# Manual de usuario Versión 0.1

## CIAF - Subdirección de Catastro

## DinoSoil Toolbox – Mapeo Digital Suelos

IGAC
INSTITUTO GEOGRÁFICO
AGUSTÍN CODAZZI

Centro de Investigación y Desarrollo - CIAF

# Manual de usuario Versión 0.1

## CIAF - Subdirección de Catastro

## DinoSoil Toolbox – Mapeo Digital Suelos

IGAC
INSTITUTO GEOGRÁFICO
AGUSTÍN CODAZZI

**Proyecto de Innovación**

# DinoSoil Toolbox – Mapeo Digital Suelos

Manual de usuario Versión 0.1


**Equipo técnico CIAF:**

Alejandro Coca Castro (Líder técnico)

Victoria Camacho

Patricia Escudero

Andrés Felipe López

Pedro Karin Serrato

Yesenia Vargas

**Supervisores CIAF:**

Carlos Franco y Diana Galindo

**Equipo técnico subdirección Agrología:**

Ricardo Devia

Juan Camilo García

Joan Sebastián Gutiérrez

Daniela Esmeralda Prieto

**Supervisores Agrología:**

Janeth González y Napoleón Ordoñez



Bogotá D.C. diciembre 2020

# CONTROL DE CAMBIOS

| | |
|---|---|
| **Proyecto:** | Propuesta de proyecto de innovación<br>Mapeo Digital de Suelos -MDS |
| **Identificador:** | Documento Preliminar |
| **Nombre del Documento:** | Herramienta DinoSoil Toolbox |
| **Tipo de documento:** | Manual de Usuario |
| **Fase/Etapa:** | Desarrollo |
| **Responsable:** | Oficina CIAF, Instituto Geográfico Agustín Codazzi – IGAC. |
| **Revisado por:** | Carlos Franco |

| Versión | Fecha | Descripción del cambio | Elaborado Por | Aprobó |
|---|---|---|---|---|
| 1 | 2020/12/07 | Elaboración del manual | Alejandro Coca Castro<br>Victoria Camacho<br>Patricia Escudero<br>Andrés Felipe López<br>Pedro Karin Serrato<br>Yesenia Vargas | |
| 2 | 2020/12/28 | Ajustes con la versión 0.1 publica en GitHub | Alejandro Coca Castro | |

# Abreviaturas y Acrónimos

CSV - Comma Separate Values

DEM - Modelo Digital de Elevación

GEE - Google Earth Engine

MDS - Mapeo Digital de Suelos

MRRTF - Índice de planitud superior de cresta multi-resolución

MRVBF - Índice de resolución múltiple de planitud del fondo del valle

NDVI - Normalized Difference Vegetation Index

# Tabla de Contenidos


**1 Introducción ..... 9**

**1.1 Software R ..... 9**

**1.2 Problema de investigación y contribuciones ..... 10**

**2 Objetivos ..... 11**

**2.1 General ..... 11**

**2.2 Específicos ..... 11**

**3 Pre-requisitos de software ..... 11**

**3.1 R y RStudio ..... 11**

3.1.1 Instalación ..... 11

3.1.2 Introducción a R y Rstudio ..... 13

3.1.3 Sesiones en R ..... 14

**3.2 Google Earth Engine ..... 14**

**4 Configuración del proyecto ..... 16**

**4.1 Directorio config ..... 16**

4.1.1 Información del proyecto ..... 18

4.1.2 Rutas ..... 18

4.1.3 Variables objetivo ..... 18

4.1.4 Covariables ..... 19

4.1.5 Aprendizaje de Maquinas ..... 20

**4.2 Directorio datos (entrada) ..... 20**

4.2.1 Base de datos ..... 20

4.2.2 Covariables ..... 21

4.2.3 Limite ..... 23

**4.3 Generación de los derivados del DEM en el software SAGA GIS ..... 23**

4.3.1 Basic Terrain Analysis ..... 24

4.3.2 Multiresolution Index of Valley Bottom Flatness (MRVBF/ MRRTF) ..... 27

4.3.3 Topographic Openness ..... 28

4.3.4 Terrain Ruggedness Index ..... 29
4.3.5 Topographic Position Index ..... 29
4.3.6 Sky View Factor ..... 30
**4.4 Control de calidad de las capas ..... 32**
**5 Panel de la herramienta ..... 33**
**5.1 Carga de las rutas de los scripts y configuraciones del proyecto ..... 34**
**5.2 Preprocesamiento de la base de datos ..... 35**
**5.3 Generación de la matriz de datos ..... 39**
**5.4 Selección de las covariables ..... 40**
**5.5 Exploración y análisis estadístico ..... 42**
**5.6 Ejecución de los algoritmos configurados ..... 44**
**5.7 Selección del mejor modelo ..... 46**
**5.8 Error e incertidumbre ..... 47**
**5.9 Predicción ..... 49**
**6 Mantenimiento de la herramienta ..... 49**
**Referencias ..... 51**
**Anexos ..... 52**

# Lista de Figuras

**Figura 3-1** Ambiente de trabajo de Rstudio. ..... 13
**Figura 4-1** Estructura propuesta de un proyecto ejemplo (proyecto_cesarmagdalena) para el funcionamiento de la herramienta dinoSOIL-toolbox ..... 16
**Figura 4-2** Contenido del archivo conf.txt. Cada componente se debe configurar como se indica en la siguiente página ..... 17
**Figura 4-3** Ejemplo de los archivos EXCEL que pueden disponerse dentro de la carpeta `originales` subdirectorio `0_basededatos` ..... 20

**Figura 4-4** Ejemplo de los archivos GeoTIFF que pueden disponerse dentro de los subdirectorios `dem` y `dem_derivados` ..... 21
**Figura 4-5** Ejemplo de múltiples archivos Shapefile ESRI que pueden disponerse dentro de la carpeta `clima` ..... 22
**Figura 4-6** Ventana inicial del Software SAGA GIS ..... 23
**Figura 4-7** Visualización del DEM en el SAGA GIS ..... 24
**Figura 4-8** Herramienta Basic Terrain Analysis ..... 26
**Figura 4-9** Visualización de los productos de la herramienta Basic Terrain Analysis ..... 26
**Figura 4-10** Herramienta Multiresolution Index of Valley Bottom Flatness MRVBF. ..... 27
**Figura 4-11** Herramienta Topographic Openness ..... 28
**Figura 4-12** Herramienta Ruggedness Index ..... 29
**Figura 4-13** Herramienta Topographic Position Index ..... 30
**Figura 4-14** Herramienta Sky View Factor ..... 31
**Figura 4-15** Visualización de las ventanas para exportar los archivos GeoTIFF derivados del DEM ..... 32
**Figura 5-1** Ejemplo del archivo principal de la herramienta, *dinoSOIL-toolbox.R*, abierto en el ambiente de RStudio ..... 33
**Figura 5-2** Ejemplo de la adición de las variables entre las líneas 21 a 31 del archivo principal de la herramienta, *dinoSOIL-toolbox.R*, abierto en el ambiente de RStudio ..... 35
**Figura 5-3** Ejemplo de la consulta de argumentos en el modulo de Preprocesamiento y almacenamiento de estos en el espacio de trabajo ..... 36
**Figura 5-4** Ejemplo de los archivos de salida del modulo de Preprocesamiento ..... 37
**Figura 5-5** Ejemplo de conversión de texto del archivo CSV a columnas en MS Excel. 37
**Figura 5-6** Ejemplo del archivo CSV de la base de datos vertilcalizada convertido en columnas en MS Excel ..... 38
**Figura 5-7** Ejemplo del modelo entrenado y calibrado de ranger (Random Forest) de la variable Gran Grupo del proyecto ejemplo de Cesar/Magdalena ..... 46

# Lista de Tablas

**Tabla 1-1** Recursos disponibles sobre uso del software R en análisis espaciales y MDS. ..... 10
**Tabla 3-1** Vista general del ambiente de trabajo de RStudio ..... 14
**Tabla 4-1** Derivados de la herramienta Basic Terrain Analysis ..... 25
**Tabla 4-2** Derivados de la herramienta Multiresolution Index of Valley Bottom Flatness MRVBF ..... 27
**Tabla 4-3** Derivados de la herramienta Topographic Openness ..... 28
**Tabla 4-4** Derivados de la herramienta Terrain Ruggedness Index ..... 29
**Tabla 4-5** Derivados de la herramienta Topographic Position Index ..... 30
**Tabla 4-6** Derivados de la herramienta Sky View Factor ..... 31
**Tabla 5-1** Descripción por argumento y ejemplo del componente de generación de los datos ..... 39
**Tabla 5-2** Descripción de los archivos de salida del componente de generación de los datos ..... 40
**Tabla 5-3** Descripción por argumento y ejemplo del componente de generación de los datos ..... 41
**Tabla 5-4** Descripción de los archivos de salida del componente de la selección de covariables ..... 41
**Tabla 5-5** Comparación de las salidas gráficas de los algoritmos RFE con datos originales y balanceados de la variable Gran Grupo del proyecto Cesar/Magdalena ..... 42
**Tabla 5-6** Descripción por argumento y ejemplo del componente de generación de los datos ..... 43
**Tabla 5-7** Descripción de los archivos de salida del componente exploratorio y estadístico ..... 44
**Tabla 5-8** Descripción por argumento y ejemplo del componente de generación de los datos ..... 45
**Tabla 5-9** Descripción de los archivos de salida del componente exploratorio y estadístico ..... 47

**Tabla 5-10** Descripción de los archivos de salida del componente exploratorio y estadístico ..... 48
**Tabla 5-11** Descripción de los archivos de salida del componente exploratorio y estadístico ..... 49

# 1 Introducción

Para la comprensión por parte del usuario externo e interno sobre el manejo de la herramienta dinoSOIL-toolbox versión 0.1, se implementó el Manual de Usuario que describe los lineamientos para utilizar de forma práctica las funcionalidades que dispone la herramienta, y así conocer, interactuar y explorar la adecuada operación de los componentes de la misma.

El manual describe cada una de las funcionalidades organizadas por etapas y sus resultados en el sistema luego de ejecutarlas. De esta manera, el usuario al finalizar la lectura, obtendrá los conocimientos básicos que potencializan cada una de las funcionalidades de la herramienta para su uso adecuado. Se espera el presente documento apoye la optimización de generación de información digital de suelos de la Subdirección de Agrología así como futuros proyectos con otras entidades. Las siguientes secciones resumen aspectos del software R, el cual fue el esqueleto en el cual la herramienta fue implementada, así como un repaso del problema de investigación y contribuciones.

## 1.1 Software R

La herramienta dinoSOIL-toolbox se encuentra implementada en su totalidad en el software R. Este manual no pretende profundizar sobre este software, no obstante, es importante mencionar algunas de sus características principales así como sugerir recursos para su consulta.

R, una plataforma de análisis estadístico con herramientas gráficas muy avanzadas, es un referente en el análisis estadístico desde hace décadas. Se puede obtener y distribuir R gratuitamente debido a que se encuentra bajo la Licencia Pública General (GPL por sus siglas en inglés) del proyecto colaborativo de software libre GNU. R está disponible para los sistemas operativos Linux, Windows y Mac. Existen varias formas de obtenerlo e instalarlo. RStudio es una interface gráfica muy útil para utilizar R y es la recomendada en este manual para ejecutar la herramienta dinoSOIL-toolbox. Los detalles de la instalación e introducción a R y RStudio se encuentran en la **Sección 3.1** siendo esta información mayoritariamente extraídos del libro de Mas (2018).

La **Tabla 1-1** presenta una lista de recursos en español e inglés que describen R tanto para análisis generales con enfoque espacial así como para el Mapeo Digital de Suelos (MDS).

**Tabla 1-1** Recursos disponibles sobre uso del software R en análisis espaciales y MDS.

| Tema | Autor(es) | Descripción de métodos y resultados |
|---|---|---|
| Análisis Espaciales | Mas (2018) | Este libro introduce el software R y aplicaciones para estudios con enfoque espacial. |
| MDS | Hengl & MacMillan (2019) | Este repositorio es del Libro: "Predictive Soil Mapping with R". Presenta varias implementaciones para MDS. |
| MDS | Malone *et al.* (2017) | Este repositorio es del Libro: "Using R for Digital Soil Mapping". Presenta varias implementaciones para MDS. |

## 1.2 Problema de investigación y contribuciones

El desarrollo de la herramienta dinoSOIL-toolbox surge como una necesidad de optimizar los procesos de MDS dentro de la Subdirección de Agrología del IGAC. Cabe indicar este desarrollo fue facilitado gracias a los adelantos previos de varios profesionales de la Subdirección quienes basados en el software R lograron contribuir en proyectos como el Mapa Nacional de Carbono Orgánico, Mapa de Salinidad, Estudio detallado en Sibundoy, entre otros. Basado en estas experiencias, surge la oportunidad de ofrecer un desarrollo que optimiza las tecnologías de la información para su potencial adopción en los procesos de generación y toma de información en campo sobre suelos en las áreas donde sea prioritarias estas actividades

Cabe mencionar que el nombre de esta herramienta surge de su implementación inicial en un área prioritaria de Política de Tierras en Cesar/Magdalena. La delimitación de esta área conserva una forma similar a la de un dinosaurio por lo que se decidió personalizar el nombre de la herramienta con el prefijo *dino* seguido de su principal objetivo como una caja de herramientas de estudios del suelo o en inglés SOIL-toolbox.

La principal contribución de la herramienta es ofrecer un proceso organizado y sistemático de manejo de datos (entrada y salidas) así como procesamiento eficiente que faciliten aplicar MDS en cualquier área de estudio.

# 2 Objetivos

## 2.1 General

Ofrecer al usuario una guía práctica y detallada para utilizar los componentes de la herramienta dinoSOIL-toolbox versión 0.1, que le permitan la interacción de forma adecuada.

## 2.2 Específicos

- Ilustrar de manera didáctica la ejecución de cada uno de los componentes y el conjunto de pasos para la obtención exitosa de resultados en las funcionalidades que componen la herramienta dinoSOIL-toolbox versión 0.1.
- Brindar a la Subdirección de Agrología recursos ejemplo para la exploración y prueba de la herramienta.

# 3 Pre-requisitos de software

## 3.1 R y RStudio

### 3.1.1 Instalación

Como se indicó la **Sección 1.1**, R puede instalarse en varios sistemas operativos como se describe a continuación (Mas, 2018):

#### *Windows*

Para obtener R para Windows entre en la página del Comprehensive R Archive Network (CRAN) https://cran.r-project.org/mirrors.html. Escoja el espejo de su preferencia (CRAN mirrors). Clique en *Download R for Windows* e *Install R for the first time*. Clique en Download R 4.0.2 for Windows (o la versión más actualizada disponible), salve el archivo de R para Windows y ejecutarlo.

El ejecutable para instalar la versión gratis de RStudio Desktop para Windows puede bajarse de la página web de RStudio (https://www.rstudio.com/products/rstudio/download/).

### *Linux*

R está incluido en los repositorios de la mayoría de las distribuciones de Linux. Por ejemplo, en Ubuntu, se instala fácilmente utilizando el centro de software. Estos repositorios no tienen siempre la última versión de R. Para instalar la última versión, se puede seguir los pasos a continuación: Eliminar las versiones anteriores (en caso de existir instalaciones previas):

```
sudo apt-get remove --purge r-base*
```

Actualizar el sistema:

```
sudo apt-get update && apt-get -y upgrade
```

Importar la clave pública:

```
gpg --keyserver keyserver.ubuntu.com --recv-key E084DAB9
gpg -a --export E084DAB9 | sudo apt-key add
```

Añadir el repositorio de R en el archivo `/etc/apt/sources.list` (la línea de comando abajo es para la versión Ubuntu Xenial (Ubuntu 16.04 VPS), adaptar a su propia versión de Linux):

```
sudo echo "deb http://cran.rstudio.com/bin/linux/ubuntu xenial/" | sudo
tee -a /etc/apt/sources.list
```

Instalar la última versión de R:

```
sudo apt-get update
sudo apt-get install r-base r-base-dev
```

En https://www.rstudio.com/products/rstudio/download/, se encuentran los archivos de instalación de RStudio Desktop para diferentes distribuciones de Linux. La instalación puede llevarse a cabo utilizando programas como Synaptic, el centro de software o en la terminal.

### *Mac*

Los usuarios de Mac encontrarán los archivos de instalación de R en la página https://cran.r-project.org/bin/macosx/ y los de RStudio Desktop en http://www.rstudio.com/products/rstudio/download/.

## 3.1.2 Introducción a R y Rstudio

R se ejecuta a través de la escritura de comandos. Aunque resulta al principio complejo familiarizarse a este mecanismo de comandos, este resulta después de la practica flexible y permite la programación. Es importante tomar en cuenta algunos detalles: R es case-sensitive, es decir sensible a la diferencia entre minúsculas y mayúsculas ) y por lo tanto ”Nombre” es diferente de ”nombre”. El separador de decimales es el punto ”.”, la coma se usa para separar los elementos de una lista1. Se recomienda evitar el uso de acentos en las rutas y los nombres de archivos. RStudio es una interfaz gráfica de R que permite la programación y visualización de resultados. La **Figura 3-1** presenta la interfaz de RStudio con sus respectivos componentes principales descritos en la**Tabla 3-1**.

