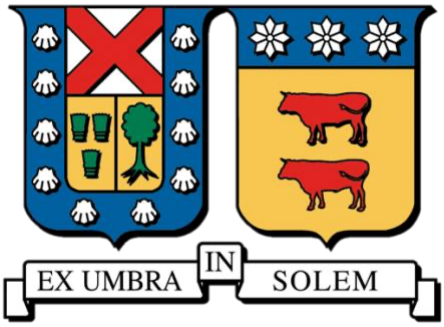
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INGENIERÍA COMERICAL




# Evaluación de modelos estadísticos y de aprendizaje automático para la estimación de propiedades del suelo en función de índices espectrales satelitales.

**Memoria presentada por**
**Nicolás Valdés Astargo**
**INGENIERÍA COMERCIAL**
**MARZO 2026**

**Profesor guía: Rodrigo Ortega Blu**

## CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

**1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO**

**Tipo de monografía (marcar una opción):** ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

**Título del trabajo: _Evaluación de modelos estadísticos y de aprendizaje automático para la estimación de propiedades del suelo en función de índices espectrales satelitales.** _____

**Nombre del candidato(a): _Nicolás Eduardo Valdés Astargo** _____

**Carrera / Grado: ___Ingeniería Comercial** _____

**Campus:** ____Vitacura_____ **Departamento:** __Ingenieria Comercial _____

**2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS**

Yo, Rodrigo Ortega Blu _____, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

**3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)**

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

**Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):**

_____

_____

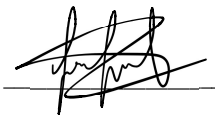
_____

**4.- FIRMAS**

**Profesor(a) guía o director(a) de memoria o tesis:**

**Fecha:** 5 de mayo 2026 **Firma:** 

**Estudiante o Candidato(a):**

**Fecha:** 5 de mayo 2026 **Firma:** 

**Agradecimientos**

Primero que todo quiero agradecer a Dios por darme la fuerza y sabiduría para afrontar estos años de estudio, con distintas adversidades y piedras en el camino que sin duda fueron grandes enseñanzas y experiencias.

Agradecer a mis padres, Sabina y Sergio quienes siempre me apoyaron, motivaron y acompañaron en cada situación siempre con mensajes emotivos y correctos ante cada situación. A mis hermanos Cristóbal y Valentina, quienes siempre me ayudaron de una u otra forma, ya sea con distracciones, trabajos, motivación, escuchar.

Agradecer a mis amigos de la universidad, quienes eran mi motivación de asistir a la universidad, verlos, estudiar juntos, compartir con ellos era lo mejor de ir, mencionar a quienes tuvieron este rol en mis años de estudio Daniel Parra, David Useche, Felipe Andueza, Fernanda Recabarren, Consuelo del Solar, Sofía Belmar, Cristóbal Lazo, Daniel Urrejola, Silvana Veliz, María José Vidal e Isidora Lobos, gracias a cada uno de ellos.

Agradecer a mis amigos de siempre, quienes me ayudaban con sus conversaciones, apoyaban en cada problema que tenía, con quienes me podía distraer y hablar de lo que sea, gracias Benjamín León, Vicente Navarro y José Urtasun.

Agradecer a mi profesor guía Rodrigo Ortega, quien tuvo toda la disposición de ayudarme con el proyecto, dándome ideas, siempre con la mejor voluntad y enseñando, gracias profesor por el tema que fue sumamente interesante y distinto.

Por último agradecer a toda persona que me ayudó durante mi vida universitaria.

## Índice

## 1. Introducción

La calidad del suelo es uno de los pilares fundamentales de la productividad agrícola y de la sostenibilidad de los agroecosistemas. Las decisiones relacionadas con fertilización, riego y manejo de cultivos dependen en gran medida, de un conocimiento adecuado del estado físico y químico del suelo. Tradicionalmente, esta información se ha obtenido mediante análisis de laboratorio, los cuales, aunque son precisos, implican altos costos, tiempos prolongados y cobertura espacial limitada.

En este contexto, las tecnologías de teledetección satelital han abierto nuevas posibilidades para el monitoreo del suelo y la vegetación, permitiendo evaluar grandes extensiones de terrenos de manera periódica, no invasiva y a bajo costo. Particularmente los índices espectrales como el NDVI (Normalized Difference Vegetation Index) y el SAVI (Soil-Adjusted Vegetation Index), han demostrado ser útiles para representar el vigor de la vegetación, la cobertura del suelo y otras variables relacionadas con la salud de los cultivos. No obstante, aún existe un vacío en la literatura respecto a la posibilidad de utilizar estos índices para estimar variables edáficas, lo que permite revertir el enfoque tradicional de estimación a aplicar técnicas de inteligencia artificial para anticipar las condiciones del suelo en función de la respuesta espectral de la vegetación y/o del suelo.

El desarrollo de modelos predictivos que permiten estimar variables fisicoquímicas del suelo a partir de índices espectrales representa un avance significativo en la agricultura de precisión, ya que podría facilitar la toma de decisiones agronómicas basada en datos, optimizando la asignación de recursos y reduciendo los costos operacionales. Además, el uso de herramientas como QGIS y RStudio, junto con imágenes satelitales gratuitas como Sentinel-2, hace que esta propuesta sea escalable, replicable y aplicable en distintos territorios agrícolas, incluso en contextos con recursos limitados.

En este sentido, la presente investigación busca aportar evidencia empírica y una metodología integrada que combine imágenes satelitales, análisis de suelo, procesamiento geoespacial y

modelos de predicción, contribuyendo al desarrollo de soluciones más eficientes y sostenibles para el monitoreo de la calidad del suelo en la agricultura moderna.

## 2. Estado del Arte

Un suelo de buena calidad permite la producción de alimento de calidad, lo que es esencial para la nutrición humana y animal. También un suelo de buena calidad contribuye a la sostenibilidad agrícola al mejorar la calidad del agua, aumentar la biodiversidad y mitigar el cambio climático (Nix, año).

*Nix, J. (s/f). Impacto de la Salud del Suelo en la Sostenibilidad Agrícola.*

### Fuentes de datos satelitales

Existen numerosos satélites que adquieren datos multiespectrales desde el espacio, siendo los más conocidos Sentinel-2, Landsat 8, Airbus Spot 6 y Pléiades.

Las imágenes obtenidas tienen diferente resolución espacial, Landsat brinda datos con una resolución espacial de 30 metros, Sentinel-2 de 10 metros, Spot 6 hasta 1,5 metros y Pléiades hasta 50 centímetros.

La resolución temporal para Landsat 8 es cada 16 días, mientras que para Sentinel-2 es cada 3/5 días dependiendo de la zona, Spot 6 y Pléiades se pueden solicitar según las necesidades.

A través del portal Copernicus, se tiene acceso a los datos del satélite Sentinel-2 y los integra para proporcionar índices de vegetación a los usuarios.

Las bandas Sentinel-2 se procesan para calcular múltiples índices de vigor, estrés hídrico y clorofila. Se proporcionan índices para las fechas que estén disponibles, excluyendo automáticamente las imágenes con nubosidad.

### Índices de Vegetación

Al momento de hablar de imágenes satelitales, es necesario introducir el concepto de índice de vegetación para entender cómo se hace el seguimiento remoto a la salud de los cultivos.

Los índices de vegetación son una herramientas claves para la agricultura inteligente, el uso de datos satelitales y su correcta interpretación optimiza las intervenciones en el campo y

hacen que una actividad estructurada de exploración de cultivos sea sostenible desde el punto de vista económico.

*Tipos de Índices de Vegetación*

Existe una gran diversidad de índices de vegetación (Sgargi, 2022), siendo los más comunes:

NDVI: Permite evaluar el estado de salud de la vegetación y muestra las diferencias en el vigor de las plantas analizando la reflectancia de la vegetación en las bandas Roja y NIR.

SAVI: Permite evaluar las condiciones de desarrollo de la vegetación en las fases de emergencia y desarrollo temprano, ya que aplica una corrección al suelo.

LAI: Índice de área foliar que se correlaciona con la superficie foliar de la planta expresada en $m^2$ por $m^2$ derivado del índice EVI.

TCARI/OSAVI: Índice de clorofila que permite identificar posibles zonas cloróticas dentro del campo y obtener una visión general del estado nutricional de las plantas.

WDRVI: Analiza la salud de la vegetación y es especialmente útil cuando la vegetación está bien desarrollada y es exuberante.

GNDVI (Green NDVI): Es un índice de vigor adicional que reduce el efecto de saturación cuando la vegetación está especialmente desarrollada.

NDMI: Índice específico que evalúa el contenido de agua de la vegetación, por lo tanto utilizable solo con vegetación desarrollada.

NMDI: Puede usarse para evaluar el contenido de agua del suelo, en el caso del suelo desnudo, un valor de índice alto indica suelo seco. En presencia de vegetación, un valor alto del índice indica que la planta no está bajo estrés hídrico.

Los 4 pasos principales para la interpretación de los índices de vegetación permiten el análisis multitemporal y la comparación entre índices:

1) Determinar el estado fenológico de la planta.
2) El análisis de la tendencia histórica de los índices para evaluar si existen anomalías y si están relacionadas con fenómenos conocidos.
3) La identificación de los índices a comparar.
4) La comparación entre índices para identificar las áreas problemáticas a verificar en el campo.

*Sgargi, C. (2022, mayo 2). Imágenes de satélite para la agricultura.*

## QGIS

QGIS es un sistema de información geográfica de código abierto que permite visualizar, editar y analizar datos geoespaciales.
QGIS permite trabajar con una amplia gama de formatos de datos geográficos, incluyendo datos vectoriales (líneas, puntos, polígonos), datos ráster (imágenes satelitales, mapas de elevación) y bases de datos espaciales (CITA).
*Mergin Maps en directo. (s/f).*

## RStudio

Es un entorno de desarrollo integrado (IDE) de código abierto, que se utiliza principalmente para trabajar con el lenguaje de programación R. Es una herramienta que facilita la programación, el análisis de datos y la creación de visualizaciones graficas (CITA).
*IBM watsonx as a Service. (2025, junio 19).*

QGIS se utiliza para la visualización de datos espaciales, mientras que RStudio se pueden usar para análisis estadísticos y de código personalizados que luego se pueden aplicar en

QGIS. La integración se facilita a través de plugins como el Processing R Provider en QGIS, que permite ejecutar un código R dentro de QGIS.

**Casos Similares**

A nivel nacional el servicio aerofotogramétrico (SAF) de la Fuerza Aérea de Chile, desarrolló el estudio "Uso de imágenes satelitales para el análisis y generación de reportes automáticos en zonas de sequias en el territorio nacional".

El SAF cuenta con un departamento donde se trabajan imágenes satélites y aerofotogramétricas. Con estas se pueden determinar índices de vegetación y por tanto de sequía en el territorio captado.
Se desarrolló Varda, un prototipo de software que trabaja con imágenes satelitales para obtener índices de vegetación de un periodo específico por un operador.

Las imágenes fueron obtenidas mediante los satélites Sentinel 2, a través de la plataforma Google Earth Engine, ya que cuenta con imágenes de distintos sensores en una sola plataforma y de forma libre.

Ya cuando se tenían estas imágenes, eran procesadas para obtener el índice de vegetación específicamente Normalized Difference Vegetation Index (NDVI) y Soil Adjusted Vegetation Index (SAVI) y un NDVI categórico, el cual permite clasificar el terreno estudiado.

También se utilizó la Application Programming Interface (API) del sitio OpenWeather, para obtener índices meteorológicos de interés para tratar la sequía, más en concreto el porcentaje de humedad, velocidad del viento y temperatura.

Ya con cada uno de los datos obtenidos, se usan para generar un reporte automático. Esto permite estudiar varias zonas de sequía a la vez, teniendo una visión más general del territorio nacional.

Para validar el funcionamiento de este software Varda, se utilizó System Usability Scale (SUS) a funcionarios del SAF, quienes testearon y comentaron su experiencia.

Otro caso similar, fue un estudio desarrollado en el marco del proyecto AgroLens, se propuso una metodología de aprendizaje automático para predecir parámetros del suelo a partir de imágenes satelitales, datos meteorológicos, tasas de rendimiento agrícola y embeddings generados por un modelo fundacional de observación terrestre. La investigación utilizó un conjunto de datos "Lucas 2018 TOPSOIL" (base de datos que reúne aproximadamente 19.000 muestras de suelo tomadas en la Unión Europea y el Reino Unido) e imágenes Sentinel-2, con el objetivo de estimar variables como pH, fósforo, nitrógeno y potasio sin depender exclusivamente de análisis de laboratorio. Para este estudio se llevaron a cabo distintos modelos de regresión, entre ellos XGBoost, Random Forest y redes neuronales. Los resultados mostraron que los modelos extendidos, especialmente aquellos que incorporan píxeles vecinos, clima y rendimiento de cultivos, lograron reducir el error de predicción , destacando un mejor desempeño para las variables fósforo, nitrógeno y potasio, y en Random Forest para pH. El estudio igual presentó limitaciones importantes, particularmente en África, debido a la escasez de datos. Este trabajo, confirma el potencial del uso combinado de teledetección y aprendizaje automático para la estimación de propiedades del suelo, aunque también evidencia que la calidad, disponibilidad y temporalidad de datos siguen siendo factores críticos para la robustez de los modelos.
*AGROLENS PROJECT et al. (2025)*

Un caso similar es el estudio en una zona árida del centro de Irán, donde se utilizaron imágenes Landsat 8, variables topográficas de 96 muestras de suelo para predecir propiedades químicas como EC, pH, pOH, carbonato, bicarbonato, sodio y cloro. Se compararon modelos como GLM, GAM, CART, SVM, Random Forest y un ensemble, observándose que este ultimo alcanza el mejor desempeño predictivo, seguido por Random Forest. El estudio concluye que la teledetección y aprendizaje automático es una herramienta eficaz para el mapeo digital de suelo en ambientes áridos.
*Molaeinasab et al. (2025)*

Otro caso similar es un estudio realizado en el norte de Irán, donde se emplearon 317 muestras del suelo, imágenes Landsat 8, variables topográficas y climáticas para predecir propiedades físicas y químicas del suelo mediante modelos de Deep Learning. Se compararon arquitecturas CNN, RNN y CNN-RNN, observándose que el modelo híbrido CNN-RNN obtuvo mejor rendimiento global. El estudio concluye que la integración de datos de teledetección, relieve y clima mejora la precisión en la cartografía digital de suelos y permite representar mejor la variabilidad espacial de sus propiedades.
*Hosseini et al. (2023)*

**Brechas y Oportunidades**

Según Riquelme (2023), existen numerosas brechas y oportunidades que pueden llenarse a través del uso de tecnologías satelitales, entre ellas:

*Falta de información sobre la salud del suelo*

No se tiene una comprensión completa de las propiedades del suelo y cómo se relacionan con productividad agrícola, la biodiversidad y la resistencia a factores ambientales.
Cambio de uso de suelo, como la expansión de áreas urbanas o agrícolas, puede tener impactos negativos en la biodiversidad y la provisión de servicios ecosistémicos.

