cargue de la base de datos del mgc2023.gdb

Paso 6. Realizar el recorte de una de las capas del MGC como **Fallas** con la capa de **departamento**. La herramienta a usar es **cortar** del grupo de proveedores de software de nombre **Superposición vectorial** de **QGIS**.

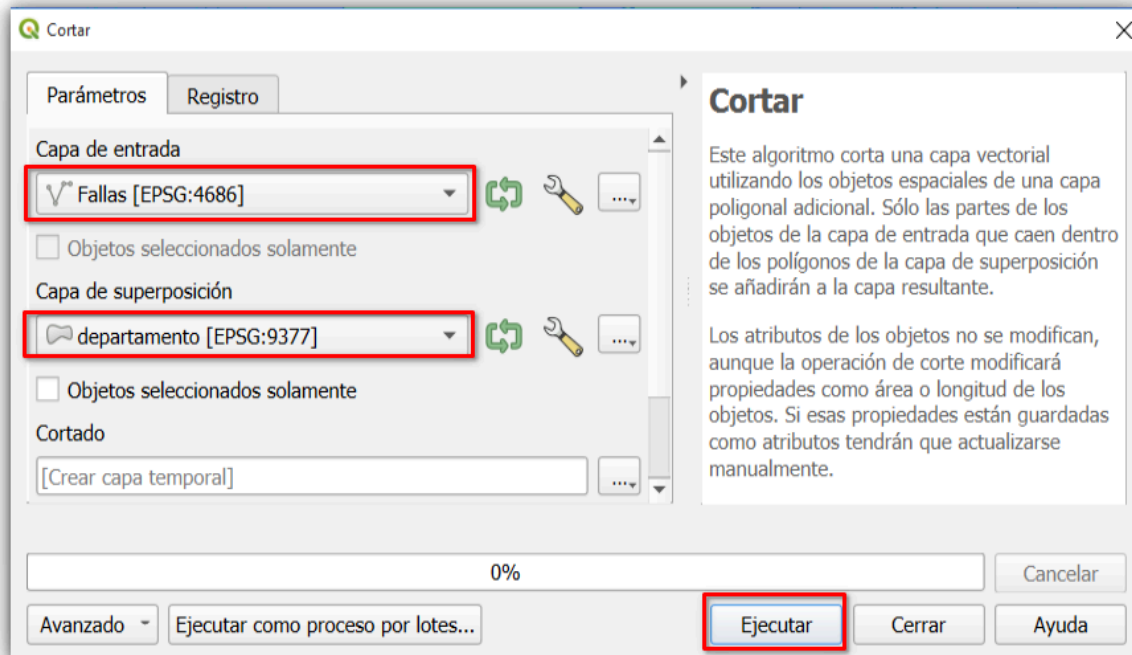


Imagen 6. Herramienta de cortar capas vectoriales

El algoritmo no puede ejecutarse debido a que la capa de **departamento** contiene geometrías inválidas que necesitan ser corregidas. Para solucionarlo, se debe utilizar la herramienta llamada **Corregir geometrías** en la rama de **Geometría vectorial** en la caja de herramientas de **QGIS**. Además, es necesario habilitar la herramienta **Editar objetos de la capa activa** en la caja de herramientas, ya que es necesario editar directamente la capa de **departamento**. Antes de ejecutar la herramienta, es importante seleccionar la capa de **departamento** como capa activa en el panel de capas.

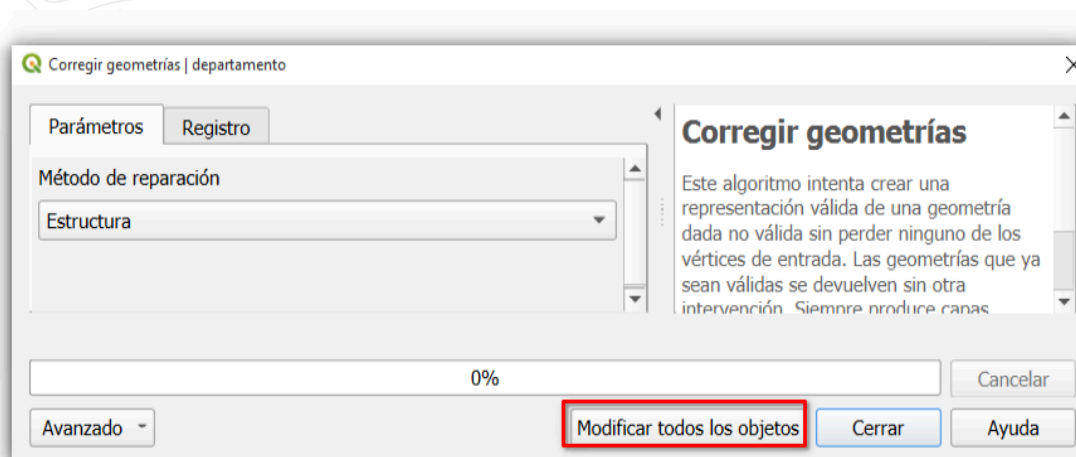


Imagen 7. Corregir geometrías en capa activa del departamento

Aceptar los cambios de corrección conmutando la edición en la capa del **departamento**. Ejecutar nuevamente la herramienta de recorte y recuerde deshabilitar la opción de **Editar objetos de la capa activa** para mostrar todas las herramientas disponibles. Observar que el resultado se mostrará en un nuevo grupo de nombres **Resultados** en el panel de capas.

Paso 7. Guardar el proyecto de **QGIS** con nombre `semana1.qgz` en la carpeta de los datos.

5. Procesamiento por lotes

QGIS facilita el procesamiento masivo de capas con su herramienta de procesamiento por lotes. Esta herramienta permite ejecutar múltiples procesos en una sola ventana para un gran número de capas de entradas y salidas, simplificando el trabajo con grandes conjuntos de datos. La clave de su eficiencia radica en la utilización de constructor de expresiones para generar autocompletado en todos los parámetros requeridos por una herramienta. Esto elimina la necesidad de escribir manualmente los valores para cada capa, ahorrando tiempo y reduciendo errores.

5.1. Conversión de coordenadas por lotes de las capas de una base de datos

Se plantea en el ejercicio el flujo de procesamiento de recorte de las capas de la base de datos espacial del **MGC 2023** en el contexto geográfico de un departamento, comenzando con la re-proyección masiva de coordenadas geográficas a coordenadas planas **MAGNA-SIRGAS Origen Nacional**, seguido de la corrección de geometrías y, finalmente, el recorte de las capas con una capa de polígonos. Todo esto con el objetivo de almacenar la información en una única base de datos en formato **GeoPackage**.

Paso 8. Ejecutar desde la caja de herramienta el algoritmo de nombre **Vector General** » **Reproyectar capa** y hacer clic en el botón **Ejecutar como proceso por lotes...** Otra forma de acceder a estas opciones es haciendo clic derecho sobre la herramienta en la caja de herramientas y luego seleccionar la opción en el menú contextual. Observar que la interfaz cambia a un despliegue tabular, donde cada columna define los parámetros y las filas representan el procesamiento para cada capa vectorial. Para completar el diligenciamiento de todas las capas de entrada de la vista del mapa, es necesario seleccionar la opción de **(1) Autorellenar...** del parámetro de la capa de entrada y **(2) Seleccionar de Open Layers...**. Esto desplegará una ventana donde se podrán seleccionar las capas del MGN 2023, recordando excluir la capa resultado del **recorte** y el **departamento**. Por último, **(3)** hacer clic en **Aceptar**.

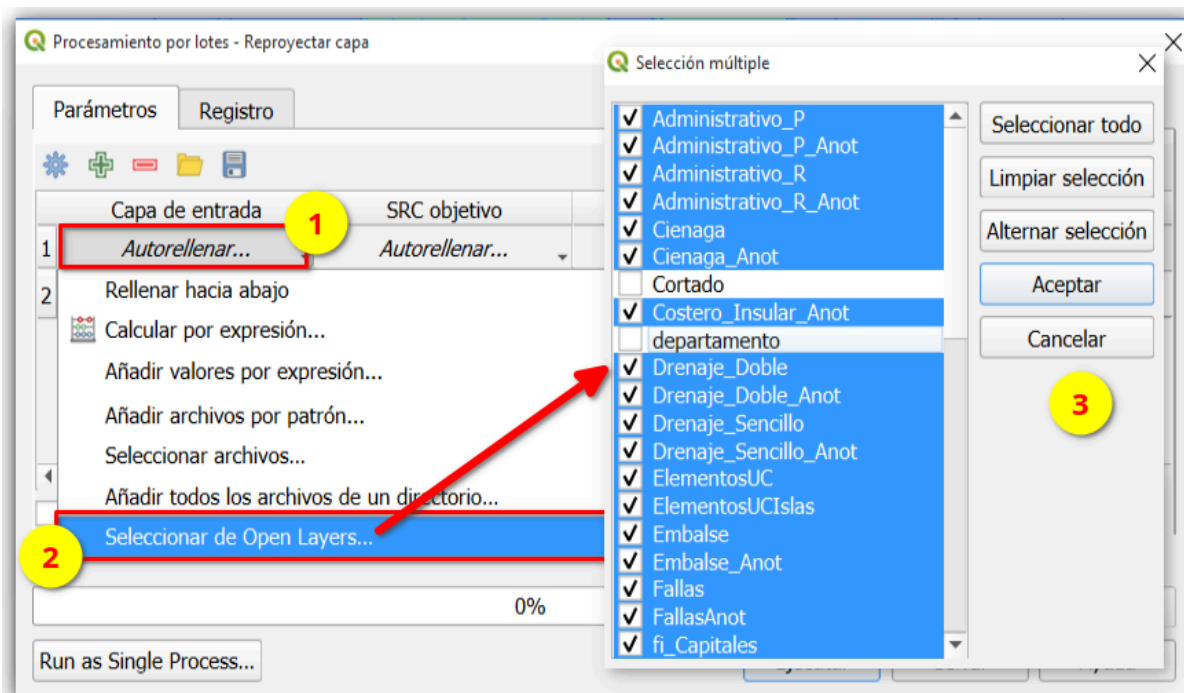


Imagen 8. Autorellenado de los parámetros de entrada en el modo de procesamiento por lotes

Para establecer el parámetro de **SRC Objetivo**, debes seleccionar en la primera fila el código **EPSG:9377** y luego hacer clic en el botón de **Autorellenar**, eligiendo la opción **Rellenar hacia abajo**. De esta manera, se definirán automáticamente todos los parámetros necesarios para las capas. De igual forma, para la primera capa especificar la salida del parámetro **Reproyectada** en formato **GPKG**. Posteriormente, aparecerá una ventana en la que se puede elegir la manera de **Rellenar con valores de parámetro** y seleccionar la opción de **Capa de entrada**. Aunque se va a usar una expresión para redireccionar la salida a un único **GeoPackage**, lo anterior es necesario para lograr autocompletar todos los valores.

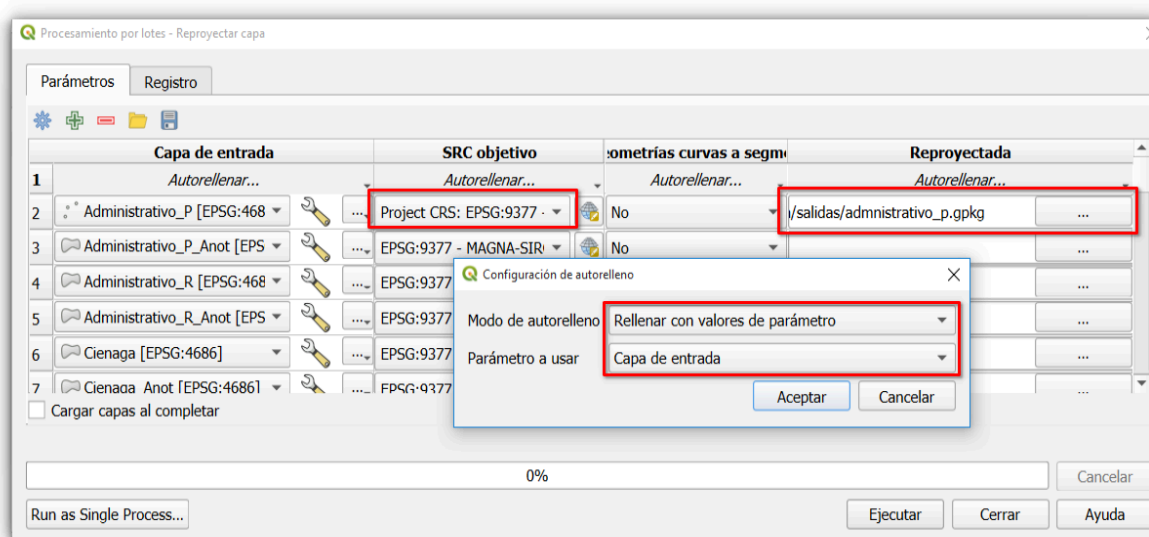


Imagen 9. Rellenar parámetros de salida

Para generar la salida, es necesario crear una carpeta de nombre **salidas** en el directorio del proyecto. Para asegurarse de que los nombres de las salidas de la capa sean correctos, es necesario utilizar el constructor de expresiones, específicamente en los parámetros de **Reproyectada**, **Autorellenar** y la opción **Calcular por expresión**. Para ello, ingrese la siguiente expresión en el constructor de consultas:

```
'ogr:dbname=\' || @project_folder || '/salidas/reproyectada.gpkg\'
table=\' || lower(layer_property( @INPUT, 'name')) || \' (geom)'
```

La anterior expresión crea una base de datos de **GeoPackage** en la carpeta salidas del proyecto y agrega todas las salidas del procesamiento con el nombre en minúsculas de la capa de entrada.

Verificar en la ventana de constructor de expresiones, que la previsualización entrega una salida correcta de la capa y la base de datos.

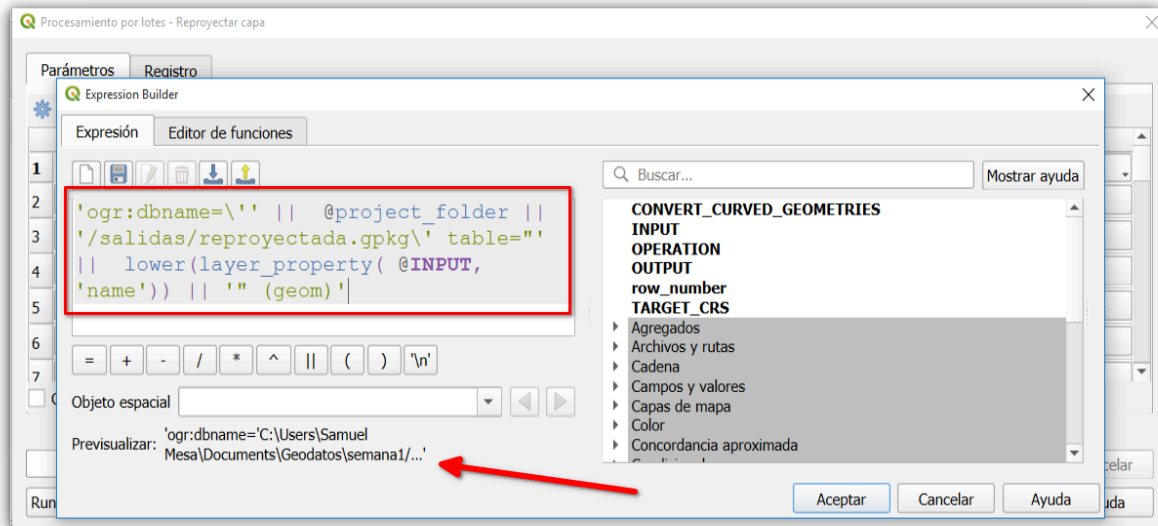


Imagen 10. Expresión para redirigir la salida a una base de datos GPKG

Finalmente, hacer clic en el botón **Ejecutar** para realizar el procesamiento masivo de los datos. Es necesario verificar en el **Registro** del log de salida, los errores o avisos generados.

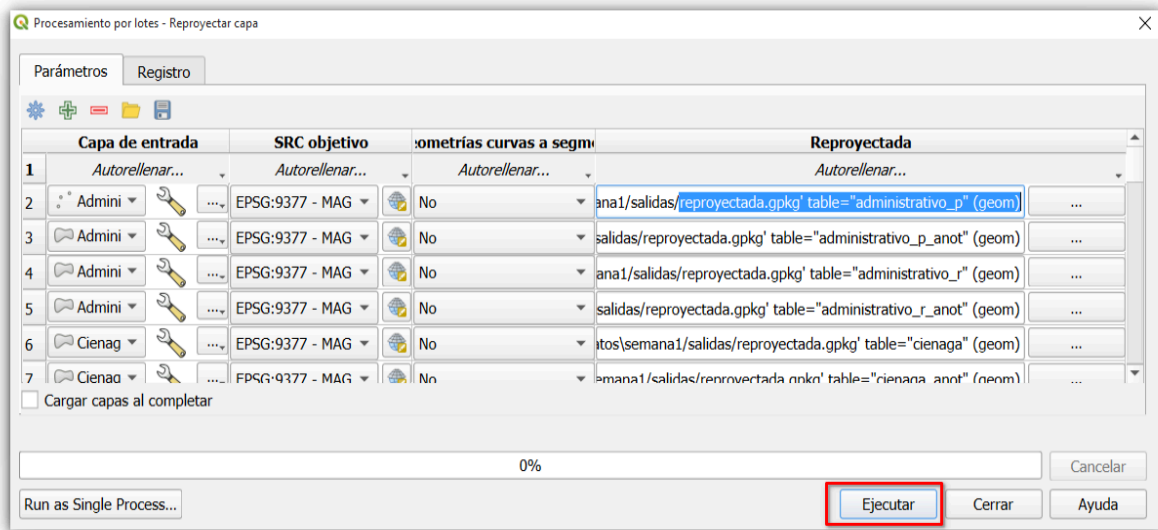


Imagen 11. Configuración de la reproyección en modo de procesamiento por lotes

Paso 9. Para continuar con el flujo de procesos, quitar el grupo de las capas de entrada del `mgc2023.gdb` y agregar la nueva base de datos `reproyectada.gpkg` al proyecto, este definirá la entrada para la siguiente herramienta en modo de procesamiento por lotes.

6. Entregable del ejercicio

Continuando con el flujo de procesos y procesamiento por lotes para generar el recorte masivo de la base de datos de `mgc2023.gdb` en el contexto de la capa de departamento se requiere seguir con los siguientes pasos:

- Sobre la base de datos **GeoPackage** `reproyectada.gpkg` generada en el Paso 8, realizar la corrección de geometrías en el modo de procesamiento por lotes y almacenar las salidas en una única base de datos de nombre `corregida.gpkg`.
- Para la base de datos obtenida anteriormente de corrección de geometrías realizar el recorte final con la capa de departamento. La entrega final del ejercicio corresponde a esta base de datos de nombre `mgc2023-cundinamarca.gpkg`, las especificaciones técnicas que se deben cumplir en este entregable son:
 - Sólo se entrega la base de datos **GeoPackage** que corresponde al recorte de todas las capas del MGN 2023 en el contexto de la capa departamento, el nombre del contenedor de las salidas es `mgc2023-cundinamarca.gpkg`.
 - Todas las capas del entregable deben corresponder con el Sistema de Coordenadas MAGNA SIRGAS Origen Nacional.
 - Todas las capas del entregable deben tener geometrías válidas.

Ejercicio 2

Modelador gráfico de flujos de procesos

1. Introducción

Dentro de las diversas funcionalidades que ofrece el entorno de procesamiento de **QGIS**, destaca el modelador gráfico de flujo de procesos. Esta herramienta resulta útil, ya que permite construir un modelo que representa la secuencia lógica de herramientas interconectadas, con sus respectivos parámetros de entrada y salida. Gracias a esto, se logra una solución eficiente para la ejecución de una serie de algoritmos de forma secuencial.

En este ejercicio se aborda el uso general del diseñador gráfico de modelos de procesos, la configuración de almacenamiento y despliegue, especificar parámetros de entradas y salidas del modelo, integrar un flujo de algoritmos y procesos, ramas condicionales de control, especificar variables y sustitución, documentación y almacenamiento del modelo en el proyecto.

2. Datos, software y recursos necesarios

Para el desarrollo correcto del presente ejercicio práctico se requiere la revisión de la sesión de conceptos teóricos de la semana 2 del Nivel Avanzado de **QGIS**:


- Revisión de los contenidos teóricos de la segunda semana (grabación disponible en la **Plataforma de Aulas Virtuales del SGC**)([Enlace web](#)).
- **Datos del ejercicio**: Corresponde a los datos de la semana 1 del Mapa Geológico de Colombia del **SGC** y del Marco Geoestadístico Nacional del **DANE**: Descargar [datos](#).

3. El modelador de flujo de procesos

Continuando la exploración de funcionalidades del entorno de procesamiento de **QGIS** para el recorte y almacenamiento masivo de una base de datos espacial, incluyendo los geoprocesamientos de reproyección del **SRC** y corrección de geometrías, se propone integrar esta secuencia de algoritmos en un modelo gráfico de flujo de procesos. Este modelador permitirá comprender la construcción de nuevas herramientas y la ejecución eficiente de tareas de gestión y análisis de datos.

3.1. Configuración inicial

Paso 1. Guardar el proyecto de nombre `semana2.qgz` en la carpeta de los datos. Recordar que este directorio constituirá el proyecto configurado con las rutas relativas, lo cual permite la apertura en otros entornos de **QGIS**. Agregar los datos vectoriales de departamento del **DANE** y la base de datos del **MGC2023** del **SGC**.

Paso 2. Iniciar el modelador gráfico de procesos, desde el menú de **Procesos** » **Diseñador de modelos..** o desde la barra de la caja de herramientas del entorno de procesamiento y seleccionar el botón de Modelos .

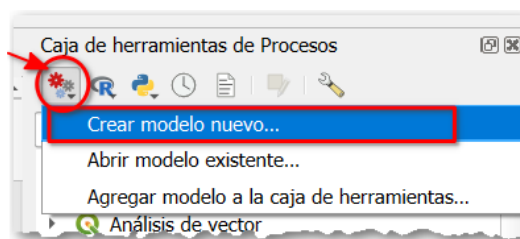


Imagen 1. Crear un modelo de flujo de procesos

Paso 3. En el modelador de flujos de procesos, ingresar el nombre del modelo como **Geoprocesamiento recortado** y el grupo o rama de almacenamiento como **SGC** en el panel de propiedades del modelo (Ver Imagen 2). Esto permitirá almacenar, organizar el despliegue y buscar el modelo en la caja de herramientas del entorno de procesamiento.

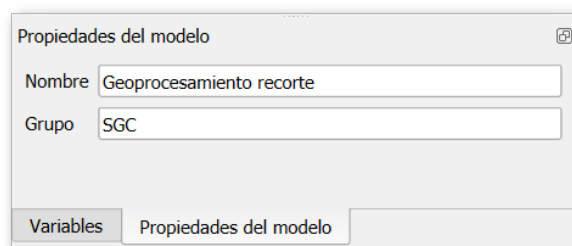



Imagen 2. Propiedades del modelo

Para almacenar el modelo en el proyecto de **QGIS**, seleccionar el menú **Modelo** » **Guardar modelo en el proyecto** o hacer clic en el botón correspondiente en la barra de herramientas . De esta manera, el modelo será almacenado y embebido directamente en el proyecto de **QGIS**, lo que permite acceder a él al abrir el proyecto. De preferir almacenarlo en el perfil de usuario, seleccionar **Guardar** en el formato ***.model3**.

3.2. Entradas de los modelos

El modelador de procesos cuenta con la mayoría de los controles gráficos de la interfaz de las herramientas de geoprocésamiento, incluyendo capas de entrada para vectores, raster, mallas, nubes de puntos, números, cadenas de texto, opciones de selección, tablas y extensiones geográficas, entre otros. Esto permite a los usuarios configurar y personalizar los parámetros de la herramienta y enlazarlos a los diferentes flujos del modelo.

Paso 4. Desde el panel de entradas hacer doble clic o arrastrar al lienzo del modelo el parámetro de nombre **Capa vectorial** (1) y luego definir en (2) la descripción como **Capa a recortar**, tipo de geometría a **Cualquier tipo de geometría** y **Obligatorio** (ver Imagen 3).

En la pestaña de **Comentarios** de cada parámetro, se pueden agregar comentarios detallados sobre los parámetros del modelo, lo que resulta útil para la auto-documentación de la herramienta o para exportar el proceso en formatos de imagen. Estos comentarios permiten proporcionar información adicional sobre el flujo de procesos del algoritmo generado, lo que ayuda a otros usuarios a entender cómo funciona la herramienta y cómo se utiliza.