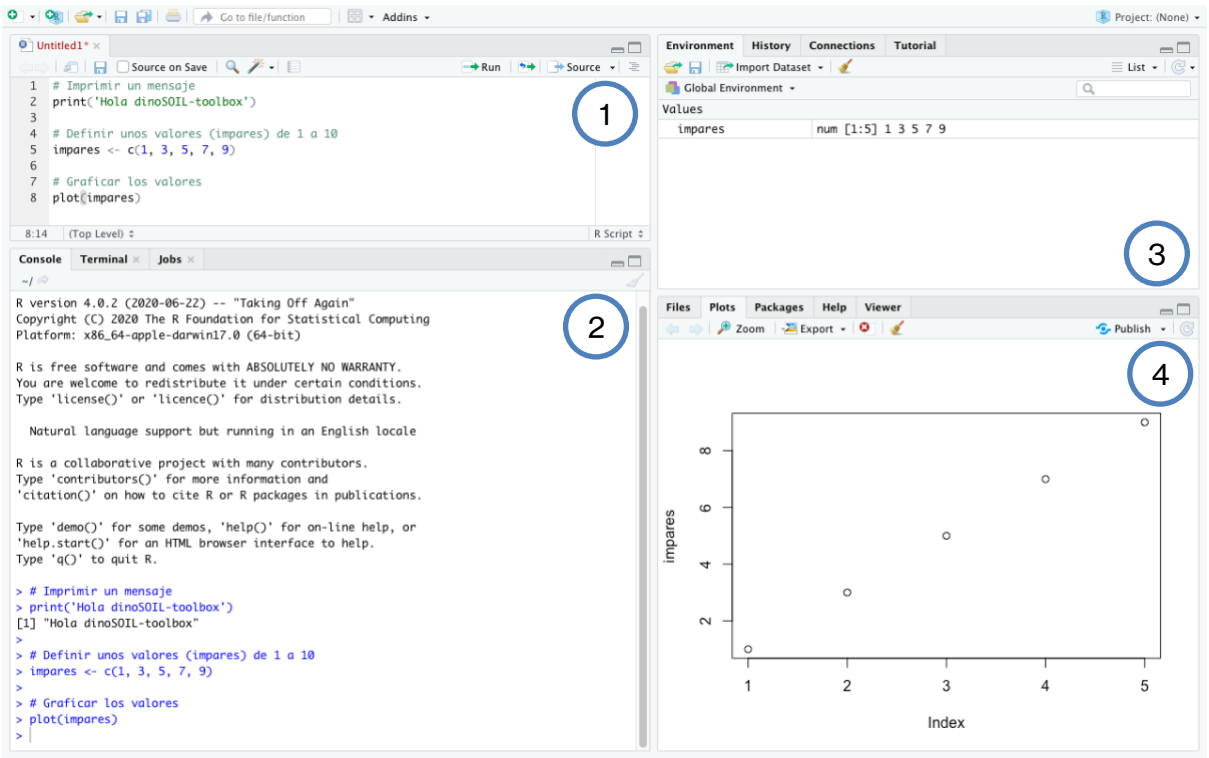



**Figura 3-1** Ambiente de trabajo de Rstudio. Fuente: Elaboración propia.

**Tabla 3-1** Vista general del ambiente de trabajo de RStudio.

| No. | Nombre y descripción |
|---|---|
| 1 | *Editor texto de los scripts.* Un script es un archivo de texto con una serie de instrucciones. Se puede ejecutar una sola línea del script, un conjunto (bloque) de líneas o bien el script entero. |
| 2 | *Consola*. Donde se digitan los comandos de línea. Ella interpreta cualquier entrada como un comando a ser ejecutado. Estos comandos y su sintaxis proporcionan una forma bastante natural e intuitiva de acceder a datos y realizar procesamientos y operaciones estadísticas. Sin embargo, es más fácil escribir su código como un guión (script) en la sección de editor de texto. |
| 3 | *Área de trabajo y el histórico.* En esta se observan las variables creadas y sus valores almacenadas dentro del entorno de R. |
| 4 | *Gráficos*. Se despliega cualquier tipo de gráfico generado en la consola o editor del texto. |

### 3.1.3 Sesiones en R

Cuando se trabaja con R todos los datos están generalmente en la memoria viva de la computadora. A pesar de la capacidad de las computadoras actuales, eso puede ser un problema cuando se manejan bases de datos muy grandes. La función `rm()` permite borrar un objeto para liberar memoria. Por ejemplo, `rm(mapa)` eliminará el objeto llamado mapa. `rm(list=ls())` borrará todos los objetos del espacio de trabajo.

Al momento de salir de una sesión de R, se le preguntará *Save workspace image?*, lo cual permite salvar los datos y utilizarlos en una futura sesión. En Windows, para que RStudio haga un despliegue correcto de los acentos (comentarios en español), es necesario volver a abrir el script con File > Reopen with encoding (UTF-8). En RStudio, seleccionar las líneas de comando que se desea ejecutar con el ratón, y dar un clic en el ícono de ”Run”. Para correr la script línea por línea, se debe poner el cursor en la línea deseada, y dar clic en ”Run” (o alternativamente, seleccionando las teclas control y R).

## 3.2 Google Earth Engine

El procesamiento en la nube es una tecnología que permite el análisis de grandes volúmenes de datos con una infraestructura (múltiples servidores alojados por un proveedor) de programas y servicios para su consumo y uso masivo en múltiples aplicaciones. Para el sector de la teledetección, la compañía Google tiene un producto

denominado Earth Engine (GEE, por sus siglas en inglés). Este producto contiene un catálogo con petabytes de imágenes satélite y conjunto de datos geoespaciales con capacidades de análisis a escala planetaria que están a disposición de científicos, investigadores y desarrolladores. Una mayor información de la bondades y aplicaciones de esta plataforma se encuentra en Gorelick *et al.* (2017).

El catálogo de imágenes corresponde a una importante colecta de conjuntos de datos de diferentes sensores entre ellos Landsat, Sentinel, MODIS e imágenes de alta resolución espacial como Planet SkySat y NAIP (Programa Nacional de Imágenes para Agricultura). Estas bases de datos se actualizan a medida que toman nuevas imágenes (cerca de 6000 nuevas escenas diarias), creando así un enorme catálogo de datos geoespaciales (Gabriel Alejandro Perilla, 2020). Los repositorios de imágenes se pueden consultar a través de varios criterios como calidad, localización y fechas.

En cuanto al ambiente para el desarrollo, se dispone de un editor de código que opera en línea, donde se juntan todos los elementos. Acá, a través de Scripts se hace el llamado a los datos, sean estos imágenes o vectores, se integran las API (Application Program Interface) de desarrollo para el uso de algoritmos de investigación a partir de los datos, se crean aplicaciones a partir de funciones de interfaz de usuario y se generan los procesos de trabajo en la nube. Estas API están generadas en lenguaje JavaScript y Python.

Para efectos de la herramienta dinoSoil-toolbox, GEE puede ser usado para adicionar covariables ambientales derivadas de datos satelitales. Estas se pueden descargar mediante la plataforma de code editor en JavaScript o el API de Python. Alternativamente, para la versión 0.1 de la herramienta también se puede generar un archivo NDVI de cierto periodo a partir de la colección de imágenes Landsat 8, específicamente el producto de reflectancia de la superficie (LANDSAT/LC08/C01/T1_SR). La librería rgee es la que permite la integración de GEE con la herramienta dinoSoil-toolbox. Una mayor información de esta herramienta puede encontrarse en Aybar *et al.* (2020).

# 4 Configuración del proyecto

El proyecto se debe alojar bajo una carpeta con un nombre y ubicación de preferencia del usuario (preferiblemente con un espacio libre mayor de 10 GB). De la carpeta del proyecto, se ramifican dos directorios principales `config` y `datos`. En `config` se aloja un archivo de texto con las configuraciones principales del proyecto. En `datos` se disponen los datos de entrada como la base de datos (observaciones, perfiles y coordenadas) en formato Excel y covariables en formato raster y/o vector. La **Figura 4-1** presenta el esquema del proyecto ejemplo en Cesar/Magdalena (ver rutas para el acceso al proyecto en el **Anexo 1**) con el cual se pretende dar guía de los componentes y estructura requerida de los archivos de entrada para el uso de la herramienta. Como se va a percibir con el avance del manual, la estructura propuesta permite un manejo adecuado de los datos e información del proyecto por carpetas. El correcto funcionamiento de la herramienta depende de que se mantenga una estructura inicial como la planteada en el proyecto ejemplo.

```
proyecto_cesarmagdalena
├── config
└── datos
    └── entrada
        ├── 0_basededatos
        │   └── originales
        ├── 1_covariables
        │   ├── raster
        │   │   ├── dem
        │   │   └── dem_derivados
        │   └── vector
        │       ├── clima
        │       └── relieve
        └── 2_limite
```

**Figura 4-1** Estructura propuesta de un proyecto ejemplo (proyecto_cesarmagdalena) para el funcionamiento de la herramienta dinoSOIL-toolbox.

## 4.1 Directorio config

Este directorio aloja el archivo de configuración `config.txt` en el cual se definen las principales configuraciones del proyecto (ver **Figura 4-2**). Este archivo debe ser modificado previo al inicio del uso de la herramienta con la descripción, ruta, variable(s) objetivo y ajustes, lista de covariables y ajustes, y aspectos del modelado con aprendizaje de maquinas del proyecto. El archivo puede modificarse posteriormente si

se desea agregar nuevas variables objetivo sin necesidad de crearse otro proyecto, a excepción de que se tenga una base de datos diferente.

```
##################################################
Informacion del proyecto
##################################################
Proyecto: PT Cesar-Magdalena
Por: Equipo CIAF
Ciudad, Pais: Bogota, Colombia
Fecha: Septiembre 2020
########################

########################
Rutas
########################
proyecto.dir = F:/IDI_MDS_Agrologia/proyecto/proyecto_cesarmagdalena

########################
Variables objetivo
########################
variables.categoricas = FAMILIA_TE;GRANGRUPO
variables.continuas = pH

########################
Ajustes Variables Objetivo
########################
categoricas.minobservaciones = 5
continuas.profundidades = 0-30;30-10

########################
Covariables
########################
covariables.vector = clima;relieve
covariables.raster = dem;dem_derivados;ndvi

########################
Ajuste Covariables
########################
vector.atributos = Denominaci;TIPO_RELIE
NDVI.fechas = 2019-01-01;2020-08-30

########################
Aprendizaje de Maquinas
########################
modelos.categoricas = ranger;mlp;xgbTree;glmnet;multinom
modelos.continuas = ranger;xgbTree;svmRadial;cubist;glmnet
metricas.categoricas = Accuracy;Kappa
metricas.continuas = RMSE;Rsquared
##################################################
```

**Figura 4-2** Contenido del archivo conf.txt. Cada componente se debe configurar como se indica en la siguiente página.

### 4.1.1 Información del proyecto

Se debe incluir detalles pertinentes sobre el proyecto, autor(es), ciudad y país y fecha de creación. En el ejemplo se incluyen los detalles usados para el proyecto demo de cesar/magdalena.

### 4.1.2 Rutas

Se debe incluir la ruta completa al directorio del proyecto. Esta ruta debe ir con los separadores de barra / y no en barra oblicua invertida \. En el caso del sistema operativo Windows, es importante verificar que la ruta comienza con la unidad de disco donde se aloja el proyecto. En el ejemplo seria la unidad `F:`, seguido de la ruta completa.

### 4.1.3 Variables objetivo

Se debe declarar las variables objetivo de acuerdo a su grupo, categóricas o continuas. Los nombres de las variables deben estar separadas por punto y coma (;) y coincidir con el nombre de los archivos de las bases de datos de perfiles y/o observaciones. Adicionalmente, en la sección de `Ajustes Variables Objetivo` se puede definir aspectos del modelado según el tipo de variable. De esta manera, para variables categóricas se puede definir en `categoricas.minobservaciones` el número mínimo de observaciones que debe tener cada categoría o grupo para poder ser considerado. Para las variables continuas, como pH, se puede definir en `continuas.profundidades` hasta dos profundidades para interpolar, y que posteriormente serán usadas para el respectivo modelado. Es importante tener en cuenta que los valores de estas dos profundidades pueden ser modificado según el requerimiento del proyecto, y se aplica para todo el listado de variables continuas.

En el ejemplo de la **Figura 4-2**, se configuraron como variables categóricas las variables FAMILIA_TE y GRANGRUPO que corresponden al nombre dado en las bases de datos para Familia Textural y Gran Grupo, respectivamente. En el caso de las variables continuas, únicamente se tiene pH, y este nombre corresponde al mismo que se tiene en las bases de datos de entrada.

### 4.1.4 Covariables

Se debe declarar las covariables según su formato de origen (raster o vector) para usarse en el proceso de modelado. El nombre de estas debe coincidir con los nombres de las carpetas alojadas en el subdirectorio `1_covariables`. En `Ajuste Covariables` se puede configurar propiedades de las capas con formato vectorial (para su conversión a formato raster) así como de la temporalidad de los insumos satelitales. En el ejemplo de la **Figura 4-2**, se configuran las covariables `clima` y `relieve` así como `dem`, `dem_derivados` y `ndvi` de acuerdo a su formato vector y raster, respectivamente. Para el caso de las capas vectoriales `clima` y `relieve` se debe declarar el atributo con el cual se va rasterizar, en este caso `Denominaci` y `TIPO_RELIE`, respectivamente.

Sobre las covariables en formato raster, es importante tener en cuenta que el archivo `dem` (en la ruta datos > entrada > 1_covariables > raster > dem) debe estar bajo el nombre dem.tif. En caso de tenerse otro nombre la herramienta arrojará un error indicando verificar el nombre del archivo. Adicionalmente, si no se tiene dem no se continua el proceso ya que este define la resolución espacial de trabajo, y por ende de las otras covariables. En caso de incluirse archivos derivados del dem (en la ruta datos > entrada > 1_covariables > raster > dem_derivados) estos deben ser declarados en el config.txt como `dem_derivados`. En lo referente al `ndvi` este se puede generar como parte del proceso de la herramienta usando la librería de Google Earth Engine (se requiere autenticación con usuario de esta plataforma y cuenta Gmail) o añadirse de manera externa. En caso de añadirse de manera externa, se debe declarar el nombre de archivo como `ndvi.tif`. En caso de usarse la funcionalidad de generar el NDVI mediante la herramienta, es importante que la fecha inicial y final del NDVI se declare en el campo `NDVI.fechas`. En lo posible la fecha debe coincidir con la fecha de colección de los datos de suelos en campo que esta consignada en la pestaña SITIO de la base de datos o es de conocimiento del analista. La herramienta también permite agregar otras capas en formato raster que se hayan procesado, sin necesidad de hacer la conversión de vector a raster. Si el raster representa una variable tipo categórico como niveles de clima, la codificación de dichas clases con los valores de los pixeles deben declararse dentro de un archivo CSV (ver **Anexo 2**).

### 4.1.5 Aprendizaje de Maquinas

Se debe indicar los modelos por grupo (categórica o continua) en el modelado con aprendizaje de maquinas. Los nombres de los modelos deben estar separados por punto y coma (;). El nombre de estos modelos debe coincidir con los disponibles en la librería caret siendo algunos específicos según el tipo de variable a modelar. Adicionalmente, se debe indicar las métricas usadas para evaluar el desempeño de los modelos. Solamente serán válidas aquellas métricas aceptadas por caret.

Cuando se desconoce que modelos y/o métricas usar, se sugiere dejar los valores por defecto del archivo plantilla de `config.txt`. Como se explicará posteriormente existe la alternativa de usar los modelos que por defecto ofrece la librería caret . Estos modelos se caracterizan por que permiten graficar con facilidad la importancia de las covariables. A diferencia, si se declara manualmente un modelo, es importante verificar que caret reconoce como graficar la importancia de las covariables (ver el **Anexo 3**).

## 4.2 Directorio datos (entrada)

### 4.2.1 Base de datos

En la carpeta denominada `originales` del subdirectorio `0_basededatos` debe ubicarse los archivos de la base de datos de observaciones, perfiles y coordenadas en formato XLSX. La **Figura 4-3** presenta un ejemplo de archivos EXCEL de la base de datos usadas para el proyecto de plantilla de Cesar/Magdalena. Estas bases de datos deben contar en lo posible con una inspección de calidad y estar en arreglo horizontal. Este tipo de arreglo es luego transformado en vertical (o proceso conocido verticalización) para su posterior uso en la herramienta.

```
proyecto_cesarmagdalena
└── datos
    └── entrada
        ├── 0_basededatos
        │   └── originales
        │       ├── BD_CATEGORICAS_COOORDENADAS.xlsx
        │       ├── BD_OBSERVACIONES_MODELAMIENTO_PT_2020.xlsx
        │       └── BD_PERFILES_MODELAMIENTO_PT_2020.xlsx
```

**Figura 4-3** Ejemplo de los archivos EXCEL que pueden disponerse dentro de la carpeta `originales` subdirectorio `0_basededatos`.