*Mejora de la productividad agrícola*

Al tener mayor conocimiento del suelo permite optimizar la fertilización, la gestión del agua y la selección de cultivos, lo que contribuye a aumentar la producción y rentabilidad.
Promoción de la inversión en tecnologías innovadoras, la aplicación de tecnologías de diagnóstico de suelos y la automatización de procesos puede mejorar la eficiencia y sostenibilidad de la agricultura.
*Riquelme Adriasola, D. A. (2023).*

## 3. Hipótesis

Las propiedades fisicoquímicas del suelo, específicamente el pH, materia orgánica (OM), el contenido de potasio (K) y fósforo (P), pueden ser estimadas a partir de índices espectrales NDVI y SAVI satelitales, mediante modelos estadísticos y de aprendizaje automático, siendo los modelos no lineales, como Random Forest, más efectivos que los lineales para capturar estas relaciones.

La estimación remota de propiedades edáficas puede contribuir a una mejor asignación de recursos en la toma de decisiones agrícolas, reduciendo costos operacionales y optimización de prácticas de manejo del suelo.

## 4. Objetivos

### 4.1. Objetivo General

Evaluar la capacidad predictiva de los índices espectrales NDVI y SAVI sobre las variables fisicoquímicas del suelo (pH, materia orgánica, potasio y fosforo), mediante modelos estadísticos y de aprendizaje automático, como base para la estimación remota de la calidad del suelo y su aplicación en la toma de decisiones agrícolas.

### 4.2 Objetivos Específicos

- Construir una base de georreferencia de datos del suelo a partir de puntos de muestreo y capas satelitales, integrando variables edáficas y valores de NDVI y SAVI mediante herramientas SIG como QGIS.

- Aplicar modelos de regresión lineal y random forest para predecir las variables del suelo (pH, OM, K y P) a partir los valores de NDVI y SAVI.

- Comparar el desempeño de los modelos predictivos, evaluando métricas como $R^2$, RMSE y MAE, y analizando el impacto de los índices estudiados, NDVI y SAVI, y del tipo de modelo, lineal o no lineal.

- Discutir el potencial uso agronómico de los modelos generados, destacando su utilidad para estimar remotamente la calidad del suelo, optimizar el monitoreo agrícola y reducir costos operativos en la toma de decisiones.

## 5. Materiales y Métodos

### 5.1 Análisis utilizando el índice NDVI

Para llevar a cabo este proyecto se realizaron diversos procedimientos para lograr obtener los datos exactos con los que se trabajó.

Antes de iniciar con la recopilación de datos, se descargó los dos principales programas que se llevarían a cabo en el proceso, estos programas fueron QGIS y RStudio.

Primero se recolectaron datos de 25 sitios distintos. Los datos fueron compartidos por la empresa Neoag Agricultura de Precisión (www.neoag.cl). Los datos entregados de cada sitio, correspondientes a distintos cultivos, incluían, por una parte el sitio en forma de perímetro, donde se mostraba una carpeta con cinco archivos correspondiente a ESRI shape, donde solo nos interesaba el archivo en formato shapefile (.shp), este archivo para poder ser visualizado debía cargarse en el programa QGIS y por otra parte había un archivo donde se incluían las coordenadas geográficas de los puntos de muestreo y sus respectivas mediciones de propiedades del suelo, este archivo fue fundamental para la georeferenciación y el análisis espacial en conjunto con imágenes satelitales, proceso que se explicará más adelante.

Lo primero que se hizo fue cargar el archivo shapefile en el programa QGIS, como una capa vectorial. Una vez cargado el archivo, se observó la geometría correspondiente al perímetro exacto del sitio que se investigó. Con esto fue posible visualizar la zona de estudio de manera georreferenciada, asegurar que todos los análisis posteriores se aplicaran exclusivamente dentro de los límites del ensayo.
*(La figura 1 muestra cómo añadir Shapefile.)*
*(La figura 2 muestra cómo se vería Shapefile en QGIS.)*

El siguiente paso fue la obtención de imágenes satelitales del sitio que se estaba investigando, para lo que se utilizó la plataforma Copernicus browser, para usar esta página web, se debe iniciar sesión, en el caso de no tener cuenta hay que registrarse, con el objetivo de obtener una descarga precisa de imágenes satelitales desde la plataforma Copernicus Browser, se generó un archivo en formato KML (Keyhole Markup Language) la cual delimita el área de estudio. Este archivo fue exportado directamente desde QGIS, a partir del shapefile que contenía el polígono del perímetro del sitio.

*(La figura 3 muestra cómo exportar Shapefile.)*

*(La figura 4 muestra qué formato dar al archivo.)*

Una vez cargado el archivo shapefile en QGIS y se verificó que el polígono representara correctamente los límites del área de estudio, a través de la opción exportar, guardar como, se selecciona el formato que se requería en este caso usamos KML y se guardaba, este archivo KML generado fue utilizado para cargar la zona de estudio en la plataforma Copernicus, donde se buscaron imágenes de una fecha adecuada para el estudio, en este caso se optó por el día 23 de enero del año 2022, de esta forma obteníamos una imagen clara y precisa.

*(La figura 5 muestra cómo se cargaría KML en Copernicus.)*

*Copernicus Browser. (s/f)*

Al realizar este proceso se obtuvieron varias imágenes satelitales sentinel-2, estas imágenes fueron descargadas en forma de banda, donde se obtuvieron bandas 1, 2, 3, 4, 5, 6, 7, 8, 9,11 y 12. Para nuestro estudio se utilizó la banda 4 (B04) y la banda 8 (B08), donde la banda 4 corresponde a banda roja (Red) y la banda 8 corresponde a banda infrarrojo cercano (NIR), estas bandas fueron añadidas en QGIS, como capa ráster.

*(La figura 6 muestra cómo descargar las bandas solicitadas.)*

Para este estudio se quería identificar el NDVI (Índice de Vegetación de Diferencia Normalizada), para identificar el NDVI, se hizo mediante la herramienta calculadora ráster en QGIS, en esta calculadora se utilizó la formula NDVI = (B08 – B04) / (B08 + B04).

*(La figura 7 muestra cómo cargar las bandas.)*

*(La figura 8 muestra cómo se vería en QGIS.)*

*(La figura 9 muestra la calculadora ráster con su respectiva fórmula de NDVI.)*

El resultado fue un ráster NDVI, lo que permitió evaluar espacialmente la salud y vigor de la vegetación en el área de estudio, este ráster fue recortado utilizando el polígono del sitio (archivo shapefile) para asegurar que el análisis se limitara exclusivamente al sitio de estudio.

Posterior a esto se cargó el archivo que incluía las coordenadas geográficas de los puntos de interés, junto con las variables fisicoquímicas del suelo recolectadas en terreno.
*(La figura 10 muestra cómo cargar el archivo.)*

La carga se realizó mediante la opción de agregar texto delimitado, con un sistema de referencia espacial adecuado, en este caso fue de EPSG:9056 – WGS 84 (G1674). Una vez proyectados, los puntos quedaron superpuestos al polígono del área de estudio y las bandas espectrales previamente cargadas.
*(La figura 11 muestra cómo se vería en QGIS.)*

Este paso tuvo como propósito ubicar espacialmente los sitios de muestreo sobre las imágenes satelitales y dentro del límite del sitio, permitiendo verificar su correcta distribución y asegurar la información geográfica entre las mediciones de campo y la información satelital.

Una vez visualizados en QGIS los puntos de muestreo del archivo con las coordenadas, se generaron zonas de influencia mediante la herramienta “BUFFER”, con el propósito de representar espacialmente el entorno inmediato de cada punto.

Este procedimiento se realizó a través del menú vectorial, herramientas de geoproceso, BUFFER, donde se hizo con la configuración:

Capa de entrada: Archivo con las coordenadas y variables fisicoquímicas.
Distancia de 0,00003 grados, equivalente a 3 metros, aproximadamente, en coordenadas geográficas.

Segmentos por cuadrante: 12, para lograr una geometría más suavizada y circular.
Estilo de terminación y de ángulos: Redondo.

El resultado obtenido fue una capa con polígonos circulares de aproximadamente 3 metros de radio, centrados en cada punto de muestreo. Esta representación espacial permitió delimitar un área homogénea alrededor de cada muestra, facilitando su inspección visual y su ubicación dentro del polígono de estudio.
*(Figura 12 muestra la configuración utilizada en QGIS para generar los BUFFERS)*
*(La figura 13 muestra cómo se vería en QGIS.)*

Posteriormente, se utilizó la herramienta "Píxeles ráster a puntos" en QGIS, con el fin de transformar el ráster generado a partir del índice NDVI en una capa vectorial de puntos. Esta operación permite representar cada pixel del ráster como un punto con valor asociado, lo que facilita su análisis estadístico y espacial en otras plataformas con RStudio.

Esta herramienta fue configurada con los siguientes parámetros:

Capa ráster: Cálculo de NDVI, obtenido mediante la calculadora ráster.
Número de Bandas: Banda 1 (Gray), correspondiente a las bandas 4 y 8.
Nombre del campo: Value, que representa el valor numérico del NDVI asociado a cada píxel.
Salida: Capa temporal de puntos vectoriales, generada automáticamente tras la ejecución del algoritmo.

Esta transformación resultó en una capa de puntos en la que cada entidad representa el centro de un píxel de ráster original, con su valor de NDVI correspondiente. Esta capa fue fundamental para permitir el cruce con otras capas vectoriales, como los BUFFERS generados en torno a los puntos de muestreo.
*(La figura 14 muestra la configuración utilizada en la herramienta "Pixeles Ráster a Puntos" en QGIS.)*
*(La figura 15 muestra cómo se vería en QGIS.)*

Con el objetivo de vincular los valores de NDVI a las zonas de influencia definidas alrededor de los puntos de muestreo, se aplicó la herramienta intersección en QGIS. Esta operación permitió extraer únicamente aquellos píxeles (convertidos previamente a puntos vectoriales) que se encontraban dentro de los BUFFERS de 3 metros generados en etapas anteriores.

La herramienta intersección fue configurada de la siguiente forma:

Capa entrada: Puntos vectoriales, correspondiente a los puntos generados a partir del ráster NDVI.
Capa de superposición: Hecho BUFFER, que contenía los polígonos de 3 metros alrededor de cada punto de muestreo.
Salida: Se creó una capa temporal con los resultados de intersección.

Este procedimiento permitió conservar solo los puntos que coincidan espacialmente con las zonas de muestreo, asignándoles además los atributos de las capas involucradas.

El resultado fue una capa de puntos con valores de NDVI espacialmente coincidentes con la muestra del suelo, lista para ser exportada y utilizada en el análisis estadístico posterior en RStudio.
*(La figura 16 muestra la configuración utilizada para realizar la intersección en QGIS.)*

Una vez obtenida la capa resultante de la intersección entre los puntos vectoriales (con valores de NDVI) y los BUFFERS de muestreo, se procedió a exportar la información en formato Excel.

El procedimiento de exportación se llevó a cabo de la siguiente forma:

Se selecciona la capa intersección, se seleccionó exportar, guardar objeto como y ahí se abre una pestaña, en ella se seleccionó el formato Hoja de cálculo de MS Office Open XML (XLSX), de esta forma se guardara como un archivo Excel.
*(La figura 17 muestra el proceso de exportación)*

*(La figura 18 muestra el formato para guardar el archivo.)*

Debido a que los 25 sitios de muestreo correspondían a distintos cultivos, con diferentes valores de NDVI, los valores de NDVI no eran directamente comparables entre sitios. Para permitir un análisis conjunto y homogéneo, se aplicó una normalización por valor máximo a los valores de NDVI.

La fórmula utilizada fue: NDVI_Observado / NDVI_Máximo_del_Sitio

Esta transformación provocó que cada NDVI de cada sitio estaba en una escala de 0 a 1, donde 1 representaba el valor máximo observado dentro del sitio respectivo. De esta forma se logró estandarizar la magnitud del índice entre sitios, sin eliminar la variabilidad interna, facilitando la comparación y modelamiento conjunto de los datos.

El procedimiento completo desde el archivo shapefile hasta la estandarización de NDVIs fue repetido para cada uno de los 25 sitios incluidos en el estudio.

Cada sitio fue tratado como una unidad de análisis independiente, respetando su geometría específica y utilizando su propio conjunto de coordenadas y ráster de NDVI. Luego de realizar la estandarización mencionada a cada sitio, se preparó una base de datos, donde se incluía un promedio por zona de cada sitio de las variables a estudiar (NDVI, pH, K, P y MO), donde para el sitio 1 hay 6 zonas, donde se obtuvo un promedio de todas las mediciones por zona para cada una de las variables, este proceso se repitió con todos los sitios, separados por hoja de Excel, mostrando en una hoja toda la información necesaria del sitio y así con los 25 sitios.

Este archivo consolidado facilitó la revisión, control de calidad y el posterior análisis estadístico en RStudio, permitiendo importar los datos de manera estructurada y consistente.

Una vez consolidados los datos por sitio en un archivo Excel, se procedió a realizar el análisis estadístico en RStudio, comenzando con la evaluación de la relación entre propiedades del suelo y los índices de vegetación obtenidos mediante imágenes satelitales.