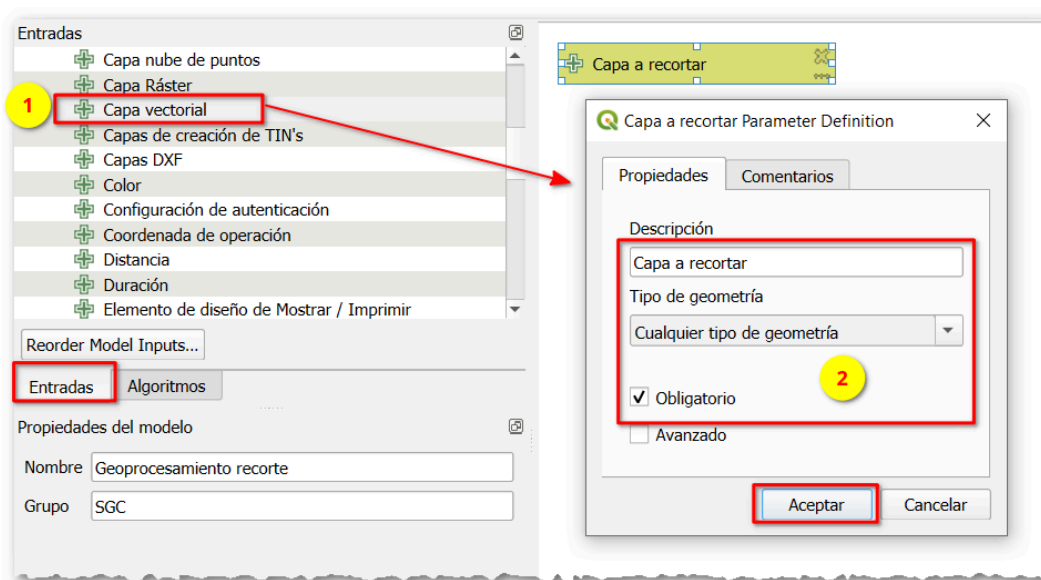


Imagen 3. Definir parámetro de entrada del modelo

3.3. Algoritmos del modelo

El modelador gráfico de procesos soporta la mayor parte de los algoritmos de la caja de herramientas, principalmente los nativos del propio software. Aquellos no soportados se deben básicamente por algunas propiedades y sintaxis de la herramienta. De igual forma, se habilitan otra serie de herramientas que son propias del modelador, orientadas al control, definir variables en el modelo y condicionales del flujo de control.

3.4. Especificar un algoritmo en el modelo

Similar a la definición de parámetros del modelo de entradas, para especificar una herramienta consiste en arrastrarla al lienzo o hacer doble clic sobre el algoritmo para integrarla al flujo de procesos, definiendo por lo mínimo los parámetros obligatorios.

Paso 5. Hacer doble clic sobre el algoritmo de **Comprobar validez** y definir como capa de entrada la **Capa a recortar**, luego definir las salidas de nombres temporales para los resultados de **válido** y **no válido**. Estas últimas se desplegarán directamente en la vista de mapa, lo que permite verificar y depurar el modelo.

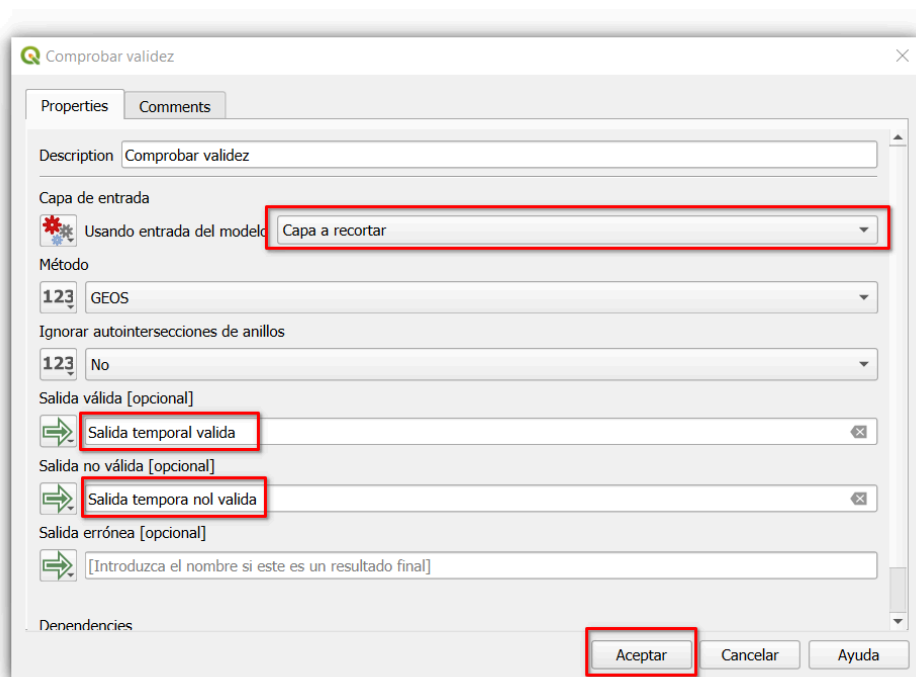


Imagen 4. Algoritmo de comprobar validez

4. Ejecutar el modelo y comprobar las salidas

Al definir ramas en el flujo de procesos, se recomienda validar y evaluar las salidas, siguiendo el enfoque del desarrollo de software en el que se compila el programa,

por último se verifican o validan los resultados. Esto facilita la depuración y ajuste del modelo.

Paso 6. Guardar el modelo en el proyecto y ejecutar desde el menú **Modelo** » **Ejecutar modelo...**, o hacer clic en el icono correspondiente ►. Esto abrirá una ventana de algoritmos que solicitará el parámetro de entrada. En este caso, comprobar las capas de **UC** y **Fallas** del **MGC2023**, para los demás parámetros almacenar por defecto y de forma temporal. Verificar los resultados en la vista del mapa y la tabla de atributos para cada resultado, lo que permitirá identificar qué capa requiere corrección geométrica y cuál no.

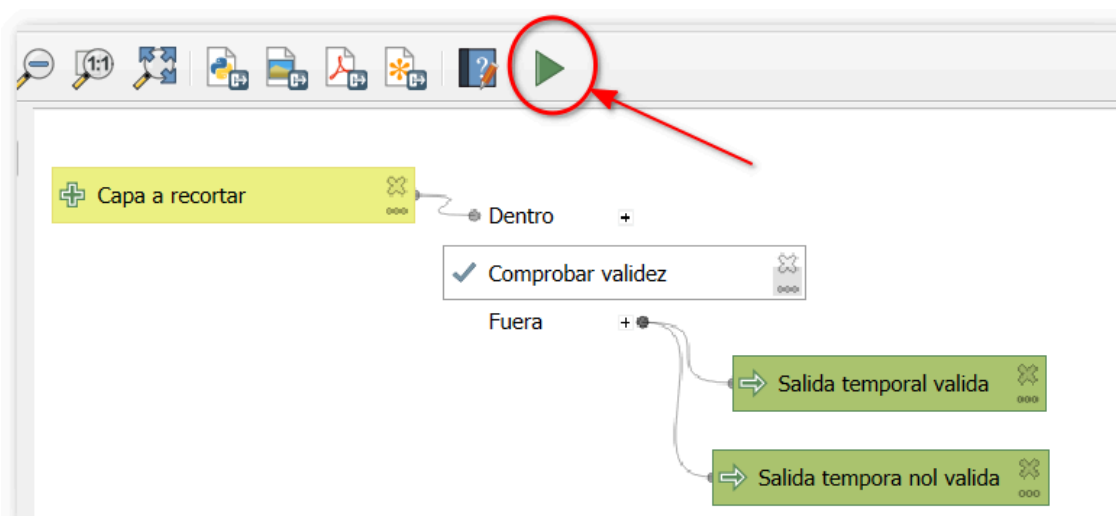


Imagen 5. Rama del flujo de procesos de comprobar validez

4.1. Rama condicional

Corresponde a uno de los algoritmos propios del modelador gráfico de procesos, y permite bifurcar ramas de salidas a partir de expresiones condicionales de los parámetros del modelo. La definición de la secuencia posterior de algoritmos para cada rama, requiere habilitar la opción de **Dependencias** y seleccionar la correspondiente rama condicional.

Paso 7. Agregar al flujo el algoritmo del modelador de nombre **Rama condicional**. Como se muestra en la Imagen 6 definir en (1) la descripción correspondiente a la herramienta de nombre **Rama condicional valido**, en (2) con el botón de **Añadir dos entradas**, ingresar los nombres de las ramas en (3) como **Valido** y **NoValido**. En (4) ingresar al constructor de expresiones y definir la expresión correspondiente de cada rama (5), esto a partir de la variable de salida del algoritmo, para este caso como **@Comprobar_validez_INVALID_COUNT**. En el botón de dependencias (6) seleccionar el algoritmo de **Comprobar validez** para conectar la secuencia en el flujo de procesos.

Condición de comprobación de geometrías inválidas

Para continuar la secuencia lógica del flujo de procesos, es necesario conectar las herramientas correspondientes en cada rama condicional.

Paso 8. Agregar el algoritmo de **Corregir geometrías** al modelo y seleccionar los parámetros como se muestra en la Imagen 7. Se recomienda eliminar las salidas anteriores y temporales del modelo. Depurar la rama con una salida temporal y seleccionar como dependencia la rama condicional de **NoValido**.

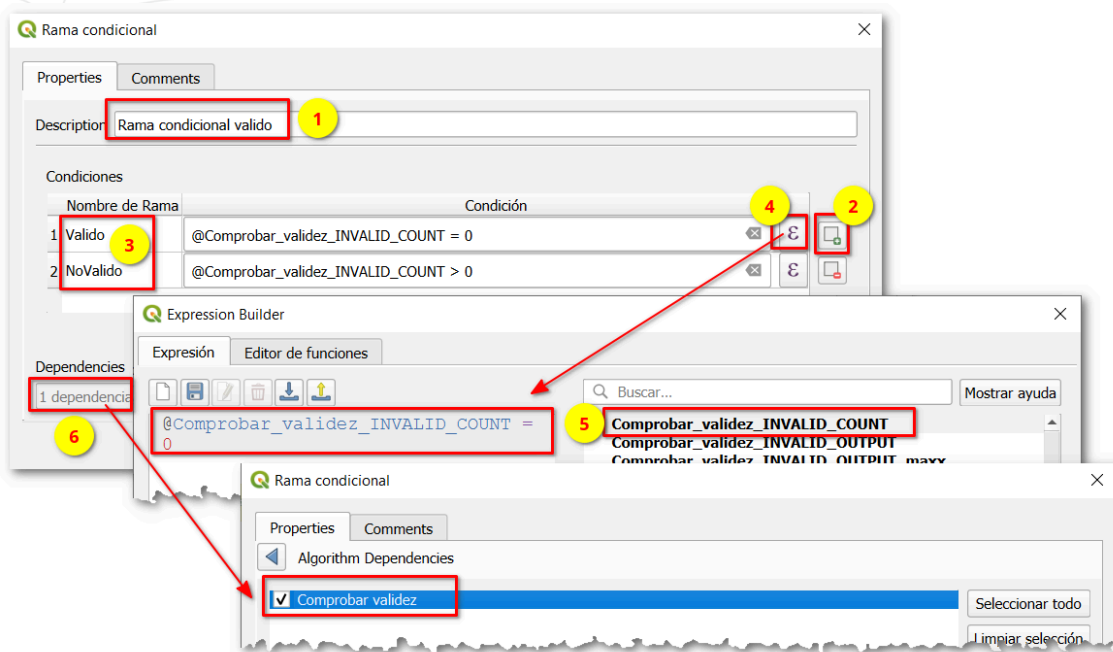


Imagen 6. Rama condicional de geometrías válidas.



Imagen 7. Corregir geometrías en rama condicional no válido.

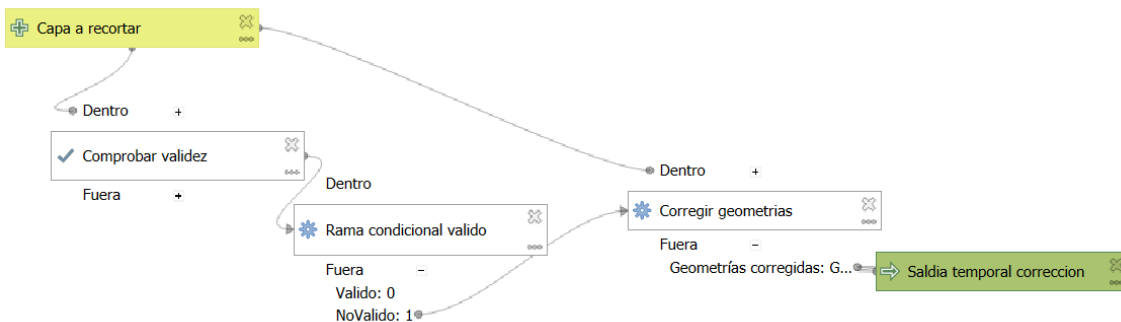


Imagen 8. Flujo de procesos de corrección de geometrías

Condición de comprobación de geometrías válidas

Para la condición de retornar la capa con geometrías válidas, es necesario realizar una copia de la capa de entrada desde el constructor de expresiones.

Paso 9. Agregar el algoritmo **Extraer por expresión**, establecer como nombre “**Extraer por expresion valido**”, seleccionar como capa de entrada la del modelo, ingresar la expresión **\$id IS NOT NULL** para realizar una copia de la capa de entrada. La dependencia de algoritmos corresponde a la rama de condición “**válido**”, y en la salida de objetos coincidentes, ingresar “**Salida temporal corrección válida**”. Borrar las otras salidas.

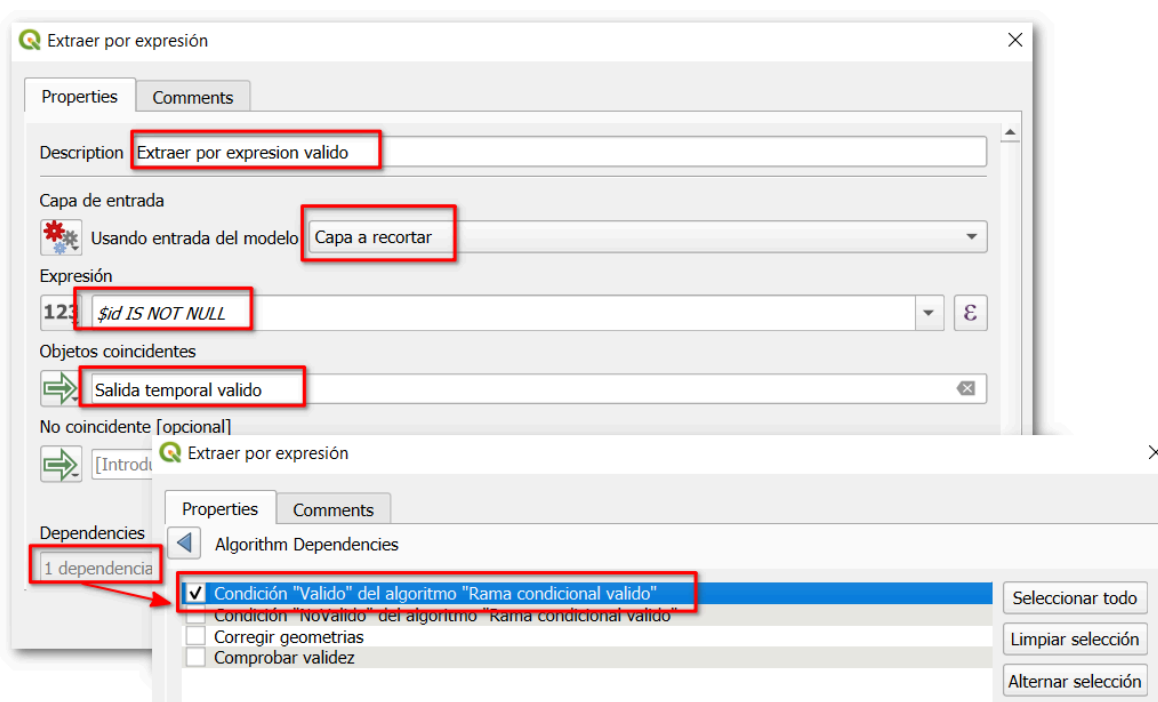


Imagen 9. Extraer por expresión en rama condicional valida

El flujo de proceso hasta la rama condicional de comprobación de validez de las geometrías se muestra en la Imagen 10.

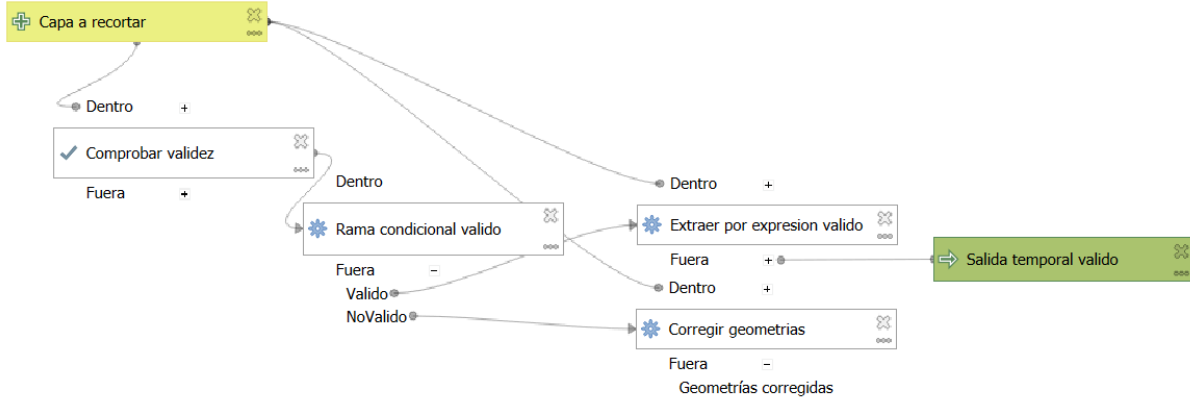


Imagen 10. Flujo de procesos de extraer por expresión geometrías válidas

4.2. Integrar salidas de las ramas condicionales.

Para integrar las salidas de diferentes ramas condicionales, se puede utilizar el algoritmo de combinar geometrías vectoriales.

Paso 10. Agregar al modelo el algoritmo de nombre “Unir capas vectoriales”, ingresar un nombre que identifique la herramienta en el modelo como “Unir capas vectoriales valido”. En capas de entrada seleccionar “Valor precalculado” y usar la expresión de `array(@Extraer_por_expresion_valido_OUTPUT, @Corregir_geometrias_OUTPUT)` para definir la integración de las capas de salida de las ramas condicionales.

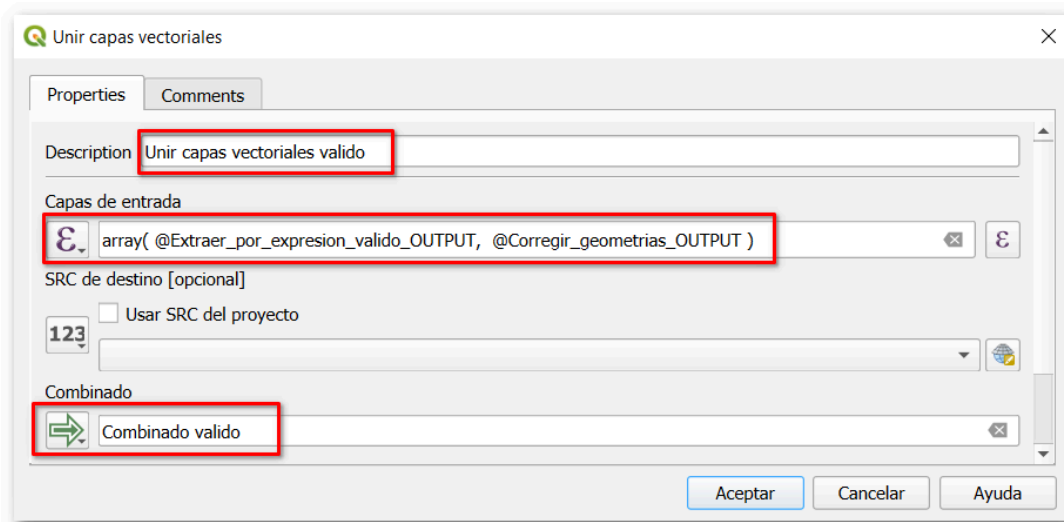


Imagen 11. Combinación de salidas de la rama condicional de valido

El flujo de proceso hasta la combinación de capas de salidas de las ramas condicionales se muestra en la Imagen 12.

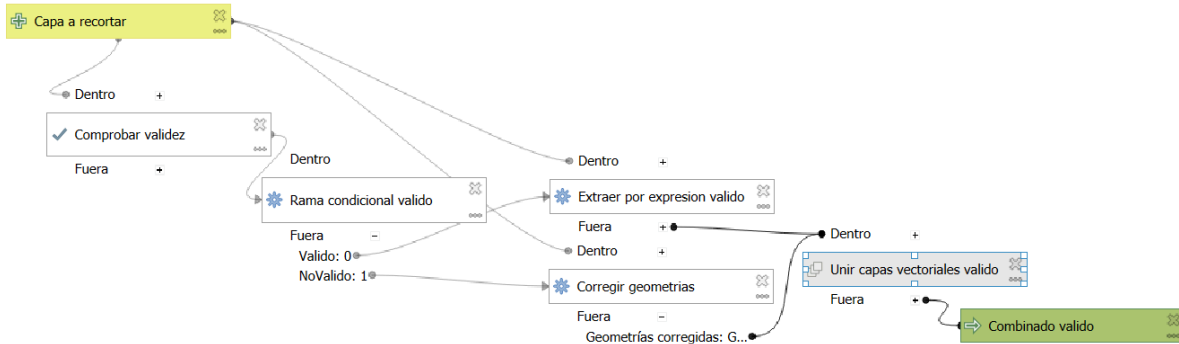


Imagen 12. Flujo de procesos de combinar salidas de las ramas condicionales

Paso 11. Continuar el flujo de procesos agregando una nueva entrada al modelo, usando el control de nombre **“Objetos vectoriales”** con la descripción de **“Capa de recorte”** y de geometría polígono. Enlazar esta entrada con la herramienta de **“Cortar”** de la categoría de **“Superposición vectorial”**. La capa entrada de la herramienta corresponde la salida del algoritmo de **“Combinado desde el algoritmo Unir capas vectoriales valido”** y la capa de superposición corresponde a la capa de entrada de nombre **“Capa de recorte”**.

La Imagen 13 permite comprobar la secuencia del flujo de procesos de cortar.

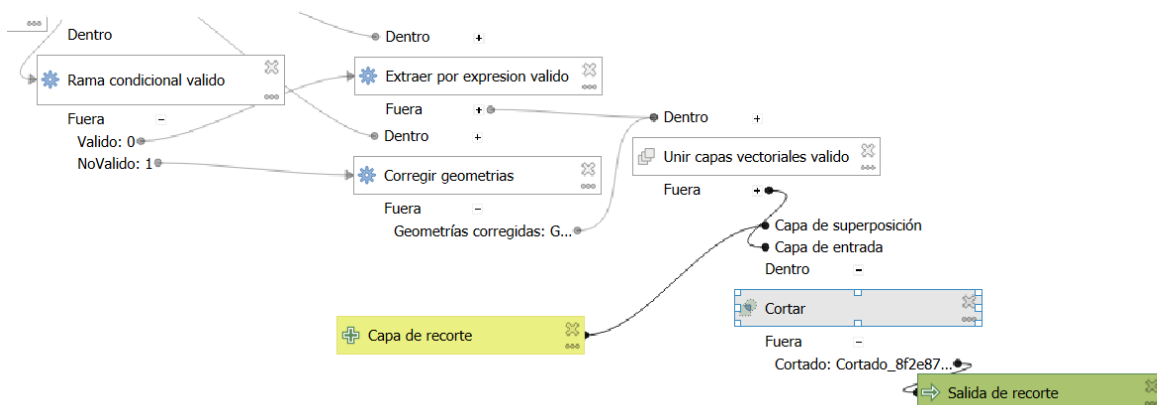


Imagen 13. Flujo de procesos de cortar capa vectorial

4.3. Establecer variables en el modelo.

Las variables del modelo desempeñan un papel importante en la configuración y sustitución de valores en las entradas, salidas y otros parámetros del algoritmo. Para este ejercicio, se almacenará la definición del **SRC** de la capa de entrada a recortar para especificar la conversión a la proyección cartográfica de **Origen Nacional**.

Paso 12. Agregar el algoritmo de nombre **“Establecer variable de proyecto”**, definir los parámetros de descripción como **“Establecer variable de proyecto SRC”**, el nombre la variable como **“capa_a_recortar_src”** y como valor de variable

la expresión `layer_property(@capa_a_recortar, 'crs')`. La dependencia del algoritmo seleccionar a **“Cortar”**.

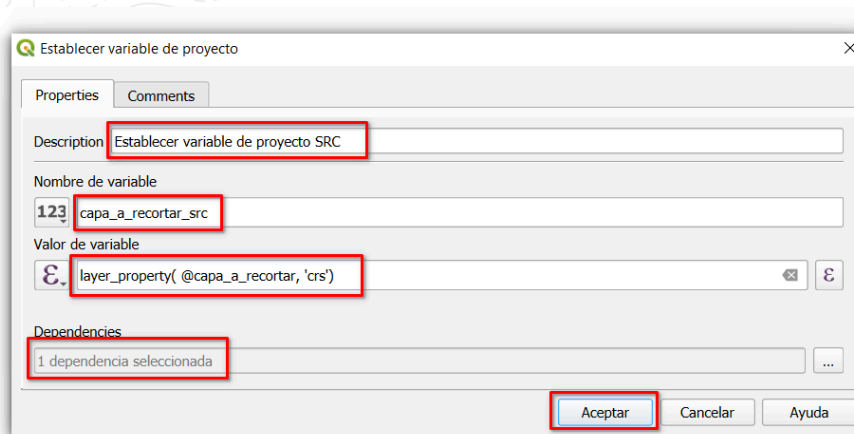


Imagen 14. Algoritmo de establecer variable de proyecto de SRC

4.4. Salida final del modelo

Continuando con el flujo de proceso del modelo, se establece a continuación un parámetro adicional de entrada de caja de chequeo, para especificar por parte del usuario la reproyección de la capa, así como la comprobación del SRC de la capa de entrada a partir del SRC.