### 4.2.2 Covariables

Debe alojarse los archivos de las covariables y ubicarlos de acuerdo a su formato de origen en la carpeta de `raster` o `vector`.

#### *raster*

Esta carpeta debe alojar aquellas covariables que se encuentran en formato raster GeoTIFF. Si se encuentra en otro formato diferente a GeoTIFF se debe hacer su conversión a este formato para el correcto funcionamiento de la herramienta.

Como punto de partida, se sugiere adicionar el modelo digital de elevación (DEM por sus siglas en ingles) en la carpeta `dem`. Este es usado como referencia para establecer la resolución espacial de modelado y predicción de las variables objetivo. Adicionalmente, si se van a emplear covariables derivadas del DEM, ya sea generadas por el software SAGA (ver Sección 4.3) y/u otro software de preferencia, estas deben ser alojadas dentro de la carpeta dem_derivados. La **Figura 4-4** muestra un ejemplo de los archivos ambas subcarpetas.

```
proyecto
└── datos
    └── entrada
        └── 1_covariables
            ├── raster
            │   ├── original
            │   │   └── dem.tif
            │   ├── dem_derivados
            │   │   ├── AnalyticalHillshading.tif
            │   │   ├── Aspect.tif
            │   │   ├── ChanneNetworkBaseLevel.tif
            │   │   ├── ConvergenceIndex.tif
            │   │   ├── CrossSectionalCurvature.tif
            │   │   ├── FlowAccumulation.tif
            │   │   ├── MRRTF.tif
            │   │   ├── NegativeOpenness.tif
            │   │   ├── SkyViewFactor.tif
            │   │   ├── Slope.tif
            │   │   ├── TerrainRuggednessIndex.tif
            │   │   ├── TopographicPositionIndex.tif
            │   │   ├── TopographicWetnessIndex.tif
            │   │   ├── ValleyDepth.tif
            │   │   ├── VerticalDistanceChannelNetwork.tif
            │   │   └── VisibleSky.tif
```

**Figura 4-4** Ejemplo de los archivos GeoTIFF que pueden disponerse dentro de los subdirectorios `dem` y `dem_derivados`.

Adicional al dem, se pueden adicionar otras variables en formato raster. Tal es el caso del ndvi, que se puede derivar de observaciones de satélite. Si se cuenta con este tipo de datos se puede adicionar dentro de la carpeta `ndvi`. Caso contrario, este archivo puede generarse mediante internamente en la herramienta, no obstante se requiere configurar una cuenta de Google Earth Engine como es detallado en la **Sección 3.2**.

### *vector*

Esta carpeta debe alojar aquellas covariables que se encuentran en formato vectorial Shapefile ESRI. Si se encuentra en otro formato diferente a Shapefile ESRI se debe hacer su conversión a este formato para el correcto funcionamiento de la herramienta.

Las capas vectoriales de `clima` y `relieve` son un ejemplo de covariables que pueden ir en esta carpeta. Es importante mencionar que la herramienta es capaz de reconocer varias capas Shapefile ESRI por covariable que cubran el área de estudio. Se recomienda guardar cada una de estas con nombre diferente. La **Figura 4-5** presenta un ejemplo de la covariable `clima` la cual contiene múltiples archivos Shapefile ESRI los cuales cubren el área prioritaria de política de tierras del proyecto ejemplo en los departamentos de Cesar y Magdalena.

```
proyecto
└── datos
    └── entrada
        └── 1_covariables
            └── vector
                ├── clima
                │   ├── CLIMA_CESAR_20170830VF.dbf
                │   ├── CLIMA_CESAR_20170830VF.prj
                │   ├── CLIMA_CESAR_20170830VF.sbn
                │   ├── CLIMA_CESAR_20170830VF.sbx
                │   ├── CLIMA_CESAR_20170830VF.shp
                │   ├── CLIMA_CESAR_20170830VF.shp.xml
                │   ├── CLIMA_CESAR_20170830VF.shx
                │   ├── CLIMA_MAGDALENA_20200124VF.dbf
                │   ├── CLIMA_MAGDALENA_20200124VF.prj
                │   ├── CLIMA_MAGDALENA_20200124VF.sbn
                │   ├── CLIMA_MAGDALENA_20200124VF.sbx
                │   ├── CLIMA_MAGDALENA_20200124VF.shp
                │   ├── CLIMA_MAGDALENA_20200124VF.shp.xml
                │   └── CLIMA_MAGDALENA_20200124VF.shx
```

**Figura 4-5** Ejemplo de múltiples archivos Shapefile ESRI que pueden disponerse dentro de la carpeta `clima`.

### 4.2.3 Limite

Debe alojarse el límite del área de estudio a en formato vectorial Shapefile ESRI. Si se encuentra en otro formato diferente a Shapefile ESRI se debe hacer su conversión a este formato para el correcto funcionamiento de la herramienta. No existe limitación respecto al nombre del archivo, no obstante solamente un archivo Shapefile ESRI es permitido en esta carpeta. En caso de haber más de un archivo, se generará error.

## 4.3 Generación de los derivados del DEM en el software SAGA GIS

Una alternativa para generar los productos derivados del DEM es el software de acceso libre SAGA GIS. Una vez instalado el programa en el computador es posible acceder a sus funciones a través del menú Geoprocessing.

Los productos derivados se generan con el módulo Terrain Analysis. Muchas de las herramientas allí dispuestas utilizan el DEM como entrada y permiten almacenar las salidas que se requieran. Inicialmente, para cargar el DEM, se utiliza la opción Open del menú File. A continuación, se ubica la ruta de almacenamiendo y cargamos el raster de elevación. Este aparecerá en la ventana “Tree” del panel de contenido del programa (ver **Figura 4-6**).

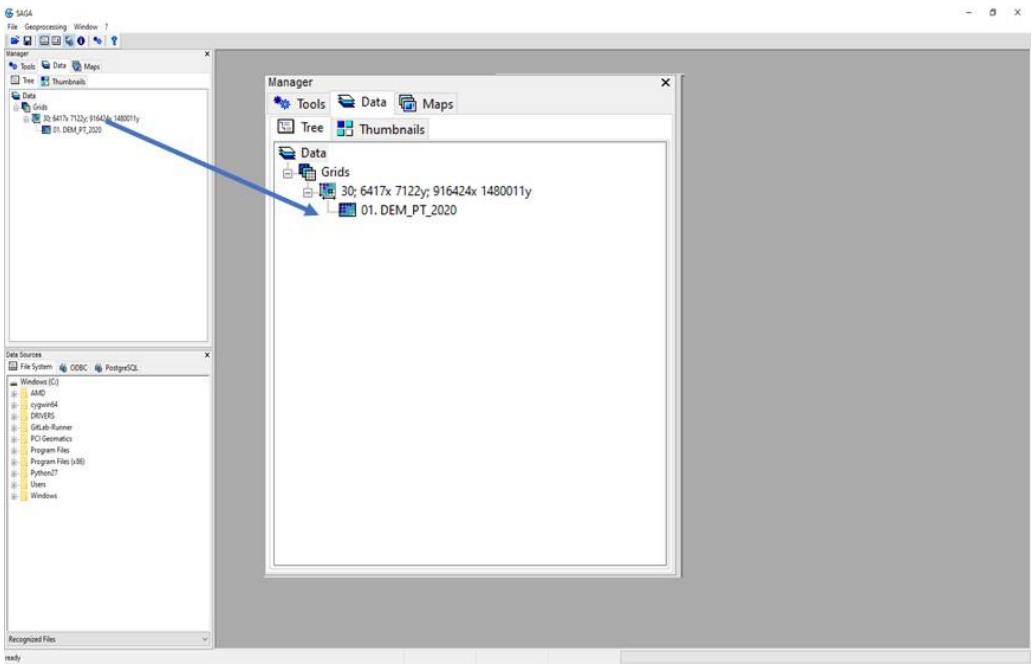



**Figura 4-6** Ventana inicial del Software SAGA GIS. Fuente: Elaboración propia.

Una alternativa para visualizarlo es pulsar clic derecho sobre el archivo y seleccionar la opción “Add to map”. De esta forma, se abrirá una nueva ventana con el archivo raster desplegado (ver **Figura 4-7**).

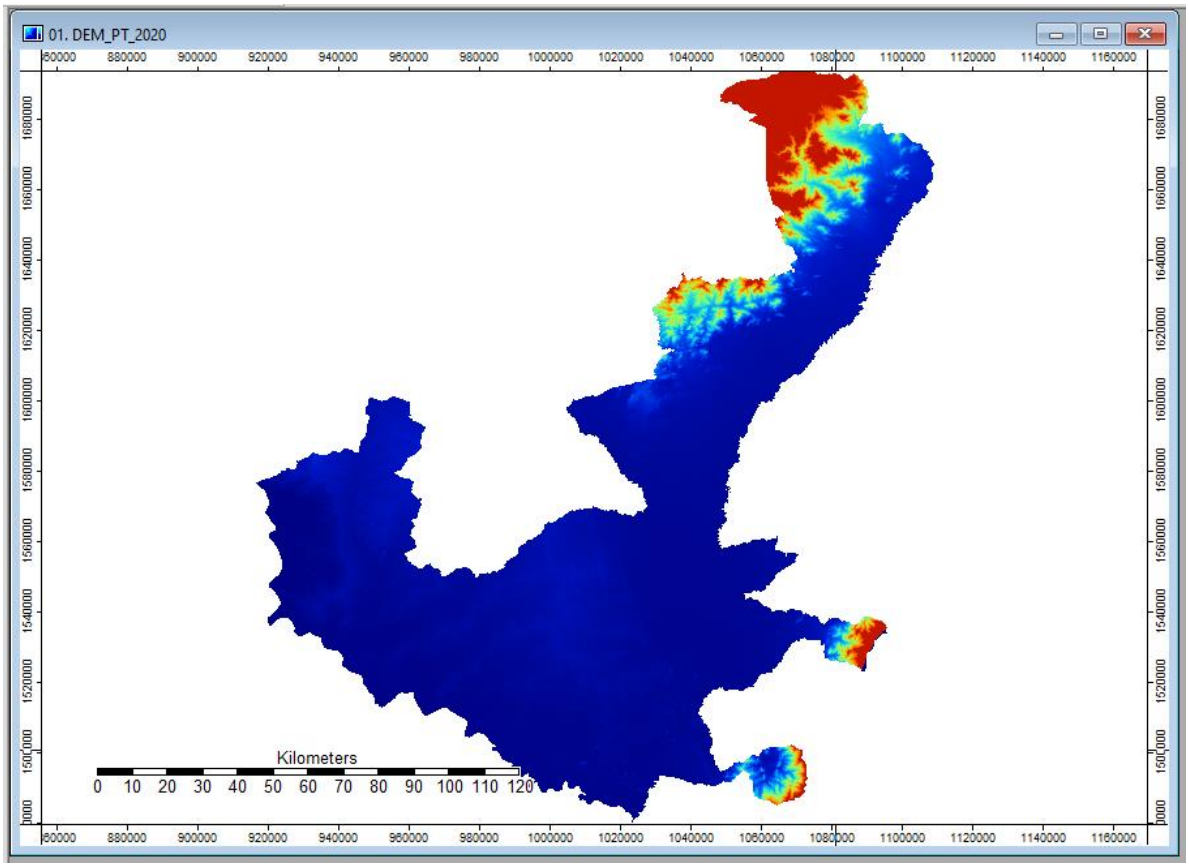



**Figura 4-7** Visualización del DEM en el SAGA GIS. Fuente: Elaboración propia.

### 4.3.1 Basic Terrain Analysis

En el menú **Geoprocessing >> Terrain Analysis** ubicamos la herramienta **Basic Terrain Analysis,** a través de la cual podemos calcular muchas de las covariables citadas. La **Tabla 4-1** indica las covariables que se pueden derivar con dicha herramienta. Un total de 14 covariables pueden ser derivadas, cada una de ellas describiendo alguna propiedad particular del paisaje.

**Tabla 4-1** Derivados de la herramienta Basic Terrain Analysis.

| Derivados del DEM | Función del SAGA | Descripción |
|---|---|---|
| Sombreado analítico (HILLSAHD) | Analytical Hillshading | Simula el efecto de la luz natural sobre la superficie terrestre, bajo algunos supuestos necesarios y simplificaciones que hacen que el resultado sea utilizable para fines cartográficos |
| Pendiente (DECL) | Slope | Expresa la inclinación de la superficie del terreno desde la horizontal o un nivel base local |
| Aspecto de la ladera (ASPECTO) | Aspect | Componente direccional del vector gradiente y es la dirección del máximo gradiente de la superficie en un punto dado |
| Curvatura longitudinal (LOGCURV) | Longitudinal Curvature | Describe la forma de las características del paisaje (convexidad / concavidad) |
| Curvatura de la sección transversal (CSC) | Cross Sectional Curvature | Describe la forma de las características del paisaje (convexidad / concavidad). Es la curvatura en una sección transversal de un valle o un cañón |
| Índice de convergencia (CONVINDEX) | Convergence Index | Calcula un índice de convergencia / divergencia con respecto al flujo terrestre |
| Depresiones cerradas | Closed Depressions | Depresiones y morfologías de tipo plano hundidas por encima del nivel general y que en la presencia de precipitaciones intensas o prolongadas se acumulan aguas de escorrentía |
| Área de captación total (CATAREA) | Total Catchment Area | Área de descarga con contribución de las celdas pendiente arriba |
| Índice de humedad topográfica (TWI) | Topographic Wetness Index | Produce una cuadrícula que muestra la acumulación de agua. Útil para el mapeo de suelos o inundaciones. |
| Factor LS | LS Factor | Combinación de la pendiente y la longitud de la pendiente como un atributo útil para predecir el potencial de erosión. |
| Nivel base de la red de canales (CNBL) | Channel Network Base Level | Nivel de base de una red fluvial es la elevación por debajo de la cual la corriente fluvial o el río principal (colector) no erosiona su cauce |
| Distancia de la red de drenajes (CND) | Vertical Distance Channel Network | Distancia vertical desde la red de drenaje a las alturas adyacentes |
| Profundidad del valle (VALLEYDEP) | Valley Depth | Distancia vertical al nivel base de una red de canales. |
| Posición relativa de pendiente (RSP) | Relative Slope Position | Representa la posición de la pendiente de la celda y su posición relativa entre el fondo de un valle y la cima de una cresta |

Tal y como se observa en la **Figura 4-8** el parámetro de entrada para esta herramienta son los datos de Elevación provistos por el DEM; por lo que, en la opción “Elevation” se

selecciona el DEM previamente cargado. En las otras opciones se deja por defecto la opción <create>, dado a que estas son las salidas que se buscan generar.

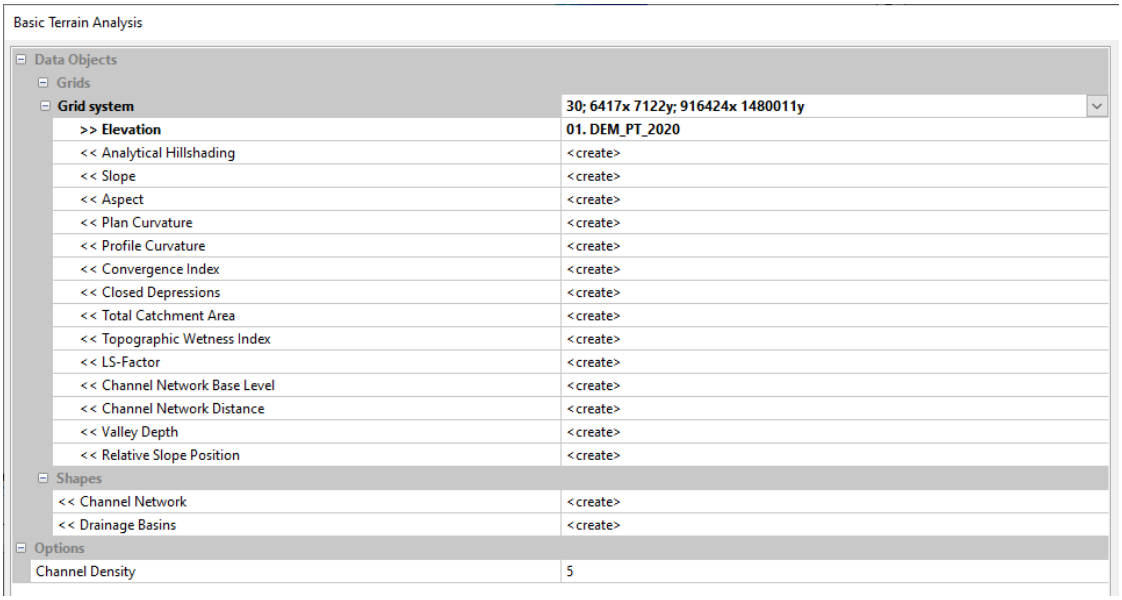



**Figura 4-8** Herramienta Basic Terrain Analysis. Fuente: Elaboración propia.

Finalmente, se pulsa el botón de “Okay” y el proceso entra en ejecución. En el extremo inferior derecho de la ventana observará una barra de progreso que da cuenta de que el software efectivamente está ejecutando la operación. Además, en el menú de contenido también se añadirán todos los productos generados (ver **Figura 4-9**).