En primera instancia se analizó la relación entre el pH (nivel de acidez) del suelo y NDVI normalizado, utilizando un modelo de regresión lineal simple. El objetivo fue determinar si

existía una asociación significativa entre ambas variables, considerando que el NDVI puede reflejar condiciones edáficas que influyen en la productividad vegetal.

La relación fue expresada como: pH = $f$(NDVI_normalizado) = β0 + β1 * NDVI

El modelo permitió estimar la capacidad predictiva del NDVI sobre el pH. También se calcularon las métricas de ajuste como $R^2$, el error cuadrático medio (RMSE) y el error absoluto medio (MAE).

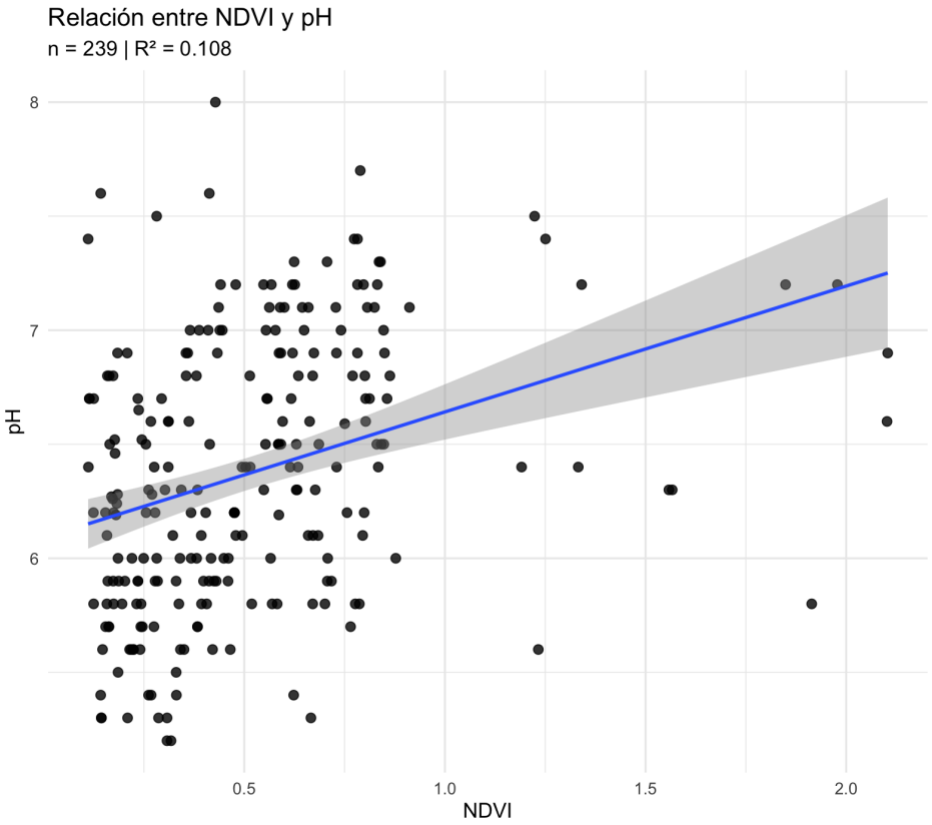



*Figura 1. Relación entre NDVI estandarizado y pH del suelo (Regresión Lineal)*

Con el objetivo de evaluar la capacidad predictiva no lineal del pH sobre el índice NDVI del suelo, se aplicó un modelo de Random Forest en RStudio. Este enfoque permitió modelar relaciones más complejas entre variables sin requerir supuestos de linealidad, siendo especialmente útil en contextos donde los procesos naturales no siguen una estructura estrictamente lineal. También se registraron valores de $R^2$, RMSE y MAE.

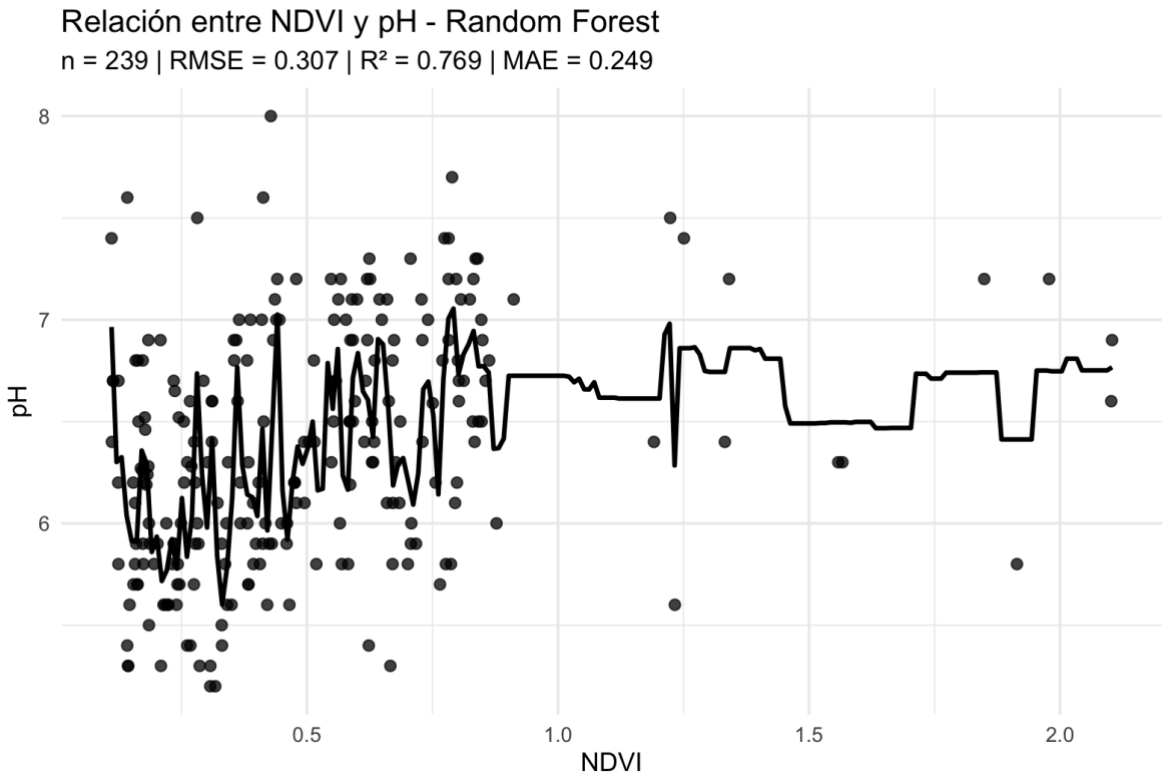



*Figura 2. Relación entre NDVI estandarizado y pH del suelo (Random Forest)*

Para la validación del modelo se utilizó el procedimiento automático validación cruzada en RF (10 fold).

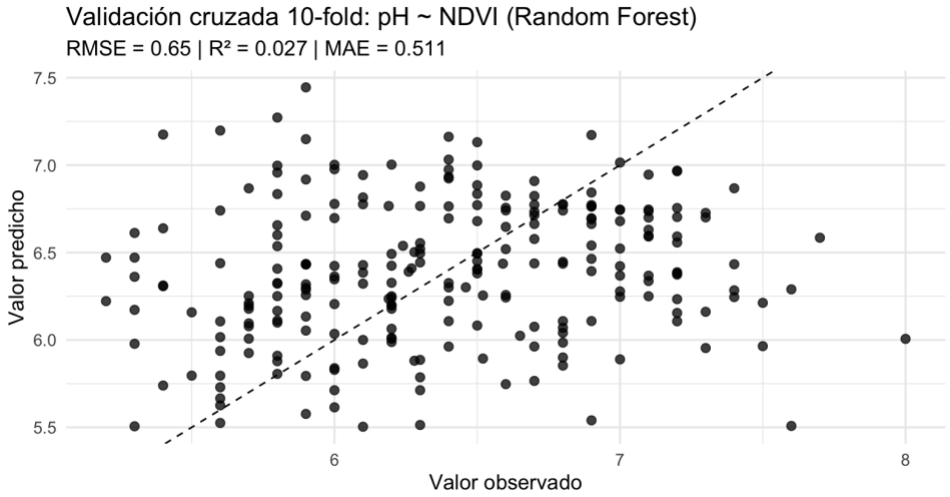



*Figura 3. Relación entre NDVI estandarizado y pH del suelo (validación cruzada)*

Este procedimiento se repitió con cada una de las variables de estudio K, P y MO.

*(Gráficos para cada variable en anexos)*

## 5.2 Análisis utilizando el índice SAVI

Además del índice NDVI, se aplicó el mismo procedimiento metodológico utilizando el índice SAVI (Índice de Vegetación Ajustado al Suelo). Este índice es particularmente útil en zonas agrícolas o semiáridas donde el suelo desnudo puede influir significativamente en su reflectancia.

Para su cálculo, se utilizaron las mismas bandas espectrales del satélite Sentinel-2, la banda 8 (NIR) y la banda 4 (RED).

La fórmula que utilizó en la calculadora ráster de QGIS fue:

SAVI = ((B8 – B4) / (B8 + B4 + L)) * (1 + L)
*(La figura 19 muestra la calculadora ráster con su respectiva fórmula de SAVI.)*

En esta fórmula L es igual a 0,5, siendo este un factor de corrección estándar para áreas con cobertura vegetal intermedia.

Una vez generado el ráster SAVI, se repitieron exactamente los mismos pasos metodológicos aplicados anteriormente con NDVI.

Creación de BUFFERS alrededor de los puntos de muestreo, conversión del ráster de SAVI a puntos vectoriales en QGIS, exportación de los puntos interceptados con valores de SAVI a un archivo Excel, consolidación de los 25 sitios en un único archivo con una hoja por sitio y la aplicación de modelos de regresión lineal y Random Forest en RStudio para predecir SAVI a partir de las variables fisicoquímicas del suelo (pH, OM, K y P).

En este caso el modelo fue representado como:
Variable del suelo = $f(\text{SAVI\_normalizado}) = \beta 0 + \beta 1 * \text{SAVI}$

Primero se evaluó la relación entre el índice SAVI y el pH, en un modelo de regresión lineal. Este análisis permitió determinar si el índice SAVI puede ser un predictor confiable de pH del suelo.

La relación fue expresada como: pH = $f$(SAVI_normalizado) = β0 + β1 * SAVI

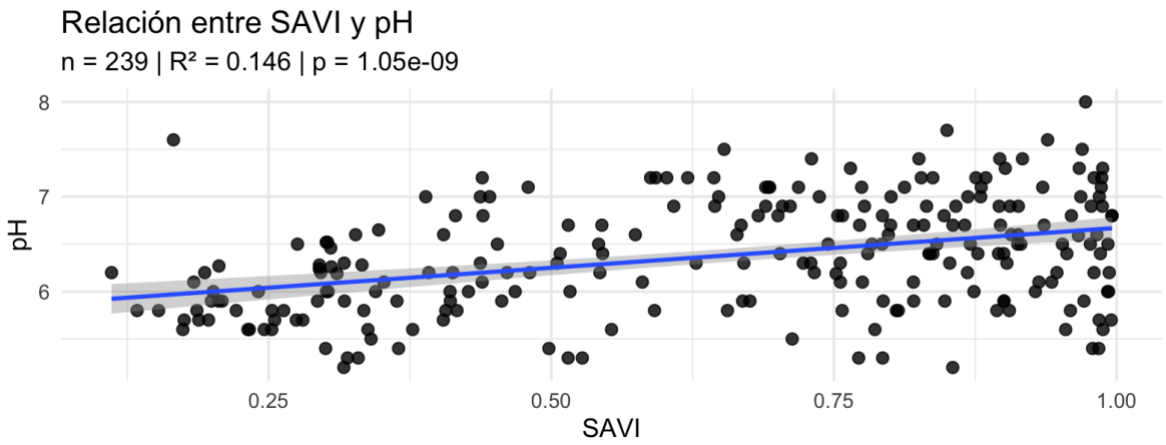



*Figura 4. Relación entre SAVI estandarizado y pH del suelo (Regresión Lineal)*

Para capturar posibles relaciones no lineales entre el índice SAVI y la variable pH, se ajustó un modelo de Random Forest. Este enfoque permite detectar patrones complejos sin necesidad de asumir una forma funcional específica.

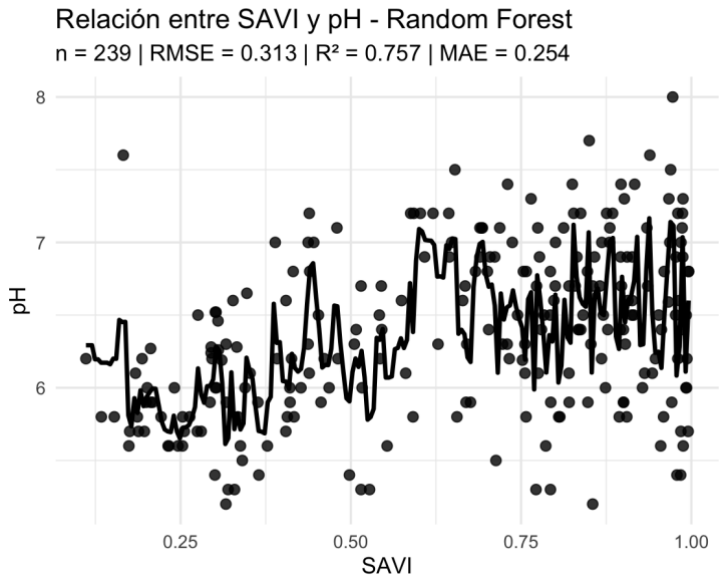



*Figura 5. Relación entre SAVI estandarizado y pH del suelo (Random Forest)*

Para la validación del modelo se utilizó el procedimiento automático validación cruzada en RF (10 fold).