Para decidir la selección de herramientas, se conecta una rama condicional con los algoritmos de reproyección y extracción por expresión, y se continúa combinando estos resultados en la salida final del proyecto. La Imagen 15 muestra la representación de la secuencia final de procesos para integrar.

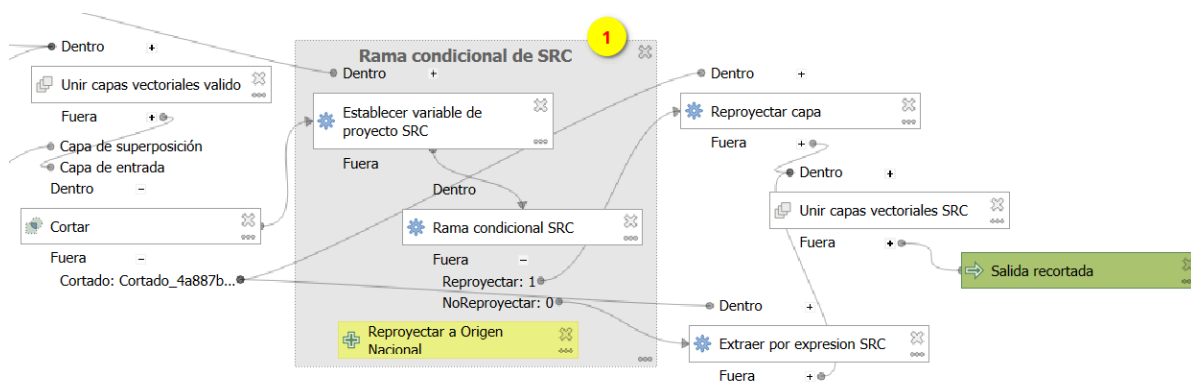


Imagen 15. Flujo de proceso final del algoritmos con conexión de rama condicional, reproyectar, extraer por expresión y combinar vectores

Para documentar una rama en el modelador gráfico, seleccionar un conjunto de elementos en (1) y agrupar los mismos desde el menú **Editar » Add Group Box**.

Paso 13. Agregar una entrada al modelo de tipo **Booleano** con descripción **“Reproyectar a Origen Nacional”**. Agregar una rama condicional llamada **“Rama condicional SRC”**. Para este último, agregar dos ramas denominadas **“Reproyectar”** y **“NoReproyectar”**.

Establecer las expresiones para cada rama según la selección del usuario de la entrada a reproyectar, y la validación del **SRC** de la capa de entrada a recortar a través de la variable del modelo. Seleccionar como dependencia de algoritmo al **“Establecer variable de proyecto SRC”** (ver Imagen 16).



Imagen 16. Rama condicional de SRC

Paso 14. En la rama condicional de **SRC** de **“Reproyectar”**, continuar el flujo de procesos con la herramienta de **Vector General** llamada **“Reproyectar capa”**. La entrada corresponde a la salida del algoritmo **“Cortado”**, y el **SRC objetivo** es el Origen Nacional o **EPSG:9377**. Recordar conectar la secuencia lógica del flujo a la dependencia del algoritmo **“Condición Reprojectar”** (Ver Imagen 17).

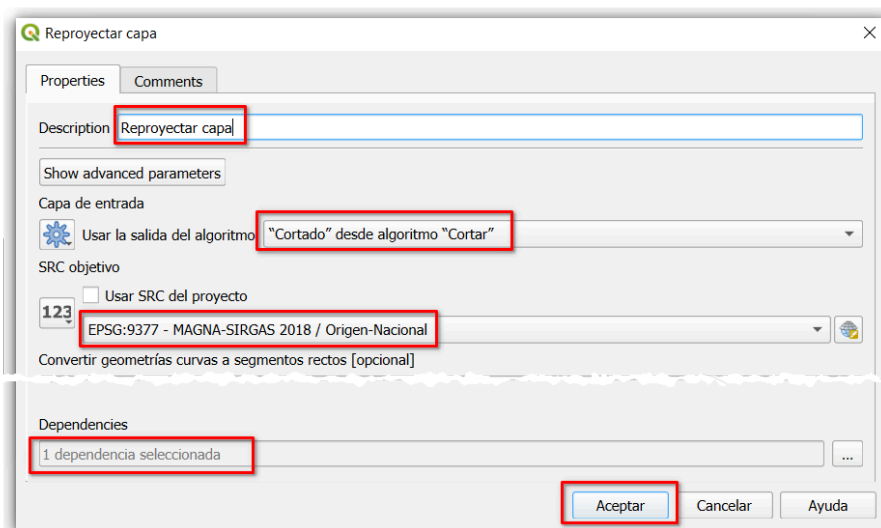


Imagen 17. Reprojectar capa a Origen Nacional

Paso 15. En la rama condicional de **SRC** de **“NoReproyectar”**, continuar con la copia de la capa recortada utilizando el algoritmo de **“Extraer por expresión”**. Seleccionar como dependencia la rama de **“NoRoyectar”** (Ver Imagen 18).



Imagen 18. Copiar capa de cortar en la rama condicional de NoReproyectar

Paso 16. Combinar los resultados de las ramas condicionales con el algoritmo de **“Unir capas vectoriales”**. Finalmente, la salida de este algoritmo constituirá la salida final del modelo.

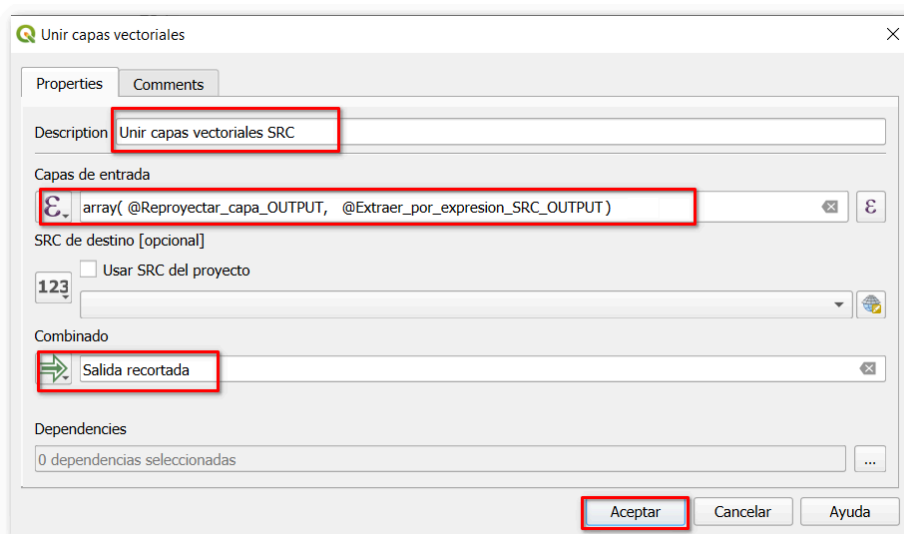



Imagen 19. Combinar salidas de las ramas condicionales y salida final del modelo

4.5. Documentación del modelo

La documentación permite asignar una descripción del flujo de procesos, así como el versionamiento y autoría del algoritmo. La documentación se construye como una página web que se despliega a la derecha durante la ejecución del modelo.

Paso 17. Documentar el modelo desde el menú **Modelo » Edit Model Help...** o directamente en el botón . Realizar el diligenciamiento de la ayuda como:

- **Descripción del algoritmo:** El modelo permite realizar el geoprocetamiento de recorte de una capa vectorial y adicionalmente corregir las geometrías y reproyectar opcionalmente al Origen Nacional.
- **Descripción corta:** Recortar, reproyectar y corregir geometrías.
- **Parámetros de entrada y Parámetros de salida.**
- **Autor y versionamiento.**

Al ejecutar el modelo final se despliega la siguiente interfaz gráfica:



Imagen 20. Interfaz gráfica del modelo final

4.6. Ejecutar el modelo en modo procesamiento por lotes

El modelo se ejecuta desde la interfaz de **QGIS** en el menú **Proyecto » Modelos** y desde la caja de herramientas en la categoría **“Modelos del proyecto”**. Al hacer clic derecho sobre el modelo, permite editarlo o ejecutarlo en modo de procesamiento por lotes. Para la versión de **QGIS 3.34.3**, es necesario marcar la opción **“Cargar capas al completar”** para ejecutar el modelo sin errores.

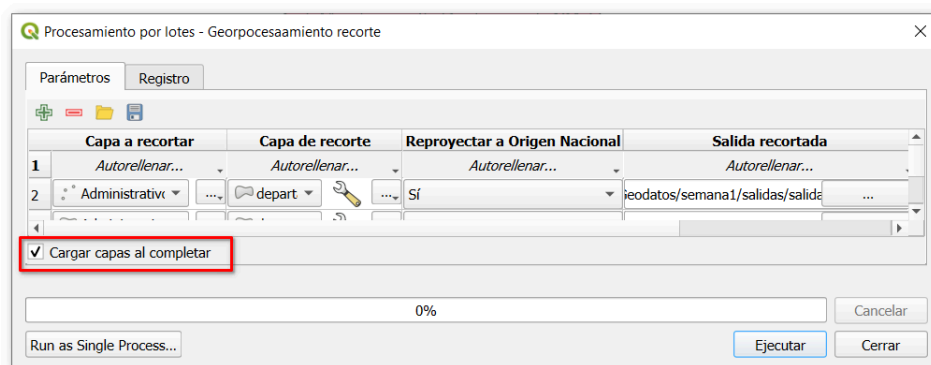


Imagen 21. Procesamiento por lotes del modelo

5. Entregable del ejercicio

Construir el modelo de geoprocесamiento para el ejercicio:

- Almacenar el modelo gráfico de procesos en el proyecto de **QGIS** y que cumpla con las siguientes especificaciones:
 - Una (1) capa de entrada de cualquier geometría de “**Capa a recortar**”, una (1) capa de entrada de polígono de “**Capa de recorte**”, así como una (1) caja de chequeo para especificar opcionalmente la reproyección de la capa.
 - Verificar la validez de las geometrías de la capa de entrada de recorte y definir dos ramas condicionales que permita, por una parte, corregirlas y otra, realizar una copia de la entrada de capa a recortar.
 - Recortar la entrada de “**Capa a recortar**” con la “**Capa de recorte**”.
 - Especificar una variable en el modelo para almacenar el **SRC** de la capa de entrada a recortar. A partir de esta y la selección opcional de usuario para reproyectar, definir una rama condicional para cada caso.
 - Almacenar únicamente una salida en el modelo, que corresponda al geoprocесamiento final de recorte, corrección de geometrías y reproyección.
- Documentar los metadatos básicos del proyecto de **QGIS** con una descripción, el nombre y fecha de entrega.
- Documentar el modelo con una descripción detallada, descripción corta, autor, versión y los parámetros del modelo.
- Exportar el modelo en formato de imagen, agregando comentarios en los algoritmos, cajas de agrupamiento y expandiendo los parámetros de las herramientas.
- Guardar una copia del modelo en formato ***.model3**.
- Realizar el procesamiento masivo de la base de datos del **MGC2023** y almacenar es una (1) sola base de datos de **GeoPackage**.

* Nota 1

La entrega corresponde a la carpeta comprimida del proyecto donde se almacena el **QGZ**, los datos, salidas y modelo solicitado. El archivo comprimido no debe superar más de los 20 MB en disco.

Ejercicio 3

Desarrollo de Scripts con **Python** en el entorno de **QGIS**

1. Introducción

Una de las interfaces clave en entornos **SIG** es el desarrollo de *scripts*, cuya finalidad es automatizar y facilitar la ejecución de tareas. Un *script* o **guion** consiste en una serie de instrucciones o sentencias ejecutadas por un motor especializado. Principalmente, se utiliza para automatizar o simplificar tareas, siendo ellos pequeños algoritmos diseñados para realizar acciones específicas.

En este ejercicio se examinan dos lenguajes de programación más empleados para generar *scripts* en **QGIS**. El primer lenguaje es **Python**, que interactúa con toda la interfaz mediante la definición de expresiones, el desarrollo de guiones y complementos, en resumen, permite personalizar y extender el software para adaptarse a las necesidades del usuario. El segundo lenguaje es **R**, que destaca en el análisis estadístico y la visualización de datos. Ambos lenguajes son altamente destacados en el ámbito del software libre y la ciencia de datos.

2. Datos, software y recursos necesarios

Para el desarrollo correcto del presente ejercicio práctico se requiere la revisión de la sesión de conceptos teóricos de la semana 3 del Nivel Avanzado de **QGIS**:

- Revisión de los contenidos teóricos de la tercera semana (grabación disponible en la **Plataforma de Aulas Virtuales** del **SGC**)([Enlace web](#)).
- **Datos del ejercicio**: Corresponde a los datos de la semana 3 del Mapa Geológico de Colombia del **SGC** y del Marco Geoestadístico Nacional del **DANE** y los *scripts*: Descargar [datos](#).
- Opcional el libro de **Python** para Ingenieros:

<https://jorgedelossantos.github.io/apuntes-python/intro.html>

3. Conociendo la sintaxis de Python

Python es un lenguaje de programación interpretado, interactivo, mutiparadigma. Con una gran comunidad de usuarios y variedad de bibliotecas y *frameworks*. Ofrece una sintaxis clara lo que facilita su aprendizaje y uso en todo tipo de proyectos desde el desarrollo web, ciencia de datos, inteligencia artificial, automatización.

En el entorno de **QGIS**, **Python** desempeña un papel crucial como el lenguaje de programación predeterminado para el desarrollo de *scripts*. Facilita la creación de expresiones personalizadas, la automatización de tareas y la extensión mediante el desarrollo de complementos o *plugins*.

La sintaxis de un lenguaje de programación se refiere a las reglas y estructuras que determinan cómo se deben escribir las instrucciones o comandos en ese lenguaje para que el programa sea válido y pueda ser interpretado correctamente por el ordenador. Es fundamental seguir la sintaxis adecuada para evitar errores de compilación o de interpretación por parte del intérprete o compilador del lenguaje.

El presente ejercicio no pretende ser un manual del lenguaje de programación de **Python**, solo introduce a la sintaxis básica para realizar llevar a cabo algunas operaciones de procesamiento en **QGIS**.

3.1. Configurar e iniciar el intérprete o consola de Python

Python, al ser un lenguaje interpretado, requiere que se inicie el intérprete de comandos correspondiente. En **QGIS**, este proceso se simplifica mediante un panel dedicado a la consola, que permite ingresar instrucciones en **Python** y proporciona otras herramientas útiles.

Paso 1. Crear una copia del proyecto del ejercicio con el nombre de `semana3.qgz` en la carpeta de datos, este debe incluir el modelo gráfico de proceso creado.

Paso 2. Antes de comenzar a interactuar con el intérprete de **Python**, es importante revisar algunas configuraciones proporcionadas por **QGIS**. Esto se puede hacer desde el menú `Configuración » Opciones...`. Comprobar desde la pestaña **IDE** algunas opciones como el despliegue del esquema de color y fuente.

Consejo 1

Para la generación de *scripts*, aproveche las posibilidades actuales de la inteligencia artificial generativa mediante herramientas de software libre como khoj.dev que apoya este proceso de desarrollo.

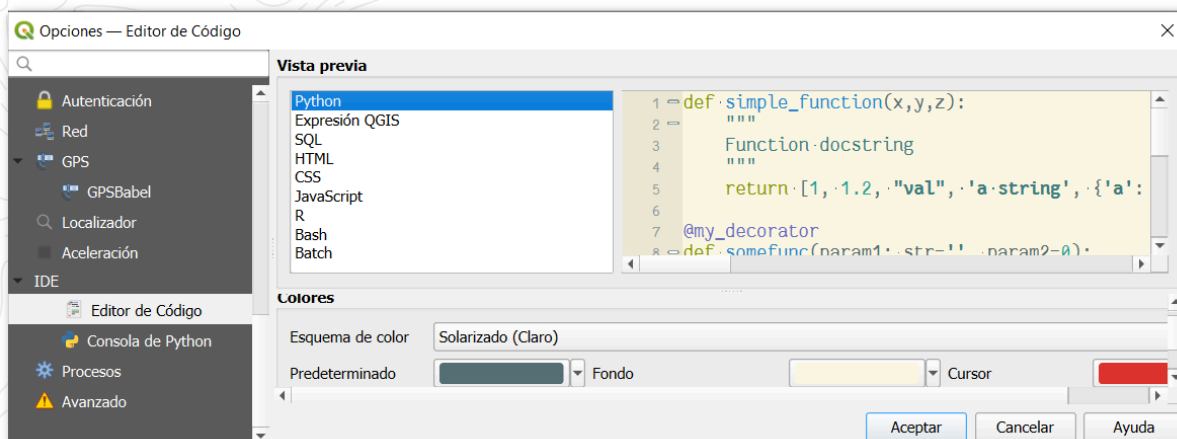


Imagen 1. Configuración IDE y consola de Python

Paso 3. Iniciar el panel de consola de Python desde el menú **Complementos** » **Consola de Python**. La consola se despliega en la parte inferior, presentando una interfaz dividida en dos secciones: la ventana de resultados y avisos, y el prompt “>>>”, donde puedes ingresar directamente los comandos. Usualmente, esta característica te permite ingresar y probar fragmentos de código de manera sencilla, así como inspeccionar los resultados de manera inmediata.

Para comenzar con **Python**, es importante comprender algunos aspectos clave de su sintaxis que se listan a continuación y que se explicaran con algunos ejercicios:

- **Indentación:** corresponde a los espacios en blanco al principio de una línea y es crucial para definir bloques de código.
- **Sensibilidad a mayúsculas y minúsculas:** Python distingue entre mayúsculas y minúsculas. Por lo tanto, «Variable» y «variable» son tratadas como entidades diferentes en Python.
- **Comentarios:** Puedes incluir comentarios en tu código utilizando el símbolo “#” para una sola línea o utilizando triple comillas (“""" ””) para comentarios de varias líneas.
- **Variables y tipos de datos:** En Python, las variables pueden contener diferentes tipos de datos como números enteros, flotantes, cadenas de texto, listas, tuplas y diccionarios. No es necesario declarar el tipo de variable explícitamente.
- **En Python, todo es un objeto.** Esta es una característica fundamental del lenguaje y es una de las razones por las que Python es conocido como un lenguaje de programación orientado a objetos.

Para iniciar desde la consola de Python, desde el prompt **(1)** (>>>) ingresar la instrucción `print("Hola mundo desde QGIS!")`. Observar en **(2)** la salida o resultado del comando.



Imagen 2. Consola de Python

En (3) se muestra la barra de herramientas de la consola de **Python**, en orden son: Limpiar consola, Ejecutar orden, Mostrar Editor, Opciones.. y *Dock Code Editor*.

3.2. Variables

En **Python**, las variables se definen dinámicamente, lo que significa que no es necesario declararlas con un tipo específico antes de asignarles un valor. Simplemente, se utiliza el operador de asignación (=), y **Python** determina automáticamente el tipo de datos, que puede ser texto, números u otras estructuras. Además, las variables en **Python** pueden ser sobrescritas, lo que implica cambiar tanto el valor como el tipo asociado a la variable.

Tipos de datos

Paso 4. Comprobar las salidas de las variables de **Python** a través del Prompt de la consola de **Python**.

- Cadenas de texto (**str**): `base_mgc = "C:\geodatos\mgc2023.gpkg"`
- Enteros (**int**): `iteraciones = 3`
- Flotantes (**float**): `buffer = 10.2`
- Booleanos (**bool**): `reproyectar_magna = True`
- Listas (**list**): `capas = ['Pliegues', 'Fallas', 'UC', 'Volcanes']`
- Tuplas (**tuple**): `zoom = (iface.mapCanvas().zoomOut(), iface.mapCanvas().zoomIn())`
- Diccionarios (**dict**):
`parametros = {'INPUT': 'capa', 'BUFFER': buffer, 'ITERATE': iteraciones, 'OUTPUT': 'salida.gpkg'}`

Use la función **print(nombre_variable)** para acceder e imprimir el valor de cada variable.

Para acceder a un elemento de una lista o tupla, se usa el índice correspondiente. Por ejemplo, `capas[0]` accede al primer elemento. En el caso de los diccionarios, se usa el nombre de la llave para acceder al valor, como en `parametros["INPUT"]`.

Utilizar algunos operadores como suma (+) para realizar algunas operaciones aritméticas, ejemplo `iteraciones + buffer`, `reproyectar_magna + iteraciones` y `base_mgc + iteraciones`.

Sustitución de variables

La sustitución de variables en **Python** se refiere al proceso de reemplazar los marcadores de posición `{}` dentro de una cadena de texto con los valores de las variables correspondientes. **Python** proporciona varias formas de realizar la sustitución de variables, las más comunes son mediante el uso de formateo de cadenas y utilizando **f-strings** (literales de cadena formateados).

```
print("El valor de iteraciones es {} y buffer es: {}".format(iteraciones, buffer))
print(f"El valor de iteraciones es {iteraciones} y buffer es: {buffer}")
```

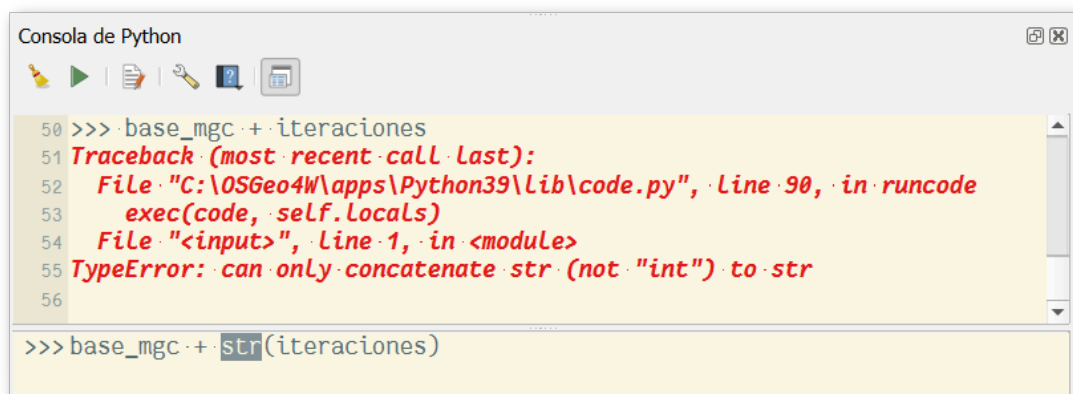
python

Observar para ambos casos se sustituye el valor en la cadena de texto lo que se encuentra entre los marcadores de posición `{}`.

3.3. Errores en Python

En la consola de **Python**, los errores se despliegan mediante mensajes de error que proporcionan información sobre lo que salió mal durante la ejecución del programa. Estos mensajes de error incluyen detalles sobre el tipo de error, la ubicación donde ocurrió y a menudo una traza de la pila que muestra el camino que siguió el programa hasta llegar al error.

Cuando ocurre un error en **Python**, la consola imprime el mensaje de error en la salida estándar. Aquí hay un ejemplo de un mensaje de error típico en **Python**:



```
Consola de Python
50 >>> base_mgc + iteraciones
51 Traceback (most recent call last):
52   File "C:\OSGeo4W\apps\Python39\Lib\code.py", line 90, in runcode
53     exec(code, self.locals)
54   File "<input>", line 1, in <module>
55   TypeError: can only concatenate str (not "int") to str
56
>>> base_mgc + str(iteraciones)
```

Imagen 3. Mensajes de error en Python

Paso 5. Para corregir el anterior error debe realizarse una conversión del tipo de dato, ejemplo de numérico a cadena de texto `base_mgc + str(iteraciones)`

3.4. Estructuras de control

El flujo de control en **Python** te permite escribir programas flexibles y potentes, los cuales pueden adaptarse dinámicamente a diferentes situaciones y condiciones. Es fundamental comprender cómo funcionan las declaraciones condicionales y los bucles para construir programas eficientes y funcionales, a continuación los más importantes:

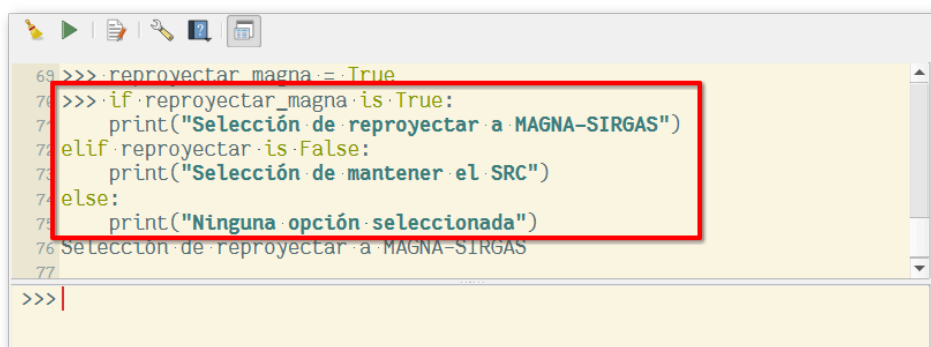
Condicional if-elif-else

(**if**) Permite ejecutar un bloque de código si una condición es verdadera; (**elif**) Se utiliza para verificar condiciones adicionales después de la primera, y (**else**) Se ejecuta cuando ninguna de las condiciones anteriores es verdadera.