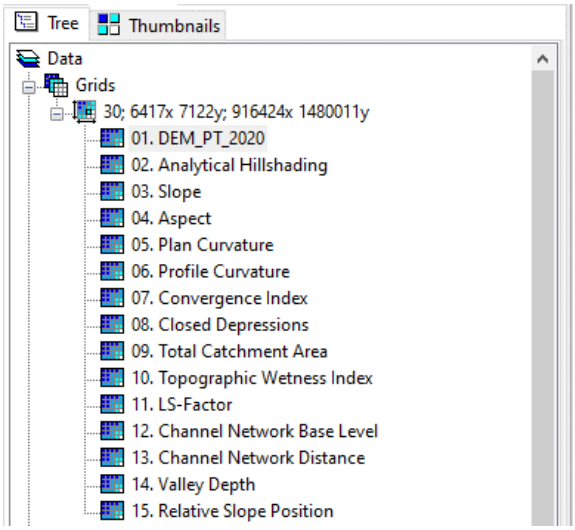



**Figura 4-9** Visualización de los productos de la herramienta Basic Terrain Analysis. Fuente: Elaboración propia.

Muchos de estos derivados pueden también generarse a través de herramientas individuales ubicadas principalmente en los submenús Morphometry, Lighting, Channels e Hydrology del menú Terrain Analysis. Sin embargo, la herramienta Basic Terrain Analysis permite el cálculo de estos 14 derivados automáticamente.

### 4.3.2 Multiresolution Index of Valley Bottom Flatness (MRVBF/ MRRTF)

En el menú **Geoprocessing >> Terrain Analysis >> Morphometry** ubicamos la herramienta **Multiresolution Index of Valley Bottom Flatness MRVBF**. La **Tabla 4-2** muestra los derivados que se pueden calcular a partir de esta herramienta.

**Tabla 4-2** Derivados de la herramienta Multiresolution Index of Valley Bottom Flatness MRVBF.

| Derivados del DEM | Función del SAGA | Descripción |
|---|---|---|
| Índice de planitud superior de cresta multi-resolución (MRRTF) | MRRTF | Índice topográfico diseñado para identificar áreas planas altas en un rango de escalas |
| Índice de resolución múltiple de planitud del fondo del valle (MRVBF) | MRVBF | Índice topográfico diseñado para identificar áreas de material depositado en fondos de valles planos |

Como archivo de entrada se require el modelo digital de elevación (Elevation) y otros parámetros adicionales dejando los valores por defecto (ver **Figura 4-10**). El resultante de esta operación se agrega automáticamente en la ventana “Tree” del SAGA GIS.

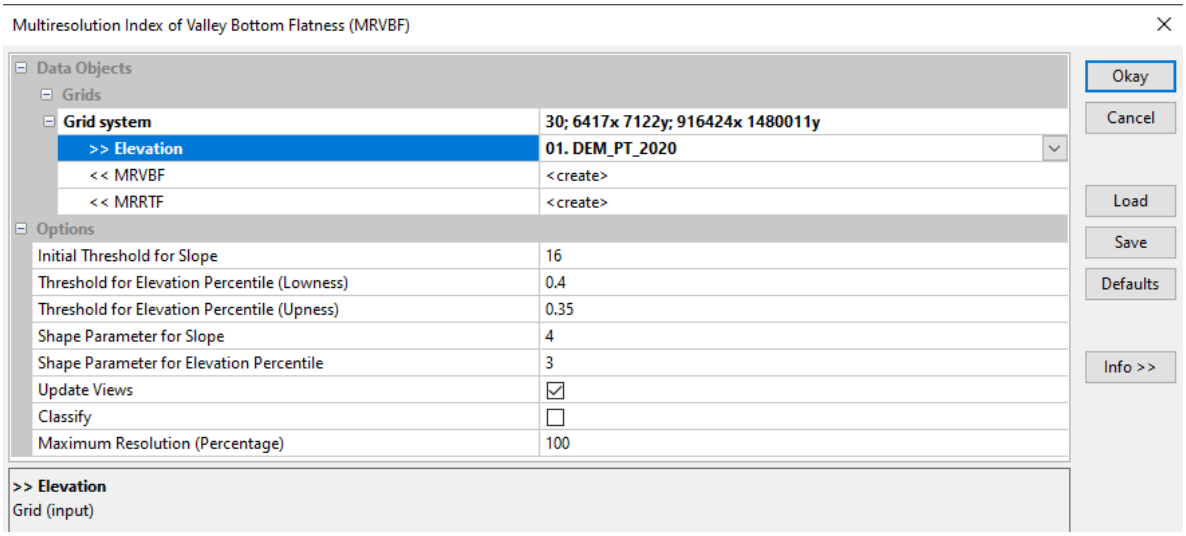



**Figura 4-10** Herramienta Multiresolution Index of Valley Bottom Flatness MRVBF. Fuente: Elaboración propia.

### 4.3.3 Topographic Openness

En el menú **Geoprocessing >> Terrain Analysis >> Lighting** ubicamos la herramienta **Topographic Openness**. La **Tabla 4-3** muestra los derivados que se pueden calcular a partir de esta herramienta.

**Tabla 4-3** Derivados de la herramienta Topographic Openness.

| Derivados del DEM | Función del SAGA | Descripción |
|---|---|---|
| Apertura negativa (NEGOP) | Negative Openness | La apertura topográfica expresa el dominio (positivo) o el cerramiento (negativo) de una ubicación de paisaje. Se ha relacionado con la amplitud de un paisaje desde cualquier posición |
| Apertura positiva (POSOP) | Positive Openness | La apertura topográfica expresa el dominio (positivo) o el cerramiento (negativo) de una ubicación de paisaje. Se ha relacionado con la amplitud de un paisaje desde cualquier posición |

Esta herramienta recibe como entrada el modelo digital de elevación (Elevation) y genera los raster de Positive Openness y Negative Openness. Adicionalmente, la ventana de diálogo solicita otros parámetros adicionales los cuales se dejan con los valores por defecto (ver **Figura 4-11**). El resultante de esta operación se agrega automáticamente en la ventana “Tree” del SAGA GIS.

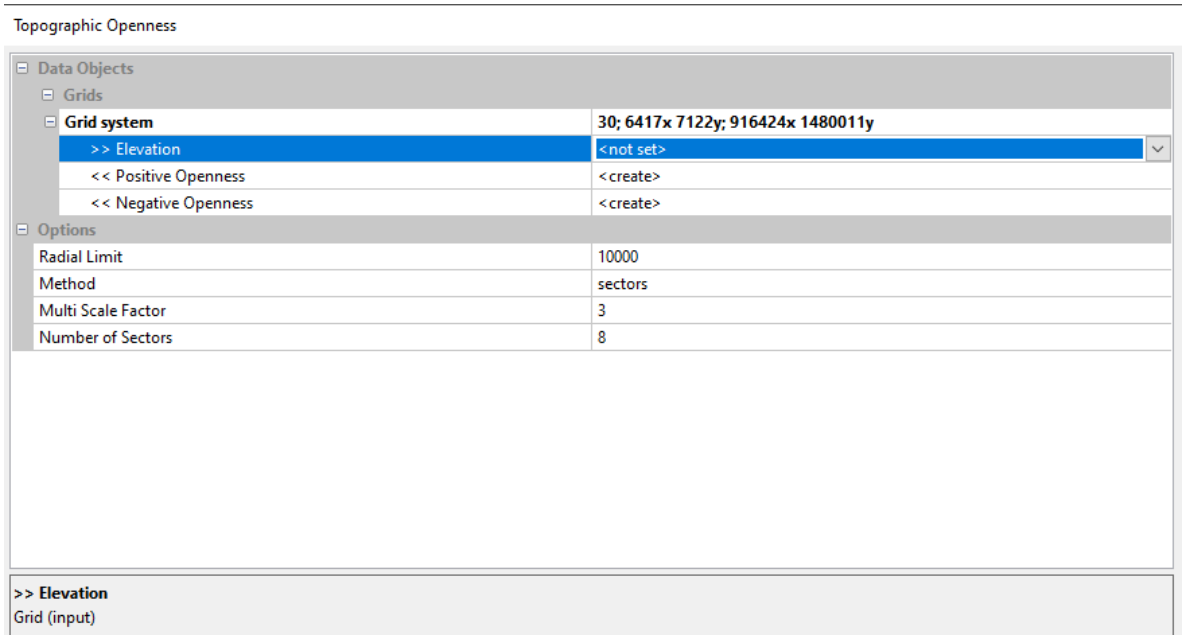



**Figura 4-11** Herramienta Topographic Openness. Fuente: Elaboración propia.

### 4.3.4 Terrain Ruggedness Index

En el menú **Geoprocessing >> Terrain Analysis >> Morphometry** ubicamos la herramienta Terrain Ruggedness Index. La **Tabla 4-4** muestra los derivados que se pueden calcular a partir de esta herramienta.

**Tabla 4-4** Derivados de la herramienta Terrain Ruggedness Index.

| Derivados del DEM | Función del SAGA | Descripción |
|---|---|---|
| Índice de rugosidad del terreno (TRI) | Terrain Ruggedness Index | Medida rápida y objetiva de heterogeneidad del terreno |

Esta herramienta admite como entrada el modelo digital de elevación (Elevation) y genera el raster de Terrain Ruggedness Index. Adicionalmente, la ventana de diálogo solicita otros parámetros adicionales los cuales se dejan con los valores por defecto (ver **Figura 4-12**). El resultante de esta operación se agrega automáticamente en la ventana "Tree" del SAGA GIS.

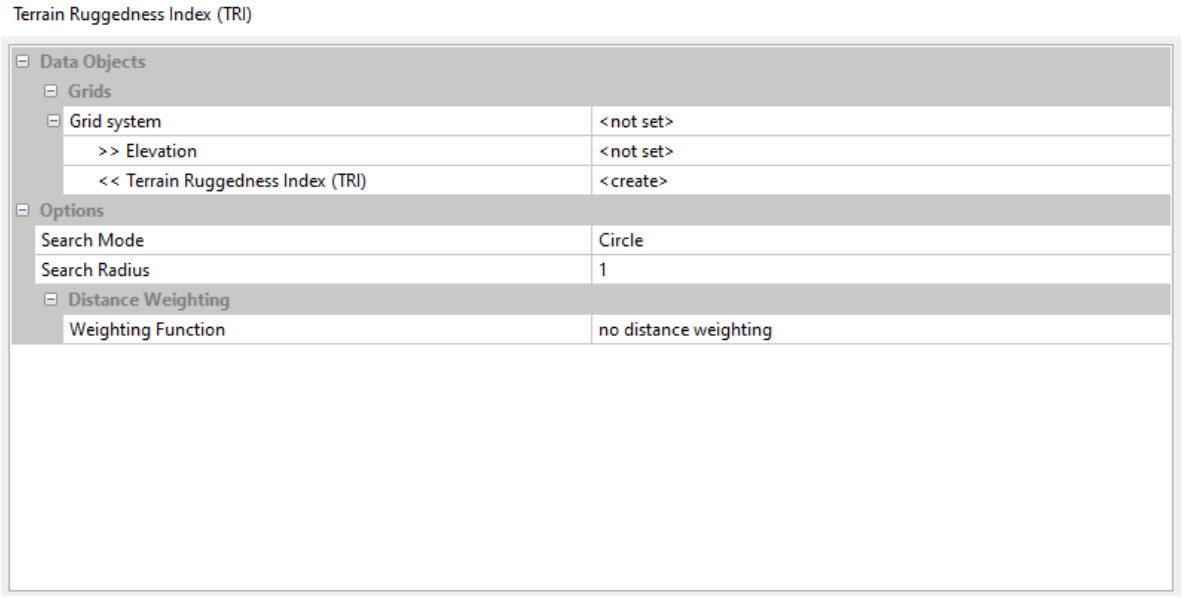



**Figura 4-12** Herramienta Ruggedness Index. Fuente: Elaboración propia.

### 4.3.5 Topographic Position Index

En el menú **Geoprocessing >> Terrain Analysis >> Morphometry** ubicamos la herramienta **Topographic Position Index**. La **Tabla 4-5** muestra los derivados que se pueden calcular a partir de esta herramienta.

**Tabla 4-5** Derivados de la herramienta Topographic Position Index.

| Derivados del DEM | Función del SAGA | Descripción |
|---|---|---|
| Índice de posición topográfica (TPI) | Topographic Position Index | Diferencia entre la elevación en una celda y la elevación promedio de un conjunto de celdas que la rodean (dentro un radio predeterminado). Los valores de TPI por encima de cero muestran ubicaciones que son más altas que el promedio de la ventana local |

Esta herramienta admite como entrada el modelo digital de elevación (Elevation) y genera el raster del Topographic Position Index. Adicionalmente, la ventana de diálogo solicita otros parámetros adicionales los cuales se dejan con los valores por defecto (ver **Figura 4-13**). El resultante de esta operación se agrega automáticamente en la ventana "Tree" del SAGA GIS.

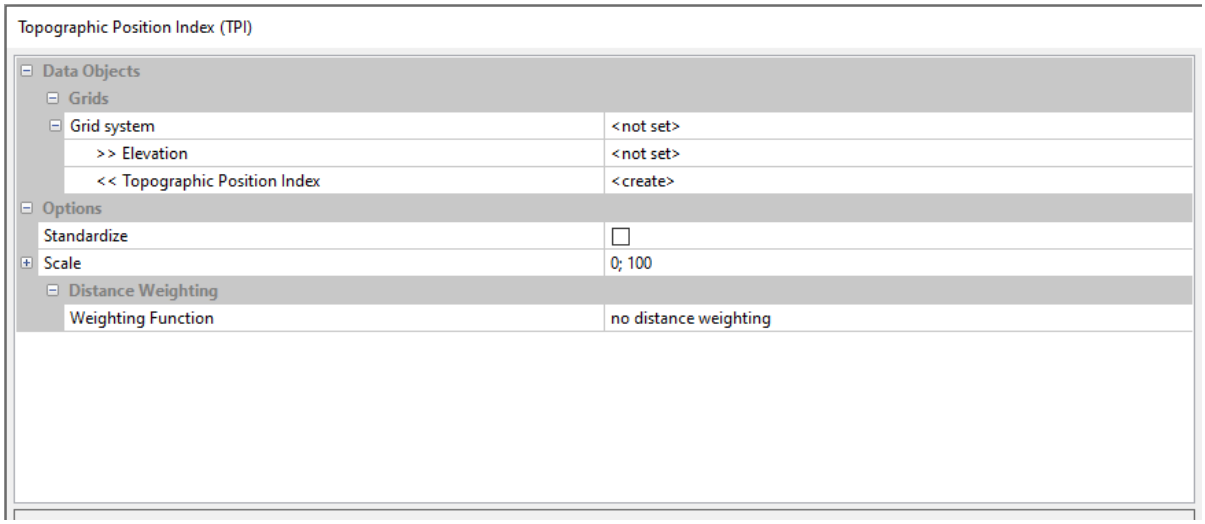



**Figura 4-13** Herramienta Topographic Position Index. Fuente: Elaboración propia.

### 4.3.6 Sky View Factor

Finalmente, en el menú **Geoprocessing >> Terrain Analysis >> Lighting** ubicamos la herramienta **Sky View Factor**. La **Tabla 4-4** muestra los derivados que se pueden calcular a partir de esta herramienta.

**Tabla 4-6** Derivados de la herramienta Sky View Factor.

| Derivados del DEM | Función del SAGA | Descripción |
|---|---|---|
| Factor de cielo visible | Sky View Factor | Parámetro adimensional con valores entre cero y uno. Representa la fracción de cielo visible en un hemisferio que se encuentra centrado sobre la ubicación analizada |
| Cielo visible | Visible Sky | Describe el porcentaje del hemisferio despejado por encima de una determinada ubicación |

Esta herramienta recibe como entrada el modelo digital de elevación (Elevation) y genera los raster de Sky View Factor y Visible Sky. Adicionalmente, la ventana de diálogo solicita otros parámetros adicionales los cuales se dejan con los valores por defecto (ver **Figura 4-14**). El resultante de esta operación se agrega automáticamente en la ventana "Tree" del SAGA GIS.