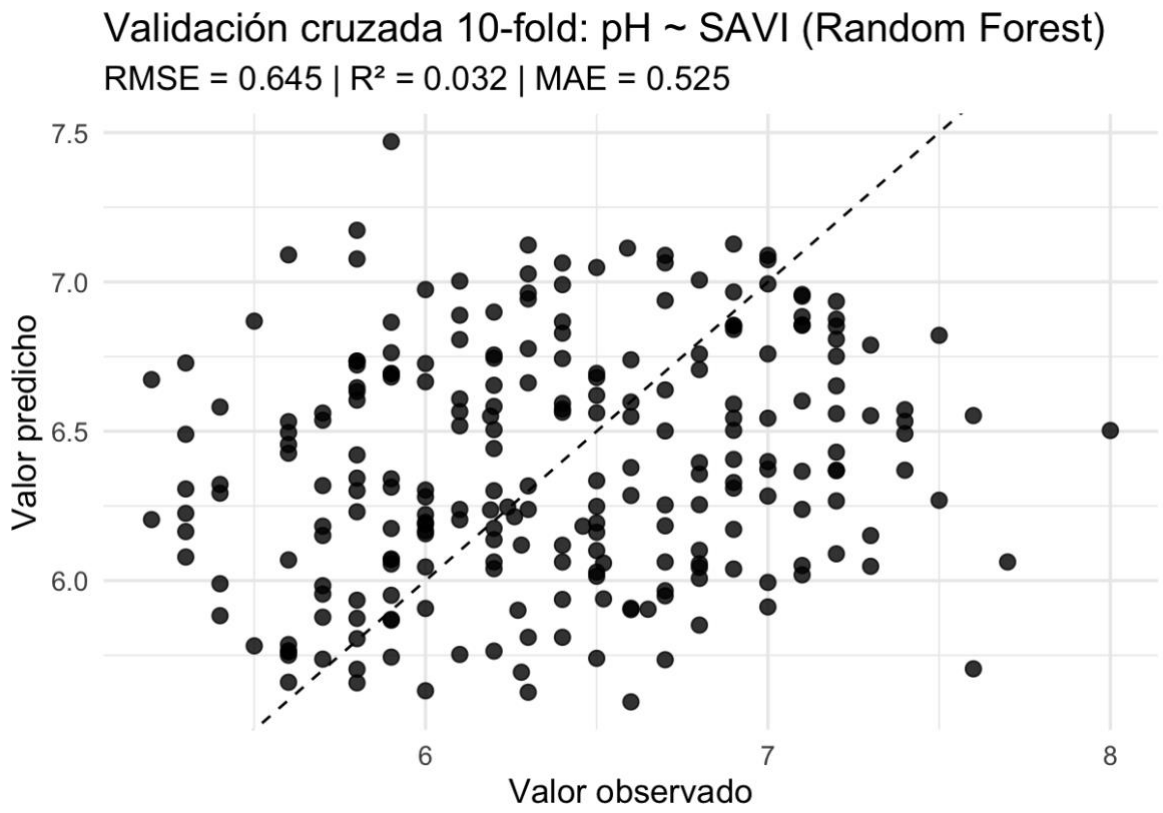



*Figura 6. Relación entre SAVI estandarizado y pH del suelo (Cross-validation)*

Este procedimiento se repitió con cada una de variables de estudio K, P y MO.

*(Gráficos para cada variable en anexos)*

## 5.3 Análisis utilizando índices NDVI y SAVI

Por último se utilizaron ambos índices, se juntaron ambas bases de datos, de NDVI y SAVI, este análisis se realizó solo con la validación cruzada, primero se evaluó la relación entre ambos índices y el pH.

Con este análisis se permitió determinar si los índices NDVI y SAVI en conjunto pueden ser predictores confiables de pH del suelo.

La relación fue expresada como: pH = $f(\text{SAVI}_{\text{normalizado}}\ y\ \text{NDVI}_{\text{normalizado}})$ = β0 + β1 * SAVI y NDVI

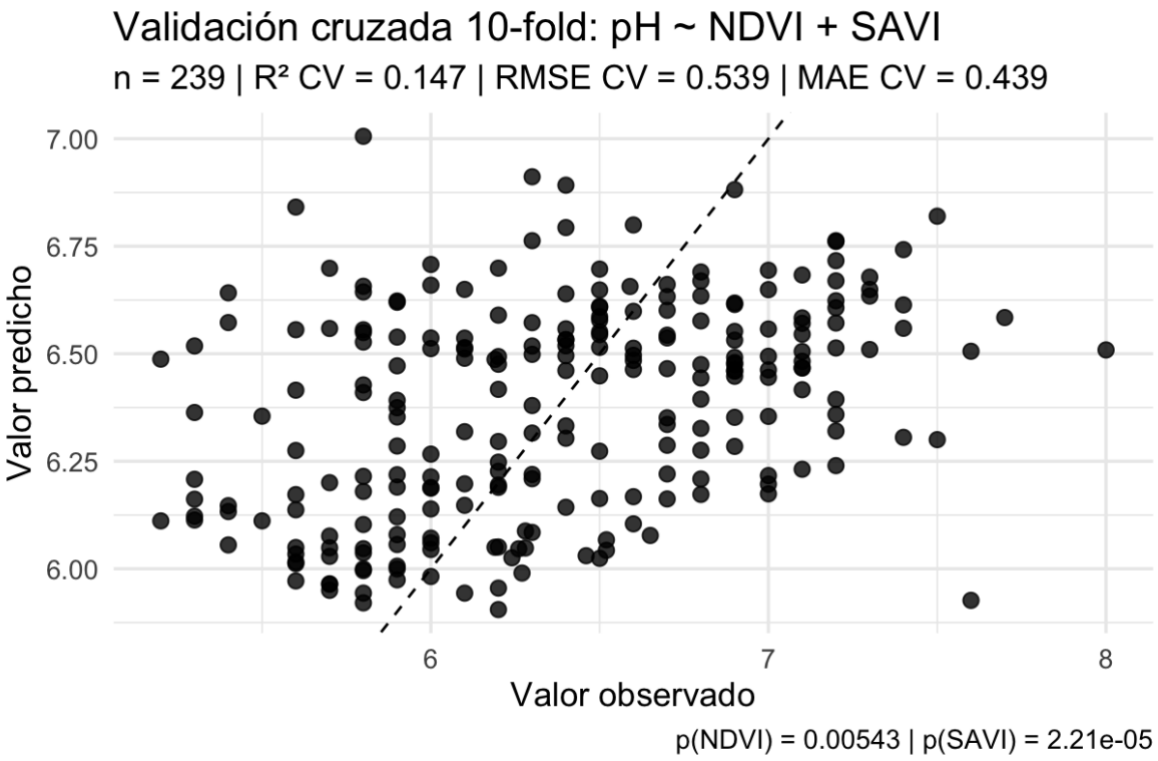



*Figura 7. Relación conjunta entre SAVI y NDVI estandarizado y pH del suelo (validación cruzada)*

Este procedimiento se repitió con cada una de variables de estudio K, P y MO.

*En el anexo "Códigos", se muestran los códigos implementados en RStudio para cada índice y modelo de regresión.*

## 6. Resultados y discusión

El presente estudio evaluó la capacidad de los índices de vegetación, NDVI y SAVI, relacionados con valores fisicoquímicos del suelo, pH, potasio, fosforo y materia orgánica. Se utilizaron modelos de regresión lineal, Random Forest y validación cruzada con el fin de comparar enfoques paramétricos y no paramétricos. La siguiente tabla muestra solo los valores obtenidos con validación cruzada, que fueron un poco más precisos.

| Variable | Modelo | Índice(s) | $R^2$ | RMSE | MAE | Detalle |
|---|---|---|---|---|---|---|
| pH | Regresión lineal | NDVI | 0.177 | 0.552 | 0.460 | Buen desempeño relativo; mejor $R^2$ CV entre los modelos simples |
| pH | Random Forest | NDVI | 0.027 | 0.650 | 0.511 | Bajo desempeño en validación; evidencia de sobreajuste |
| pH | Regresión lineal | SAVI | 0.138 | 0.542 | 0.439 | Menor error entre los modelos simples |
| pH | Random Forest | SAVI | 0.032 | 0.645 | 0.525 | Bajo desempeño en validación |
| pH | Regresión lineal múltiple | NDVI + SAVI | 0.147 | 0.539 | 0.439 | Mejor RMSE global en pH; empata mejor MAE |
| K | Regresión lineal | NDVI | 0.086 | 201.578 | 149.523 | Mejor opción para K |
| K | Random Forest | NDVI | 0.024 | 233.486 | 162.852 | Peor que lineal |
| K | Regresión lineal | SAVI | 0.022 | 209.540 | 151.684 | Inferior a NDVI lineal |
| K | Random Forest | SAVI | 0.009 | 231.003 | 163.446 | Muy bajo desempeño |
| K | Regresión lineal múltiple | NDVI + SAVI | 0.030 | 208.689 | 150.743 | No supera a NDVI lineal |
| P | Regresión lineal | NDVI | 0.114 | 26.507 | 17.368 | Mejor opción para P |
| P | Random Forest | NDVI | 0.054 | 34.221 | 19.973 | Más error que lineal |

| | | | | | | |
|---|---|---|---|---|---|---|
| P | Regresión lineal | SAVI | 0.069 | 29.939 | 17.961 | Intermedio, pero no mejor |
| P | Random Forest | SAVI | 0.029 | 35.203 | 20.013 | Bajo desempeño |
| P | Regresión lineal múltiple | NDVI + SAVI | 0.087 | 29.660 | 17.430 | Mejor que SAVI solo, pero no supera a NDVI lineal |
| MO | Regresión lineal | NDVI | 0.062 | 2.713 | 1.896 | Mejor opción para MO |
| MO | Random Forest | NDVI | 0.001 | 3.500 | 2.283 | Muy bajo desempeño |
| MO | Regresión lineal | SAVI | 0.008 | 2.931 | 1.916 | Débil |
| MO | Random Forest | SAVI | 0.005 | 3.223 | 2.125 | Bajo desempeño |
| MO | Regresión lineal múltiple | NDVI + SAVI | 0.035 | 2.860 | 1.925 | No supera a NDVI lineal |

## 6.1 Desempeño de los modelos

Los resultados muestran que los modelos de regresión lineal presentaron un desempeño más consistente y estable que los modelos de Random Forest al ser evaluados mediante la validación cruzada. Aunque los modelos Random Forest alcanzaron valores altos de ajuste sobre la muestra original, su rendimiento disminuyó de forma importante al pasar a validación, lo que sugiere un problema de sobreajuste. Esto indica que, si bien fueron capaces de adaptarse bien a los datos observados, no lograron mantener esa capacidad al predecir datos no utilizados en el entrenamiento.

Los modelos lineales, aun cuando mostraron valores de $R^2$ moderado o bajos, demostraron un comportamiento más robusto en la validación cruzada. Esto es relevante en el objetivo del estudio, ya que permite concluir que los modelos lineales ofrecen una mejor capacidad de generalización y, por lo tanto, una base metodológica más sólida para la estimación de propiedades del suelo a partir de índices espectrales.

La incorporación de ambos índices no produjo una mejora generalizada del desempeño. El uso combinado de NDVI y SAVI solo mostró una mejora marginal en la predicción de una de las variables (pH), mientras que las demás no supero el desempeño de los modelos lineales simples.

## 6.2 Comparación entre índices NDVI y SAVI

Comparando el comportamiento entre NDVI y SAVI, se observó que NDVI presentó mejor comportamiento en las variables K, P y MO. Para estas variables, los modelos lineales construidos con NDVI como predictor único alcanzaron mejores valores de $R^2$ en validación cruzada y menos errores de predicción en comparación con SAVI.

En el caso de SAVI, mostró un mejor comportamiento para la variable pH, más en concreto en términos de error de predicción. Por lo que SAVI captó de manera más eficiente ciertas condiciones relacionadas con la variable pH.

## 6.3 Variable por variable

La variable pH fue la que presentó el comportamiento más favorable en comparación con las otras variables, mostrando la relación más clara con los índices espectrales. El modelo con SAVI mostró menores errores de predicción que el modelo NDVI, mientras que el modelo NDVI junto a SAVI logro una mejora marginal adicional en RMSE, manteniendo un MAE similar. Sin embargo el modelo lineal con NDVI conservó un $R^2$ de validación un poco superior.

La variable K, el mejor desempeño correspondió al modelo lineal con NDVI. Aunque la capacidad explicativa fue baja, este modelo supero a todos lo modelos aplicados. Los resultados indican que la relación entre los índices espectrales y la variable K es débil, por lo que estos índices por sí solos no resultan suficientes para generar predicciones de alta precisión.

La variable P, tuvo comportamientos similares a la variable K. El modelo lineal NDVI alcanzó el mejor desempeño en validación cruzada, con menores errores y mayor capacidad explicativa frente a los otros modelos. Aun así, con una capacidad predictiva moderada.

La variable MO, mostro el desempeño más bajo de todas las variables. Tanto NDVI como SAVI presentaron relaciones débiles en esta variable, los modelos Random Forest no lograron mejorar los resultados. A pesar de ello, el modelo lineal NDVI volvió a ser la mejor opción comparativa, presentando menores errores y un mejor $R^2$ de validación que los otros modelos.

## 7. Conclusiones

La presente investigación permitió evaluar la capacidad de los índices espectrales NDVI y SAVI para estimar propiedades fisicoquímicas del suelo, específicamente pH, potasio (K), Fósforo (P) y materia orgánica (MO), mediante modelos de regresión lineal, Random Forest y validación cruzada. En términos generales, los resultados muestran que la estimación remota de variables edáficas a partir de imágenes satelitales es metodológicamente factible, aunque su desempeño depende de la propiedad del suelo analizada y el tipo de modelo utilizado.

En cuanto al desempeño de los modelos, se concluye que los enfoques de regresión lineal fueron más consistentes y confiables que los modelos de Random Forest. Si bien presentaron altos niveles de ajuste sobre la muestra original, su rendimiento disminuyó de forma importante al ser evaluados mediante validación cruzada, evidenciando problemas de sobreajuste. Por el contrario los modelos lineales mostraron un comportamiento más estable y una mejor capacidad de generalización, lo que los posiciona como alternativa metodológica más sólida dentro del contexto de este estudio.

Respecto a la comparación entre índices, los resultados indican que NDVI fue el índice con mejor desempeño global. Este presentó mejores resultados en las variables K, P y MO, superando a SAVI tanto en capacidad explicativa como en errores de predicción. SAVI mostró una ventaja mínima en la estimación de pH, especialmente en términos de error predictivo. A pesar de eso el uso conjunto de ambos índices no produjo una mejora generalizada, ya que solo aportó una mejora mínima en pH y no supero a NDVI en las demás variables.