Paso 6. Ejecutar el siguiente código condicional del valor almacenado en la variable **reproyectar_magna**.

```
if reproyectar_magna is True:
    print("Selección de reproyectar a MAGNA-SIRGAS")
elif reproyectar is False:
    print("Selección de mantener el SRC")
else:
    print("Ninguna opción seleccionada")
```

python



```
69 >>> reproyectar_magna = True
70 >>> if reproyectar_magna is True:
71     print("Selección de reproyectar a MAGNA-SIRGAS")
72 elif reproyectar is False:
73     print("Selección de mantener el SRC")
74 else:
75     print("Ninguna opción seleccionada")
76 Selección de reproyectar a MAGNA-SIRGAS
77
>>> |
```

Imagen 4. Condicional if-elif-else

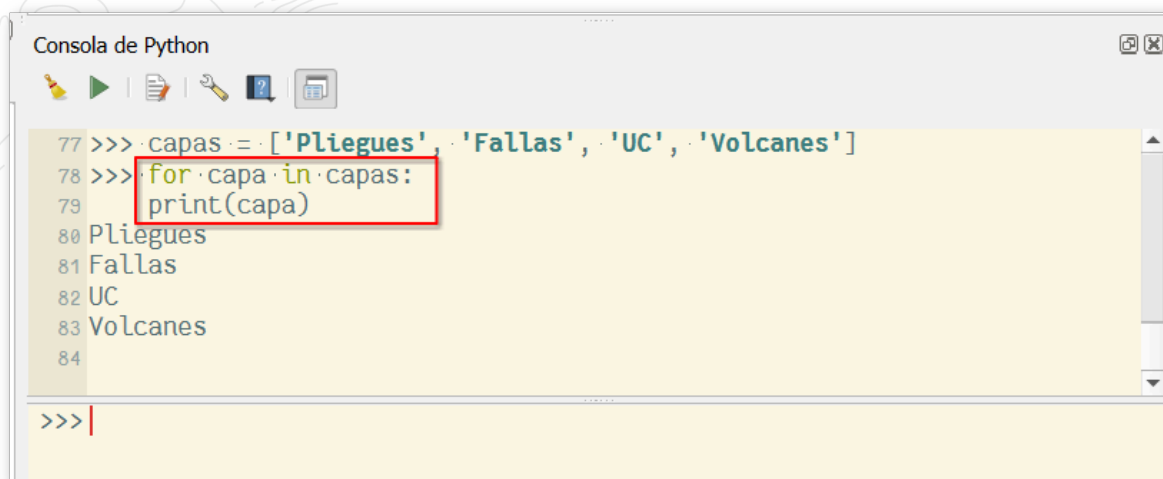
Ciclo for

Itera sobre una secuencia (como listas, tuplas, diccionarios, etc.) y ejecuta un bloque de código para cada elemento de la secuencia. Recordar que un bloque de código se realiza indentando las líneas correspondientes.

Paso 7. Ejecutar el siguiente código que permite iterar sobre cada elemento de la lista de capas e imprime su salida.

```
for capa in capas:  
    print(capa)
```

python



```
Consola de Python  
77 >>> capas = ['Pliegues', 'Fallas', 'UC', 'Volcanes']  
78 >>> for capa in capas:  
79     print(capa)  
80 Pliegues  
81 Fallas  
82 UC  
83 Volcanes  
84  
>>> |
```

Imagen 5. Ciclo o bucle **for** para las capas

3.5. Biblioteca y módulos en Python

Las bibliotecas en **Python** son conjuntos de módulos y funciones que amplían las capacidades del lenguaje base. Permiten a los desarrolladores aprovechar el trabajo de miles de contribuciones de terceros, lo que facilita la implementación de diversas funcionalidades sin tener que crearlas desde cero.

Para importar una biblioteca en **Python**, se utiliza la palabra clave **import**, seguida del nombre de la biblioteca. A continuación, te presento un ejemplo de cómo importar la librería **PyQGIS** y sus módulos de procesamiento, así como el módulo **QgsApplication**, que sirve como interfaz de entrada al núcleo de **QGIS**, permitiendo la interacción con él.

Al crear una instancia de **QgsApplication**, se inicializa el motor de **QGIS**, se cargan plugins, se registran proveedores de capas y se prepara el entorno para utilizar las funcionalidades de **QGIS**.


Para obtener más información sobre la biblioteca **PyQGIS**, se recomienda consultar la documentación del API (Interfaz de Programación de Aplicaciones), la cual ofrece una descripción detallada sobre cómo utilizar las clases, métodos y funciones disponibles para interactuar con **QGIS** desde **Python**.

Documentación del API de **QGIS**: <https://api.qgis.org/api/3.36/>

Documentación de **QGIS Python** API: <https://qgis.org/pyqgis/3.14/>

Libro de recetas para desarrolladores de **PyQGIS**:

https://docs.qgis.org/3.34/en/docs/pyqgis_developer_cookbook/index.html

Paso 8. Para facilitar el trabajo con múltiples instrucciones, se sugiere abrir el editor de código  desde la barra de herramientas de la consola de **Python**. Luego, introducir el código que permita listar los proveedores disponibles para la interfaz actual de **QGIS**. Para guardar el *script*, use el botón de **guardado** (1) y, finalmente, ejecute con (2).

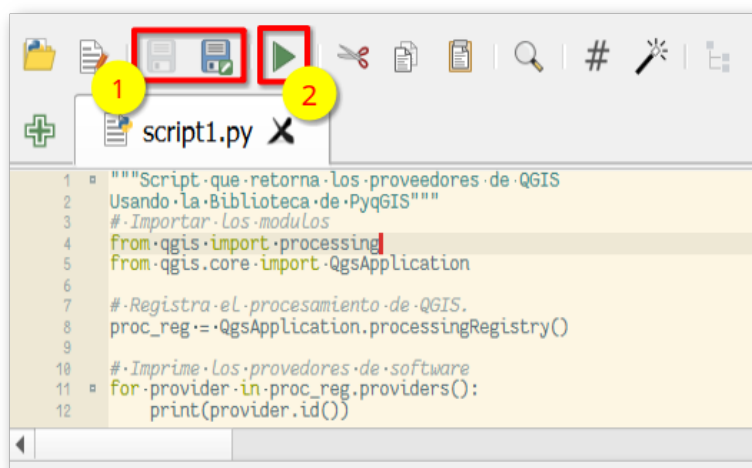
```
"""Script que retorna los proveedores de QGIS
Usando la Biblioteca de PyqGIS"""

# Importar los modulos
from qgis import processing
from qgis.core import QgsApplication

# Registra el procesamiento de QGIS.
proc_reg = QgsApplication.processingRegistry()

# Imprime los proveedores de software
for provider in proc_reg.providers():
    print(provider.id())
```

python



Para consultar los algoritmos de un proveedor específico, usar el método **proc_reg.providerById("qgis").algorithms()** para retornar el listado de algoritmos. Luego iterar sobre cada uno y mostrar su identificado **id()** y nombre **name()** conjuntamente, empleando **print(algoritmo.id(), "-> ", algoritmo.name())**.

3.6. Funciones en Python

En **Python**, una función es un bloque de código que realiza una tarea específica cuando es llamada. Las funciones pueden aceptar argumentos y devolver resultados. Se define una función utilizando la palabra clave **def**.

Paso 9. La siguiente función calcula el área de un círculo a partir del radio.

```
import math

def calcular_area_circulo(radio):
    """Calcular el area de un circulo a partir del radio"""
    area = math.pi * (radio ** 2)
    return area

calcular_area_circulo(10)
```

3.7. Desplegando la ayuda de un objeto

En **Python**, cada elemento es un objeto, es decir, una instancia de una clase. Las clases son fundamentales en la programación orientada a objetos en **Python**, ya que establecen la estructura y el comportamiento de los objetos. Un objeto en **Python** puede incluir atributos (variables asociadas) y métodos (funciones asociadas). Los atributos reflejan el estado actual, mientras que los métodos definen sus acciones.

En la consola de **Python**, se consulta la ayuda de un objeto o función utilizando la función **help()**. Escribe **help("processing")**, seguido del nombre del objeto o la función para la que deseas obtener información detallada. También puede usar la función **dir("processing")** para ver sus atributos y métodos disponibles.

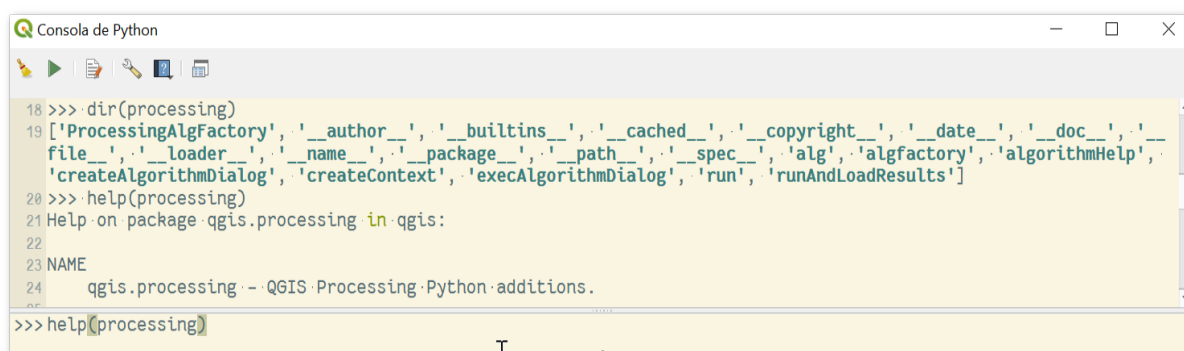


Imagen 6. Desplegando la ayuda en la consola de **Python**.

4. Procesamiento desde la consola de Python

Después de revisar los conceptos básicos y la sintaxis de **Python**, se propone el desarrollo de *scripts* para replicar el ejercicio anterior de procesamiento por lotes y la creación de modelos gráficos de flujo de procesos.

4.1. Creando un módulo de geoprociamiento simple

Para crear un algoritmo de procesamiento con **Python** en **QGIS**, es necesario seguir una sintaxis y plantilla documentada proporcionada en el [sitio web oficial](#). La propuesta de los *scripts* presentados a continuación es ofrecer una introducción básica a las capacidades de **Python** como interfaz de procesamiento, siguiendo un flujo de procesos.

Paso 10. Realizar la secuencia de herramientas para el geoprociamiento de las capas, iniciando desde validar las geometrías de una capa, luego corregir para la capa no válida, y posteriormente realizar la reproyección, dependiendo de la variable booleana y finalmente recortar con la capa de **departamento**.

A continuación, el código del *script* que procede con algunos de los flujos propuestos. En el ejercicio este *script* tiene como nombre `geoprociamiento.py`.

```
1  """Ejecutar la secuencia de algoritmos
2  para el procesamiento del MGC2023"""
3
4  # Importar el modulo para interactuar con el proyecto
5  from qgis import processing
6  from qgis.core import QgsProject, QgsProviderRegistry
7  from qgis.core import QgsVectorLayer
8
9  # Guardar como variable la ruta absoluta del proyecto
10 DIR_PROYECTO = QgsProject.instance().absolutePath()
11
12 def geoprociamientoAlgorithm(capa_a_procesar, capa_de_recorte, gpkg_salida):
13     """Esta funcion permite el geoprociamiento de
14     correccion de geometrías, indexamiento y recorte"""
15
16     ## Si la capa a procesar existe o es valida
17     if capa_a_procesar.isValid() is False:
18         print("La capa a procesar no es valida. Por favor corrija.")
19         return
20
21     ## Si la capa a procesar existe o es valida
22     if capa_de_recorte.isValid() is False:
23         print("La capa de recorte no es valida. Por favor corrija.")
24         return
25
26     #Decodificar la fuente de la capa de entrada
```

```
27 uri_capa_procesar = QgsProviderRegistry.instance().decodeUri(  
28     capa_a_procesar.dataProvider().name(), capa_a_procesar.publicSource()  
29 )  
30  
31 capa_nombre = uri_capa_procesar["layerName"]  
32  
33 # Definir variable capa de salida  
34 ruta_gpkg_salida = f"{DIR_PROYECTO}/salidas/{gpkg_salida}.gpkg"  
35 capa_salida = f'ogr:dbname=\'{ruta_gpkg_saliparametrosda}\'  
36     table="{capa_nombre}" (geom)'  
37  
38 print(f"Inicio de procesamiento para la capa {capa_nombre}")  
39 print("Paso 1: Creando el índice espacial")  
40  
41 processing.run("native:createspatialindex", {  
42     'INPUT':capa_a_procesar  
43 })  
44  
45 print("Paso 2: Validando geometrías")  
46  
47 #Use runAndLoadResults() para desplegar los resultados temporales  
48 validar_vector = processing.run("qgis:checkvalidity", {  
49     'INPUT_LAYER': capa_a_procesar,  
50     'METHOD': 2,  
51     'IGNORE_RING_SELF_INTERSECTION': False,  
52     'VALID_OUTPUT': 'TEMPORARY_OUTPUT',  
53     'INVALID_OUTPUT': 'TEMPORARY_OUTPUT',  
54     'ERROR_OUTPUT': 'TEMPORARY_OUTPUT'  
55 })  
56  
57 if validar_vector['INVALID_COUNT'] > 0:  
58     print(f"La capa {capa_nombre} requiere correccion de geometrías")  
59  
60     corregir_vector = processing.run("native:fixgeometries", {  
61         'INPUT': capa_a_procesar,  
62         'METHOD': 1, 'OUTPUT': 'TEMPORARY_OUTPUT'  
63     })  
64  
65     resultado_validar = corregir_vector['OUTPUT']  
66  
67 else:  
68     resultado_validar = capa_a_procesar  
69  
70 print("Paso 3: Recortando la capa")  
71  
72 recortar_capa = processing.runAndLoadResults("native:clip", {  
73     'INPUT': resultado_validar,  
74     'OVERLAY': capa_de_recorte,  
75     'OUTPUT': capa_salida})  
76  
77 print("Finalizado el procesamiento")
```

```
77     return recortar_capa
78
79
80 def main():
81     """Use esta funcion para comprobar la salida del_
82     algoritmo de geoprocAMIENTOAlgorithm"""
83     ## Definir las variables de entrada y salida
84     ruta_capa_procesar = f"{DIR_PROYECTO}/geodatos/mgc2023.gdb|layername=UC"
85     ruta_capa_recorte = f"{DIR_PROYECTO}/geodatos/DANE-MGN-2018.gpkg|
86     layername=departamento"
87
88     ## Crear el objeto vectorial
89     capa_procesar = QgsVectorLayer(ruta_capa_procesar, 'capa_a_procesar', 'ogr')
90     capa_recorte = QgsVectorLayer(ruta_capa_recorte, 'capa_de_recorte', 'ogr')
91
92     geoprocAMIENTOAlgorithm(capa_procesar, capa_recorte, 'mgc2023_prueba')
93
94 main()
```

El código está documentado en cada bloque. En la **línea 12** se define una función llamada **geoprocAMIENTOAlgorithm()**, diseñada para ser reutilizada como un módulo en otros *scripts*. Los argumentos aceptados por esta función son la capa vectorial a procesar, la capa de recorte y el nombre del **GeoPackage** de salida. Para validar si los datos vectoriales de las capas a procesar y recortar son válidos, es decir, si son vectores, se utiliza el método **isValid()**, como se especifica en las **líneas 17 y 22** del código.

El llamado de un algoritmo de geoprocAMIENTO se realiza mediante **processing.run()**, utilizando los parámetros que, generalmente, se pueden definir en la interfaz gráfica de la caja de herramientas (ver Imagen 7). En este código, se efectúan tres llamadas a herramientas: en la **línea 41** para crear un índice espacial, en la **línea 47** para verificar la validez de las geometrías, en la **línea 60** para corregir las geometrías, y en la **línea 72** para recortar la capa. Es importante notar el condicional en las **líneas 56 y 66**, el cual establece los flujos para cada caso de validación de geometrías. Para ambos casos, la capa vectorial resultante se almacena en una variable llamada **resultado_validar**, que luego se utiliza como entrada en la herramienta de recorte, tal como se observa en la **línea 73**.

Para llevar a cabo pruebas simples de geoprocAMIENTO, se utiliza la función **main()** con las capas de entrada de prueba definidas entre las **líneas 85 y 90**. En la **línea 91**, se invoca el algoritmo **geoprocAMIENTOAlgorithm** con los parámetros requeridos.

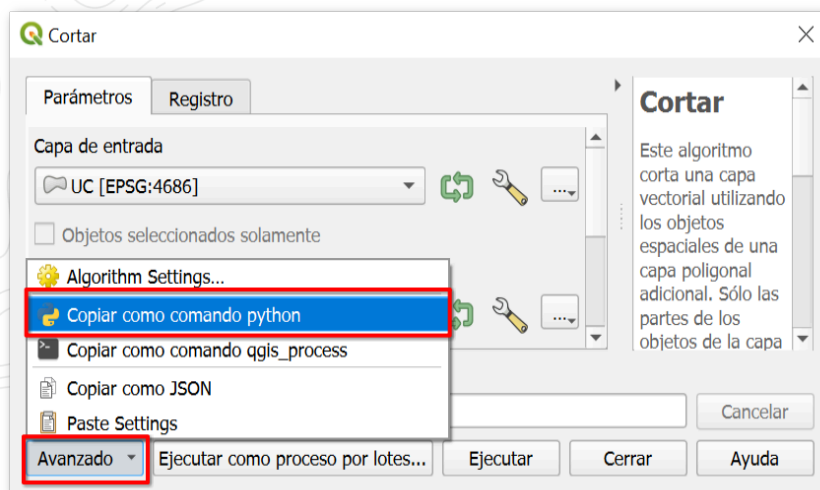


Imagen 7. Copiar el código de Python con los parámetros de una herramienta gráfica

4.2. Procesamiento en modo por lotes en Python

Siguiendo con la propuesta de una introducción básica al desarrollo de *scripts* de Python en QGIS, se lleva a cabo el procesamiento masivo de las capas que se encuentran en el grupo de capas denominado **mgc2023** en el panel de capas del proyecto de QGIS. Finalmente, los resultados del procesamiento se almacenan en un único **GeoPackage**.

Paso 11. Para llevar a cabo el procesamiento en modo por lotes desde el *script* proporcionado llamado `lotes.py`, ajustar las rutas del proyecto y el nombre del grupo de capas correspondiente al **mgc2023**.

```

1  """Ejecutar la secuencia de algoritmos
2  en procesamiento por lotes"""
3
4  #Importar librerías y el módulo de geoprocésamiento desarrollado
5  import datetime
6  import importlib
7  import geoprocésamiento
8  # Recarga la librería en edici3n del módulo
9  importlib.reload(geoprocésamiento)
10
11 # Registrar la hora actual de inicio del procesamiento
12 timeStart = datetime.datetime.now()
13
14 # Retorna las capas del panel de capas
15 root = QgsProject.instance().layerTreeRoot()
16 # Busca el nombre del grupo de capas de la base de datos agregada
17 mygroup = root.findGroup("mgc2023") #Ajuste de acuerdo al nombre del proyecto
18
19 #Define las variables de capa de recorte
20 ruta_capa_recorte = f"{geoprocésamiento.DIR_PROYECTO}/geodatos/DANE-
21 MGN-2018.gpkg|layername=departamento"
22 ## Crear el objeto vectorial de la capa de recorte
    
```

```
23 capa_de_recorte = QgsVectorLayer(ruta_capa_recorte, 'capa_de_recorte', 'ogr')
24
25 # Define una variable con el nombre del gpkg de salida
26 gpkg_salida = "mgc2023_cundinamarca"
27
28 # Contador de capas
29 total_capas = len(mygroup.children())
30 capa_i = 1
31
32 # itera todas las capas del grupo del mgc2023
33 for child in mygroup.children():
34     name = child.name().lower()
35     layer = child.layer()
36     print(f"Procesando capa {capa_i}/{total_capas}")
37     capa_i += 1
38
39     geoprosesamiento.geoprosesamientoAlgorithm(layer, capa_de_recorte, gpkg_salida)
40
41 # Registra la hora de terminacion
42 timeEnd = datetime.datetime.now()
43 print("El procesamiento por lotes fue finalizado correctamente")
44 print("Total tiempo de procesamiento: {}".format(timeEnd - timeStart))
```

Para explicar de manera general el algoritmo proporcionado, se aclara que en la **línea 7** se importa el *script* de geoprosesamiento como un módulo, el cual fue explicado en el paso anterior. El método **importlib.reload** en la **línea 9** se utiliza para recargar el primer *script* en caso de realizar modificaciones. Las **líneas 12, 42 y 44** registran el tiempo total del procesamiento. En las **líneas 20 a 26** se declaran las variables para los parámetros de la capa de recorte y el nombre del **GeoPackage** de salida de los resultados.

En las **líneas 15 a 17** se lleva a cabo la búsqueda del grupo de capas denominado **mgc2023** en el panel de capas. Posteriormente, en las **líneas 33 a 39**, se itera sobre las capas del grupo para realizar un procesamiento masivo utilizando la función **geoprosesamientoAlgorithm**.

Es importante tener en cuenta que el archivo del proyecto `semana3.qgz` debe estar ubicado en la carpeta que contiene las carpetas de **geodatos**, **salidas** y los *scripts* de **Python** del ejercicio. Además, la base de datos espacial `mgc2023.gdb` debe ser desplegada como un grupo de capas en el panel de capas. Finalmente, los resultados del procesamiento se almacenarán en la base de datos espacial `mgc2023_cundinamarca.gpkg` en la carpeta de **salidas**.

5. Entregable del ejercicio

Desarrollar los *scripts* en **Python** para el geoprocesamiento masivo del `mgc2023.gdb`.

- Actualizar los *scripts* de `geoprocesamiento.py` y `lotes.py` para incluir la reproyección de la capa al sistema de coordenadas Origen Nacional. Se debe agregar un argumento booleano en la función **geoprocesamientoAlgorithm**, el cual permitirá al usuario definir si se realizará la conversión de coordenadas o no. Por defecto, esta conversión debe estar habilitada.
- Desarrollar un nuevo *script* llamado `lote_modelo.py` para realizar el procesamiento masivo de las capas de la base `mgc2023.gdb`, utilizando los parámetros del modelo gráfico creado en la semana 2. Los resultados de este procesamiento deben guardarse en la carpeta de **salidas**, dentro de una base de datos *Geopackage* denominada `mgc2023_modelo.gpkg`.

* Nota 1

La entrega corresponde a la carpeta comprimida del proyecto donde se almacena el **QGZ**, los datos, salidas y *scripts* solicitados. El archivo comprimido **no debe superar** más de los 20 MB en disco.

Ejercicio 4

Análisis espacial vectorial, *scripts* con R

1. Introducción

Uno de los principales objetivos de los Sistemas de Información Geográfica (**SIG**) reside en su capacidad para llevar a cabo el análisis espacial, lo cual implica el proceso de extraer información con el fin de generar conocimiento a partir de un conjunto de datos espaciales, revelando patrones, tendencias y relaciones entre los mismos. El análisis espacial proporciona una visión objetiva del territorio, permitiendo a los planificadores, investigadores y tomadores de decisiones comprender mejor la dinámica de la realidad geográfica, facilitando la toma de decisiones informadas y el diseño de estrategias eficaces en una amplia gama de aplicaciones, desde la gestión del medio ambiente hasta la planificación urbana y el desarrollo socioeconómico. En resumen, el análisis espacial constituye un componente fundamental para una toma de decisiones informada y efectiva en el ámbito territorial.

Dado que el análisis se encuentra relacionado con los modelos y estructuras de almacenamiento y representación, en este ejercicio se propone el uso de las principales funcionalidades de **QGIS** para extraer información a partir de datos vectoriales. Se llevará a cabo un análisis de superposición, integrando diversos proveedores de algoritmos con el fin de respaldar la presentación mediante reportes y resúmenes estadísticos. Esta práctica ofrece una comprensión del abanico de herramientas y recursos ofrecidos por **QGIS** enfocado a las estructuras vectoriales.

2. Datos, software y recursos necesarios

Para el desarrollo correcto del presente ejercicio práctico se requiere la revisión de la sesión de conceptos teóricos de la semana 4 del Nivel Avanzado de **QGIS**:

- Revisión de los contenidos teóricos de la cuarta semana (grabación disponible en la **Plataforma de Aulas Virtuales del SGC**)([Enlace web](#)).
- **Datos del ejercicio:** Corresponde a los datos vectoriales de la semana 4: Descarga de **datos**.
- **Referencia:**
 - <https://rspatial.org/intr/index.html>
 - <https://rspatial.org>
 - <https://ggplot2-book.org/introduction>

3. Análisis espacial vectorial

QGIS incluye nativamente diversas herramientas y funciones de análisis espacial diseñadas para trabajar con modelos de datos vectoriales, es decir, permitiendo procesar tanto geometrías como sus atributos asociados. La categoría de “**Análisis vectorial**” ofrece algoritmos para realizar el agrupamiento de patrones puntuales, generar estadísticas de campos y llevar a cabo la superposición de capas de polígonos. Además, dentro de las herramientas de análisis espacial vectorial, aquellas organizadas en la rama “**Superposición vectorial**”, las cuales permiten realizar operaciones como cortar, interceptar, unir, obtener la diferencia simétrica y dividir con líneas. Adicionalmente, **QGIS** ofrece herramientas orientadas a la generación de gráficos y resúmenes estadísticos, ampliando así las capacidades de análisis y visualización de datos.

3.1. Superposición vectorial

En el presente ejercicio se plantea el desarrollo del análisis vectorial a partir de las capas de **Unidades Cronoestratigráficas** (UC) y la *División Política Administrativa* del departamento de Caldas. El objeto del análisis consiste en presentar el área y los porcentajes de UC por división política administrativa.

Paso 1. A partir del conjunto de datos de Caldas, agregar el **límite de departamento** y la capa de **UC** del **SGC**, almacenar el archivo de proyecto de **QGIS** en la misma carpeta con el nombre `Caldas.qgz`.

Paso 2. Configurar, en las propiedades del proyecto, el cálculo de mediciones en la proyección cartográfica por defecto (Origen Nacional) y establecer las unidades en hectáreas (ha).