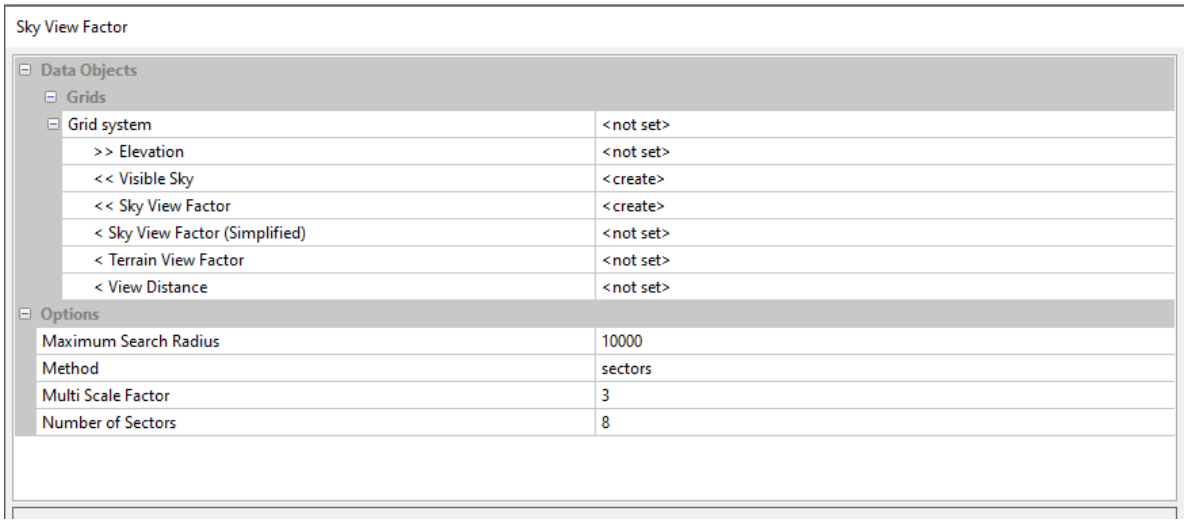



**Figura 4-14** Herramienta Sky View Factor. Fuente: Elaboración propia.

Para efectos del Mapeo Digital de Suelos, únicamente se utilizan los derivados tipo raster. Una manera de exportarlos es a través del menú **Geoprocessing >> File >> Grid >> Export >> GeoTIFF**.

En la opción Grid(s), se selecciona el producto que se quiera exportar, marcándolo en la ventana izquierda y presionando los botones (<, <<, >>, >). Finalmente, en la opción File, se selecciona la ruta de almacenamiento para el raster de salida (ver **Figura 4-15**).

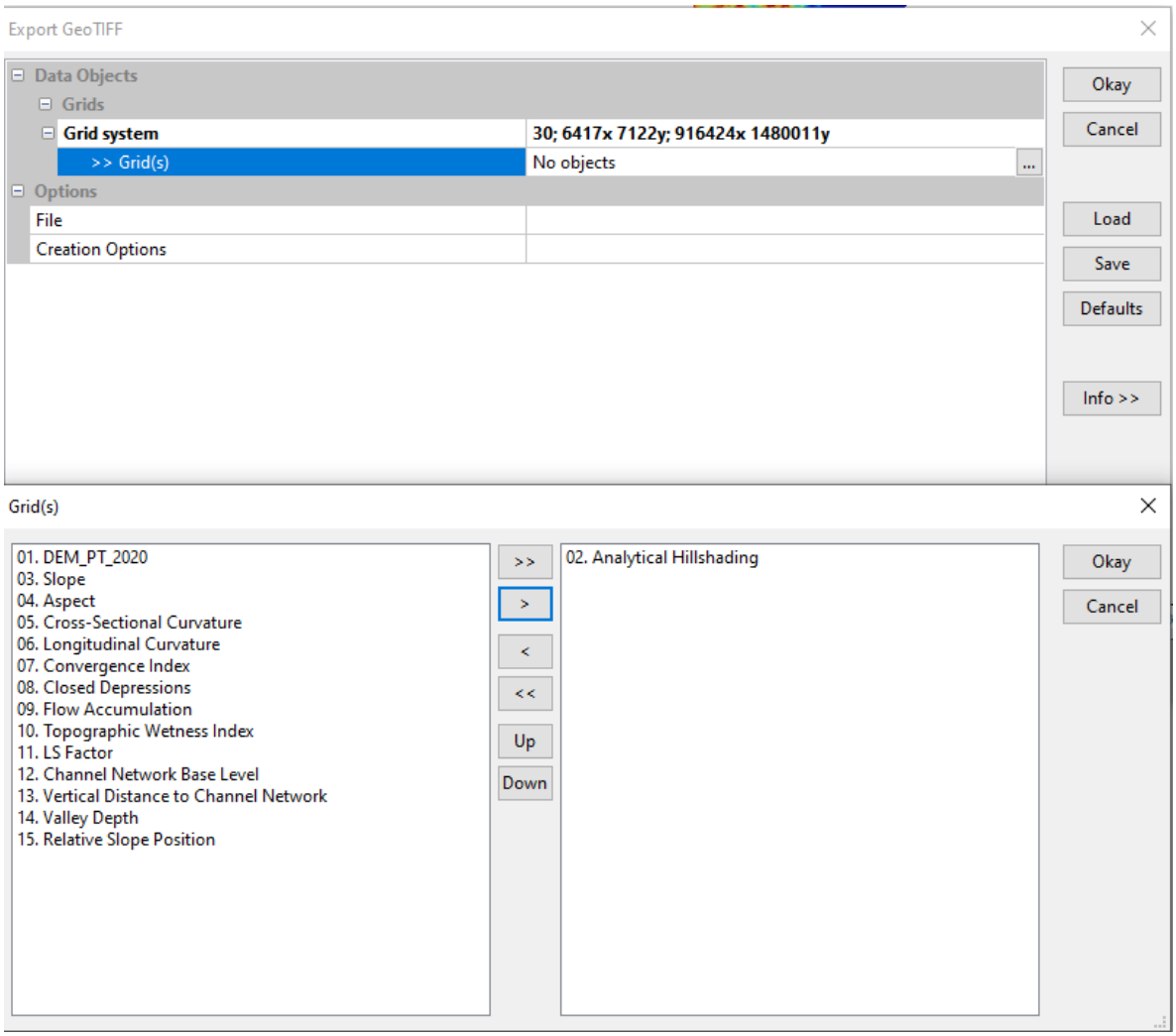



**Figura 4-15** Visualización de las ventanas para exportar los archivos GeoTIFF derivados del DEM. Fuente: Elaboración propia.

## 4.4 Control de calidad de las capas

El control de la calidad de las capas generadas es un paso importante previo a incorporar estas en la herramienta bajo la estructura de carpetas indicada. Es por tanto una buena practica revisar que los archivos de las covariables se encuentran completos y que cubren el área de interés. Esta verificación puede llevarse a cabo por medio de cualquier software SIG p.e. QGIS. Es importante, verificar que el dem y derivados se encuentran en la proyección deseada, preferiblemente en proyección con pixeles en unidades en metros y no grados. Respecto a los archivos Shapefile ESRI, estos, como se ilustra a continuación con capa cartográfica de clima, están conformados por varios subarchivos:

```
CLIMA_CESAR_20170830VF.dbf
CLIMA_CESAR_20170830VF.prj
CLIMA_CESAR_20170830VF.sbn
CLIMA_CESAR_20170830VF.sbx
CLIMA_CESAR_20170830VF.shp
CLIMA_CESAR_20170830VF.shp.xml
CLIMA_CESAR_20170830VF.shx
```

Es importante que estas diferentes extensiones están presentes ya que guardar información como el tipo de proyección, la tabla de atributos asociada, geométrias, entre otros.

# 5 Panel de la herramienta

Para hacer uso de la herramienta se debe abrir el archivo principal *dinoSOIL-toolbox.R* (ubicado dentro de la carpeta src) en RStudio (ir a menú > file > Open File). La **Figura 5-1** ilustra el archivo abierto en el ambiente de RStudio.

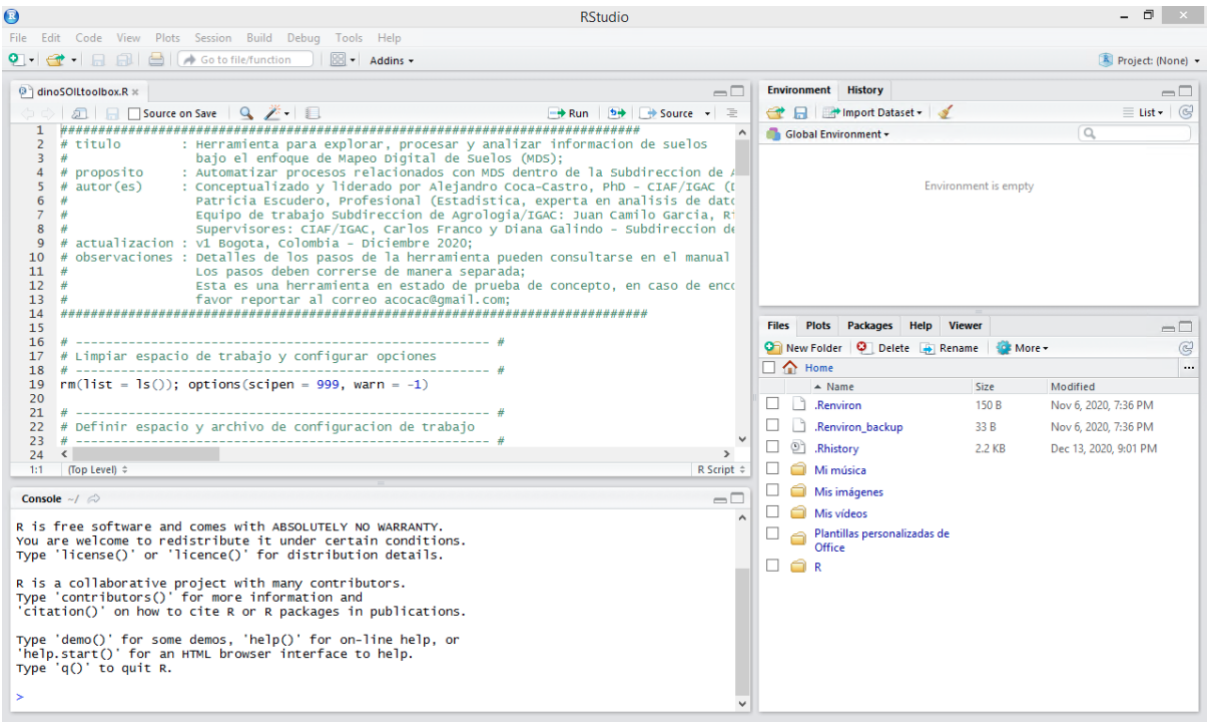



**Figura 5-1** Ejemplo del archivo principal de la herramienta, *dinoSOIL-toolbox.R*, abierto en el ambiente de RStudio.

El cabezote del script cargado contiene toda la información perteneciente a la herramienta tales como título, propósito, autores, fechas de actualización y observaciones. Este texto no debe modificarse al menos de que se realicen

modificaciones posteriores a las funcionalidades de la herramienta. En la parte inferior del cabezote entre las líneas 16 a 19 se tiene las primeras operaciones de la herramienta relacionadas con la limpieza del espacio de trabajo y configurar opciones generales:

```
# ------------------------------------------------------ #
# Limpiar espacio de trabajo y configurar opciones
# ------------------------------------------------------ #
rm(list = ls()); options(scipen = 999, warn = -1)
```

En este caso la primera parte, `rm(list=ls())` borrará todos los objetos del espacio de trabajo. Posteriormente, `options(scipen = 999, warn = -1)` , lo que hace es evitar la notación científica en grandes valores p.e. 100000000000 en vez de 1e+11 configurando `scipen = 999` y mensajes de alerta con `warn = -1`.

## 5.1 Carga de las rutas de los scripts y configuraciones del proyecto

La carga de las rutas de los scripts y configuraciones del proyecto se encuentra entre las líneas 21 a 31. En la variable `r.dir` se debe copiar la ruta a la carpeta que contiene el código fuente or source (src) de la herramienta. En la segunda parte, se debe indicar la ruta a la carpeta del proyecto dentro de la variable `proyecto.dir`. Las variables del archivo cong.txt son declaradas a través de la variable conf.file. En el momento de indicar las rutas, estas deben ir con los separadores de barra / y no en barra oblicua invertida \. En el caso del sistema operativo Windows, es importante verificar que la ruta comienza con la unidad de disco donde se aloja el proyecto. En el ejemplo seria la unidad `F:`, seguido de la ruta completa.

```
# ------------------------------------------------------ #
# Definir espacio y archivo de configuración de trabajo
# ------------------------------------------------------ #
### Directorio de los códigos R  (copiar la ruta completa donde están los códigos
o carpeta src)
r.dir = '/Volumes/Alejo/Users/ac/desarrollos/dinoSOIL-toolbox/src'

### Directorio del proyecto (copiar la ruta completa donde se aloja el proyecto
o carpeta proyecto)
# Indicar la ruta al proyecto
proyecto.dir = '/Volumes/Alejo/Users/ac/Documents/proyecto_cesarmagdalena'
# Cargar el archivo de configuración
conf.file = paste0(proyecto.dir,'/config/conf.txt')
```

La **Figura 5-2** muestra como las anteriores variables declaradas ahora se agregan al panel de Área de trabajo y el histórico, ubicado en la parte superior derecha.

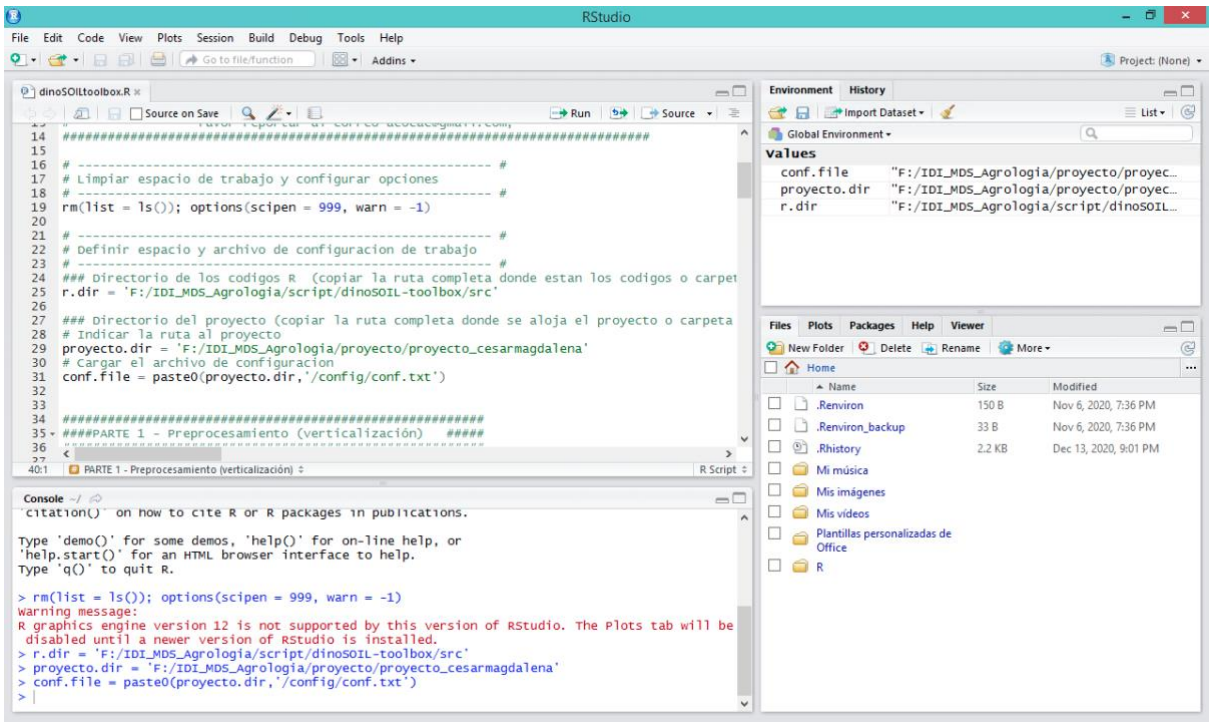



**Figura 5-2** Ejemplo de la adición de las variables entre las líneas 21 a 31 del archivo principal de la herramienta, *dinoSOIL-toolbox.R*, abierto en el ambiente de RStudio.

## 5.2 Preprocesamiento de la base de datos

Posterior a declarar las rutas de los scripts y el proyecto y su archivo `config.txt` asociado en el ambiente de trabajo, se da paso al primer componente referido al preprocesamiento de la(s) base(s) de datos, ya sean de observaciones y/o perfiles. Este componente, entre las líneas 34 y 35, se refiere a la verticalización de estas según su disponibilidad en el proyecto. Esta verticalización es principalmente útil para interpolar profundidades definidas de las variables continuas de interés, por ejemplo pH o carbono orgánico. El modulo se compone de tres líneas o subcomponentes i) cargar modulo, en la cual se carga el script específico del módulo de la carpeta modules (ubicada en el directorio src); ii) consulta de argumentos, en el cual se definen los valores de los parámetros/argumentos del script; y iii) ejecución de la función del modulo. Las líneas a continuación presentan el esquema general del componente de preprocesamiento (verticalización). Se recomienda ejecutar cada línea por separado.

```
############################################################
####PARTE 1 - Preprocesamiento (verticalizacion)   #####
############################################################

#### cargar modulo ###
source(paste0(r.dir,'/modules/1_Preprocesamiento.R'))

#### consultar al usuario argumentos del modulo ###
args_p1 <- prompt.user.part1()

#### ejecutar la funcion del modulo ###
Preprocesamiento(args_p1[[1]],args_p1[[2]],args_p1[[3]],args_p1[[4]])
```

La consulta de argumentos debe hacerse previo a la ejecución del modulo, en este caso Preprocesamiento. La **Figura 5-3** muestra ejemplo de los argumentos diligenciados. En este caso, se comienza con la base de datos de perfiles, de la cual se pide el nombre de archivo BD_PERFILES_MODELAMIENTO_PT_2020.xlsx (alojado en la ruta datos > entrada > 0_basededatos > originales). Ya que estos archivos contienen varias pestañas p.e. ORIGINAL y SITIO para el caso del proyecto Cesar/Magdalena, se consulta cual de ellas contiene la información de las profundidades. Asimismo se hace la consulta sobre la columna que contiene el ID único del perfil, en este caso COD_PERFIL.