A nivel específico, el pH fue la variable que presentó el comportamiento más favorable, mostrando la relación más clara con los índices espectrales y constituyéndose en la propiedad del suelo mejor estimada dentro del estudio.

Los resultados obtenidos no respaldan la idea de que los modelos más complejos necesariamente entregan mejores resultados predictivos. Por el contrario, la evidencia mostró que los modelos lineales simples, especialmente aquellos basados en NDVI, ofrecieron el mejor equilibrio entre ajuste, estabilidad y capacidad predictiva validada. De esta forma, el estudio demuestra que los índices espectrales pueden construir una herramienta útil para complementar el monitoreo de la calidad del suelo, pero también confirma que su potencial predictivo es variable y que debe interpretarse con cautela según la propiedad edáfica que se desea estimar.

Por último, esta investigación aporta una metodología replicable basada en el uso combinado de QGIS, RStudio e imágenes Sentinel-2, la cual puede servir como base para futuras aplicaciones en agricultura de precisión.

## 8. Literatura Citada


Nix, J. (s/f). *Impacto de la Salud del Suelo en la Sostenibilidad Agrícola*. Biomemakers.com. Recuperado el 7 de julio de 2025, de https://biomemakers.com/es/blog/como-afecta-la-salud-del-suelo-a-la-sostenibilidad-de-la-explotacion-y-a-su-rentabilidad

Sgargi, C. (2022, mayo 2). *Imágenes de satélite para la agricultura*. Agricolus; Agricolus srl. https://www.agricolus.com/es/tecnologias/imagenes-de-satelite/

*Mergin Maps en directo*. (s/f). Merginmaps.com. Recuperado el 7 de julio de 2025, de https://es.merginmaps.com/glossary/qgis

*IBM watsonx as a Service*. (2025, junio 19). Ibm.com. https://www.ibm.com/docs/es/watsonx/saas?topic=scripts-rstudio

Riquelme Adriasola, D. A. (2023). *Uso de imágenes satelitales para el análisis y generación de reportes automáticos en zonas de sequías en el territorio nacional*. Universidad de Chile. https://doi.org/10.58011/X3Q3-AF66

*Copernicus Browser*. (s/f). Copernicus Browser. Recuperado el 7 de julio de 2025, de https://browser.dataspace.copernicus.eu/?zoom=5&lat=50.16282&lng=20.78613&demSource3D=%22MAPZEN%22&cloudCoverage=30&dateMode=SINGLE

AGROLENS PROJECT, Kammerlander, C., 1, Kolb, V., Luegmair, M., Scheermann, L., Schmailzl, M., Seufert, M., Zhang, J., Dalić, D., 2, Schön, T., 3, MI4People, & Technische Hochschule Ingolstadt. (2025). Machine learning models for soil parameter prediction based on satellite, weather, clay and yield data. *Agrolens Project*.

Molaeinasab, A., Tarkesh, M., Bashari, H., Toomanian, N., Aghasi, B., & Jalalian, A. (2025). Spatial modeling of soil chemical properties in an arid region of Central Iran

using machine learning and remote sensing data. *Modeling Earth Systems and Environment*, *11*(2). https://doi.org/10.1007/s40808-025-02331-0

Hosseini, F. S., Razavi-Termeh, S. V., Sadeghi-Niaraki, A., Choi, S., & Jamshidi, M. (2023). Spatial prediction of physical and chemical properties of soil using optical satellite imagery: a state-of-the-art hybridization of deep learning algorithm. *Frontiers in Environmental Science*, *11*. https://doi.org/10.3389/fenvs.2023.1279712

Beisekenov, N., Banakinaou, W., Ajayi, A. D., Hasegawa, H., & Tadao, A. (2025a). Remote sensing-based soil organic carbon monitoring using advanced machine learning techniques under conservation agriculture systems. *Smart Agricultural Technology*, *11*, 101036. https://doi.org/10.1016/j.atech.2025.101036

Chen, N., Wei, Z., Jin, X., Lin, N., Yang, F., Zhao, L., & Wu, S. (2026). Integrating transformer-based learning and Sentinel-2 bare soil composites for soil organic carbon mapping in the black soil region of Northeast China. *Scientific Reports*, *16*(1), 3784. https://doi.org/10.1038/s41598-025-33682-4

Carbajal-Llosa, C., Barja, A., & Pizarro, S. (2025). Ensemble machine learning for digital mapping of soil pH and electrical conductivity in the Andean agroecosystem of Peru. *Frontiers in Soil Science*, *5*. https://doi.org/10.3389/fsoil.2025.1673628

Suleymanov, R., Yurkevich, M., Bakhmet, O., Popova, T., Kungurtsev, A., Zakirov, D., Vittsenko, A., Mishra, G., & Suleymanov, A. (2025). Interpretable machine learning and remote sensing data reveal soil biogeochemistry patterns in agricultural systems. *Land*, *14*(9), 1881. https://doi.org/10.3390/land14091881