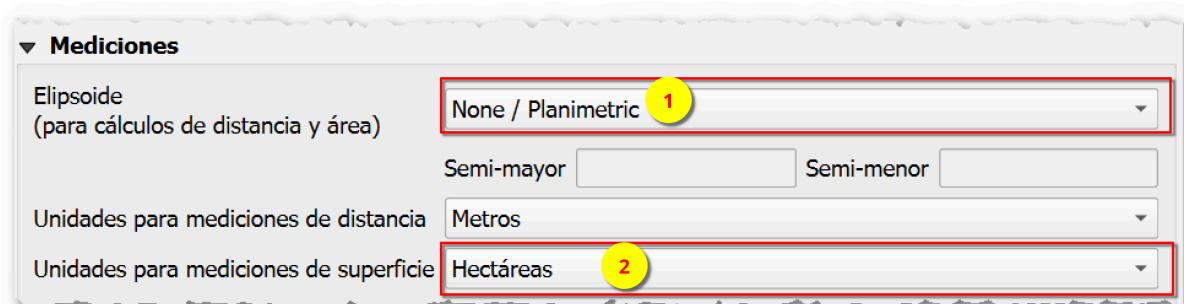


Imagen 1. Configuración de mediciones en el proyecto

Paso 3. En la tabla de atributos de la capa de *departamento* crear un nuevo campo decimal de nombre `area_dep_ha`, calculando su área con `$area`. Todos los cálculos con este operador estarán referidos a la configuración del proyecto.

Paso 4. Dado que el análisis a desarrollar es el contexto del departamento, se plantea la construcción del siguiente modelo gráfico para replicar la herramienta de *Identidad* disponible en otros software SIG. Básicamente, permite un análisis de superposición manteniendo la geometría original de la división política administrativa o capa de identidad.

Usando la herramienta de **identidad**, extraer los datos de **UC** (capa a superponer) para el departamento de caldas (capa identidad).

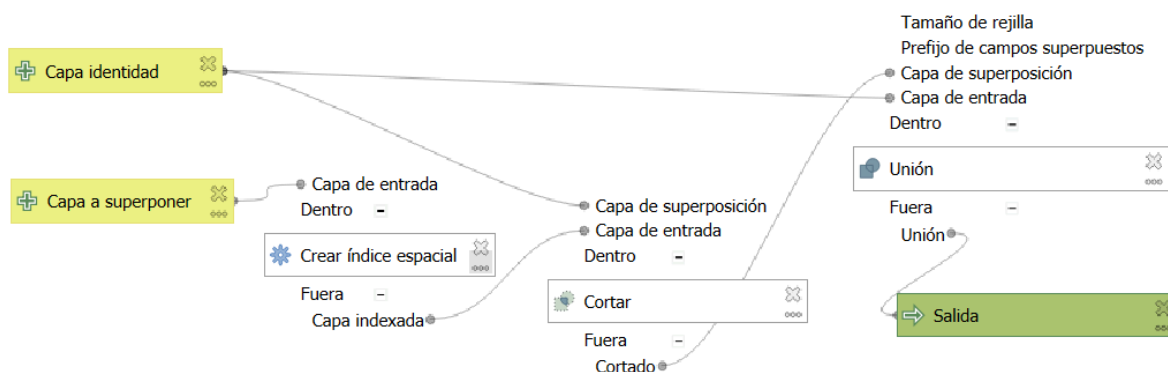


Imagen 2. Herramienta de superposición vectorial de identidad

Paso 5. Para obtener las estadísticas agregadas por un campo específico, utilizar la herramienta de **Agregar** de la categoría de **Geometría Vectorial** con el resultado de la capa en el paso anterior. En la herramienta de *agregación* seleccionar el resultado de identificación (1) luego, seleccionar el campo de **CodigoUC** (2) para agregar las estadísticas por el código de la UC. En (3) por último, borrar todos los campos y posteriormente **Añadir campo nuevo** y diligenciar de acuerdo a la Imagen 3.

Observar que los campos resultantes **area_ha** y **porcentaje** requieren definir una expresión de cálculo de área, de acuerdo a la geometría y configuración de medidas del proyecto. Para el campo de porcentaje se formula como la suma de las partes agregadas de **UC** como $100 * (\$area / area_dep_ha)$. Recordar que los cálculos de área son en ha.

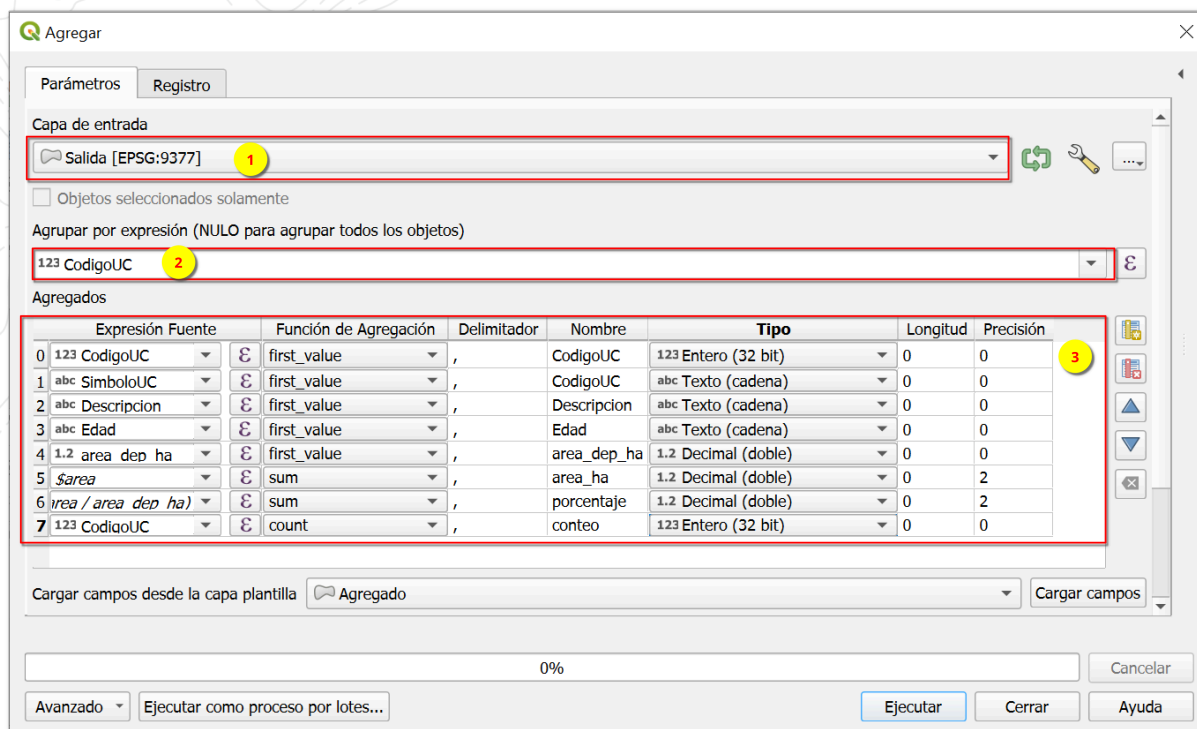


Imagen 3. Estadísticas de agregación por nivel1 de CLC en el departamento

Con el resultado de la agregación, validar con la expresión `sum(<nombre_campo>)` que los resultados de `area_ha` y `porcentaje` son correctos.

4. Generación de estadísticas y gráficos

El análisis espacial requiere la generación de reportes, resúmenes y gráficos estadísticos. QGIS ofrece algunas herramientas nativas, pero generalmente se extiende con complementos especializados de terceros, como **Data Plotly**, o la integración en el entorno de procesamiento con el proveedor de algoritmos de R.

A continuación, se muestra una de las funcionalidades más interesantes para conectar QGIS con un paquete estadístico como **GNU R**.

4.1. Instalación y configuración de librerías de R en QGIS

Uno de los puntos cruciales de la configuración del entorno de procesamiento de R en QGIS es la instalación de la versión correspondiente y los paquetes necesarios.

Para la descarga de las librerías o paquetes de R usar el siguiente enlace Web:

Descarga de [librerías de R](#).

Descomprimir los archivos en la carpeta del perfil de usuario, usando la opción el menú de QGIS `Configuración` » `Perfiles de usuario` » `Abrir carpeta del perfil activo` y extraer el contenido de la carpeta en

processing/rlibs. Es importante aclarar que todos los paquetes deben estar en la carpeta correspondiente.

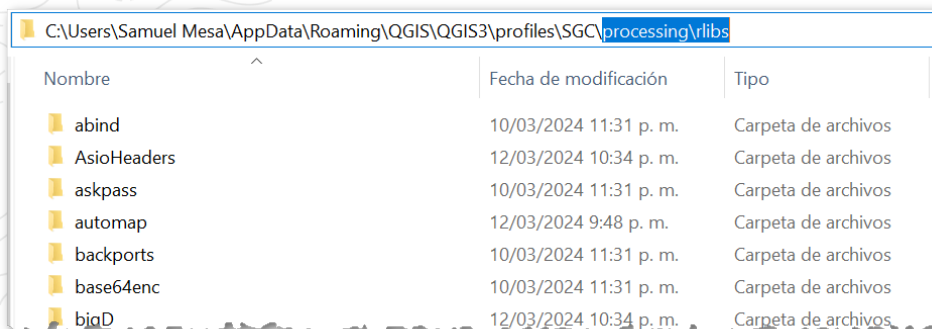


Imagen 4. Instalación de paquetes de R

Adicionalmente, instalar el complemento de **Processing R Provider**.

4.2. La sintaxis del procesamiento con el lenguaje de R en QGIS

El presente ejercicio no tiene como objetivo abarcar los conocimientos completos en el lenguaje de **R**, sino proporcionar las bases necesarias para comprender cómo se integra en el entorno de procesamiento de **QGIS**. Para mayor información, se recomienda consultar la documentación oficial del complemento del proveedor de algoritmos de **R**, disponible en: <https://north-road.github.io/qgis-processing-r/>.

R es un lenguaje de programación ampliamente utilizado en estadísticas y análisis de datos. Su popularidad radica en su capacidad para manipular, visualizar y modelar datos de manera efectiva. En el ámbito geoespacial, **R** se destaca por sus numerosas bibliotecas y paquetes especializados que permiten el análisis espacial, la cartografía y la visualización de datos geográficos. Paquetes como “**sp**”, “**rgdal**” y “**leaflet**” son herramientas para explorar patrones espaciales, realizar análisis de interpolación, crear mapas interactivos, entre otras aplicaciones. Esto hace que **R** sea una herramienta útil para profesionales y académicos que trabajan en campos como la geografía, la ecología, la planificación urbana y la epidemiología, entre otros.

Puede escribir *scripts* y llamar comandos de **R** en **QGIS**, de manera similar a como lo haría desde el entorno de **R**. Esta sección proporciona la sintaxis para utilizar comandos de **R** en **QGIS**, así como la forma de interactuar con objetos de **QGIS**, como capas y tablas, dentro de dichos *scripts*.



Recomendaciones generales de programación de *scripts* de R en QGIS

A diferencia de **Python**, **QGIS** no proporciona un intérprete interactivo para las instrucciones de **R**. En su lugar, ejecuta un *script* con una salida específica, que puede ser el registro o log de la consola, imagen en formato **HTML**, tabla, o capa

raster o vectorial. Como recomendación general, se puede emplear la integración actual de inteligencia artificial de chatbots para facilitar la generación de código con una salida específica, y luego ejecutarlo en alguno de los IDE ofrecidos de R, incluyendo servicios en línea como <https://rdr.io/snippets>.

La sintaxis de un Script de R en QGIS

Los archivos de *script* R tienen la extensión **.rsx**, y su creación es bastante sencilla si se cuenta con un conocimiento básico de la sintaxis en R. Estos archivos deben almacenarse en la carpeta de *scripts* de R, cuya ubicación se configura por defecto en la carpeta **processing/rscripts** del perfil de usuario de QGIS.

Paso 6. Para crear un *script* de R en QGIS, utilizar la barra de herramientas de procesamiento y haga clic en el icono . Luego, seleccionar “**Create New R Script...**”. En este *script*, se utilizarán las siguientes instrucciones para generar estadísticas y gráficas de Unidades Cronoestratigráficas (UC) agregadas por departamento, como resultado del Paso 5. Para integrar el código proporcionado en el archivo de `sgc_analisis_uc.rsx` en los datos del ejercicio, primero pegar o abrir el archivo en el editor de *scripts* de R en QGIS. Luego, seleccionar la opción “**Guardar script como**”  para direccionar el almacenamiento hacia la carpeta estándar de los *scripts* de R. Este paso es crucial para asegurar su disponibilidad y despliegue adecuado en la rama de R dentro de QGIS. A continuación, se explicará la sintaxis a seguir para integrar R con QGIS.

```
1  ##SGC=group
2  ##analisis_UC=name
3  ##Análisis de UC=display_name
4  ##Capa=vector
5  ##UC=Field Capa
6  ##Porcentajes=Field Capa
7  ##Area=Field Capa
8  ##output_plots_to_html
9
10 #Carga e instalación de las bibliotecas necesarias de R
11 if (!require("ggplot2")) {
12   install.packages("ggplot2")
13 }
14 library(ggplot2)
15
16 if (!require("gridExtra")) {
17   install.packages("gridExtra")
18 }
19 library(gridExtra)
20 library(grid)
21
22 #Crear dataframe a partir de los campos de Capa
```

```
23 datos <- data.frame(uc = Capa[[UC]],
24                     porc = Capa[[Porcentajes]],
25                     area = Capa[[Area]])
26
27 #Algunos ejemplos de imprimir en el log o registro
28 head(datos)
29 paste("El área total es", sum(datos$area), sep = " ")
30 summary(datos)
31
32 #Se crea una grilla con 4 filas
33 #Texto del título del resumen
34 texto_titulo <- textGrob("Reporte de UC", gp = gpar(fontsize = 12, fontface = "bold"))
35
36 #Título para la tabla
37 titulo_tabla <- textGrob("Tabla de UC", gp = gpar(fontsize = 12))
38
39 #Gráfico 1: Diagrama de barras con rampa de color de amarillo a rojo,
40 #leyenda horizontal y etiquetas de categorías rotadas a 45 grados
41 grafico <- ggplot(datos, aes(x = uc, y = porc, fill = porc)) +
42   geom_bar(stat = "identity") +
43   labs(title = "Gráfico de barras", x = "UC", y = "%") +
44   scale_fill_gradient(low = "yellow", high = "red", name = "Porcentaje") +
45   theme(
46     plot.margin = unit(c(0.1, 0.1, 0.1, 0.1), "lines"),
47     plot.title = element_text(hjust = 0.5),
48     legend.position = "bottom",
49     axis.text.x = element_text(angle = 45, hjust = 1, size = 8)
50   ) +
51   guides(fill = guide_legend(direction = "horizontal"))
52
53 #Ordena los datos en orden descendente del porcentaje de cobertura
54 tabla = datos[order(Capa[[Porcentajes]], decreasing = TRUE), ]
55 # Crear tabla con tema de fuente ajustado
56 tabla_grafico <- tableGrob(head(tabla))
57
58 # Ajustar el tamaño de fuente de la tabla
59 tabla_grafico$grobs[[1]]$gp <- gpar(fontsize = 8)
60
61 #Organizar el resumen y gráfico en una columna con cinco filas
62 grid.arrange(texto_titulo, titulo_tabla, tabla_grafico, grafico,
63             nrow = 4, heights = c(0.25, 0.25, 1.7, 3.2))
64
65 #Documentación del algoritmo
66 #' Capa: Capa vectorial
67 #' Coberturas: Campo de UC
68 #' Porcentajes: Campo de porcentajes de la capa vectorial
69 #' RPLOTS: Salida de la imagen en HTML
70 #' ALG_DESC: Este script permite la salida de un resumen y
71 #'           : gráfico de barras de las UC
72 #' ALG_CREATOR: Samuel Mesa
```



```
73 #' ALG_HELP_CREATOR: Samuel Mesa
74 #' ALG_VERSION: 1.0
```

Metadatos

Las primeras líneas (1 al 3), que inician con un doble signo de comentario de **Python** y **R** (**##**), definen el nombre para mostrar (Análisis de UC) y el grupo del *script* (**SGC**), e informan a **QGIS** sobre sus entradas y salidas. A continuación, se detallan los metadatos requeridos de acuerdo a la documentación oficial del [proveedor de R en QGIS](#).

`##script_name=name` es el nombre del *script*; bajo este nombre, se listará en la caja de herramientas de procesamiento. También se utiliza para definir el identificador del *script* en el proveedor de procesamiento de R.

`##script_title=display_name` es el título del *script*; bajo este título, se listará en la caja de herramientas de procesamiento. Anula el nombre si se define después.

`##group_name=group` es el nombre del grupo del *script*, lo que permite clasificar los *scripts* en grupos en la caja de herramientas de procesamiento.

Entradas

En las líneas 4 a 7 se especifican los datos de entrada y los parámetros. Hay varios tipos de entradas:

`##Capa = vector` Capa vectorial.

`##Campo = Field Layer` donde *Layer* es el nombre de una capa vectorial entrante a la que pertenece el campo.

`##Raster = raster` Una capa raster.

`##Tabla = table` Una tabla.

`##Numero = number` Un valor numérico.

`##Texto = string` Una cadena de texto.

`##Booleano = boolean` Un campo booleano o de chequeo.

Elementos en un menú desplegable. Los elementos deben estar separados por punto y coma.; `##type=selection point;lines;point+lines`

Salidas

Al igual que para las entradas, cada salida se debe definir al principio del *script*. Ver línea 8:

`##Vectorial=output vector` Salida de una capa vectorial.

`##Raster= output raster` Salida de una capa raster

`##Tabla=output table` Salida de una tabla

`##output_plots_to_html` Salida de un gráfico en **HTML**

Para mostrar la salida de **R** en el visor de resultados, coloque `>` delante del comando cuya salida se quiere mostrar, por ejemplo, para obtener las salidas de las líneas **28**, **29** y **30**.

Cuerpo de líneas de código

Para una explicación más detallada y para mejorar el *script* propuesto, se puede recurrir al uso de la inteligencia artificial de chatbots. El cuerpo del *script* corresponde a las líneas **10 al 63**.

Las líneas **10 a 20** se usan para cargar e instalar librerías, se emplean con las funciones `install.packages("<paquete>")` y `library(<paquete>)`. Los paquetes **ggplot2** y **grid** permiten generar salidas gráficas en una plantilla de la grilla.

Las líneas **22 al 25** permiten crear una estructura muy común en el análisis de datos, denominada *Data Frame*. Este consiste en un conjunto de datos tabulares compuesto por filas y columnas, en este caso, extraído de los campos seleccionados por el usuario.

Las líneas **27 al 30** ejecutan y muestran algunos comandos en la pestaña de registro en la ventana de ejecución del *script* de **R**. La función **head** muestra las primeras filas del *data frame*, **paste** concatena cadenas de texto, y la función **summary** imprime un resumen estadístico de los datos.

Dado que la salida del *script* es un archivo **HTML**, en las líneas **34, 37, 56 y 59** se generan las imágenes del texto que se desplegarán como encabezados en la plantilla de la grilla, que se muestra en las líneas **61 al 63**.

Finalmente, en las líneas **39 a 51**, se genera el gráfico de barras utilizando la sintaxis propia de la popular biblioteca de gráficos de **R** conocida como **ggplot2**. Esta sintaxis implica una construcción por capas, donde se especifican los datos, el tipo de gráfico y la personalización del tema del gráfico.

Ayuda de la herramienta

Al seguir la sintaxis del *script* de **R**, es posible registrar, organizar, desplegar y ejecutar como una herramienta dentro del entorno de procesamiento. Para agregar la documentación en la ventana de la herramienta, se utiliza un archivo con el mismo nombre del *script* y la extensión **.rsx.help**. También se puede documentar dentro del mismo *script*, como se hace en las líneas **65 al 74** del *script*. Se emplea **"# Parameter: Description"** para especificar en el formato clave-valor la descripción de la documentación, incluyendo algunos parámetros especiales como **RPLOTS**,

ALG_DESC, **ALG_CREATOR**, **ALG_HELP_CREATOR**, **ALG_VERSION** para definir la salida, descripción, autoría y versión del *script*.

4.3. Ejecutando el *script* de R y consultar la salida

Paso 7. Una vez desarrollado y almacenado el *script*, se recomienda ejecutarlo desde la caja de herramientas, ubicada en la categoría de **R** » **SGC** » **Análisis de UC**. Al hacerlo, se abrirá la ventana estándar de **QGIS**, donde se mostrarán los parámetros de entrada, salida y otras opciones disponibles para la ejecución en modo por lotes y registro

* Nota 1

En la versión actual de **QGIS**, no es posible ejecutar la herramienta desde el editor de *scripts* de **R**. Por lo tanto, se recomienda realizar esta tarea directamente desde la caja de herramientas de procesamiento (**Caso número 74**).

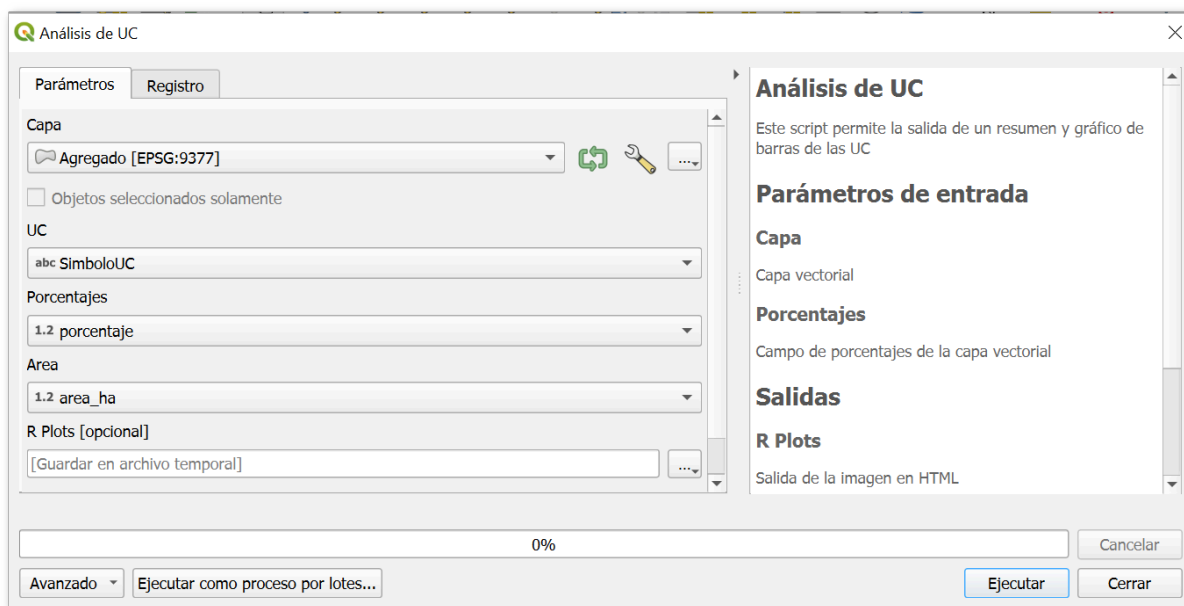


Imagen 5. Ventana de herramienta del *script* de R

Con lo anterior, se obtiene el siguiente gráfico y salida, tanto en la pestaña de registro como un enlace del archivo **HTML** que contiene la imagen de los gráficos.

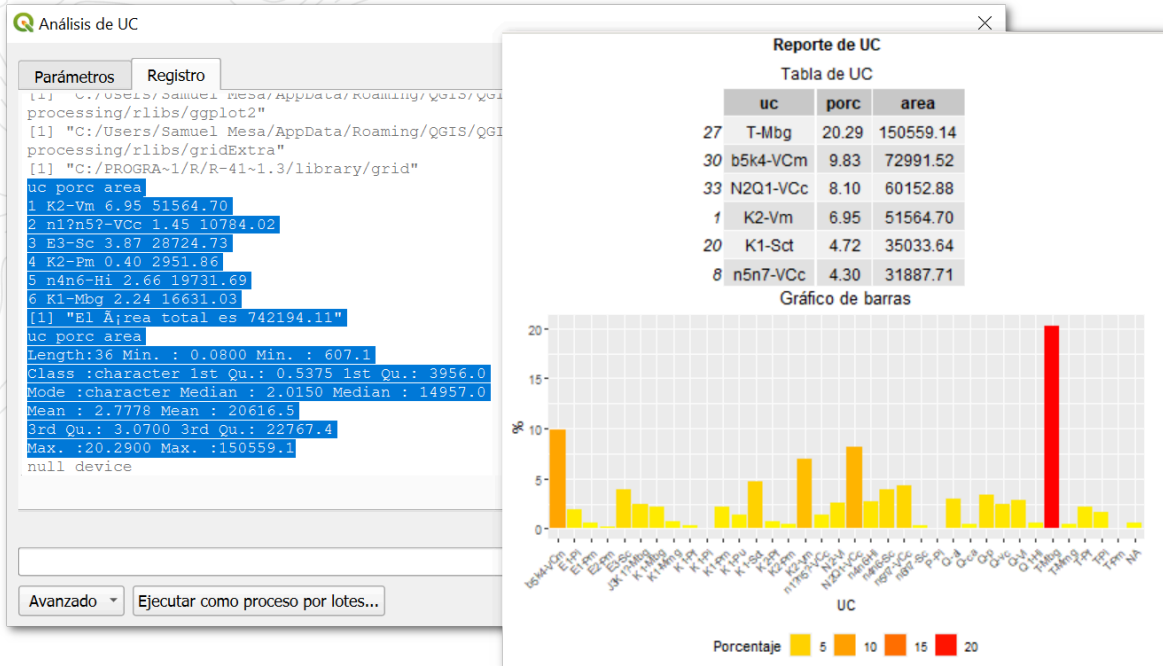


Imagen 6. Resultado del log de consola de R y gráfico del análisis de UC

Consejo 1

Se recomienda generar gráficos en *scripts* separados debido a la limitación de tamaño a 480x480 píxeles. El *script* anterior tiene como objetivo simplificar la adición de plantillas de grilla.