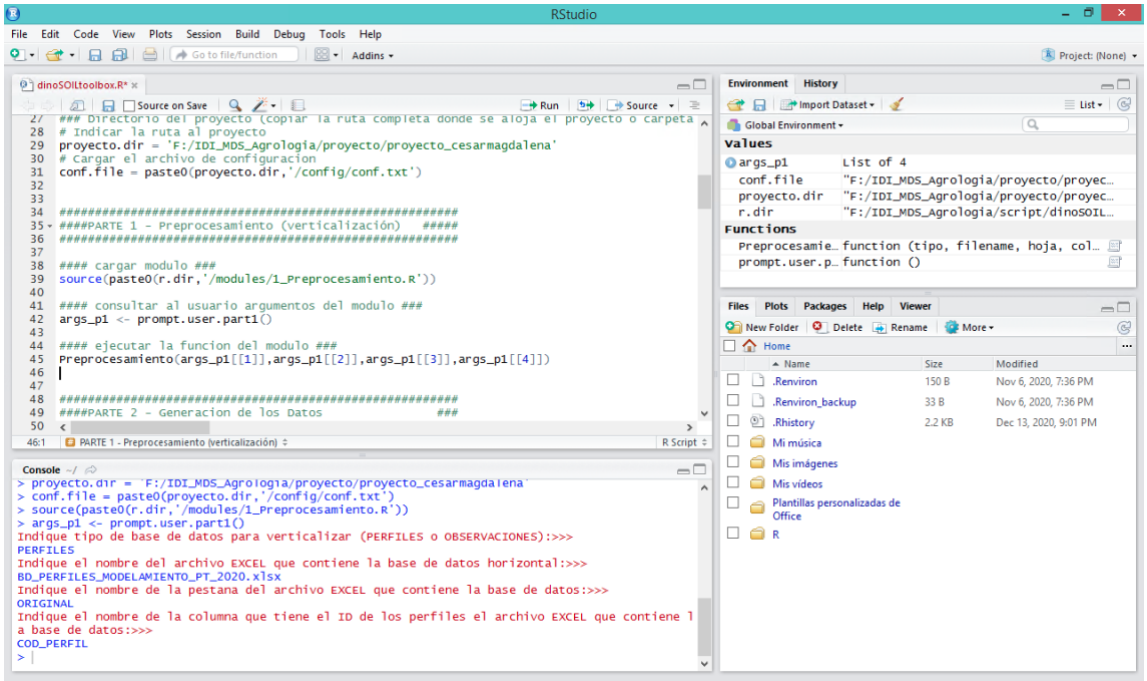



**Figura 5-3** Ejemplo de la consulta de argumentos en el modulo de Preprocesamiento y almacenamiento de estos en el espacio de trabajo.

Como resultado de la ejecución se genera un archivo separado por comas (CSV) el cual contiene la base de datos verticalizada, ya sea de perfiles (BDP) u observaciones (BDO), ubicado en la ruta (datos > entrada > 0_basededatos > derivados > verticalizado) (ver **Figura 5-4**). El formato CSV fue elegido debido a su interoperabilidad y fácil lectura en cualquier sistema operativo.


**Figura 5-4** Ejemplo de los archivos de salida del modulo de Preprocesamiento.

En Windows, los archivos CSV pueden abrirse con facilidad en MS Excel. Para configurarlo en formato de columnas, se debe usar la función del asistente para convertir texto en Columnas (en la pestaña Datos > Texto en columnas) (ver **Figura 5-5**). En la sección de Separadores, señalar la casilla de Coma, y posteriormente dar clic en Finalizar.

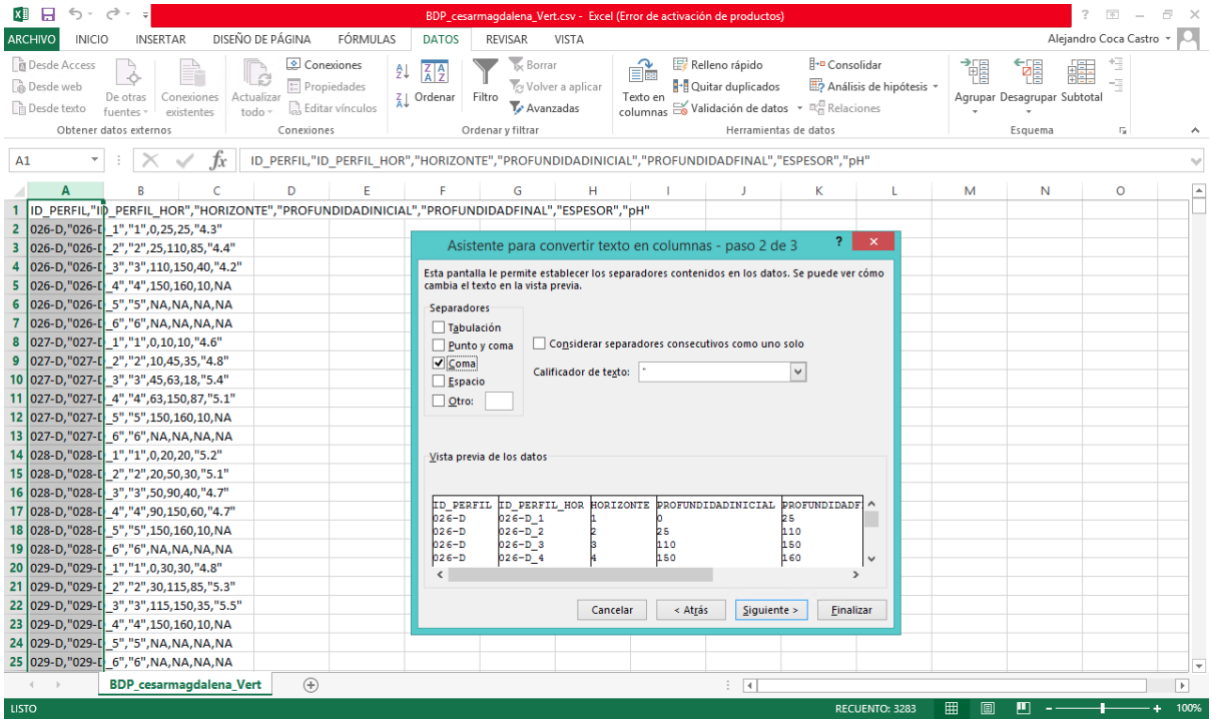



**Figura 5-5** Ejemplo de conversión de texto del archivo CSV a columnas en MS Excel.

Como resultado de la separación de columnas, la base de datos verticalizada tiene un ID único de perfil (ID_PERFIL) con cada horizonte (ID_PERFIL_HOR) el cual se tiene información de la profundidad inicial, final y espesor. Adicionalmente, cada horizonte tiene valores de la variable continua de interés definida en el archivo config.txt, en este caso pH. Posterior a la visualización de esta base de datos, se cierra el archivo y se le indica la opción de No guardar. Esto para evitar que el archivo CSV pierda su formato original y se pueda ser leído correctamente en los componentes restantes de la herramienta dinoSoil toolbox.

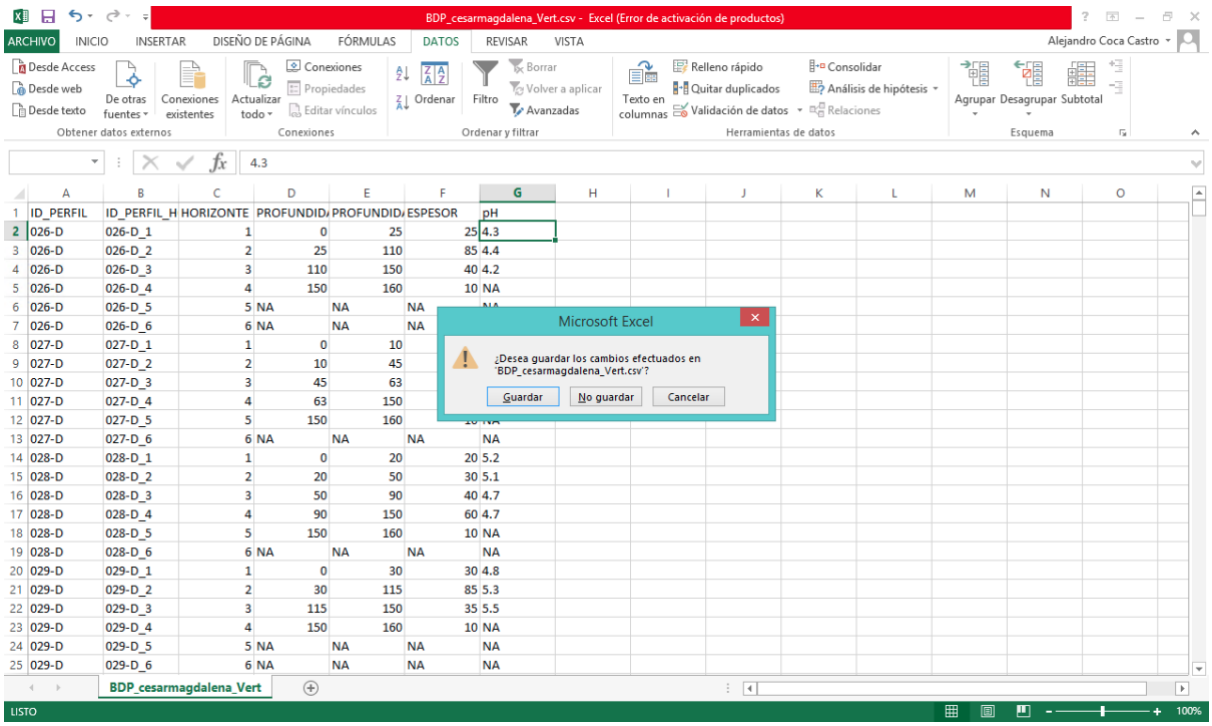



**Figura 5-6** Ejemplo del archivo CSV de la base de datos vertilcalizada convertido en columnas en MS Excel. Se sugiere no guardar los cambios para evitar problemas de lectura del archivo.

Alternativamente, para aquellos usuarios con experiencia en R, estos archivos pueden cargase para su verificación mediante el comando read.csv. Se recomienda hacer esta inspección en un nuevo script, evitando en lo posible cualquier edición sobre el script principal de la herramienta.

## 5.3 Generación de la matriz de datos

Este componente, entre las líneas 48 a 59, tiene como tarea generar el archivo CSV de la matriz general de datos, MatrixDatos.csv (datos > salida > 0_matriz). Igualmente, se genera un archivo raster multi-banda en formato GeoTIFF y el nombre de las bandas (o covariables) en formato RDS en la ruta datos > salida > 1_covariables. Este archivo raster contiene todas las covariables ambientales indicadas en el archivo `config.txt`. Este raster es usado para extraer los valores según las coordenadas de los perfiles de las bases de datos, perfiles y/o observaciones, así como para la espacialización de las variables objetivo en el área de estudio. Las líneas de este componente son:

```
#########################################################
####PARTE 2 - Generacion de los Datos                ###
#########################################################

#### cargar modulo ###
source(paste0(r.dir,'/modules/2_Datos.R'))

#### consultar al usuario argumentos del modulo ###
args_p2 <- prompt.user.part2()

#### ejecutar la funcion del modulo ###
Datos(args_p2[[1]], args_p2[[2]], args_p2[[3]])
```

La **Tabla 5-1** describe cada uno de los argumentos de consulta de este componente, similar al mostrado en la **Figura 5-3**. El archivo solicitado de Excel debe estar ubicado dentro de la ruta datos > entrada > 0_basededatos > originales.

**Tabla 5-1** Descripción por argumento y ejemplo del componente de generación de los datos.

| Argumento | Ejemplo |
|---|---|
| Indique el nombre del archivo EXCEL con las variables categoricas y coordenadas:>>> | BD_CATEGORICAS_COOORDENADAS.xlsx |
| Indique el nombre de la pestana del archivo EXCEL con las variables categoricas y coordenadas:>>> | Hoja1 |
| Indique el nombre de la columna que tiene el ID de los perfiles del archivo EXCEL las variables categoricas y coordenadas:>>> | COD_PERFIL |

Los archivos de salida de este componente con su ubicación en la estructura de carpetas del proyecto son descritos en la **Tabla 5-2**. Como resultado se tienen dos archivos finales usados en componentes restantes.

**Tabla 5-2** Descripción de los archivos de salida del componente de generación de los datos.

| Carpeta | Archivo(s) | Formato | Ruta y ejemplo(s) |
|---|---|---|---|
| entrada | Interpolado de la variable objetivo | CSV | datos > entrada > 0_basededatos > derivados > interpolados<br>BD_cesarmagdalena_pH.csv |
| entrada | Rasterizado y tabla de codificación de los valores de las capas vectoriales | GeoTIFF<br>CSV | datos > entrada > 1_covariables > raster > clima<br>clima.tif<br>clima.csv |
| entrada | Archivo del NDVI generado por la librería rgee (opcional) | GeoTIFF | datos > entrada > 1_covariables > raster > ndvi<br>ndvi2019-01-01_2020-08-30.tif |
| salida | Matriz general de datos | CSV | datos > salida > 0_matriz<br>MatrixDatos.csv |
| salida | Raster multi-banda con nombre de las bandas (o covariables) | GeoTIFF<br>RDS | datos > salida > 1_covariables<br>covariables.tif<br>covariables.rds |

## 5.4 Selección de las covariables

A partir de este componente, entre las líneas 62 a 73, resulta especifico el procesamiento de los datos por variable de interés. De esta manera, a partir de la matriz general del componente anterior, se trata la variable de interés con sus respectivas covariables ambientales. A continuación se relacionan las líneas de este componente:

```
##########################################################
#####PARTE 3 - Seleccion variables (RFE y Boruta) ######
##########################################################

#### cargar modulo ###
source(paste0(r.dir,'/modules/3_SelVariables.R'))

#### consultar al usuario argumentos del modulo ###
args_p3 <- prompt.user.part3()

#### ejecutar la funcion del modulo ###
SelVariables(args_p3[[1]], args_p3[[2]])
```

La **Tabla 5-3** describe cada uno de los argumentos de consulta de este componente. Es importante indicar que como parte del primer argumento aparece un mensaje con las variables disponibles para el modelo separadas por el símbolo |. El segundo

argumento consulta si se quiere hacer por separado o en conjunto (AMBAS) la identificación de las covariables de importancia según la base de datos de observación y/o perfiles.

**Tabla 5-3** Descripción por argumento y ejemplo del componente de generación de los datos.

| Argumento | Ejemplo |
|---|---|
| Indique el nombre de la variable objetivo de acuerdo al listado superior:>>> | GRANGRUPO |
| Indique el tipo de base de datos para modelar (AMBAS, PERFIL, OBSERVACION):>>> | AMBAS |

Posterior al ingreso de los argumentos, una serie de tratamiento de los datos se hacen en la ejecución del componente. Un primer proceso se refiere a que los valores extremos son filtrados mediante la retención de aquellos valores entre el percentil 5 y 95 por covariable. Posterior a este proceso, se hace una partición de los datos en entrenamiento (70%) y evaluación (30%). La selección de las covariables se hace sobre los datos de entrenamiento mediante los algoritmos de RFE y Boruta. Los archivos de salida de este componente con su ubicación en la estructura de carpetas del proyecto son descritos en la **Tabla 5-4**.