## 9. Anexos

*Figura 1.*

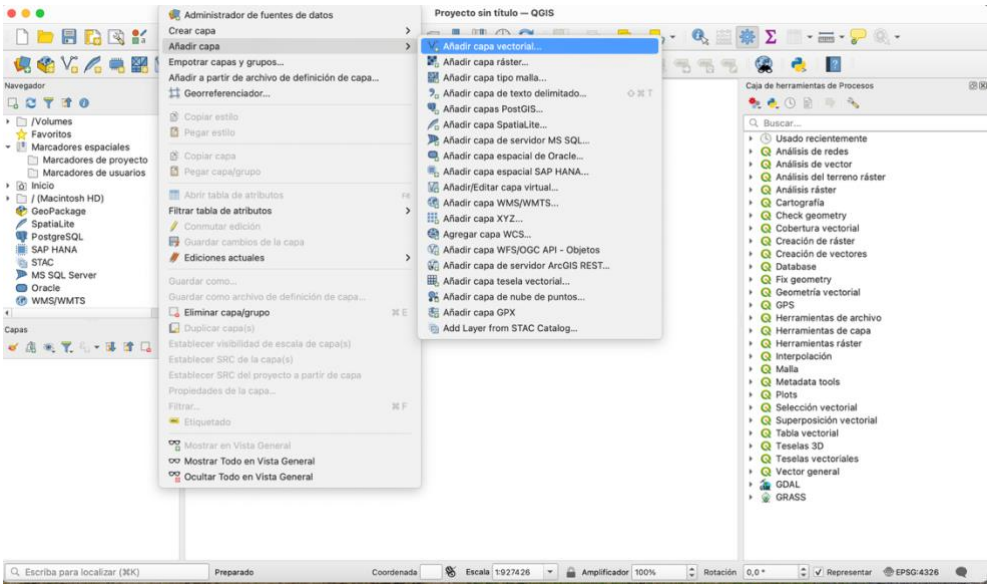



*Figura 2.*

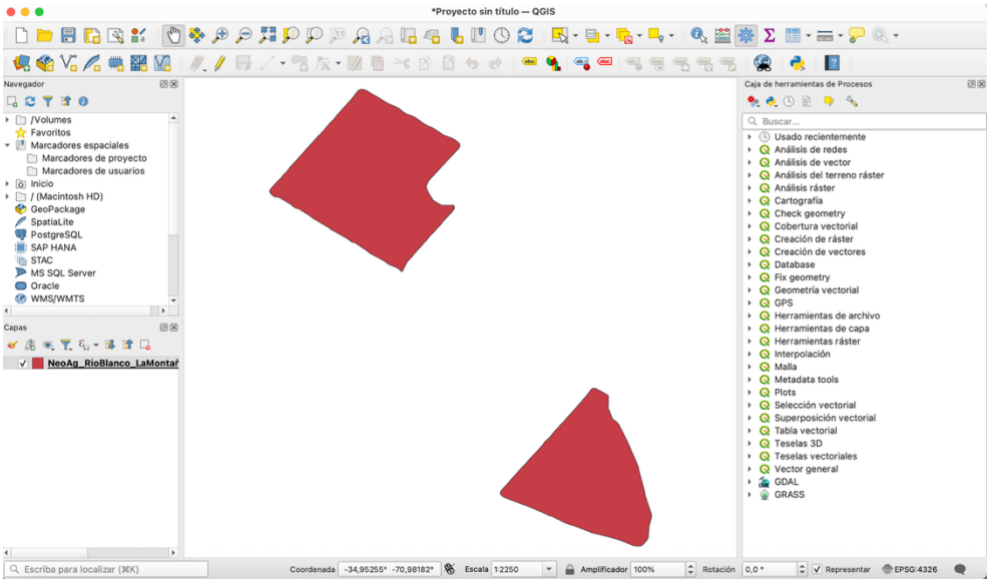


*Figura 3.*

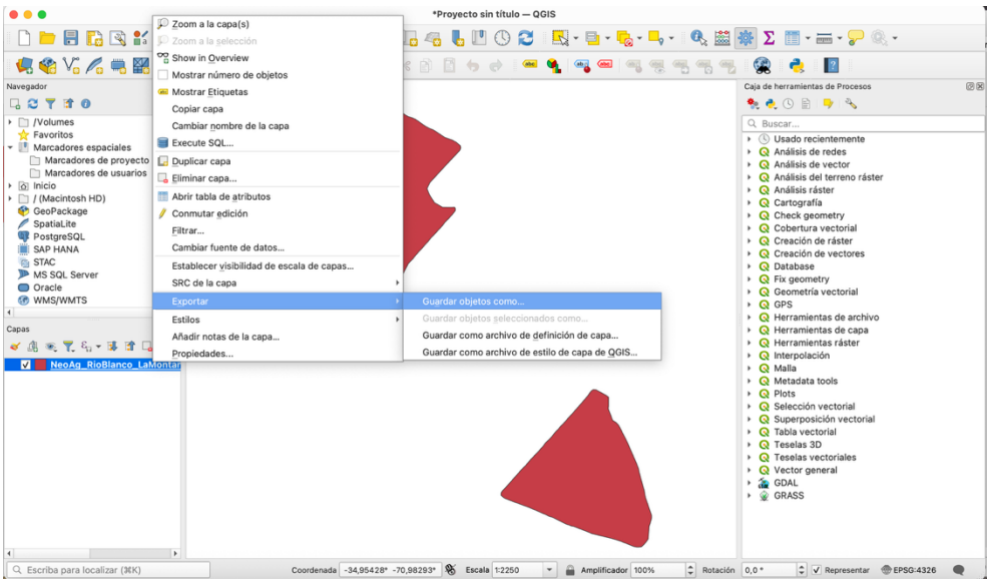



*Figura 4.*

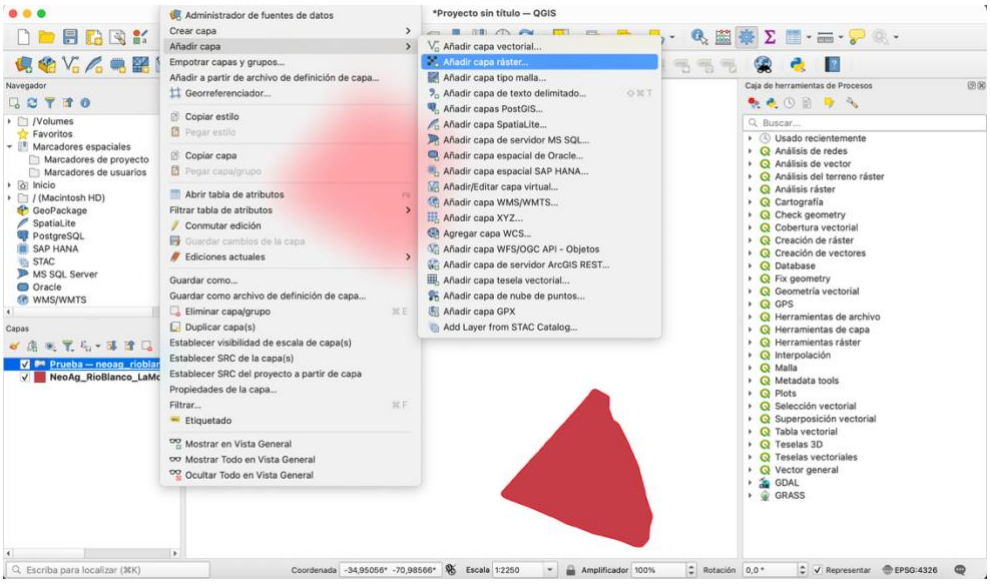


*Figura 5.*

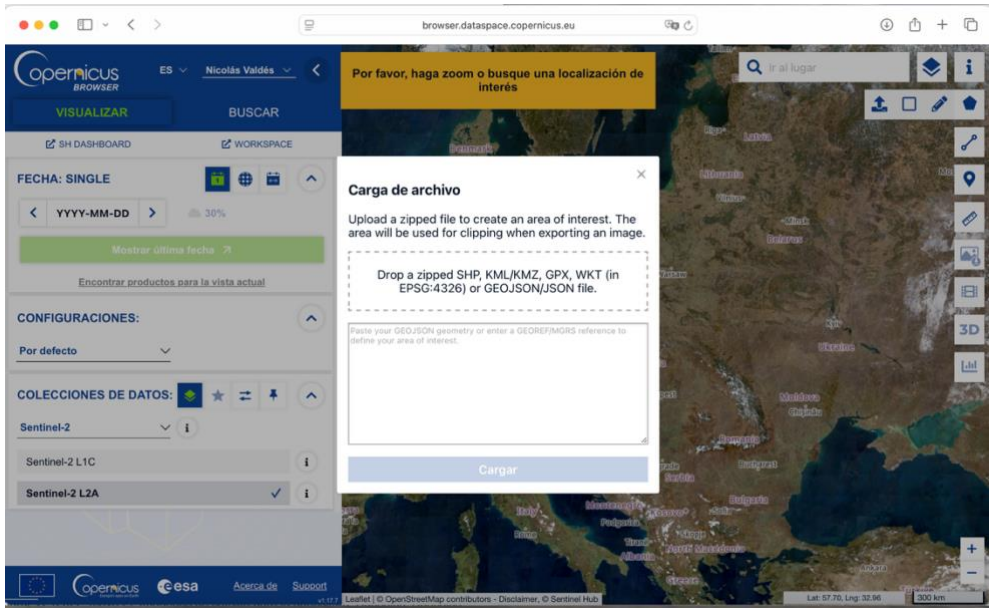



*Figura 6.*

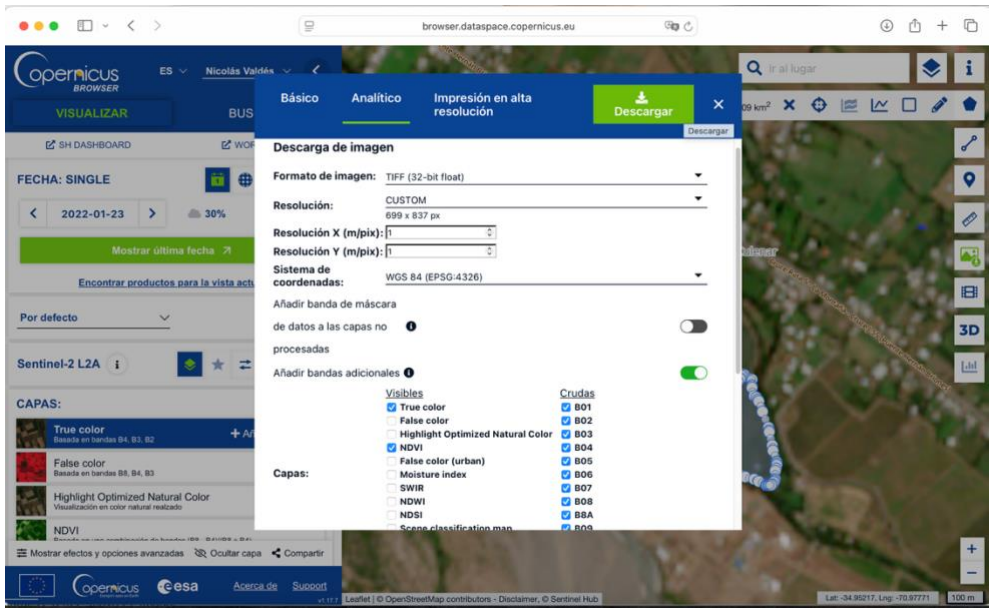


*Figura 7.*

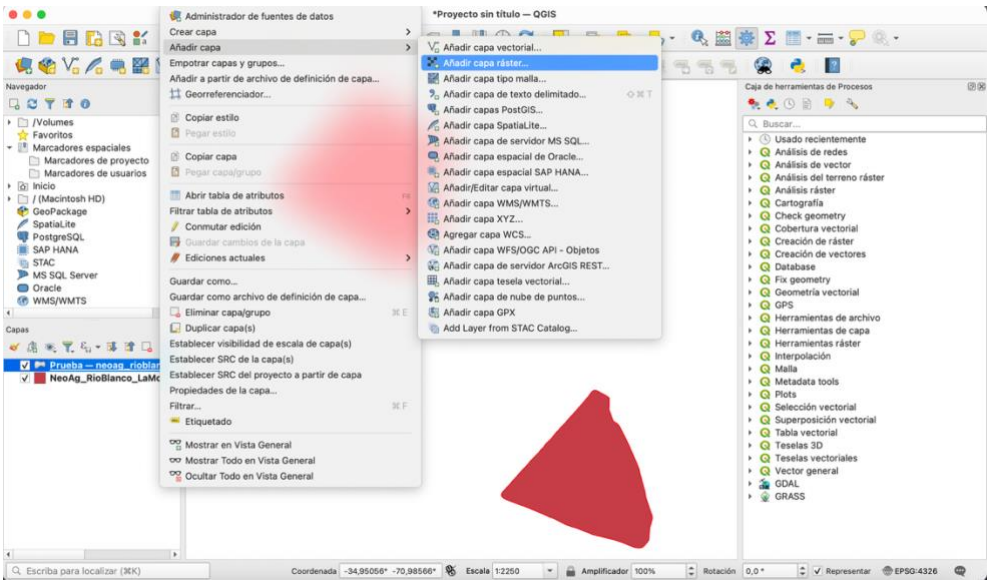



*Figura 8.*

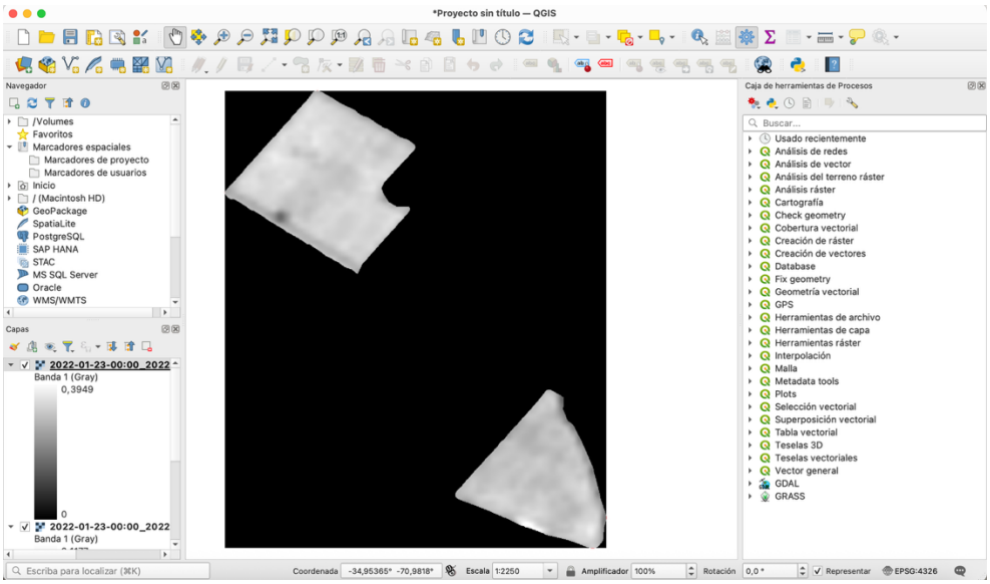


*Figura 9.*

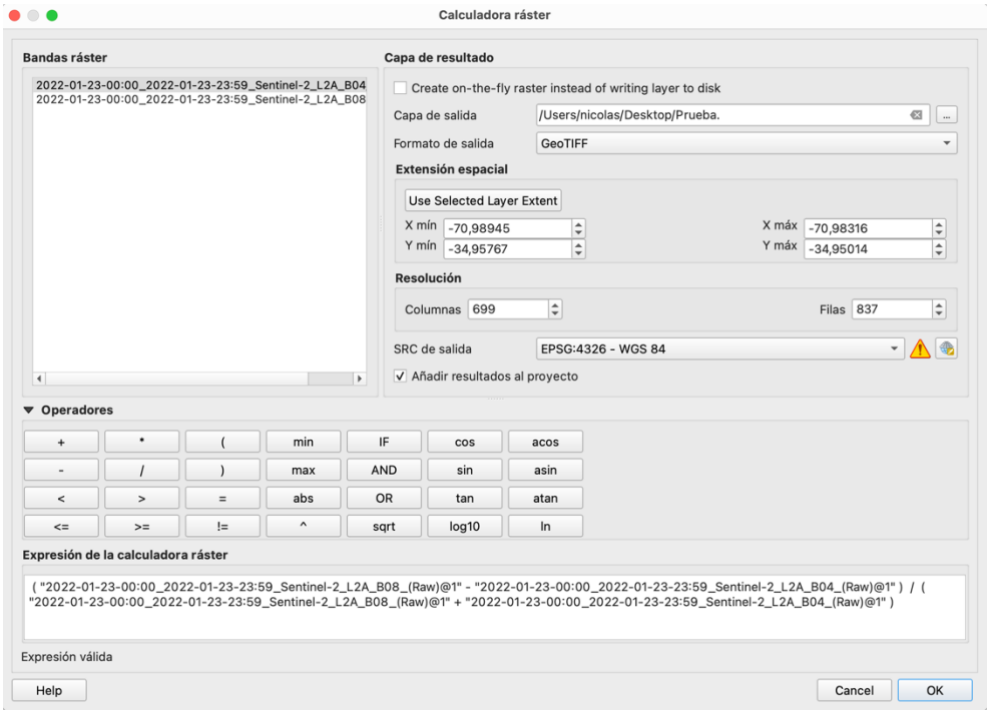



*Figura 10.*

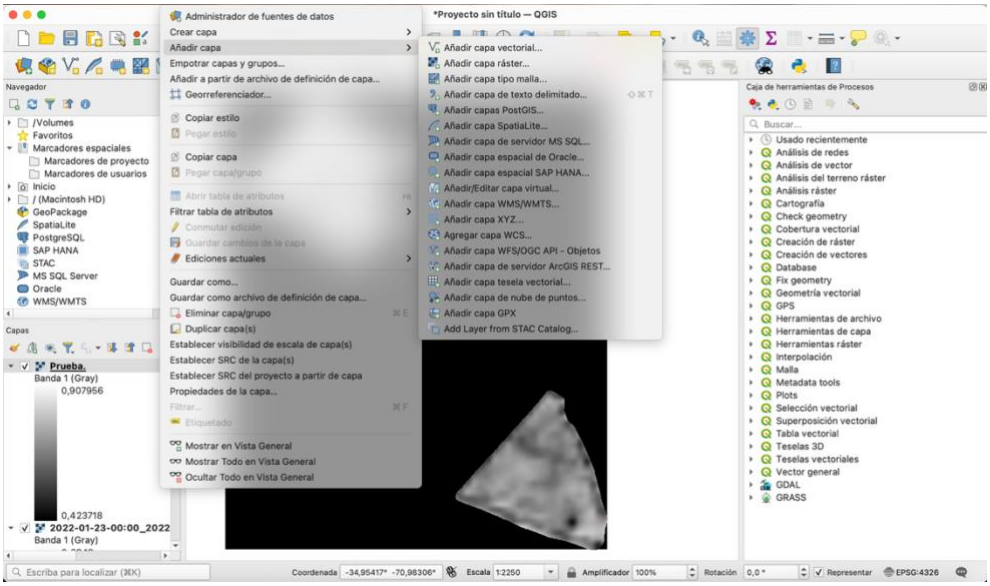


*Figura 11.*

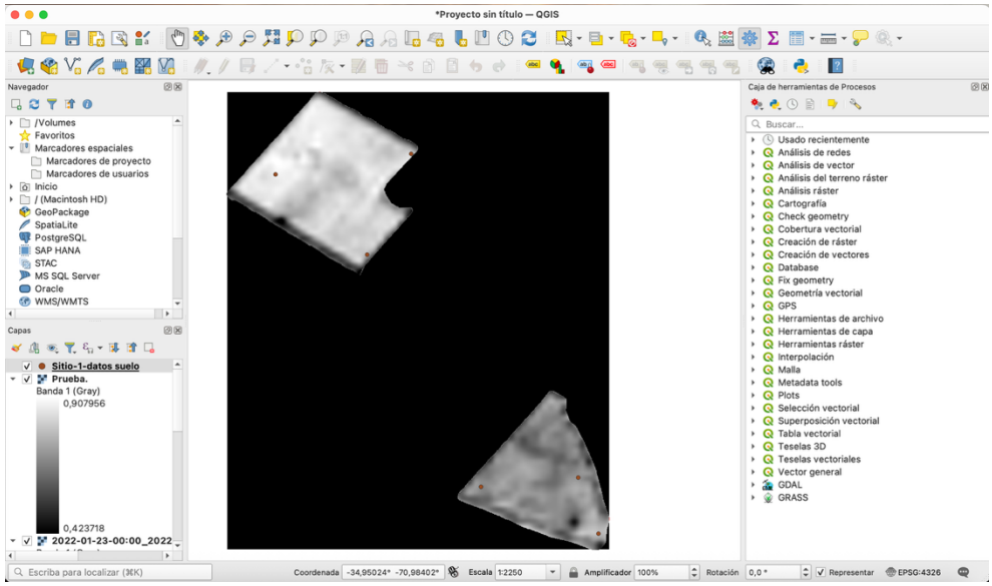



*Figura 12.*