5. Entregable del ejercicio

Realizar el ejercicio de análisis vectorial y desarrollo del *script* de R.

- Extender el *script* con un parámetro de texto para permitir que el usuario ingrese el nombre del departamento o municipio del análisis. Concatenar este texto con el título del gráfico (**texto_titulo**) para desplegarlo en el gráfico generado.
- Obtener los resultados de la agregación de Unidades Cronoestratigráficas (UC) para el departamento en el geopackage llamado `caldas.gpkg`, y la salida del *script* de R en formato **HTML** con el nombre `caldas.html`, ubicado en la carpeta denominada **plots**.
- Seleccionar un municipio de Caldas y realizar la agregación de Unidades Cronoestratigráficas (UC) en él. Guardar la salida en un geopackage y un archivo **HTML** con el nombre del municipio seleccionado. Utilizar la carpeta **plots** para almacenar la salida de los gráficos de R.

* Nota 2

La entrega corresponde a la carpeta comprimida con los resultados anteriores. Esto corresponde a los geopackages de salida por departamento y municipio seleccionado, así como la carpeta **plots** con los archivos de los gráficos. El archivo comprimido **no debe superar** más de los 20 MB en disco.

Ejercicio 5

Análisis espacial raster

1. Introducción

El modelo de datos raster se basa en estructuras matriciales numéricas simples, lo que permite la aplicación de una amplia variedad de algoritmos enfocados en el álgebra de mapas, análisis del terreno y análisis hidrológico. Los primeros sistemas de información geográfica (**SIG**) se centraron en el manejo de esta estructura y la generación de funciones de análisis espacial raster. En **QGIS**, muchos de los proveedores de algoritmos (*GRASS GIS, PC Raster, GRASS GIS, Whitebox Tools, SAGA GIS*) proporcionan acceso principal a estas herramientas.

En el presente ejercicio se muestra la integración de las herramientas raster proporcionadas por los diferentes proveedores de algoritmos configurados en **QGIS**. Además, se explorarán las herramientas disponibles en el álgebra de mapas, calculadora raster, análisis del terreno y el apoyo al análisis espacial de evaluación multicriterio.

Como objetivo del ejercicio de análisis espacial raster, se lleva a cabo la generación de **zonificación de susceptibilidad por movimientos en masa**, usando Evaluación Multicriterio (EMC). De acuerdo al **SGC** *“El término susceptibilidad a movimientos en masa, se define como el grado de disposición, fragilidad o propensión de un terreno a generar procesos de este tipo. Esta se evalúa a partir de las características físicas intrínsecas y ambientales que el terreno posee.”* El método utilizado es semicuantitativo (heurístico) donde los factores de inestabilidad son calificados, según el criterio del experto ([Consultar documento de la CVC](#)).

2. Datos, software y recursos necesarios

Para el desarrollo correcto del presente ejercicio práctico se requiere la revisión de la sesión de conceptos teóricos de la semana 4 del Nivel Avanzado de **QGIS**:

- Revisión de los contenidos teóricos de la cuarta semana (grabación disponible en la **Plataforma de Aulas Virtuales del SGC**)([Enlace web](#)).
- **Datos del ejercicio**: corresponde a los datos raster de la semana 4: Descargar [datos](#).

3. Análisis espacial raster

El análisis raster es un enfoque para el procesamiento y análisis de datos geográficos que utiliza una estructura de matriz o de cuadrícula. En lugar de representar datos como entidades discretas vectoriales, los datos raster se representan generalmente como una distribución geográfica continua en el espacio en forma de celdas. Las operaciones pueden incluir operaciones aritméticas y lógicas del álgebra de mapas, el análisis del terreno basado en el modelo digital de elevaciones y el análisis hidrológico (extracción de cuencas hidrográficas y la modelización del flujo de agua). El análisis raster es una herramienta poderosa en el análisis espacial debido a su capacidad para procesar grandes conjuntos de datos y generar resultados cuantitativos y visualmente informativos.

3.1. Análisis del terreno


Se realizan una serie de operaciones basadas en la capa de relieve del terreno de la zona de estudio, utilizando las curvas de nivel de la cartografía base a escala 1:25.000 del **IGAC**, con el fin de extraer información relevante para el análisis espacial de zonificación de susceptibilidad por movimientos en masa planteado al final del ejercicio.

Interpolación espacial

La interpolación espacial es un método para estimar valores desconocidos en un área a partir de valores conocidos en puntos cercanos. Se utiliza en análisis espacial para crear superficies continuas a partir de datos discretos. Se basa en la suposición de que los datos espaciales están correlacionados y que los valores cercanos son más similares entre sí.

Paso 1. A partir de la base de datos del **IGAC-25k.gpkg** cargar las capas de curvas de nivel y la capa de zona de estudio. Guardar el proyecto de **QGIS** con el nombre **zonficacion.qgz**.

Paso 2. Realizar la interpolación de las curvas de nivel con la herramienta nativa de **QGIS** de **Interpolación** » **Interpolación TIN** con los parámetros que se muestran en la Imagen 1.

En **(1)** seleccionar la capa de **curva_nivel** con el valor de campo a interpolar correspondiente a la cota de **ALTURA_SOBRE_NIVEL_MAR**. En **(2)** agregar los campos al algoritmo, en **(3)** definir la extensión geográfica a partir de la propia capa de curvas de nivel, haciendo uso del botón . En **(4)** definir la resolución espacial en 15 metros.

Al ejecutar la herramienta se despliega el Modelo Digital del Terreno (**DTM**) a partir de las curvas de nivel.

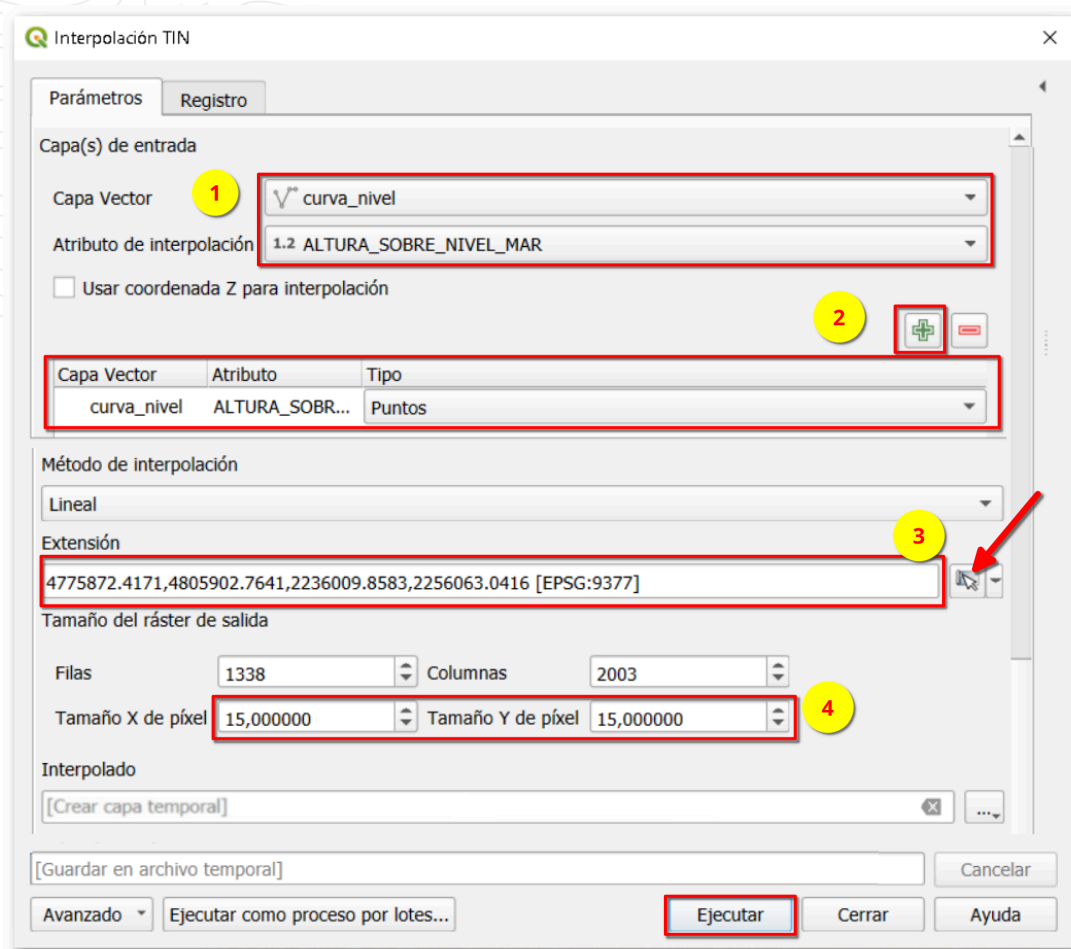


Imagen 1. Interpolación a partir de las curvas de nivel

Conversión del formato raster

Para integrar otros proveedores de algoritmos de procesamiento raster, a menudo es necesario convertir o traducir formatos y tipos de datos. En el caso de los algoritmos *PC Raster*, se requiere la conversión al formato propio con extensión **.map**. Por otro lado, la mayoría de los algoritmos como *GRASS GIS* o *SAGA GIS* admiten la conversión directa a sus formatos nativos. Sin embargo, en algunos casos, la conversión de formatos puede afectar el tiempo de procesamiento.

Paso 3. Realizar la conversión del tipo de raster usando la herramienta desde el menú **Raster** » **Conversión** » **Traducir (convertir formato)...** y seleccionar los parámetros de la capa resultante de la interpolación (1), **SRC** (2) el tipo de datos de almacenamiento (3) y finalmente, el nombre del archivo en la capeta de los datos del proyecto (4) de nombre `dtm.tif`.

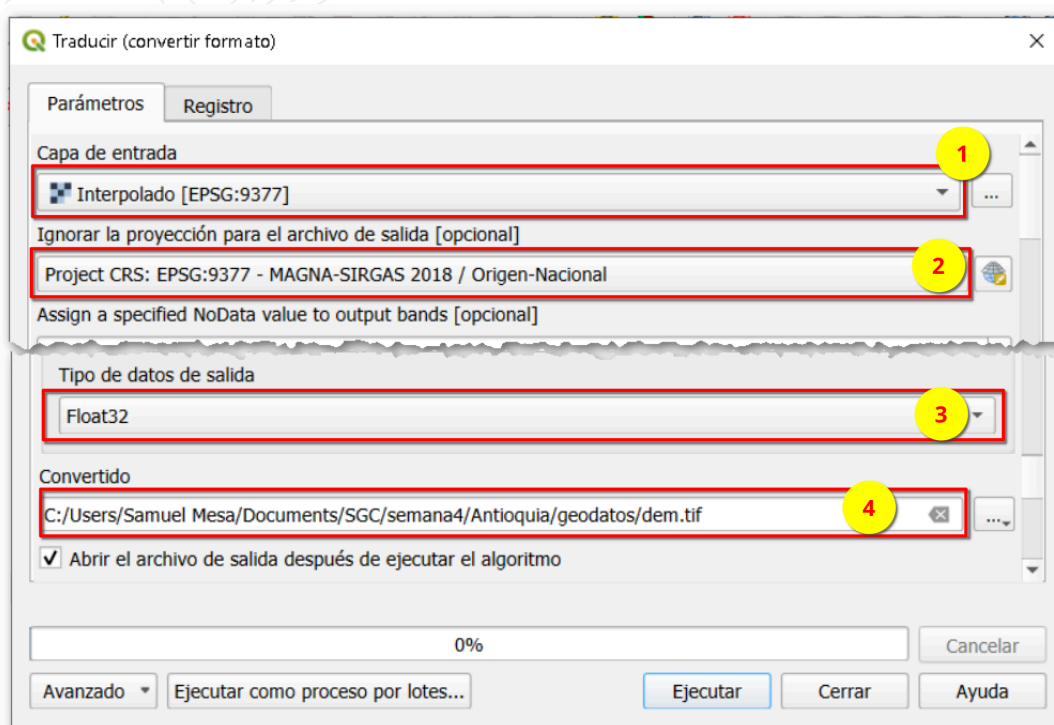


Imagen 2. Conversión el tipo de datos admitido por WBT

Lo anterior realiza la conversión del formato **GeoTiff** de *Float64* a *Float32*, este último soportado por el proveedor de algoritmos de *Whitebox Tools*.

Cálculo de las pendientes del terreno

El resultado de este proceso consiste en un paso intermedio en el análisis de zonificación de susceptibilidad por movimientos en masa planteado al final del ejercicio. La pendiente del terreno se define como la inclinación en relación con el plano horizontal o gradiente de altura topográfica. La unidad de medida de la pendiente se puede realizar en grados, porcentaje o radianes.

Paso 4. Integrar el proveedor de algoritmos de *Whitebox Tools* (también se puede instalar el complemento reciente de *Whitebox Workflows for QGIS*) para el cálculo de las pendientes en diferentes unidades. Desde **WhiteBoxTools** » **Geomorphometric Analysis** o **WhiteBox Workflows**, seleccionar la herramienta de **slope** con los parámetros como se muestra en la imagen a continuación. La capa de entrada corresponde al **DTM** y La unidad de medida de la pendiente es en **porcentajes**.

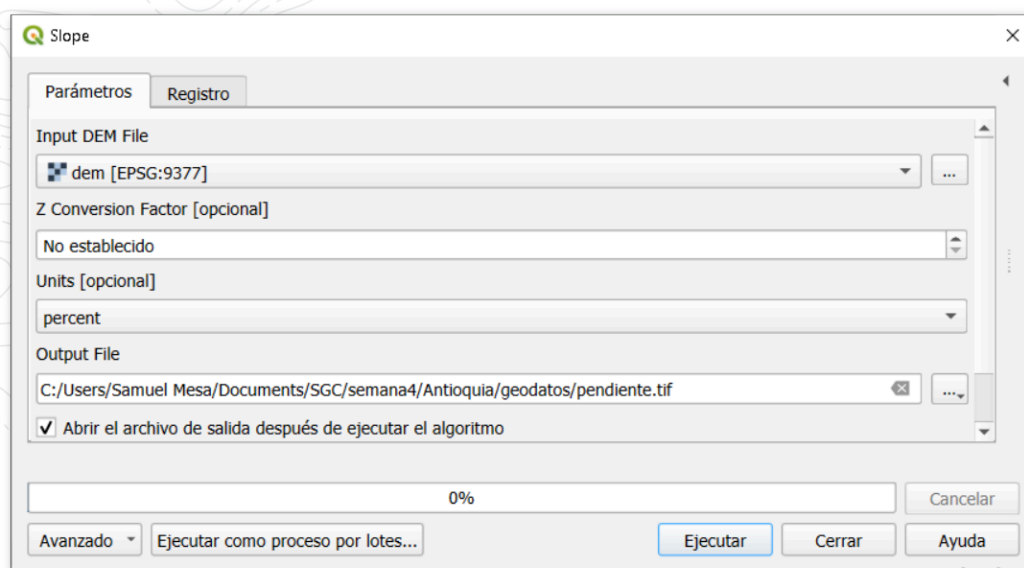


Imagen 3. Calculo de pendiente desde WhiteBox Tools

3.2. Álgebra de mapas

El álgebra de mapas es un conjunto de operaciones aritméticas y lógicas aplicadas a capas raster para generar nuevas capas a partir de ellas. Es una herramienta fundamental en el análisis raster, ya que permite realizar operaciones complejas y generar nueva información a partir de los datos originales.

Reclasificación en una capa raster

El resultado de este proceso retornará un insumo para el análisis raster. El objetivo consiste en la estandarización de los valores de la clasificación basada en la zonificación de áreas homogéneas del **IGAC**, con el fin de integrarla en la evaluación multicriterio.

Rango de pendientes	Gradiente %	Clasificación
A nivel o plano	0% al 1%	1
Ligeramente plana	1% al 3%	2
Ligeramente inclinada	3% al 7%	2
Moderadamente inclinada	7% al 12%	3
Fuertemente inclinada	12% al 25%	3
Ligeramente escarpada o ligeramente empinada	25% al 50%	4
Moderadamente escarpada o moderadamente empinada	50% al 75%	5
Fuertemente escarpada o fuertemente empinada	> 75%	5

Tabla 1. Tabla de reclasificación por susceptibilidad de pendiente (SP)

Los valores del 1 al 5 representan una clasificación de las categorías, desde la menor ocurrencia de susceptibilidad por movimientos en masa (1) hasta la más probable (5). Esta clasificación se obtiene mediante métodos heurísticos y es específica para el presente ejercicio académico.

Paso 5. Desplegar la herramienta nativa de **QGIS** » **Análisis Raster** » **Reclasificar por tabla** y realizar la reclasificación de acuerdo con la **tabla de estandarización de susceptibilidad por pendiente**.

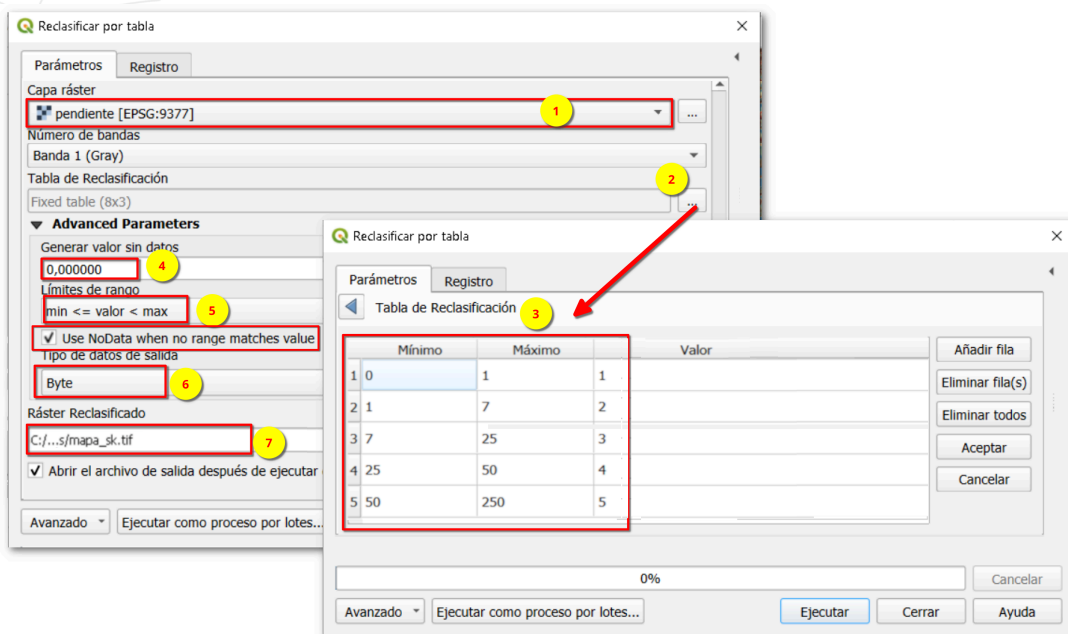


Imagen 4. Herramienta de superposición vectorial de identidad

En el proceso de análisis, se utilizan diferentes capas y parámetros para generar el primer criterio de susceptibilidad por pendiente. (1) Usar la capa de **pendiente** en porcentajes como entrada, en (2) y (3) acceder al panel de **Tabla de Reclassificación** para definir los intervalos. (4) Es importante especificar el **valor sin datos** (*NoData*) con el valor de 0 y habilitar la opción **“Use NoData when no range matches value”**. (5) Definir los límites de rango utilizados en la tabla. (6) Configurar el tipo de datos permitidos como 1 Byte (8 bits) con un límite de 256 valores. Finalmente, (7) redireccionar la salida del primer criterio de susceptibilidad por pendiente, denominado `mapa_sk.tif`. Es importante aclarar que el límite de 250% fue obtenido para los datos de pendientes de la **zona de estudio**, ya que este valor puede ser mayor en otros casos.

3.3. Evaluación multicriterio

La evaluación multicriterio en SIG es un proceso de toma de decisiones que utiliza múltiples criterios para integrar diferentes opciones o escenarios espaciales. El análisis de raster, es un proceso que implica la creación de capas raster para cada criterio, y la combinación de dichas capas mediante **álgebra de mapas** para producir una capa final evaluada. Este ejercicio tiene como objetivo generar un análisis espacial de zonificación de susceptibilidad por movimientos en masa, utilizando un método heurístico como la evaluación de criterios y calificación por expertos. Este enfoque permite integrar la experiencia y el conocimiento en la evaluación, y la definición de áreas con características similares en cuanto a litología (geología), geomorfología, cobertura de la tierra, suelos, densidad de drenajes, densidad de fallas y pendientes. Se recomienda que este proceso de selección de criterios y calificación se realice por parte de expertos.

Definición de factores

Para llevar a cabo un análisis de evaluación multicriterio, se deben definir los factores y restricciones. Para el ejercicio, se integran los siguientes criterios como **factores de susceptibilidad**, obtenidos a partir de capas de la cartografía ofrecida por el IGAC, IDEAM y el SGC: litología, cobertura de la tierra y criterio de pendiente. Se hace uso de la siguiente escala numérica de calificación de los factores de susceptibilidad definidos en valores de 1 al 5:

Calificación de susceptibilidad	Descripción de la susceptibilidad
1	Muy baja
2	Baja
3	Moderada
4	Alta
5	Muy alta

Tabla 2. Escala numérica de evaluación de susceptibilidad

Cuando se habla de una alta y baja susceptibilidad, se hace referencia al **grado de propensión** asociado con un área en particular. Una alta susceptibilidad indica que un área tiene una mayor posibilidad de ser afectada por un evento geológico adverso, mientras que una baja susceptibilidad indica que un área tiene una menor posibilidad de ser afectada.

Definición de restricciones

Las restricciones corresponden a una condición o limitación del análisis o la operación del conjunto de datos en una zona determinada, ya sea por aspectos físicos, legales, infraestructura u otros. Para el presente ejercicio, se usa las capas del **IGAC** para restringir o descartar los análisis en cuerpos de agua como drenajes dobles y embalses.

Paso 6. Extraer los cuerpos de agua del **IGAC** de la zona de estudio. Se recomienda usar **QGIS** » **Superposición vectorial** » **Diferencia múltiple** para restar las geometrías. Guardar el resultado de la capa como **zona_restriccion.gpkg**.

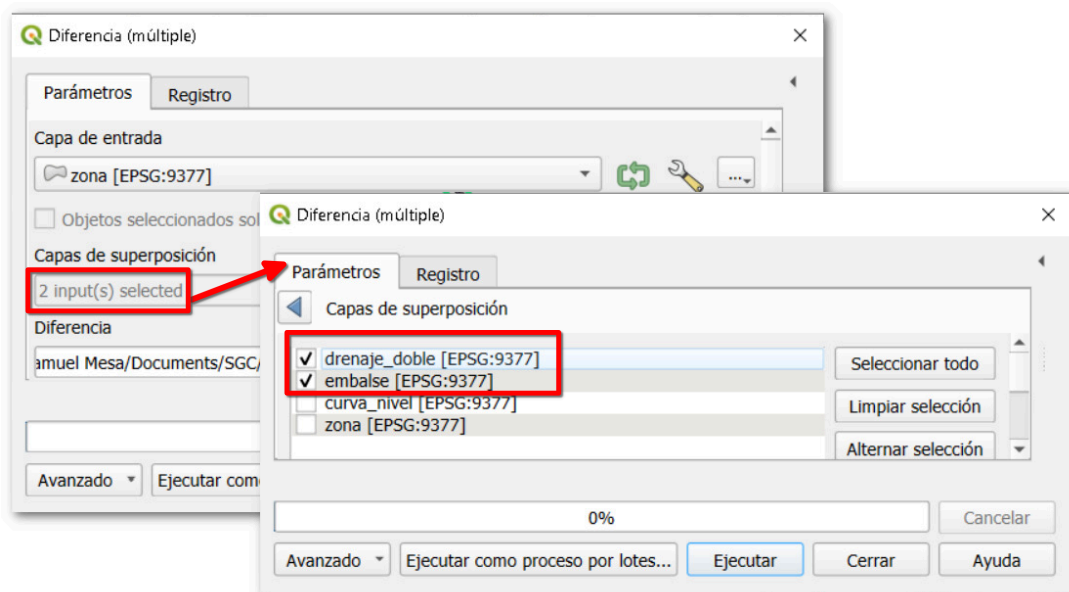


Imagen 5. Generación de restricciones en la zona de estudio

Factor de susceptibilidad por pendiente (SK)

El primer factor generado en el Paso 4 es la susceptibilidad por pendiente (**SK**). Corresponde al grado de inclinación de las laderas naturales, que puede afectar la resistencia al corte de los diferentes materiales que conforman los taludes en el área de estudio.

Recortar una capa raster por máscara

Para el mapa de susceptibilidad por pendiente se requiere recortar a la zona de estudio. Para eso se usa una máscara que permite definir los límites de análisis.

Paso 7. Usar la herramienta del proveedor de algoritmos de **GDAL** » **Extracción raster** » **Cortar raster por capa de máscara**, seleccionamos la capa de susceptibilidad de pendiente, resultado del Paso 4 y

seleccionar la máscara vectorial de zona con restricciones. Configurar el valor de **NoData** como 0 y la capa de salida de mapa de susceptibilidad de nombre **SK.tif**.