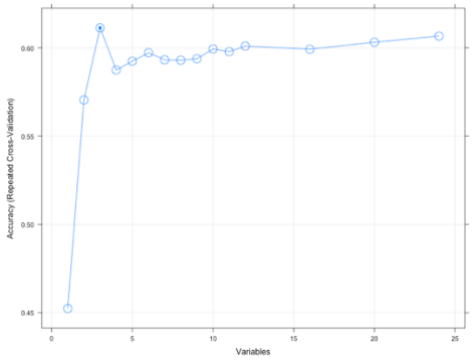
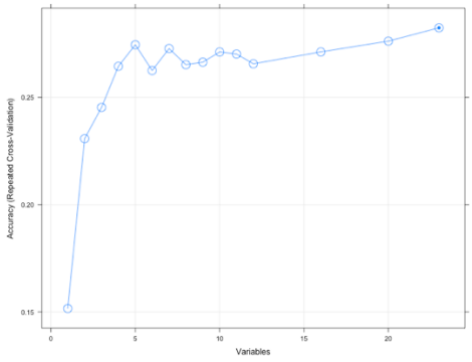
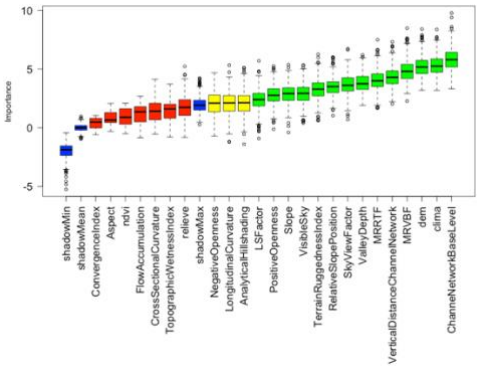
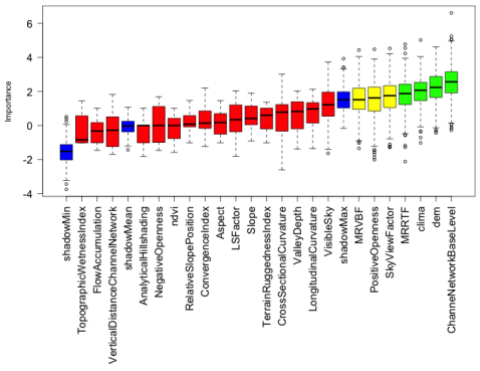
**Tabla 5-4** Descripción de los archivos de salida del componente de la selección de covariables.

| Carpeta | Archivo(s) | Formato | Ruta y ejemplo(s) |
|---|---|---|---|
| exploratorio | Modelos y gráfico de variables seleccionadas por RFE | RDS<br>PNG | exploratorio > AMBAS > rds > variable rfe.rds<br>exploratorio > AMBAS > figuras > variable rfe.png |
| exploratorio | Modelos y gráfico de variables seleccionadas por Boruta | RDS<br>PNG | exploratorio > AMBAS > rds > variable boruta.rds<br>exploratorio > AMBAS > figuras > variable rfe boruta png |
| modelos | Partición datos y sus coordenadas | CSV<br>RData | modelos > AMBAS > 0_particion > variable coordenadas.csv<br>entrenamiento.csv<br>evaluacion.csv<br>particion.RData |

Para el caso de las variables categóricas, tanto de los algoritmos RFE y Boruta también se hace la selección de covariables con el dataset balanceado usando la categoría con

menor número de observaciones. Los resultados pueden ser encontrados con el sufijo _down. La **Tabla 5-5** presenta una comparación entre las salidas gráficas del RFE y Boruta para el dataset ORIGINAL en comparación al balanceado. Es de recordar que el número óptimo de covariables es indicado con el punto y con las barras de color verde en el algoritmo RFE y Boruta respectivamente. Se observa que en este caso, es mejor conservar los datos originales sin hacer un remuestreos de acuerdo a los valores de las métricas que indican el desempeño, en este caso Exactitud General (o Overall Accuracy). Para una mejor interpretación de las salidas de los algoritmos de selección, RFE y Boruta, se recomienda consultar la investigación de Gonzales (2017).

**Tabla 5-5** Comparación de las salidas gráficas de los algoritmos RFE con datos originales y balanceados de la variable Gran Grupo del proyecto Cesar/Magdalena.

| Algoritmo | ORIGINAL | BALANCEADO |
|---|---|---|
| RFE |  |  |
| Boruta |  |  |

## 5.5 Exploración y análisis estadístico

Este componente, líneas 76-86, realiza análisis descriptivo, exploratorio y estadístico por variable de interés y sus covariables óptimas. A continuación se relacionan las líneas de este componente:

```
#########################################################
#####PARTE 4a - Modelado: Exploracion de los datos    ##
#########################################################
#### cargar modulo ###
source(paste0(r.dir,'/modules/4a_ModExploratorio.R'))

#### consultar al usuario argumentos del modulo ###
args_p4a <- prompt.user.part4a()

#### ejecutar la funcion del modulo ###
ModExploracion(args_p4a[[1]], args_p4a[[2]], args_p4a[[3]])
```

La **Tabla 5-6** describe cada uno de los argumentos de consulta de este componente. Al igual que el anterior componente, es importante indicar que como parte del primer argumento aparece un mensaje con las variables disponibles para el modelo separadas por el símbolo |. El segundo argumento consulta si se quiere hacer por separado o en conjunto (AMBAS) la identificación de las covariables de importancia según la base de datos de observación y/o perfiles. Adicional a estos argumentos, también se consulta el número optimo de covariables derivado del análisis del componente de selección de las mismas.

**Tabla 5-6** Descripción por argumento y ejemplo del componente de generación de los datos.

| Argumento | Ejemplo |
|---|---|
| Indique el nombre de la variable objetivo de acuerdo al listado superior:>>> | GRANGRUPO |
| Indique el tipo de base de datos para modelar (AMBAS, PERFIL, OBSERVACION):>>> | AMBAS |
| Indique el numero limite de covariables a considerar según interpretación del RFE y Boruta:>>> | 3 |

Posterior al ingreso de los argumentos, una serie de operaciones hacen durante en la ejecución del componente. Un primer proceso se refiere a generar tablas con información descriptiva de la variable de interés y covariables. Luego, se generan unos gráficos con fines exploratorios para el entendimiento de las relaciones de la variable objetivo y covariables. Finalmente, se llevan a cabo análisis estadísticos para confirmar o descartar la significancia de las covariables seleccionadas. Los archivos de salida (intermediarios y finales) con su ubicación en la estructura de carpetas del proyecto son descritos en la **Tabla 5-7**.

**Tabla 5-7** Descripción de los archivos de salida del componente exploratorio y estadístico.

| Carpeta | Archivo(s) | Formato | Ruta y ejemplo(s) |
|---|---|---|---|
| 1_descriptivo | Tablas con información descriptiva de la variable objetivo y covariables (categóricas y/o continuas) | CSV | modelos > AMBAS > 1_exploratorio > variable > x_covariables > 1_descriptivo<br>1_descriptivo_variableobjetivo.csv<br>2_descriptivo_covariables-continuas.csv<br>3_descriptivo_covariable-categorica_X.csv |
| 2_graficos | Gráficos de correlación de las covariables, boxplots, barras (solo cuando la variable es categórica), y dispersión (solo cuando la variable es continua), | PNG | modelos > AMBAS > 1_exploratorio > variable > x_covariables > 2_graficos<br>train_correlationmatrix.png<br>test_correlationmatrix.png |
| 3_estadistico | Evaluación de normalidad (solo cuando la variable es continua), test no-paramétrico y post-hoc (solo cuando hay significancia) | CSV | modelos > AMBAS > 1_exploratorio > variable > x_covariables > 2_graficos<br>1_normalidad.csv<br>x_kruskal_grupos.csv<br>xx_posthoc-valores_X.csv<br>x_chi-cuadrado_X.csv |

## 5.6 Ejecución de los algoritmos configurados

Este componente, líneas 89-99, realiza todo lo referido al entrenamiento y calibración de los modelos listados en el `config.txt` o seleccionados por defecto. A continuación se relacionan las líneas de este componente:

```
##########################################################
#####PARTE 4b - Modelado: Ejecucion de los modelos    ##
##########################################################
#### cargar modulo ###
source(paste0(r.dir,'/modules/4b_ModEjecutar.R'))

#### consultar al usuario argumentos del modulo ###
args_p4b <- prompt.user.part4b()

#### ejecutar la funcion del modulo ###
ModEntrenamiento(args_p4b[[1]],args_p4b[[2]],args_p4b[[3]],args_p4b[[4]],args_p4
b[[5]])
```

La **Tabla 5-8Tabla 5-6** describe cada uno de los argumentos de consulta de este componente. Al igual que el anterior componente, se pregunta la variable de interés, el tipo de base de datos a usar, el número optimo de variables según lo indicado por RFE/Boruta. Adicionalmente, para las variables categóricas, se consulta si los datos se van a manejar en su formato ORIGINAL, o con alguna técnica de balanceo, ya sea remuestrar a la categoría de menor (DOWN) o mayor (UP) observaciones. Finalmente, se debe indicar si usar los modelos declarados en el archivo `config.txt` o por defecto. Una explicación más detallada de este último argumento y su correcta integración en la herramienta se detalla en el **Anexo 3.**

**Tabla 5-8** Descripción por argumento y ejemplo del componente de generación de los datos.

| Argumento | Ejemplo |
|---|---|
| Indique el nombre de la variable objetivo de acuerdo al listado superior:>>> | GRANGRUPO |
| Indique el tipo de base de datos para modelar (AMBAS, PERFIL, OBSERVACION):>>> | AMBAS |
| Indique el numero limite de covariables a considerar según interpretación del RFE y Boruta:>>> | 3 |
| Si la variable es categórica indique la estrategia usada para balancear los datos (UP, DOWN), caso contrario que prefiera desbalanceado o la variable es continua escriba ORIGINAL:>>> | ORIGINAL |
| Indique si usar los algoritmos por DEFECTO o del archivo CONFIG:>>> | CONFIG |

Posterior al ingreso de los argumentos, una serie de operaciones hacen durante en la ejecución del componente. Un primer proceso se refiere a cargar los datos de entrenamiento. Luego cada modelo de los listados en el config.txt o por defecto es entrenado y calibrado de acuerdo a los datos de entrada. Continuando con el ejemplo de la **Tabla 5-8Tabla 5-6**, estos modelos son guardados dentro de la ruta (modelos > AMBAS > 1_exploratorio > GRANGRUPO > 3_covariables > ORIGINAL > CONFIG). Cada modelo es guardado en formato RDS, y es usado posteriormente para definir el mejor modelo según la métrica de desempeño ajustada al tipo de variable así como los mejores hiperparametros. Los archivos RDS se pueden abrir e inspeccionar en un script separado mediante el comando `load(modelo)`. Por ejemplo, la **Figura 5-7** presenta

como se carga el modelo entrenado ranger (Random Forest) para el proyecto Cesar/Magdalena de acuerdo a las configuraciones de la **Tabla 5-8Tabla 5-6**. Este modelo se encuentra dentro de la variable `modelo.ajuste`, la cual se puede inspeccionar en el panel de la consola haciendo clic en Run.

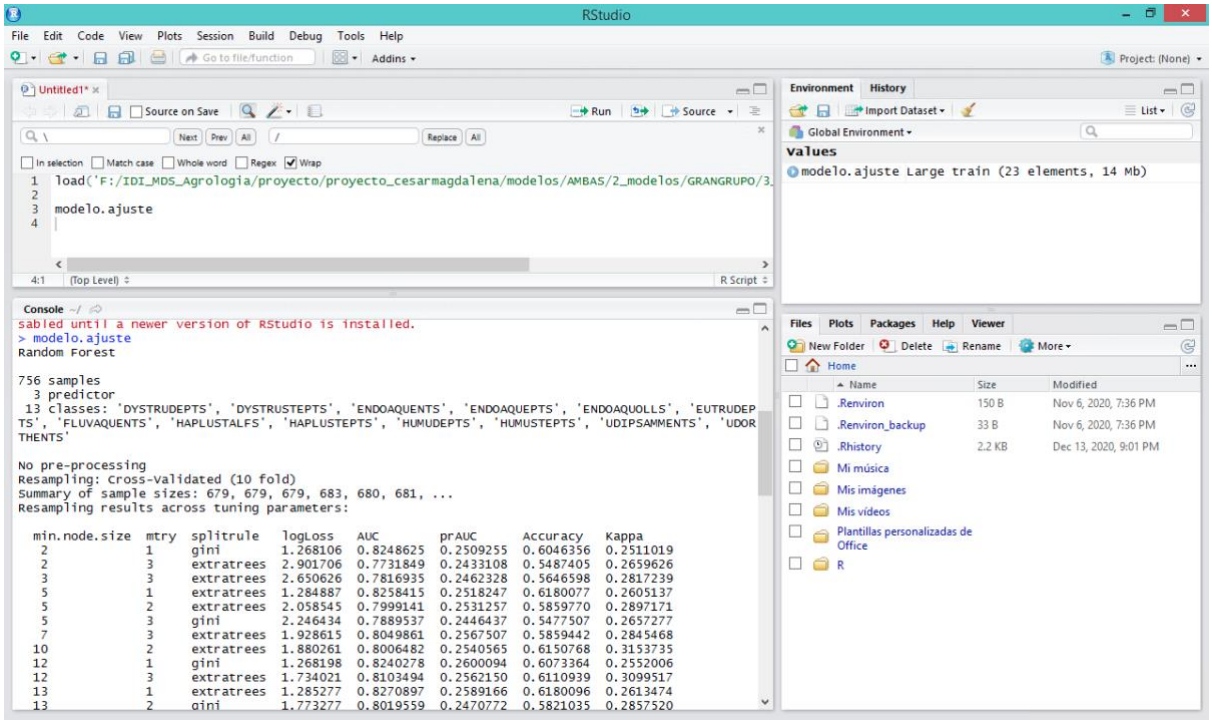



**Figura 5-7** Ejemplo del modelo entrenado y calibrado de ranger (Random Forest) de la variable Gran Grupo del proyecto ejemplo de Cesar/Magdalena. En el panel de consola se lista toda la información relevante como número de muestras, predictores, clases objetivo, si se realiza o no preprocesamiento, y resume de la validación cruzada. Posteriormente, se presenta los resultados de la combinación de hiperparametros (en este caso min.mode.size, mtry, splitrule) y los valores reportados de varias métricas, entre ellas Accuracy y Kappa.

## 5.7 Selección del mejor modelo

Posterior a ejecutar el entrenamiento y calibración de los modelos de Aprendizaje de Maquinas, se procede al componente de la selección del mejor modelo (líneas 102-113). La toma de este modelo esta principalmente basada en la métrica principal de desempeño (ajustada al tipo de variable) definida en el archivo `config.txt`. Los parámetros consultados son los mismos del componente anterior (ver **Tabla 5-8**). A continuación se relacionan las líneas de este componente:

```
############################################################
#####PARTE 4c - Modelado: Identificacion del mejor modelo #
############################################################

#### cargar modulo ###
source(paste0(r.dir,'/modules/4c_ModMejor.R'))

#### consultar al usuario argumentos del modulo ###
args_p4c <- prompt.user.part4c()

#### ejecutar la funcion del modulo ###
ModMejorModelo(args_p4c[[1]],args_p4c[[2]],args_p4c[[3]], args_p4c[[4]],
args_p4c[[5]])
```

Posterior al ingreso de los argumentos, se generan una serie de archivos tabulares y gráficos que soportan la selección del mejor modelo entre la lista de entrenados. Estos archivos se almacenan en carpetas separadas tabular y figuras en la ruta modelos > AMBAS > 3_analisis. La **Tabla 5-9** relaciona los archivos de salida en estas dos carpetas.

**Tabla 5-9** Descripción de los archivos de salida del componente exploratorio y estadístico.

| Carpeta | Archivo | Formato | Ruta y ejemplo(s) |
|---|---|---|---|
| figura | Gráficos de cajuelas comparando los diferentes modelos (y sus configuraciones) de acuerdo a las métricas de desempeño | PNG | modelos > AMBAS > 3_analisis > figura sGRANGRUPO > 3_covariables > ORIGINAL > CONFIG boxplots_modelos.png |
| tabular | Tablas con información de los mejores modelos y sus hiperparametros | CSV | modelos > AMBAS > 3_analisis > tabular GRANGRUPO > 3_covariables > ORIGINAL > CONFIG mejoresmodelos_metricas.csv mejoresmodelos_parametros.csv |

## 5.8 Error e incertidumbre

Este componente, líneas 116-127, como su nombre lo indica genera información y datos relacionados al error e incertidumbre de las predicciones del mejor modelo. Este componente es esencial para planear futuras mejoras metodológicas que apunte a mejorar las predicciones en áreas donde se tiene una alta incertidumbre. A continuación se relacionan las líneas de este componente:

```
#########################################################################
#####PARTE 4d - Modelado: Evaluacion/Incertidumbre del mejor modelo ##
#########################################################################

#### cargar modulo ###
source(paste0(r.dir,'/modules/4d_ModEvaluacion-Incertidumbre.R'))

#### consultar al usuario argumentos del modulo ###
args_p4d <- prompt.user.part4d()

#### ejecutar la funcion del modulo ###
ModEvalIncertidumbre(args_p4d[[1]],args_p4d[[2]],args_p4d[[3]], args_p4d[[4]],
args_p4d[[5]])
```

Los argumentos de consulta son los mismos usados en los componentes de ejecución de los modelos (ver **Tabla 5-8**). Posterior al ingreso de estos, se generan una serie de archivos tabulares, gráficos, espaciales (geoTIFF y mapas) relacionados con la incertidumbre del modelo. Estos archivos se almacenan en carpetas separadas 1 1_metricas, 2_figuras, 3_geotiff y 4_modelo en la ruta modelos > AMBAS > 4_incertidumbre. La **Tabla 5-10** relaciona los archivos de salida en estas dos carpetas.