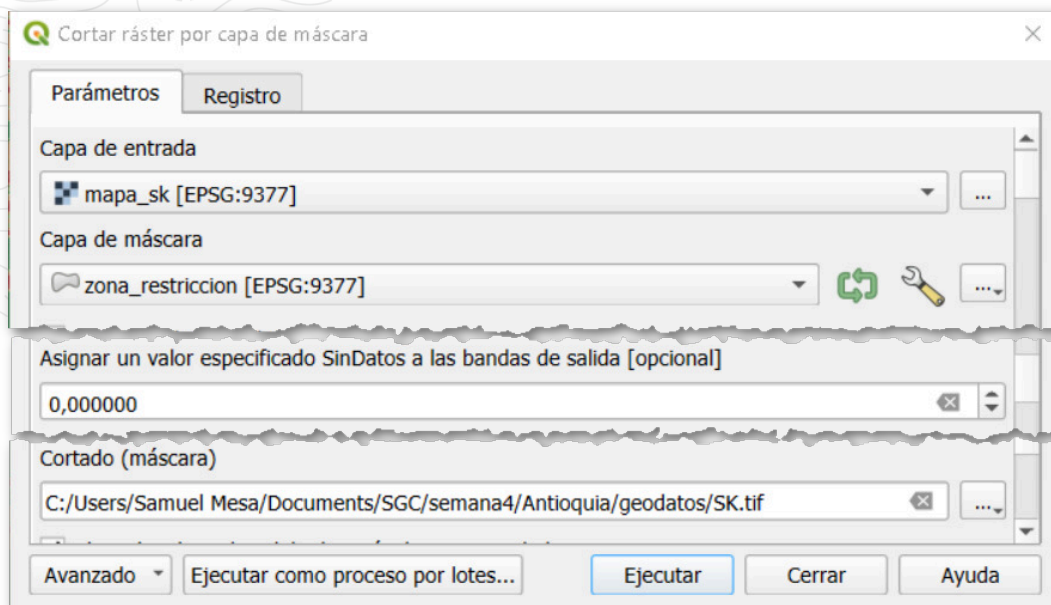


Imagen 6. Recorte por máscara de una capa raster

Factor de susceptibilidad de litología (SL)

Para la zona de estudio se compone por unidades geológicas principalmente de tipo metamórfico e ígneo y minoritariamente de tipo sedimentario. A continuación la evaluación del factor de susceptibilidad.

Litología	Símbolo	Calificación SL
Mármol	m	2
Neis Feldespático y aluminio	nf	4
Roca verde	rv	4
Cuarcita, cuarcita biotítica oscura y neis cuarzoso	nq	3
Neis	ni	4
Cuarzodiorita	Kqd	3
Cuarzo	q	2
Limo, arena y grava	Qal	4

Tabla 3. Escala numérica de evaluación de susceptibilidad

Dado que la capa de partida es vectorial, se recomienda utilizar la calculadora de expresiones para asignar la calificación a partir de una función condicional y posteriormente realizar la conversión a un formato raster.

Paso 8. Agregar la capa de geología al proyecto y crear un campo numérico entero de nombre **susceptibilidad** con la siguiente expresión en la calculadora de campos:

```
CASE
  WHEN "COD" IN ('m', 'q') THEN 2
  WHEN "COD" IN ('nq', 'Kqd') THEN 3
  WHEN "COD" IN ('nf', 'rv', 'ni', 'Qal') THEN 4
END
```

Rasterizar la capa vectorial.

Una vez calculador el campo de **susceptibilidad**, es necesario realiza la conversión de vectorial a raster.

Paso 9. Usando la herramienta del proveedor de algoritmo de **GDAL** » **Conversión vectorial** » **Rasterizar (vectorial a raster)**. Seleccionar los parámetros manejados para el proyecto, como la resolución espacial de 15 m (en unidades georeferenciadas), el valor de **NoData** en 0 y el tipo de dato en *Byte*. Para generar los valores numéricos o nivel digital de cada celda en el raster, use el campo numérico calculado anteriormente, de nombre **susceptibilidad** en la opción “*Campo a usar para un valor de marcada*”.

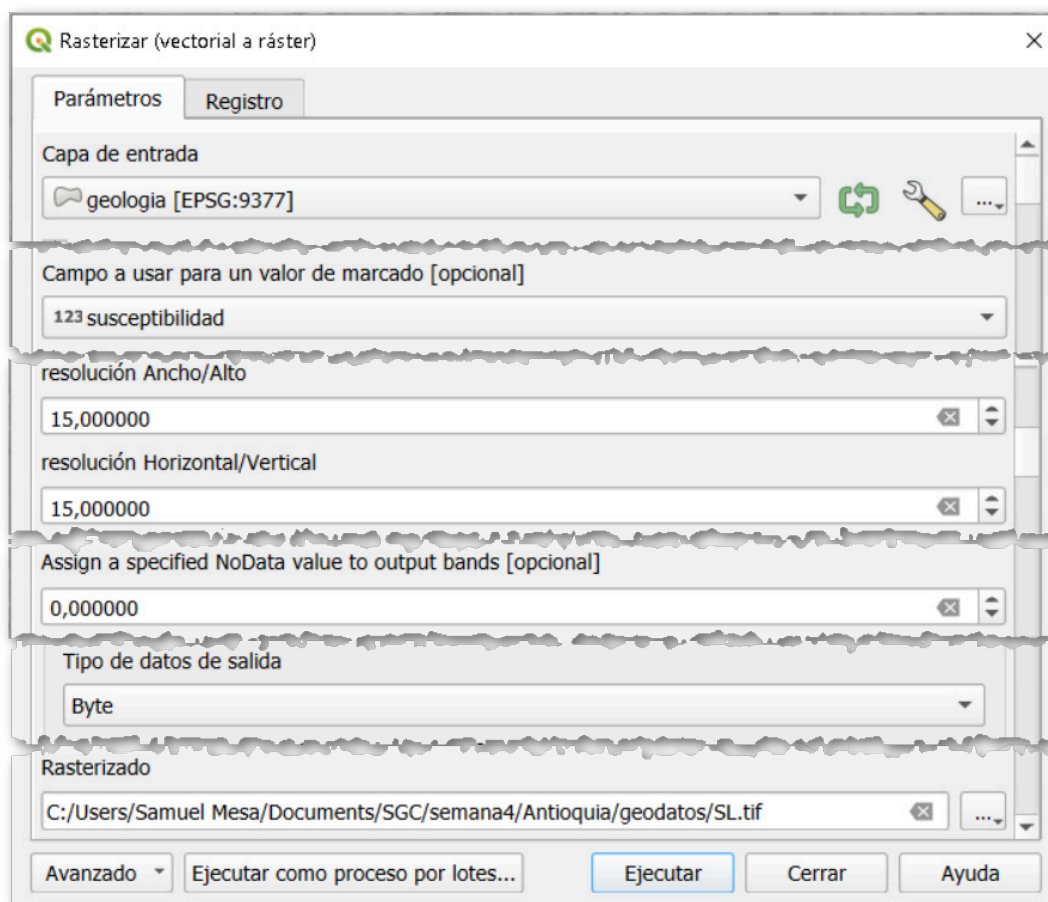


Imagen 7. Parámetros para conversión al formato raster

Paso 10. Realizar también la rasterización de la capa vectorial de la `zona_restriccion.gpkg`, y guardar la capa con el nombre `restriccion.tif`. Utilizar en este caso la opción de “**un valor fijo para marcar**” como 1 y parámetros restantes como en la Imagen 7.

Factor de susceptibilidad de cobertura de la tierra (SC)

La cobertura vegetal es un factor importante que influye en la estabilidad de los suelos de forma indirecta, ya que si esta se pierde el suelo queda desprotegido siendo más propenso a presentar erosión. A continuación el factor calificado.

Cobertura de la tierra	Código Nivel 3	Calificación SC
Bosque denso	311	2
Cuerpos de aguas artificiales	514	NA
Mosaico de cultivos, pastos y espacios naturales	243	3
Mosaico de pastos con espacios naturales	244	3
Pastos limpios	231	3
Pastos enmalezados	511	NA
Obras hidráulicas	125	NA
Vegetación secundaria o en transición	323	3
Bosque denso	311	3
Zonas industriales o comerciales	121	1

Tabla 4. Escala numérica de evaluación de susceptibilidad

Paso 11. Agregar la capa de `clc` desde la base de datos de `clc_2028.gpkg`. Plantear la expresión de la calculadora de campos y generar el mapa de susceptibilidad por cobertura terrestre de nombres **SC**. El valor de **NA** indica que *No Aplica* y debe tratarse como valor **NULL** u omitirse en la tabla.

3.4. La calculadora raster

Como estructura matricial los datos raster se pueden operar a través de la calculadora raster, herramienta que permite llevar a cabo operaciones aritméticas y lógicas e integrar diferentes capas. En la evaluación multicriterio se genera el mapa global o final evaluado a través de una regla de decisión, como una operación aritmético-estadístico que se realizan sobre cada uno de los mapas de susceptibilidad relativas. Para el presente ejercicio se usa una regla de decisión de sumatoria simple, que se especifica por la siguiente fórmula:

$$S_g = R * \frac{(SK + SL + SC... + Sn)}{n}$$

Donde:

S_g : Susceptibilidad global por movimientos en masa

R : mapa de zona de estudio con restricciones

SK : Susceptibilidad relativa de pendiente

SL : Susceptibilidad relativa de litología

SC : Susceptibilidad relativa de cobertura de la tierra

Sn : Susceptibilidad relativa n

n : número total de factores evaluados

La división entre el número total de factores evaluados se realiza para mantener la escala original del mapa global de susceptibilidad de 1 a 5.

Paso 12. Con los tres mapas de susceptibilidad relativa de pendiente (**SK**), susceptibilidad de litología (**SL**) y susceptibilidad de cobertura de la tierra (**SC**), obtener el mapa de susceptibilidad global (**Sg**) con la formulación de regla de decisión. Para esto puede usar la herramienta nativa de **QGIS** » **Análisis raster** » **Calculadora raster** y seleccionar los tres mapas con la expresión definida de suma lineal simple, incluya la máscara del Paso 8, para aplicar las restricciones sobre el resultado.

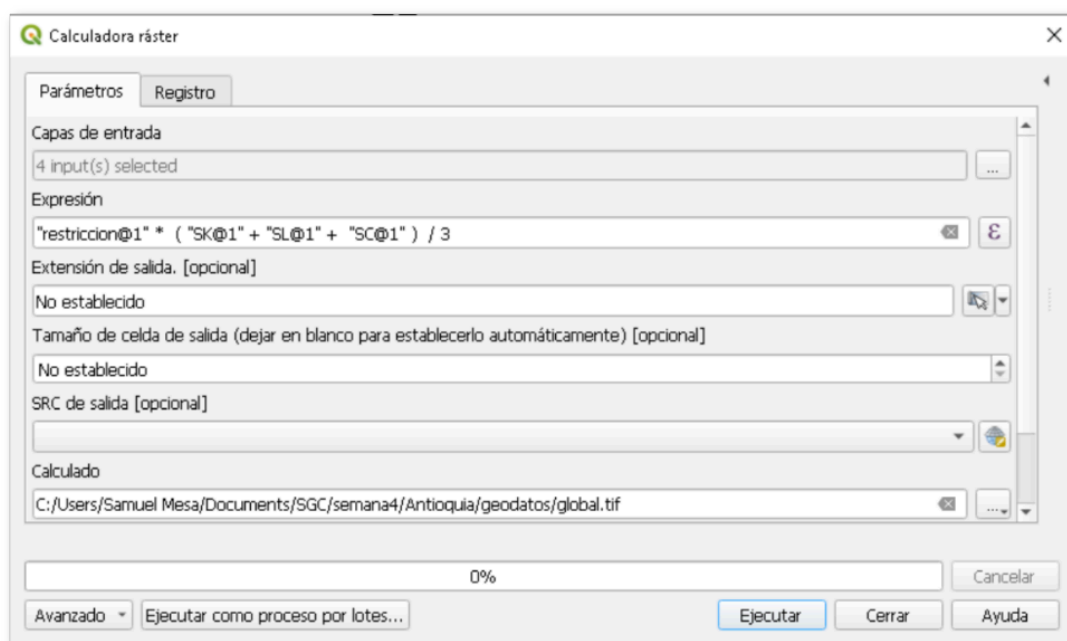


Imagen 8. Calculadora de la susceptibilidad global

Paso 13. Reclassificar los valores obtenidos del mapa anterior, para la conversión de valores decimales a discretos, utilizando reglas de clasificación adecuadas de 1 a 5, definidas en la Tabla 2. Almacenar este resultado como `susceptibilidad.tif`.

3.5. Estadísticas zonales

Paso 14. Para generar algunas métricas sobre datos raster, no es necesaria la conversión a formatos vectoriales. En su lugar, se puede utilizar la herramienta de estadísticas zonales de la capa raster. Para esto usar algunas de las herramientas nativas de **QGIS** » **Análisis raster** » **Estadísticas zonales de la capa raster**, seleccionado la capa de susceptibilidad global categorizada como capa de zona y restricción como capa de entrada de la zona de estudio.

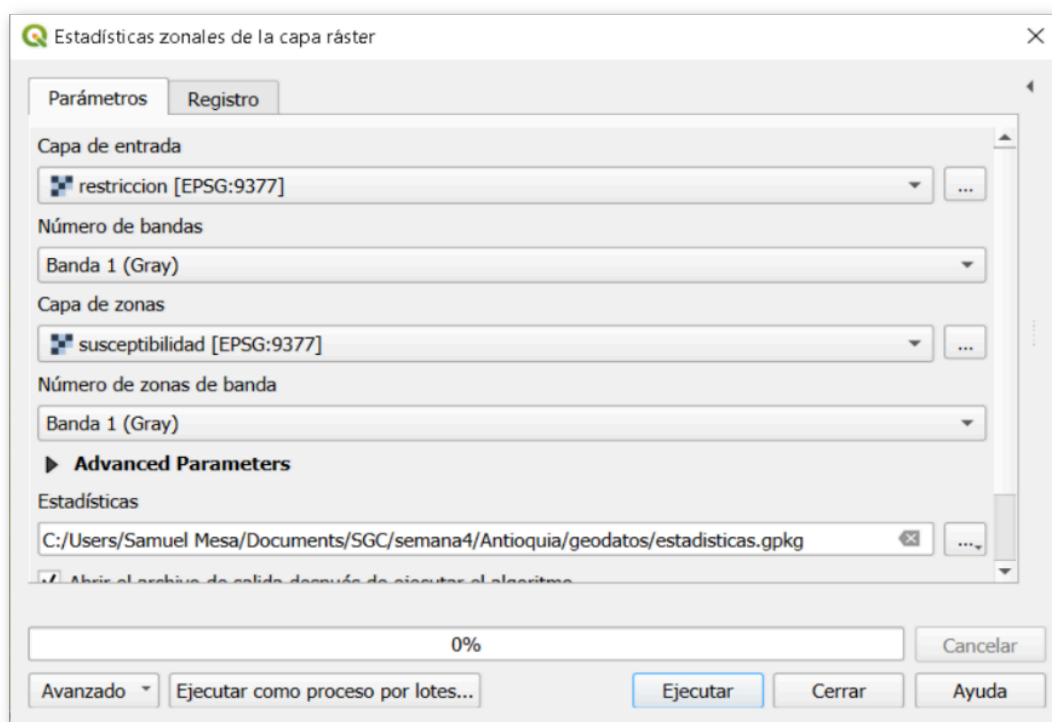


Imagen 9. Estadísticas del mapa de zonificación de susceptibilidad por movimientos en masa

Desde el proveedor de algoritmos de **Whitebox Tools** » **Math and Stats Tools** » **Raster Histogram**, también es posible generar un diagrama de histograma.

Finalmente, con la capa evaluada final, tabla e histograma se puede generar el mapa final de zonificación por movimientos en masa como muestra en la Imagen 10. En este, los cálculos de área en *ha* y *porcentajes* se obtienen a partir de las áreas calculadas por la herramienta de estadísticas zonales. El valor de porcentaje se

calcula basándose en el valor total de la zona de estudio, que corresponde a 4993.42 hectáreas.

Se recomienda utilizar la **Evaluación Multicriterio (EMC)** para valorar o ponderar los factores involucrados en la evaluación final. Una herramienta comúnmente empleada en la EMC es la comparación por pares de **Saaty** en el Proceso Analítico Jerárquico (AHP). Esta herramienta permite evaluar la importancia relativa de los criterios dentro de la evaluación final. La regla de decisión usada en la EMC en este caso se conoce como suma lineal ponderada.

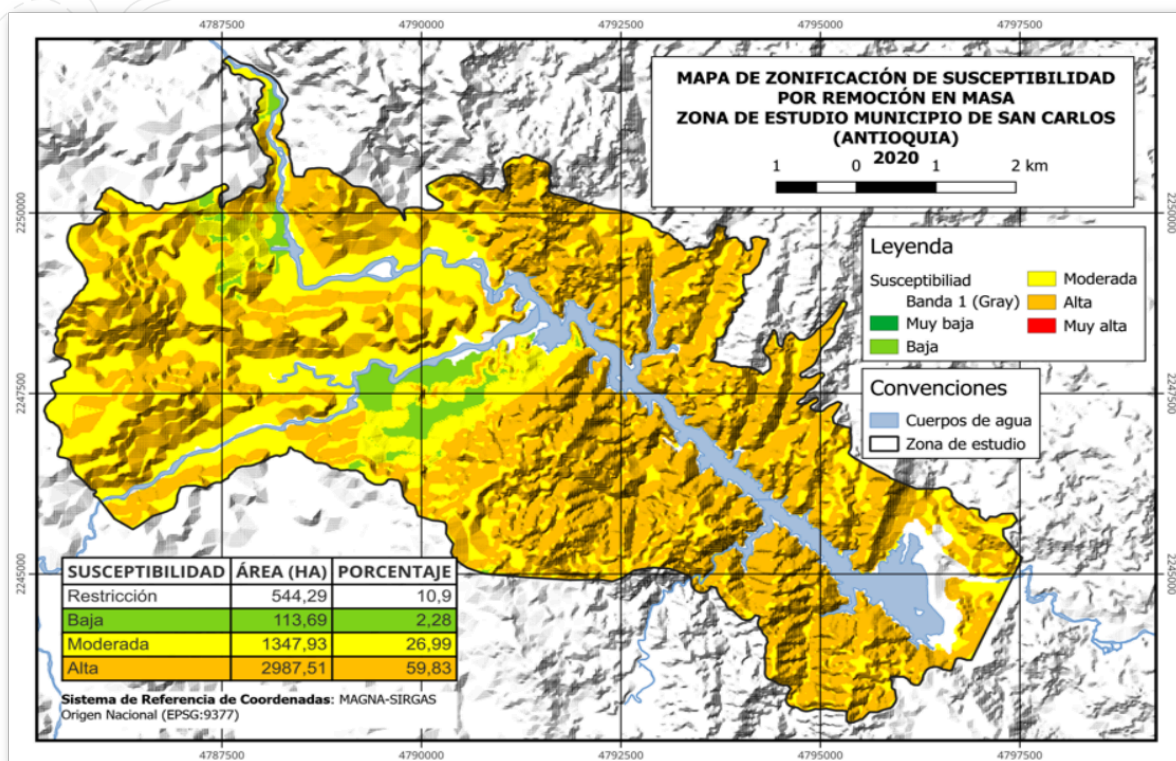


Imagen 10. Mapa de zonificación de susceptibilidad por movimientos en masa

4. Entregable del ejercicio

Realizar el ejercicio de análisis raster de evaluación multicriterio para generar el mapa de zonificación de susceptibilidad por movimientos en masa:

- Entregar el archivo comprimido en formato **ZIP** de la carpeta del proyecto de **QGIS** de nombre `zonificacion.zip`. Debe contener el proyecto en **QGZ**, las salidas de susceptibilidad global y susceptibilidades relativas. Excluya la base de datos de `IGAC-25k.gpkg` para cumplir con el tamaño de entrega.
- Generar el mapa en formato **PDF** de la zonificación de susceptibilidad por movimientos en masa. Incluya la tabla de estadísticas zonales.

* Nota 1

La entrega corresponde a la carpeta comprimida con los resultados anteriores. El archivo comprimido **no debe superar** más de los 100 MB en disco.

Ejercicio 6

Publicación de Geoservicios **WMS, WMTS, WFS y WCS**

1. Introducción

Como componente clave del ecosistema de herramientas de **QGIS**, el servidor de mapas **QGIS Server** permite la publicación de servicios web geográficos siguiendo los estándares abiertos establecidos por el **Open Geospatial Consortium (OGC)**, como **WMS, WMTS, WFS** y **WCS**. Estos servicios ofrecen la capacidad de realizar operaciones de consulta, visualización y procesamiento espacial a través de la plataforma web. Además, **QGIS Server** proporciona integración con una amplia gama de clientes de visores de mapas en la web, brindando interfaces flexibles y amigables para la publicación y consulta de datos geográficos. De esta manera, se extiende la posibilidad de acceso a través de una interfaz web, facilitando la interacción y el análisis de información geográfica a un público más amplio.

Es necesario aclarar que **QGIS Server** es el componente que se encarga de la publicación de los geoservicios, pero toda la configuración y administración de los mismos se apoya en la interfaz amigable de **QGIS** desktop.

En este ejercicio, se lleva a cabo el proceso de publicación de un proyecto de **QGIS**, abordando los requisitos técnicos necesarios para su implementación en un servidor de mapas. Además, se explora la posibilidad de desplegar un visor de mapas interactivo de manera sencilla y flexible, aprovechando tecnologías Web. Esto permite la publicación y el acceso eficiente y seguro a través de internet, facilitando la difusión y el análisis de datos geográficos en diferentes plataformas y dispositivos.

2. Datos, software y recursos necesarios

Para el desarrollo correcto del presente ejercicio práctico se requiere la revisión de la sesión de conceptos teóricos de la semana 5 del Nivel Avanzado de **QGIS**:

- Revisión de los contenidos teóricos de la quinta semana (grabación disponible en la **Plataforma de Aulas Virtuales** del **SGC**)([Enlace web](#)).
- **Datos del ejercicio:** Corresponde al proyecto del Departamento de Caldas: [datos](#).

3. Geoservicios OGC

Los **geoservicios OGC**, también conocidos como servicios web geográficos, son una forma de acceder a una amplia gama de funcionalidades ofrecidas por los Sistemas de Información Geográfica (**SIG**) a través de Internet. Esto se logra mediante el uso de software de servidor de mapas que sigue estándares definidos por el *Open Geospatial Consortium (OGC)*, lo que garantiza interoperabilidad y neutralidad con respecto a las herramientas de software utilizadas.

Estos servicios permiten acceder a diversas funcionalidades, como la visualización de mapas (**WMS, WMTS**), acceso a datos geoespaciales (**WFS, WCS**), consulta de metadatos (**CSW**) e incluso la ejecución de procesos geoespaciales en línea (**WPS**). Al seguir estándares y protocolos reconocidos, los geoservicios facilitan la integración y el intercambio de información geoespacial entre diferentes sistemas y plataformas, lo que los convierte en herramientas esenciales para la colaboración y el análisis geoespacial.

3.1. Configuración del Servidor de mapas y web

Para la publicación de geoservicios y de un visor de mapas en la web, es necesario realizar la configuración inicial el servidor de mapas (**QGIS Server**) con el servidor Web, este último tiene como función principal de almacenar, procesar y distribuir recursos web a través de internet.

Paso 1. Debido algunos errores generados por los archivos de configuración, es necesario renombrar el archivo `C:\OSGeo4W\httpd.d\httpd_qwc-services.conf` a `C:\OSGeo4W\httpd.d\httpd_qwc-services.conf.old`.

Realizar la modificación del archivo, para configurar la ruta del servidor de mapas de acuerdo a la versión instalada en el sistema, esto corresponde para la versión actual modificación de la línea 372 de `C:\OSGeo4W\apps\apache\conf\httpd.conf`. Se recomienda realizar una copia de este archivo antes de editar.

Cambiar:

```
ScriptAlias /cgi-bin/ "${SRVROOT}/cgi-bin/"
```

Por:

```
ScriptAlias /cgi-bin/ "C:/OSGeo4W/apps/qgis/bin/"
```

Modificar la línea 388 los permisos de acceso de la carpeta del *script*:

```
<Directory "${SRVROOT}/cgi-bin">  
    AllowOverride None
```



```
Options None
Require all granted
</Directory>
```

Por:

```
<Directory "C:/OSGeo4W/apps/qgis/bin">
AllowOverride None
AllowOverride None
Options ExecCGI
Require all granted
</Directory>
```


En la línea 437 habilitar y agregar las extensiones permitidas:

```
AddHandler cgi-script .cgi .fcgi .exe
```

Y en al final del archivo en la línea 547 agregar las siguientes variables de ambiente, modifique el proyecto de acuerdo a la ruta configurada de forma local:

```
SetEnv GDAL_DATA "C:/OSGeo4W/share/gdal"
SetEnv QGIS_AUTH_DB_DIR_PATH "C:/OSGeo4W/apps/qgis/resources"
SetEnv QGIS_PROJECT_FILE "C:/SGC/Caldas/caldas.qgz"
```

Paso 2. Abrir el programa de **OSGeo4W Shell** con permisos de administrador e instalar el servidor **Web de Apache**, con el comando `bin\apache-install.bat`. Es importante contar con el puerto del sistema 80 para la instalación del servidor de forma correcta.