**Tabla 5-10** Descripción de los archivos de salida del componente exploratorio y estadístico.

| Carpeta | Archivo | Formato | Ruta y ejemplo(s) |
|---|---|---|---|
| 1_metricas | Métricas de desempeño del mejor modelo según el tipo de variable | CSV | modelos > AMBAS > 4_incertidumbre > 1_metricas > GRANGRUPO > 3_covariables > ORIGINAL > CONFIG MetricasEvaluacion_ranger.csv |
| 2_figuras | Gráficas con la medición del error de las predicciones, mapas de incertidumbre e importancia de las covariables | PNG | modelos > AMBAS > 4_incertidumbre > 2_figuras > GRANGRUPO > 3_covariables > ORIGINAL > CONFIG 1_MatrizdeConfusion_ranger.png 2_ImportanciaCovariables_ranger.png 3_Probabilidad_ranger.png 4_Entropia_ranger.png |
| 3_geotiff | Archivos geoTIFF relacionados con la incertidumbre | GeoTIFF | modelos > AMBAS > 4_incertidumbre > 3_geotiff > GRANGRUPO > 3_covariables > ORIGINAL > CONFIG Entropia_ranger.tif Probabilidad_ranger.tif |
| 4_modelo (solo para variables continuas) | Archivo del modelo de regresión cuantilica | RDS | modelos > AMBAS > 4_incertidumbre > 4_modelo > GRANGRUPO > 3_covariables > ORIGINAL > CONFIG Entropia_ranger.tif Probabilidad_ranger.tif |

## 5.9 Predicción

Este componente, líneas 130-140, genera las predicciones del mejor modelo. Este componente es el paso final de la cadena de componentes de la herramienta dinoSoil-toolbox. A continuación se relacionan las líneas de este componente:

```
############################################################
#####PARTE 4e - Modelado: Uso del modelo (prediccion)    ##
############################################################
#### cargar modulo ###
source(paste0(r.dir,'/modules/4e_ModUso.R'))

#### consultar al usuario argumentos del modulo ###
args_p4e <- prompt.user.part4e()

#### ejecutar la funcion del modulo ###
ModUso(args_p4e[[1]], args_p4e[[2]], args_p4e[[3]], args_p4e[[4]],
args_p4e[[5]])
```

Los argumentos de consulta son los mismos usados en los componentes de ejecución de los modelos (ver **Tabla 5-8**). Posterior a la ejecución del modulo, se genera el mapa y archivo geoTIFF de la predicción en la carpeta prediccion. La **Tabla 5-11** relaciona los archivos de salida en estas dos carpetas.

**Tabla 5-11** Descripción de los archivos de salida del componente exploratorio y estadístico.

| Carpeta | Archivo | Formato | Ruta y ejemplo(s) |
|---|---|---|---|
| figuras | Mapa de la predicción | PNG | prediccion > AMBAS > figuras > GRANGRUPO > 3_covariables > ORIGINAL > CONFIG<br>Prediccion_ranger.png |
| geotiff | Archivo raster geoTIFF y tabla de codificación para su uso en software SIG | GeoTIFF | prediccion > AMBAS > geotiff > GRANGRUPO > 3_covariables > ORIGINAL > CONFIG<br>Prediccion_ranger.tif<br>CodificacionClases_ranger.csv |

# 6 Mantenimiento de la herramienta

El mantenimiento de la herramienta requiere un nivel de familiaridad en programación en R. En especifico, se requiere un entendimiento sobre programación en módulos, uso de bucles, funciones, asociación de scripts, entre otros aspectos avanzados. Cabe

indicar que el script principal, `dinoSOILtoolbox.R`, se encuentran enlazado con una estructura de módulos y funciones como se presenta a continuación:

```
src
├── dinoSOILtoolbox.R
├── functions
│   ├── 0_CargarConfig.R
│   ├── 1_Variables.R
│   ├── 2_Ponderado.R
│   ├── 3_Outliers.R
│   ├── 4_ConfigModelos.R
│   └── 5_Predict.R
└── modules
    ├── 1_Preprocesamiento.R
    ├── 2_Datos.R
    ├── 3_SelVariables.R
    ├── 4a_ModExploratorio.R
    ├── 4b_ModEjecutar.R
    ├── 4c_ModMejor.R
    ├── 4d_ModEvaluacion-Incertidumbre.R
    └── 4e_ModUso.R
```

La carpeta `modules` contiene cada uno de los scripts relacionados con los componentes descritos en la **Sección 5**. Cada uno de estos scripts guarda una estructura de cabezote con información de su propósito, fecha de creación, autor(es), archivos de entrada y salida, y observaciones adicionales; una función para hacer la consulta de argumentos; y la función principal del script. El siguiente ejemplo muestra la estructura del script `1_Preprocesamiento.R` donde {...} contiene el proceso para cada función, en este caso de consulta de argumentos `prompt.user.part1` y preprocesamiento (verticalización) `Preprocesamiento`.

```
################################################################################
# titulo        : Verticalizacion bases de datos de suelo;
# proposito     : Verticalizar bases de datos de suelo provistas en formato
horizontal;
# autor(es)     : Preparado por Sebastian Gutierrez (SG), IGAC-Agrologia;
Adaptado por Alejandro Coca-Castro (ACC), IGAC-CIAF;
# creacion      : Creado SG en Bogota, Colombia / Actualizado por ACC en
Septiembre 2020;;
# entrada       : Base de datos original;
# salida        : Base de datos verticalizada;
# observaciones : ninguna;
################################################################################

prompt.user.part1 <- function(){...}

Preprocesamiento <- function(tipo, filename, hoja, columna){...}
```

Adicional a los scripts de la carpeta `modules`, se tiene una serie de funciones auxiliares en la carpeta `functions`. Estas funciones se usan en uno o varios de los componentes y estas pueden ajustarse de acuerdo a los requerimientos del proyecto. Cada función se encuentra documentada con su cabezote y la mayoría de las líneas y procesos explicados para su lectura y/o modificación.

Posterior a los módulos del script `dinoSOILtoolbox.R` se deja un listado de recursos con posibles mejoras que se pueden hacer en ciertos componentes para futuras versiones de la herramienta.

# Referencias


Gabriel Alejandro Perilla, J.-F. M. (2020). Google Earth Engine (GEE): una poderosa herramienta que vincula el potencial de los datos masivos y la eficacia del procesamiento en la nube. *Investigaciones Geográficas*.

Gonzales, D. (2017). *Análisis de predictores ambientales derivados mediante teledetección y su relación con el crecimiento anual periódico de rodales de Nothofagus obliqua en la precordillera andina del maule, Chile* [Universidad de Chile]. http://repositorio.uchile.cl/handle/2250/153114

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, *202*, 18–27. https://doi.org/10.1016/J.RSE.2017.06.031

Mas, J.-F. (2018). *Análisis espacial con R : Usa R como un Sistema de Información Geográfica*. European Scientific Institute. https://eujournal.org/files/journals/1/books/JeanFrancoisMas.pdf

# Anexos

## Anexo 1: Configuración del proyecto ejemplo (demo)

Un proyecto ejemplo (demo) se ha generado como plantilla para futuros proyectos usando la herramienta dinoSoil-toolbox versión 0.1. Este demo usa las bases de datos entrada (perfiles, observaciones y coordenadas) así como los vectores cartográficos de clima y relieve del proyecto de política de tierras en Cesar/Magdalena. La única diferencia radica en el reemplazo del modelo digital de elevación (DEM) original con una resolución de 30 m por el provisto por CGIAR STRM v4 con resolución espacial de 90 m. Para acelerar la generación de resultados este fue remuestreado a 250 m y exportado en la proyección de MAGNA-SIRGAS con origen Bogotá (ESPG: 3116). Este proyecto guarda la estructura indicada en la **Sección 4**.

Los archivos del demo así como la herramienta son entregados como parte de este proyecto a los encargados de la subdirección (listados en la portada de este documento). El acceso a estos debe ser solicitado a estos profesionales.

Alternativamente, desde que el sistema de Sharepoint del proyecto tenga vigencia, en la siguiente ruta se puede descargar los archivos del demo (incluido el código fuente de la herramienta y manual de usuario):

https://igacoffice365.sharepoint.com/:f:/s/Mapeodigitaldesuelos/ElQhOoaC88ZNvu9gg3SRfUkBdeDHrS6sapv41gOMIUZZWQ?e=LeEUcR

El código fuente de la herramienta también se encuentra alojado en GitHub como repositorio privado. Si esta familiarizado con esta plataforma, favor ponerse en contacto con el administrador del repositorio a través del correo acocac@gmail.com.

# Anexo 2: Ejemplo de archivo CSV para codificar un raster agregado a las covariables

Si se agrega una covariable en formato raster de tipo categórico a la ruta datos > entrada > 1_covariables > raster esta debe estar acompañada por una tabla en formato CSV que contiene la codificación de los valores de los pixeles. La Figura 1 muestra un ejemplo de este archivo para el raster de la covariable clima, el cual fue procesado por la herramienta a partir de un archivo vector. El nombre del archivo debe ser el mismo usado que en el GeoTIFF para el correcto funcionamiento de la herramienta.

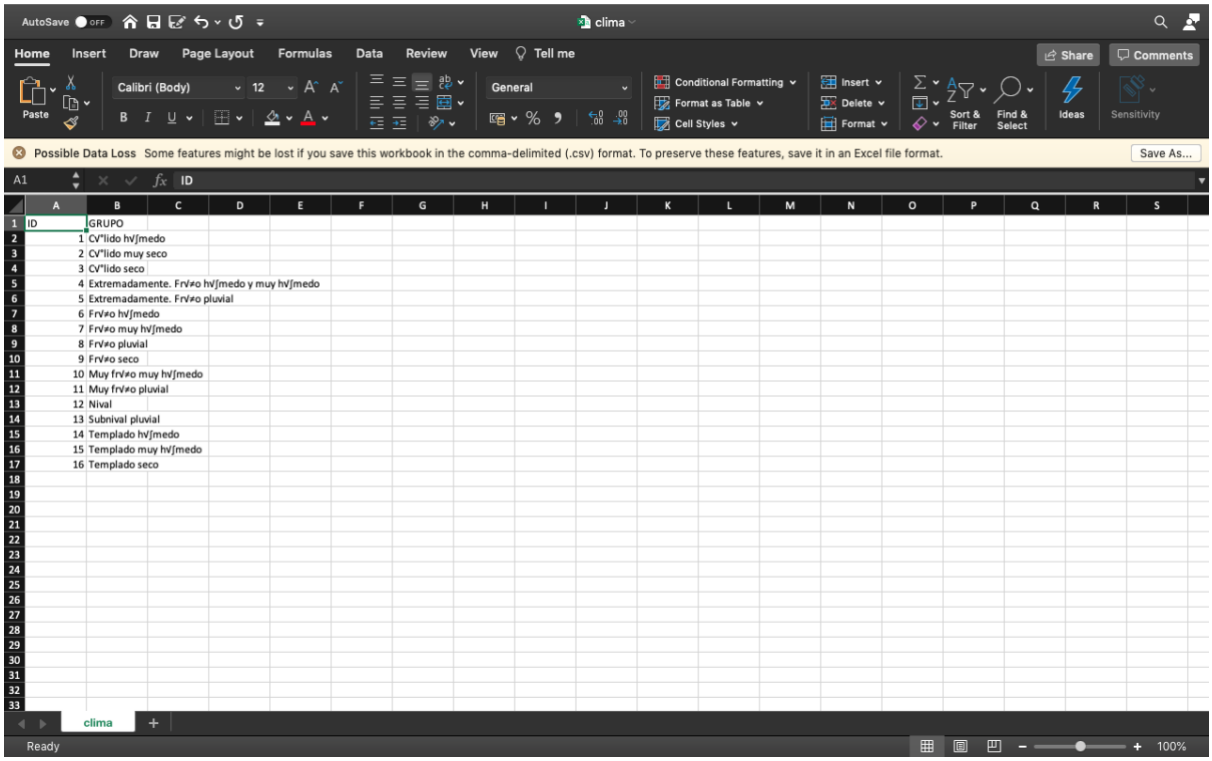



**Figura 1** Ejemplo del archivo CSV que contiene la codificación de los valores de los pixeles del archivo clima.tif.

# Anexo 3: Configuración manual de los modelos e importancia de variables

El código alojado en la ruta src > functions > 4_ConfigModelos.R contiene todo lo referido a la configuración de los modelos por defecto o config asi como la definición de funciones para estimar la importancia de las variables. Esto último es importante para aquellos modelos que los cuales la librería caret no tiene soporte para generar los gráficos de importancia de la variable. Por ejemplo, para el algoritmo ranger (Random Forest), fue necesario definir una función para graficar la importancia de las variables (ver `modelos.variables.importancia`). En caso de quererse añadir otro modelos estos se deben agregar en un condicional `else if` (ver más info en este enlace).

```
#############################################################################
# titulo        : Configuracion modelos y calculo de importancia de las covariables;
# proposito     : Configurar modelos por DEFECTO o llamados por CONFIG;
# autor(es)     : Preparado por Alejandro Coca-Castro (ACC), IGAC-CIAF;
# creacion      : Creado por ACC en Bogota, Colombia en Septiembre 2020; Actualizado por ACC
en Diciembre 2020;
# entrada       : N/A;
# salida        : Funciones;
# observaciones : Agregar a la funcion de importancia de las variables modelos que no tienen
esa opcion en la libreria caret;
#############################################################################

dict <- new.env(hash = TRUE)
Add <- function(key, val) dict[[key]] <- val

modelos.config.manual <- function(){
  modelos.lista <- c('J48', 'C5.0', 'ranger', 'svmLinear','multinom',
                     'xgbTree', 'gbm_h2o', 'glmnet', 'mlp', 'svmRadial','cubist')

  modelos.dict = mapply(Add, modelos.lista, modelos.lista)

  #opcion defecto: todos con un mismo tamano para busqueda de mejores hiperparametros
  tuneLenght_size <- rep(20, length(modelos.lista))
  tuneLenght = mapply(Add, modelos.lista, tuneLenght_size)

  ##opcion alternativa: tamano para busqueda de mejores hiperparametros por modelo
  # tuneLenght <- c('J48'=5, 'C5.0'=5, 'multinom','ranger'=20, 'svmLinear'=5, 'xgbTree'=20,
'gbm h2o'=3,
  #                 'glmnet'=5,'mlp'=5, 'svmRadial'=20)

  conflist = list(modelos.dict, tuneLenght)
  names(conflist) = c('modelos.dict','tuneLenght')

  return (conflist)

}

modelos.config.defecto <- function(){
  ##list models
  regression.models <- map(getModelInfo(), "type") %>%
    map_lgl(function(x) {
      any(x == "Regression")
```

```
    }) %>%
    {.[.]} %>%
    names() %>%
    sort()

  clasification.models <- map(getModelInfo(), "type") %>%
    map lgl(function(x) {
      any(x == "Classification")
    }) %>%
    {.[.]} %>%
    names() %>%
    sort()

  listmodels.varImp <- as.character(methods(varImp))
  listmodels.varImp <- gsub('^varImp.','', listmodels.varImp)
  listmodels.varImp <- listmodels.varImp[!listmodels.varImp %in%
c('bagEarth','bagFDA','earth',

'fda','gam','gbm','glm','JRip',

'PART','rpart','nnet','avNNet')]

  proyecto.modelos.categoricas <- clasification.models[which(clasification.models %in%
listmodels.varImp)]
  proyecto.modelos.continuas <- regression.models[which(regression.models %in%
listmodels.varImp)]

  proyecto.modelos.categoricas <- c(proyecto.modelos.categoricas,'ranger')
  proyecto.modelos.continuas <- c(proyecto.modelos.continuas,'ranger')

  dict <- new.env(hash = TRUE)
  Add <- function(key, val) dict[[key]] <- val

  proyecto.modelos <- unique(c(proyecto.modelos.categoricas, proyecto.modelos.continuas))
  tuneLenght size <- rep(20, length(proyecto.modelos))

  tuneLenght = mapply(Add, proyecto.modelos, tuneLenght_size)

  conflist = list(proyecto.modelos.continuas, proyecto.modelos.categoricas, tuneLenght)

  names(conflist) = c('modelos.continuas','modelos.categoricas','tuneLenght')

  return (conflist)
}

modelos.variables.importancia <- function(modelo,nombre){
  if (nombre == 'ranger'){

    finalModel <- modelo$finalModel

    #IMPORTANCIA VARIABLES
    imp <-as.vector(finalModel$variable.importance)
    variable <- names(finalModel$variable.importance)
    r <-data.frame(variable=variable,importance=imp)

    p <- ggplot(r, aes(x=reorder(variable,importance), y=importance,fill=importance))+
      geom_bar(stat="identity", position="dodge",fill = "darkgrey")+ coord_flip()+
      ylab("Importancia") +
      xlab("Covariable")+
      guides(fill=F) +
      theme_bw() +
      theme(text=element_text(size=18))

  } else{
    p <- NULL
  }

  return (p)
}
```

# Anexo 4: Posibles errores

### *Error: no es posible cambiar el directorio de trabajo*

Existen varias razones por las que quizás no seas capaz de cambiar el directorio de trabajo.

- Comprobar que no se haya escrito mal la ruta.
- Asegurarse de que el directorio no contenga caracteres inválidos, como acentos o espacios
- Asegurarse de que tienes permisos de administrador y/o escritura donde se almacenan los archivos de salida
- Utilizar la barra invertida doble (o la barra simple (/)

### *Error en la ejecución de algún modulo*

- Asegúrese de escribir bien los nombres indicados en la consulta de argumentos. AMBAS ≠ ambas
- Los módulos se deben ejecutar de manera secuencial ya que sus resultados están vinculados entre si.
- Cuando pregunta el número de covariables elegidas, asegúrese que corresponde en la elegida desde el paso de Exploración.

### *Error en paquete:*

- En caso de error con alguna librería, se sugiere instalarla con install.packages('libreria')