Paso 3. Iniciar el software o programa de **Apache Monitor** para gestionar los servicios instalados, seleccionar el de nombre **Apache OSGeo4W Web Server** y hacer clic en el botón **Start** para iniciar el servidor web de apache. Apache Monitor aparece en la bandeja del sistema identificado con el icono .

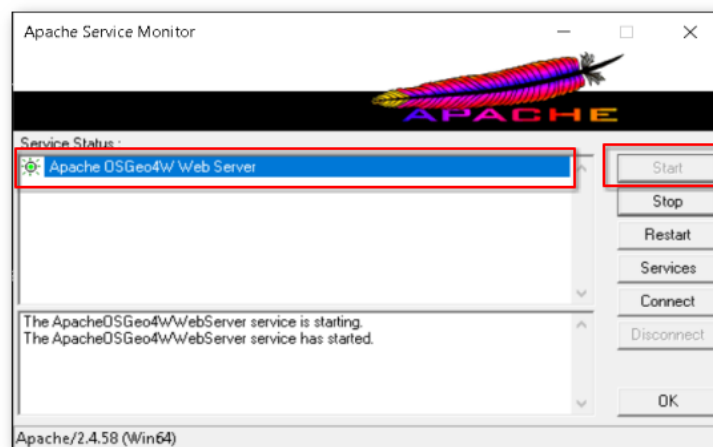


Imagen 1. Gestión de Servidor Web Apache

Paso 4. Para comprobar el funcionamiento correcto del despliegue, del servidor: http://localhost/cgi-bin/qgis_mapserv.fcgi.exe?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities

Lo que debe desplegar un archivo **XML** con el siguiente mensaje:

```
<ServerException>Project file error. For OWS services: please provide a SERVICE and a MAP parameter pointing to a valid QGIS project file</ServerException>
```

Consejo 1

“localhost” corresponde al nombre de dominio y “127.0.0.1” es la IP reservadas para establecer conexión a la máquina local, se reserva exclusivamente para la comunicación interna de la máquina.

3.2. Configuración del proyecto de QGIS

Para la publicación de geoservicios y de un visor de mapas en la web, es necesario seguir una serie de recomendaciones y especificaciones para publicar correctamente y, en consecuencia, ofrecer el acceso de los usuarios de forma eficiente e interoperable. Estas recomendaciones se orientan a la documentación de metadatos de los servicios, metadatos de las fuentes de los datos, del sistema de referencia de coordenadas, configuración de las capacidades del sistema y de las estructuras y nombres eficientes de publicación. Debido a que se trata de tecnologías en internet, es imprescindible tener en cuenta las limitaciones de latencia, es decir, el tiempo que tarda en transmitirse la información de un lugar a otro, la concurrencia, que se refiere a la capacidad del sistema para manejar múltiples solicitudes simultáneas, y la seguridad del sistema. Además, se debe considerar el encadenamiento con otros servicios web geográficos e integración con otros sistemas para garantizar compatibilidad con otras plataformas y sistemas.

Paso 5. Extraer la carpeta de datos suministrados del ejercicio y abrir el proyecto de nombre de `Caldas.qgz`. El proyecto consiste de las configuraciones de representación cartográfica generadas en la serie de ejercicios realizados durante el este desarrollo del entrenamiento.

Definiendo el Sistema de Referencia de Coordenadas Pseudo-Mercator

Una de las primeras recomendaciones es guardar tanto el proyecto como las capas utilizando el sistema de referencia de coordenadas **Pseudo-Mercator**, también conocido como *Web Mercator* (**EPSG:3857**). Esta recomendación se realiza con el propósito de establecer un estándar de facto que facilite la integración con la mayoría de los geoservicios o visores de mapas externos. Al emplear un único

SRC de coordenadas, se evita saturar el servidor de mapas con conversiones de coordenadas o reproyecciones cartográficas adicionales.

Paso 6. Todas las capas fueron reproyectadas para facilitar la integración del **SRC** a la recomendación de Web Mercator **EPSG:3857**, pero es necesario configurar o cambiar el **SRC** del proyecto para coincidir.

Configurar y documentar las capacidades del servicio

Los **geoservicios OGC** disponibles en **QGIS** encontramos de soporte nativo el *Servicio Web de Mapas (WMS)*, *Servicio Web de Entidades (WFS)* y OGC API, *Servicio Web de Coberturas (WCS)* y el *Servicio de Mapas de Teselas (WMTS)*. Cada geoservicio ofrece una serie de características que son necesarias habilitar o capacidades que se ofrecen al usuario final.

Paso 7. La configuración de las capacidades del servicio se realiza sobre las propiedades del proyecto en la pestaña Servidor de **QGIS**. En la misma ventana realizar por cada pestaña las siguientes configuraciones:

Services Capabilities:

- Nombre corto: caldas
- Título: Departamento de Caldas
- Persona: Su nombre
- Correo electrónico: Su correo
- Lista de palabras clave: **SGC**, Colombia, Caldas

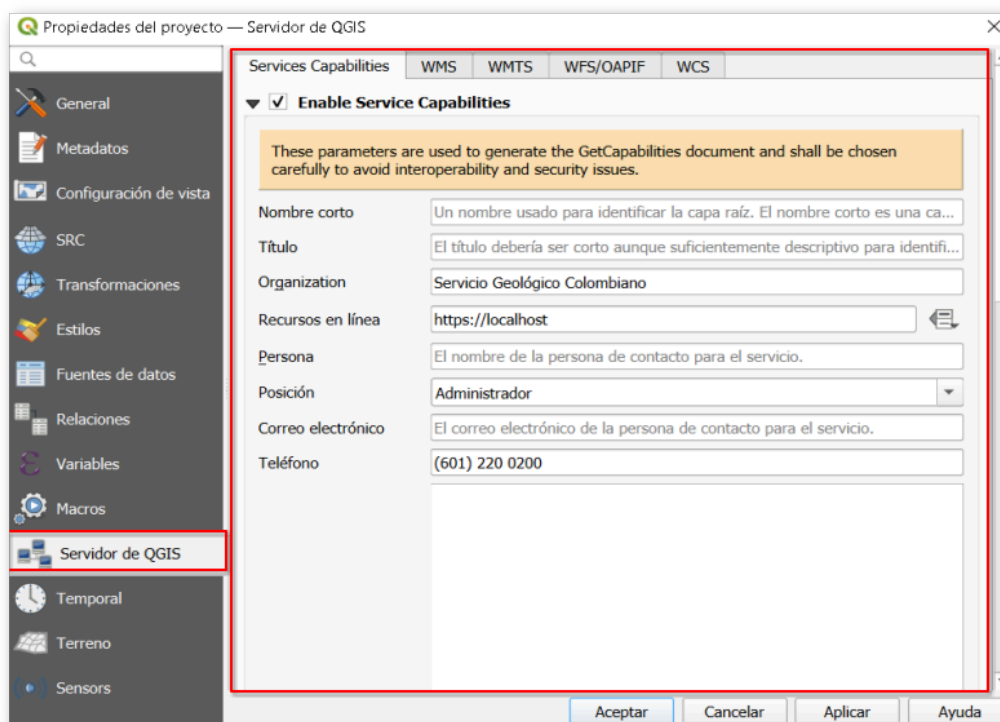


Imagen 2. Configuración de las capacidades de los geoservicios Web

WMS:

- Habilitar “*Advertised extent*” con la opción “Calcular a partir del marcador”
- Habilitar “*Restricciones del SRC*” con los **EPSG:3857**, **EPSG:4686**, **EPSG:9377**.
- Habilitar la opción “*Use attribute form settings for GetFeatureInfo response*”
- Habilitar la opción “*Añadir geometría a la respuesta del objeto*”
- Precisión de geometría GetfeatureInfo (lugares decimales): 4
- Escala predeterminada para la leyenda: 1:500000

WMTS:

- Capas publicadas
 - Mapa base: Habilitar “Publicado”, “PNG”, “JPEG”
- Unidad Cronoestratigráfica: Habilitar “Publicado”, “PNG”, “JPEG”
- Cuadrículas
 - **EPSG:3857**: Habilitar “Publicado”

WFS/OAPIF:

- Unidad Cronoestratigráfica
 - Habilitar “Publicado”
 - Precisión de geometría: 4
- Sismo
 - Habilitar “Publicar”
 - Precisión de geometría: 4
 - Habilitar “Actualizar”, “Insertar”, “Borrar”

WCS:

- DTM: habilitar “Publicado”
- Movimientos en masa 2019: habilitar “Publicado”

La configuración correspondiente a la capa de “**Sismo**” corresponde a la capacidad transaccional de edición de la misma.

Comprobación de configuración del servidor de QGIS

Para realizar la validación de las configuraciones realizadas, **QGIS** ofrece la validación del proyecto.

Paso 8. Al final de la ventana del servidor de **QGIS**, se ofrece la opción de comprobación de configuración, hacer clic en el botón **Lanzar**. Esto nos entrega información de las recomendaciones de ajustes del nombre de capas y del proyecto.

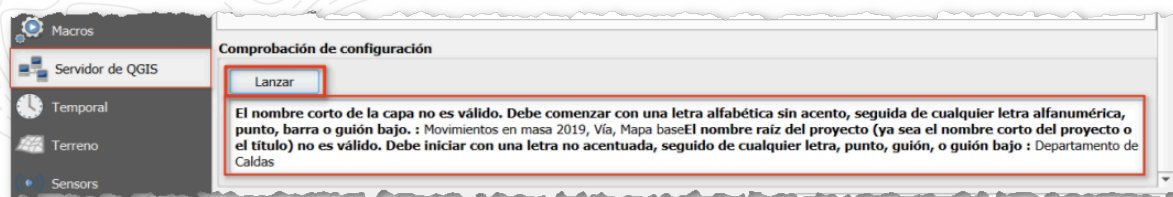


Imagen 3. Comprobación de la configuración del proyecto en el Servidor de mapas

Paso 9. Uno de los ajustes a realizar es usar un nombre corto del proyecto, en la ventana de **propiedades del proyecto** en la pestaña **General**, cambiar el Título del proyecto por **“Caldas”**.

En la misma venta de propiedades del proyecto revisar la pestaña de nombre de **“Fuente de datos”**, con las configuraciones de capacidades de identificación de capas, consultables y privadas.

Configuración de las capacidades de grupos y capas

Similar al proyecto, es posible realizar la configuración de capacidades de los geoservicios de los grupos y capas del proyecto, como los metadatos de identificación y habilitar opciones disponibles por cada geoservicio, ejemplo identificación, dimensiones de representación, etc.

Paso 10. Para el grupo de capas de **Mapa base**, agregar los metadatos de la capa en su publicación. Sobre la capa realizar clic derecho y cambiar **“Establecer datos WMS de grupo...”** y establecer como nombre corto **“mapa_base”**, título como **“Mapa base”** y Resumen **“Mapa base del Departamento de caldas”**.

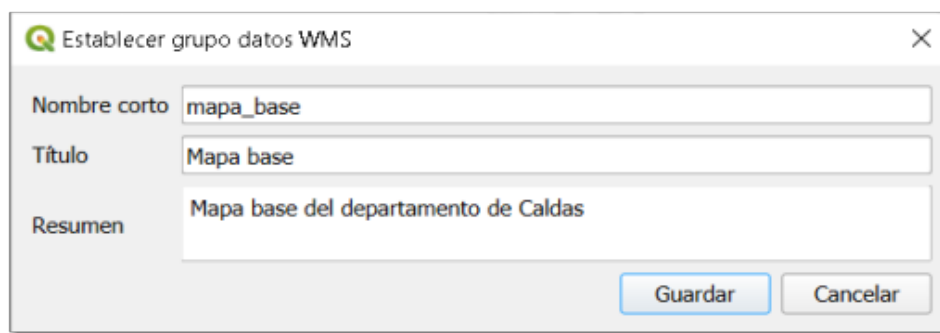


Imagen 4. Metadatos WMS de un grupo de capas

Paso 11. Ingresar a las propiedades de la capa de **“Movimientos en masa 2019”** y en la pestaña de Servidor de **QGIS**, realizar el diligenciamiento del nombre corto por **“movimientos_masa_2019”**, título **“Movimientos en masa 2019”** y resumen **“Movimientos en masa 2019 del SGC”**, lista de palabras clave **“movimientos en masa, sgc, Caldas, Colombia”**, diligenciar opcionalmente la atribución, URL de datos, atribución y metadatos.

Realizar el mismo proceso de generación de metadatos para la capa de “Unidad cronoestratigráfica” (UC).

Realizados los cambios realizar nuevamente la comprobación de configuración del servidor de mapas en las propiedad del proyecto.

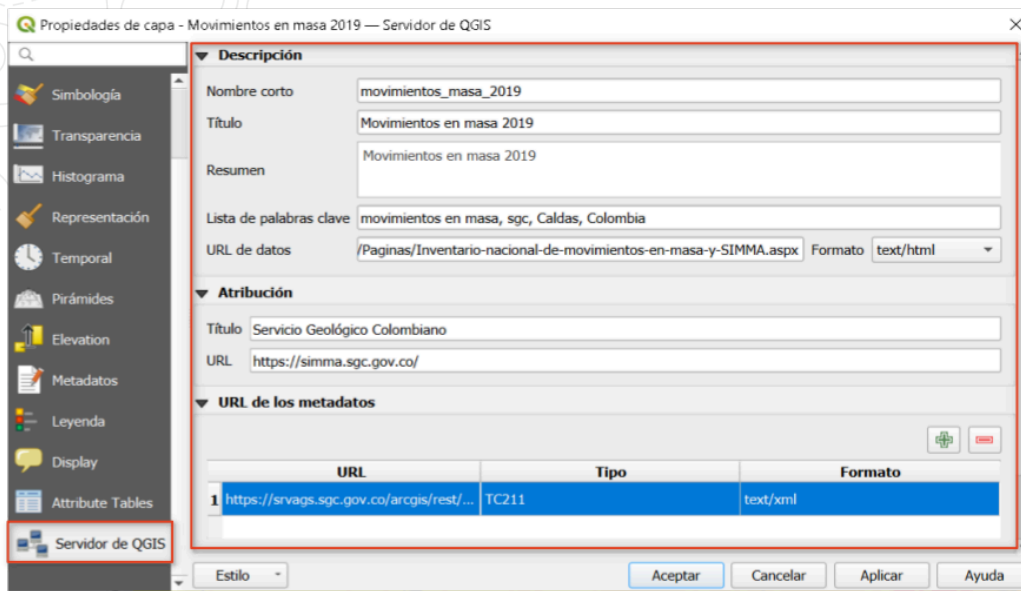


Imagen 5. Configuración de metadatos y capacidades por capa

Configuración de capacidad de dimensión de tiempo en capa geográfica

Los servicios OGC ofrece la posibilidad de configurar la dimensión de tiempo para su despliegue multitemporal en un cliente web o de escritorio.

Paso 12. Ingresar a las propiedades de Servidor de QGIS para la capa de **sismos**, diligencia los metadatos mínimos y adicionalmente agregar al final del formulario una dimensión de tiempo, seleccionado el campo correspondiente

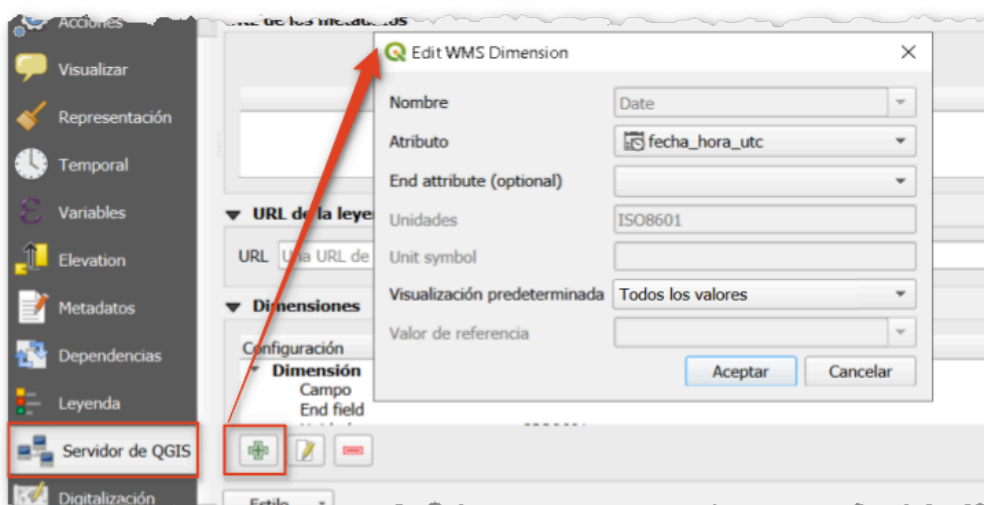


Imagen 6. Configuración de dimensión de tiempo para la capa de sismos.

Configuración de los campos y formularios de identificación del WMS

Una de las operaciones disponibles por el servicio **WMS** es el **GetFeatureInfo**, este permite retornar las propiedades de identificación en una coordenada específica. A partir de los formularios de cada capa, es posible desplegar correctamente los nombres y capas de interés.

Para configurar los campos a exponer en cada servicio **WMS** o **WFS**, sobre las propiedades de la capa y en la pestaña “**Campos**” verificar en la última columna de nombre “**Configuración**”.

Paso 13. Acceder a las propiedades de la capa de **Municipio** y configurar los alias de forma correcta para los campos de **código**, **nombre** de municipio, así como el **área** en km² y **acto administrativo**. Consultar las capas de “**sismos**” y “**Unidad cronoestratigráfica**” como ejemplo de despliegue del formulario y los nombres de los campos a través de los alias.

3.3. Despliegue de los Geoservicios OGC

Para acceder a las funcionalidades de los **geoservicios OGC** se requiere por lo mínimo un cliente ligero como un navegador web, ya que la publicación se basan en gran parte en los protocolos, estándares y formatos conocidos en la web. Para esto generalmente se usan peticiones al servidor a través de los métodos del protocolo **http**, generalmente los métodos **GET**, cuyos parámetros de consulta viajan como para de clave y valor por la **URL** o método **POST**, con parámetros a través del cuerpo del mensaje.

Peticiones de consulta de capacidades de los geoservicio OGC

Los **Servicios OGC** retorna el documento de las capacidades configuradas y publicados como un contrato con el cliente o usuario de consulta, esto permite establecer las funcionalidades configuradas en el servidor. Para realizar dicha petición se invoca la operación **GetCapabilities**.

Paso 14. Configurar la siguiente petición de acuerdo a la ruta del proyecto **QGZ**, realizar el despliegue del resultado en el navegador Web.

http://localhost/cgi-bin/qgis_mapserv.fcgi.exe?SERVICE=WMS&REQUEST=GetCapabilities

Es posible agregar el parámetro **MAP** con la ruta del proyecto **QGZ** para el despliegue, para este caso dado que se encuentra configurado la variable de entorno de nombre **QGIS_PROJECT_FILE** en el servidor **Web de Apache**, perite acceder al documento **XML** de las capacidades del servicio **WMS**.

Consejo 2

XML es un lenguaje de marcado para estructurar datos legibles tanto para humanos como para máquinas. Se usa para intercambiar datos entre sistemas, almacenar configuraciones de aplicaciones, comunicarse entre servicios web y representar documentos estructurados.

Para el resto de **geoservicios OGC** es posible retornar los metadatos cambiando el valor de la llave de nombre **“SERVICE”** por **WFS, WMTS y WCS**.



Imagen 7. Capacidades del servicio WMS y de la capa «Sismo» en formato XML

Para un despliegue en arquitectura **RESTful** con **OGC API (WFS3)**:

http://localhost/cgi-bin/qgis_mapserv.fcgi.exe/wfs3/

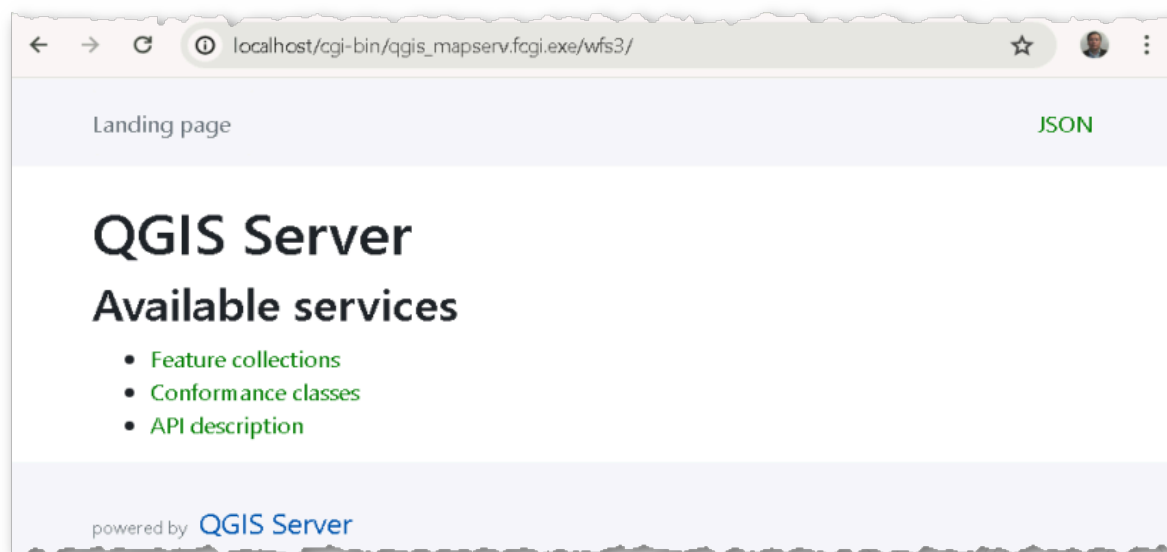


Imagen 8. Despliegue del recurso RESTful OGC API

Peticiones de retorno de representación y datos de los geoservicio OGC

Para retornar las funcionalidades de cada servicio, lo recomendable es utilizar algunos de los clientes de escritorio **SIG** que soporte la conexión a geoservicios de la **OGC**.

Paso 15. Abrir una nueva instancia de **QGIS** con un proyecto en blanco, desde allí realizar la conexión a los servicios publicados en rol de cliente. La **URL USADA** para los geoservicios son los siguientes:

WMS, WFS y WCS

http://localhost/cgi-bin/qgis_mapserv.fcgi.exe

WMTS:

http://localhost/cgi-bin/qgis_mapserv.fcgi.exe?SERVICE=WMTS&REQUEST=GetCapabilities

- Pruebe para cada caso el despliegue de capa compruebe el despliegue del mapa como servicio **WMTS**, las opciones de despliegue del **mapa base**.
- Realizar el despliegue de las capas **WFS** y comprobar las opciones de edición en la capa de **sismos**.
- Realizar el despliegue de las capas **WCS** y las opciones de descarga de capa dato en formatos conocidos.
- Realice el despliegue de las capas **WMS** de “**Sismo**”, “**Municipio**” y “**Unidades cronoestratigráfica**” y despliegue la identificación de atributos con la herramienta respectiva en **QGIS** de identificar objetos espaciales.

4. Entregable del ejercicio

Llevar cabo la configuración del proyecto y de las capacidades de los **geoservicios OGC** ofrecidos por **QGIS**. Para comprobar la entrega realizar:

- Guardar en el proyecto en una carpeta de nombre **XML**, los **GetCapabilities** de los servicios **WMS, WMTS, WFS y WCS**.
- Entregar el archivo comprimido en formato **ZIP** de la carpeta del proyecto de **QGIS** de nombre `Caldas.zip`. Debe contener el proyecto en **QGZ**, con las salidas solicitadas adicionales.

* Nota 1

La entrega corresponde a la carpeta comprimida con los resultados anteriores. El archivo comprimido **no debe superar** más de los 100 MB en disco.

Créditos y Atribuciones

Software libre y de código abierto de QGIS

QGIS es un Sistema de Información Geográfica de software libre y de código abierto. El proyecto está licenciado bajo la **GNU General Public License (GPL v2.0)**.

La comunidad global de **QGIS** invita a participar en el desarrollo continuo del proyecto a través de contribuciones financieras, desarrollo de código y actividades de promoción. Para más información visite qgis.org.

QGIS® es una marca registrada de **QGIS.org**, protegida bajo la legislación europea. Tanto el nombre **QGIS** como su logotipo están protegidos contra usos no autorizados según las [directrices oficiales de la marca registrada](#).

Datos abiertos

Los datos utilizados en este tutorial son propiedad de sus respectivas instituciones y están disponibles a través de diferentes plataformas de datos abiertos. Cada conjunto de datos está sujeto a sus propios términos de uso y licencias de datos abiertos según lo establecido por la institución proveedora. Se recomienda consultar las condiciones de uso específicas en la fuente original de los datos.

Sobre la guía de ejercicios en QGIS

Esta guía de ejercicios de **QGIS** es un recurso técnico desarrollado por el **SGC** para los procesos de entrenamiento institucional. El desarrollo y mantenimiento de este documento está bajo la coordinación del Ingeniero Jaime Alberto Garzón, profesional del Servicio Geológico Colombiano (**SGC**). Para consultas técnicas, sugerencias de mejora o contribuciones al contenido, puede comunicarse a través del correo institucional jgarzon@sgc.gov.co.

Este documento se distribuye bajo una licencia **Creative Commons Atribución-CompartirIgual 4.0 Internacional (CC BY-SA 4.0)**. Esto significa que usted es libre de compartir y adaptar el material, siempre que otorgue el crédito apropiado, proporcione un enlace a la licencia, e indique si se realizaron cambios. Si modifica o desarrolla sobre este material, debe distribuir sus contribuciones bajo la misma licencia que el original.