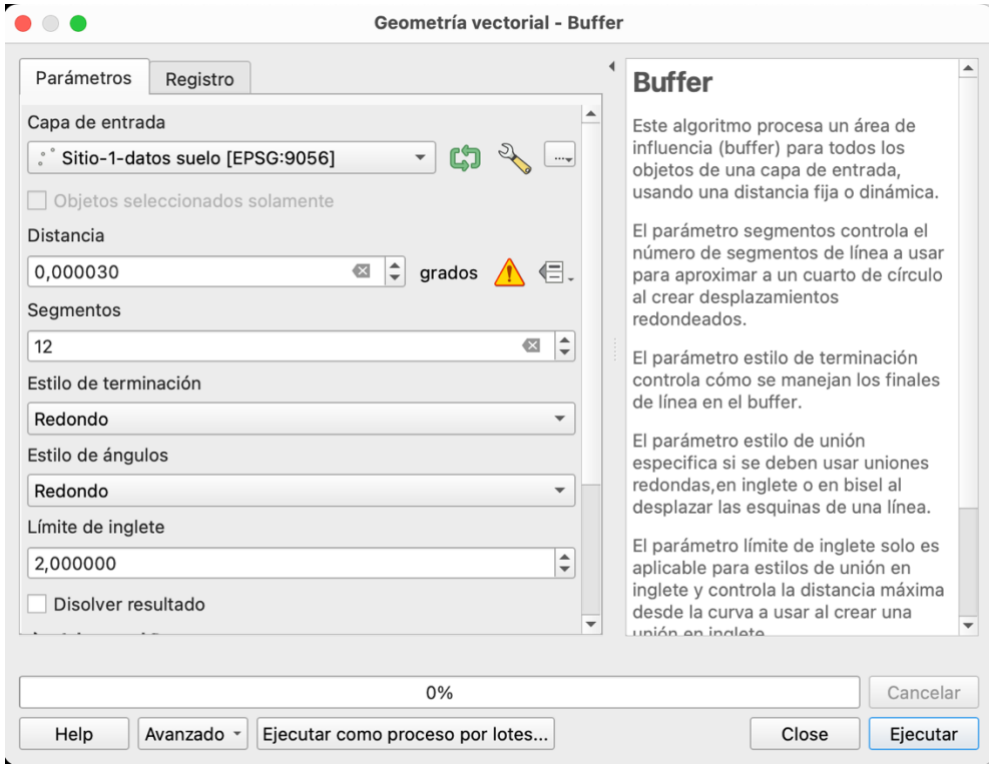


*Figura 13.*

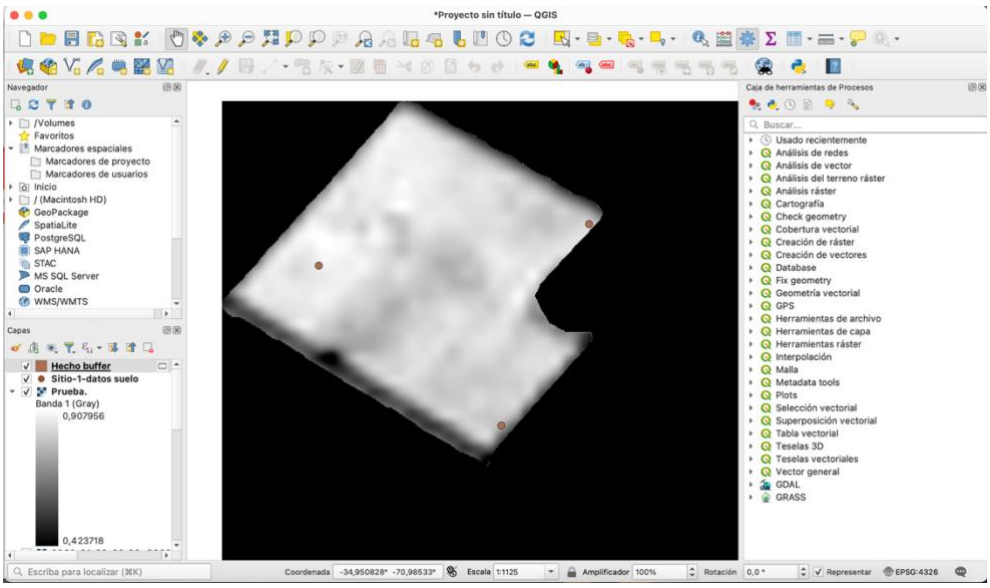



*Figura 14.*

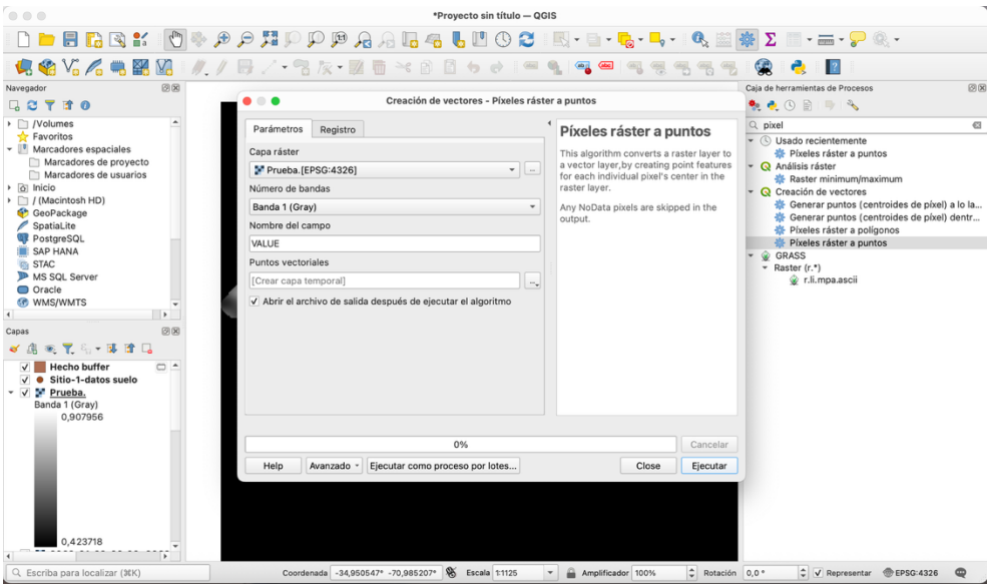


*Figura 15.*

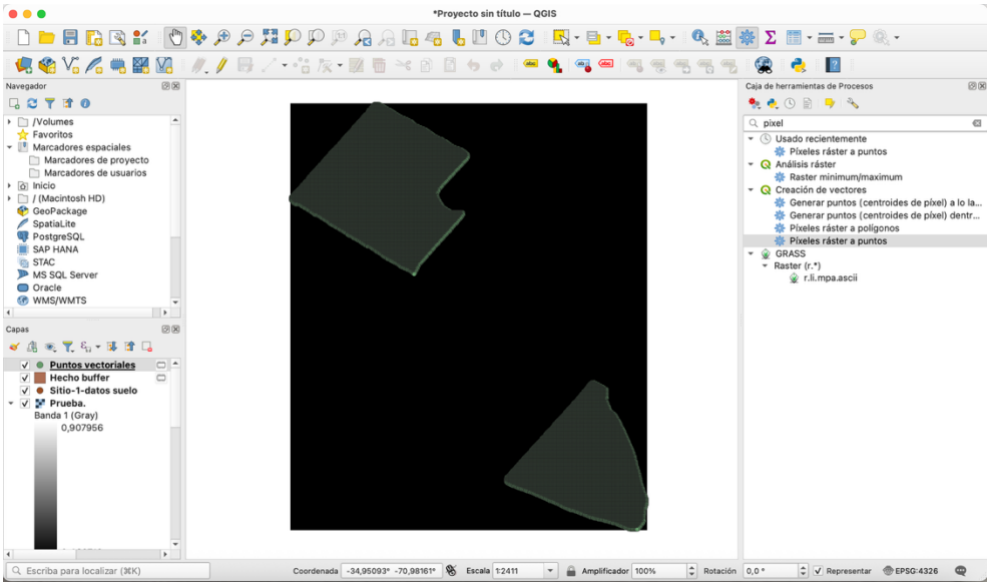



*Figura 16.*

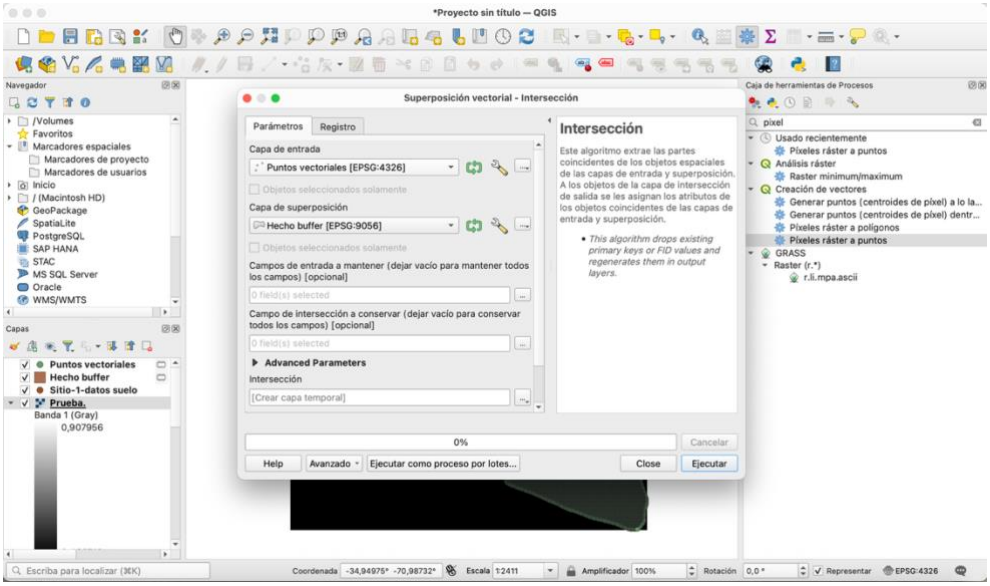


*Figura 17.*

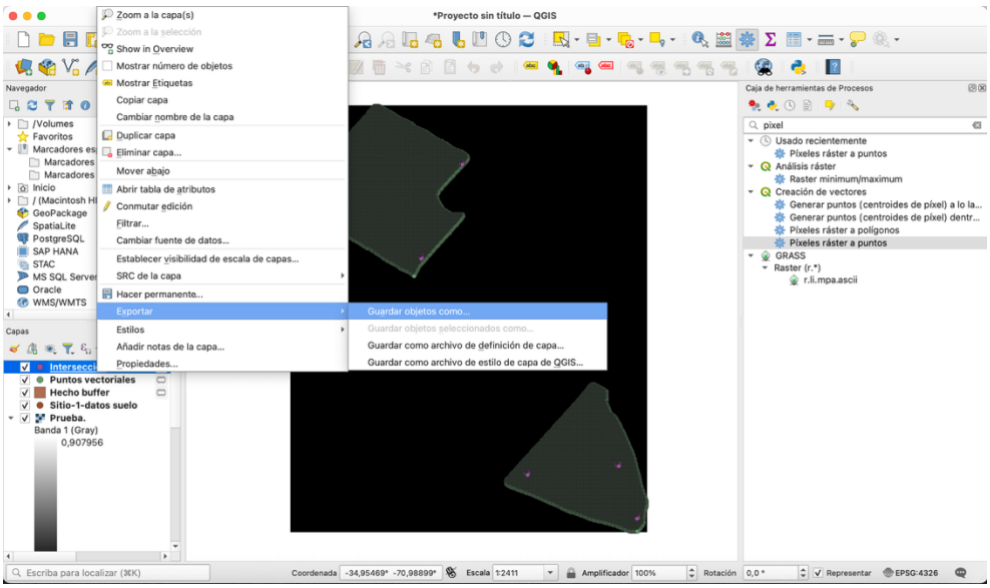



*Figura 18.*

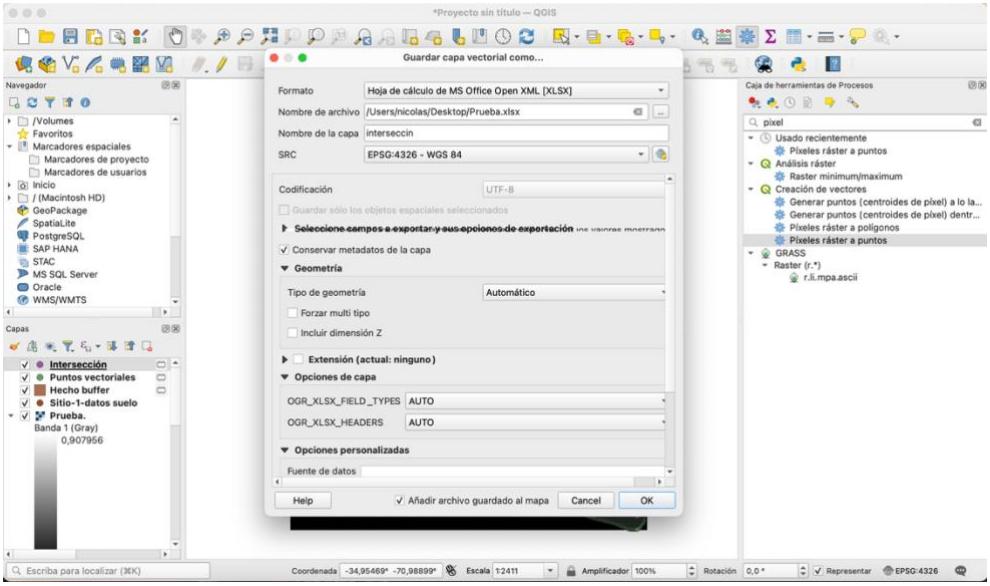
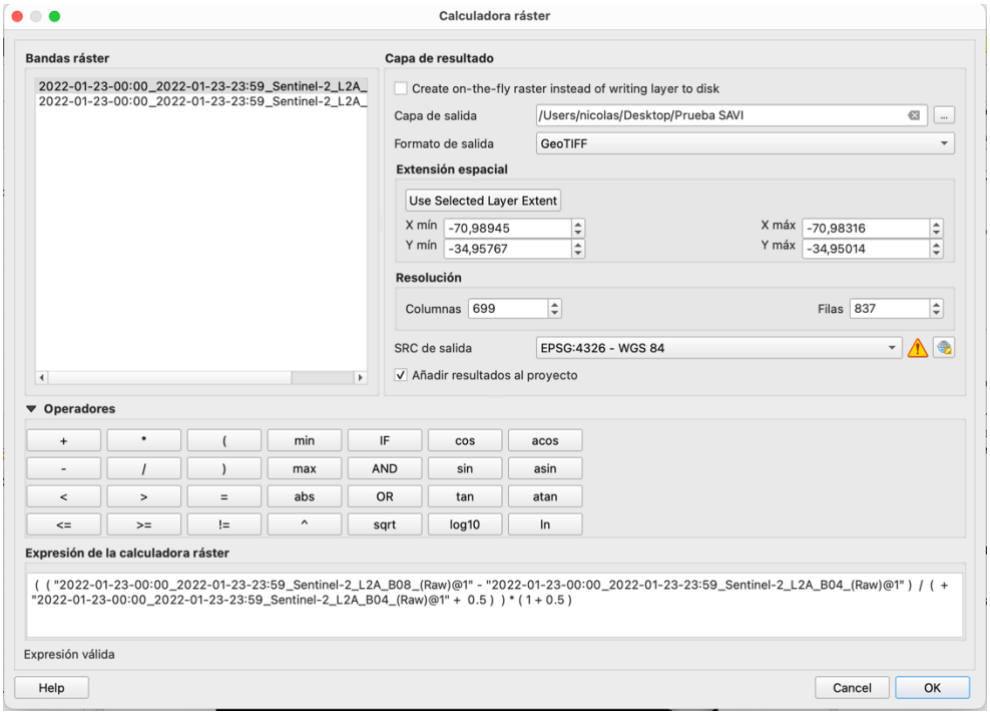
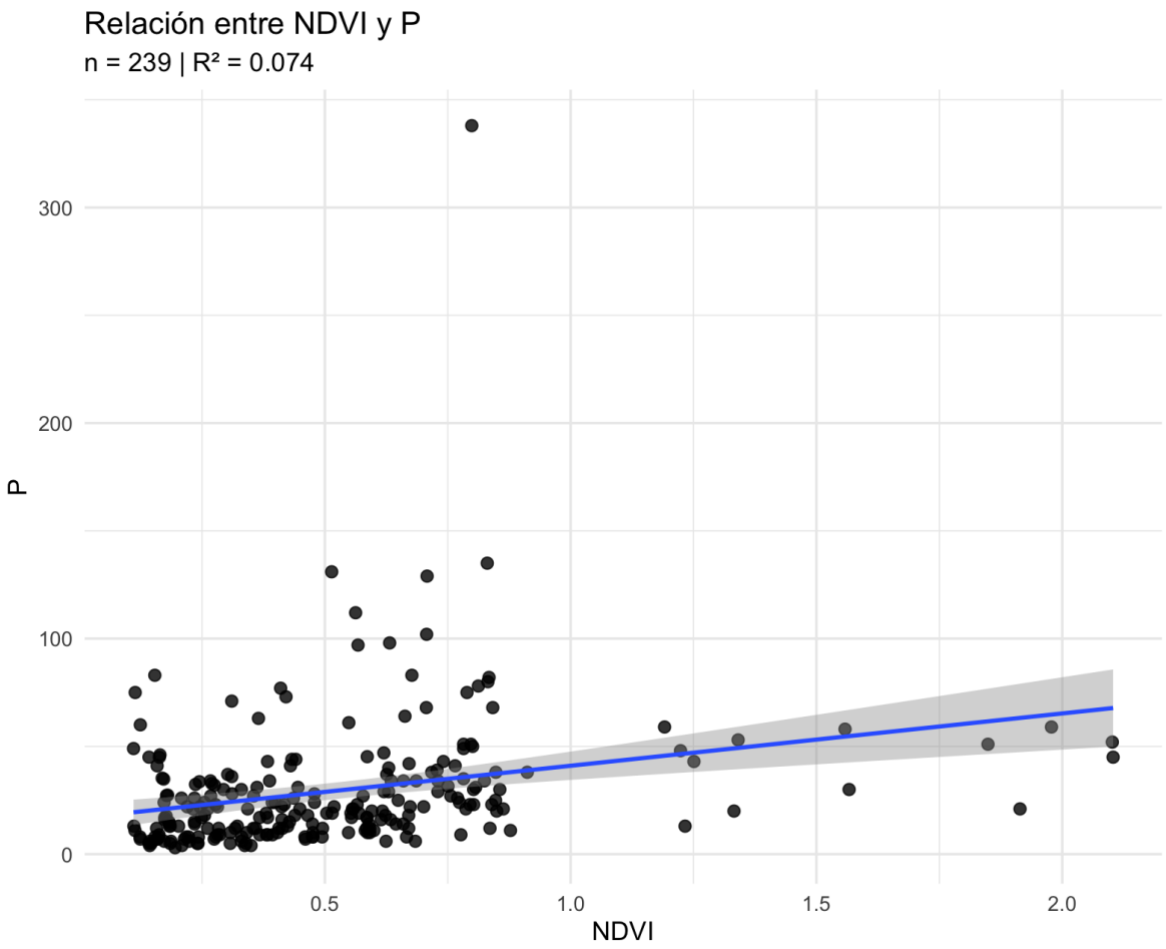
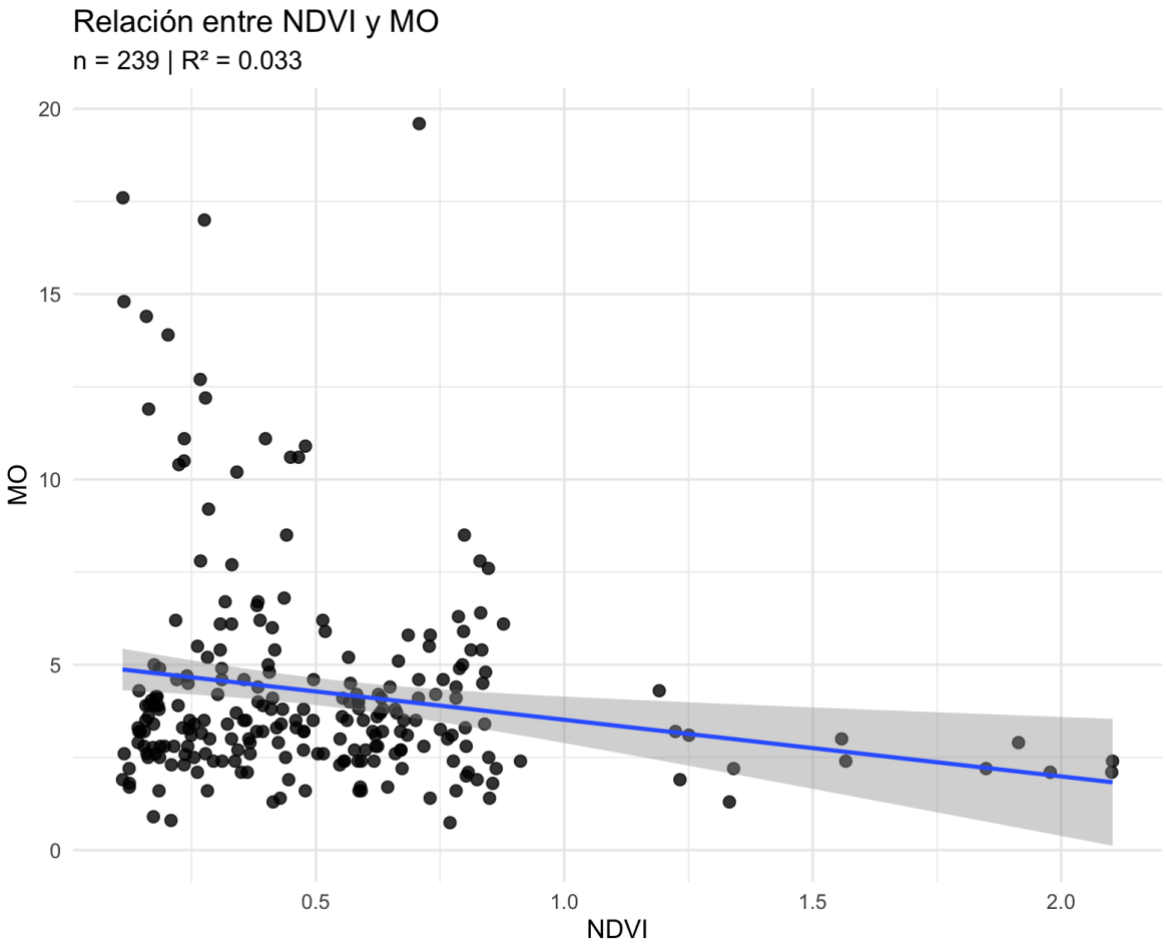


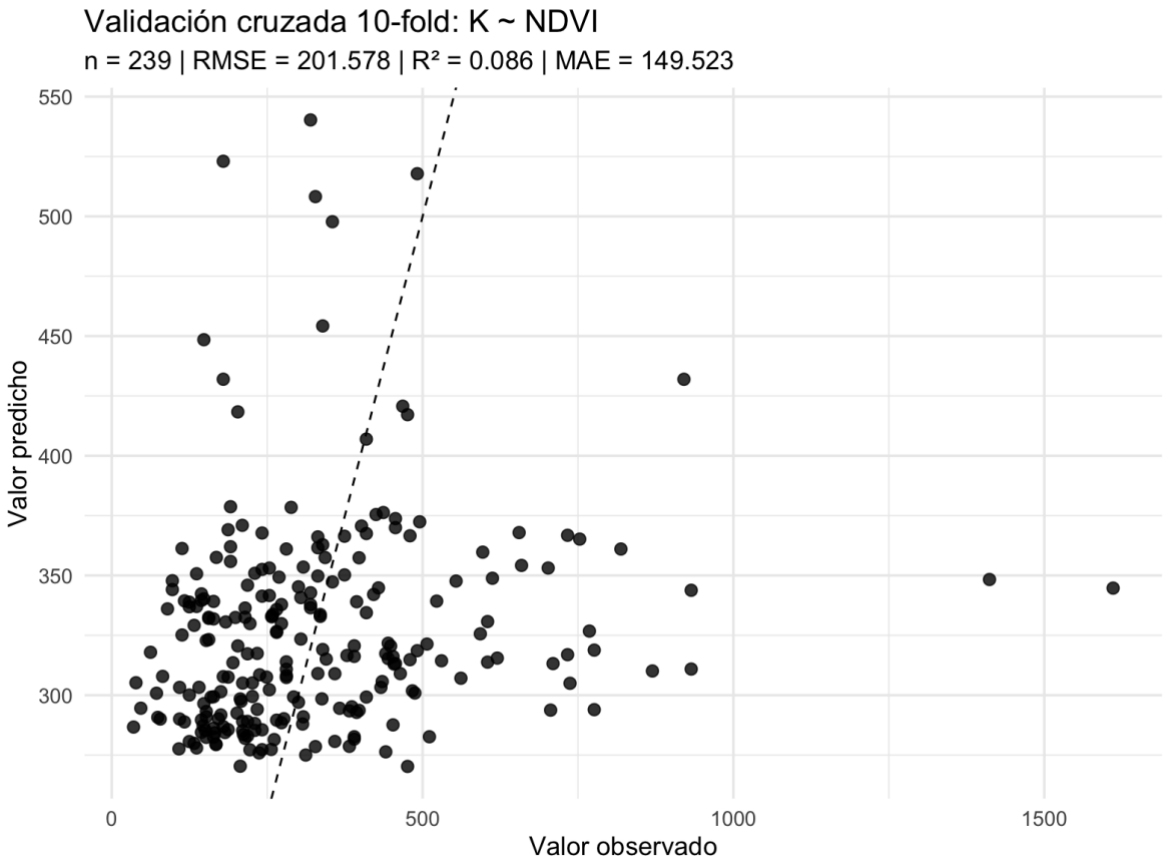
*Figura 19.*

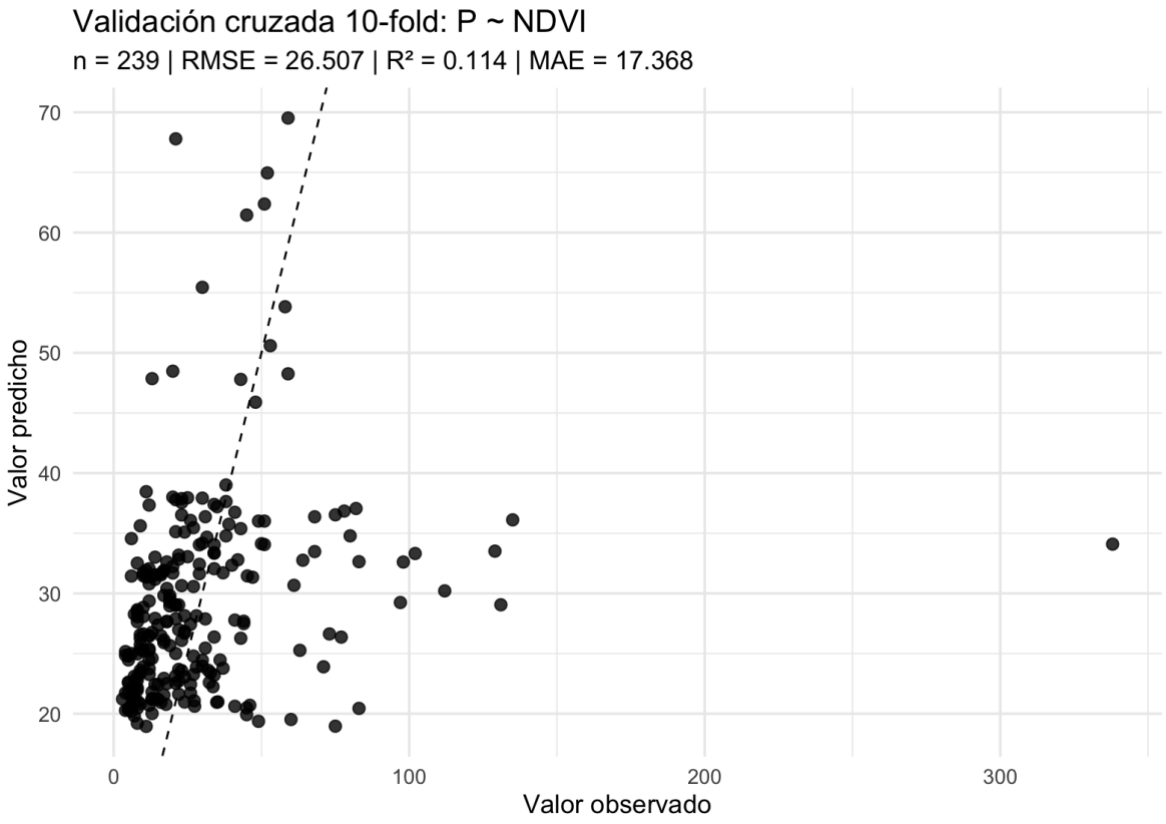

Calculadora ráster
Bandas ráster
2022-01-23-00:00_2022-01-23-23:59_Sentinel-2_L2A_
2022-01-23-00:00_2022-01-23-23:59_Sentinel-2_L2A_
Capa de resultado
Create on-the-fly raster instead of writing layer to disk
Capa de salida
/Users/nicolas/Desktop/Prueba SAVI
Formato de salida
GeoTIFF
Extensión espacial
Use Selected Layer Extent
X mín -70,98945
X máx -70,98316
Y mín -34,95767
Y máx -34,95014
Resolución
Columnas 699
Filas 837
SRC de salida
EPSG:4326 - WGS 84
Añadir resultados al proyecto
Operadores
+ * ( min IF cos acos
- / ) max AND sin asin
< > = abs OR tan atan
<= >= != ^ sqrt log10 ln
Expresión de la calculadora ráster
( ( "2022-01-23-00:00_2022-01-23-23:59_Sentinel-2_L2A_B08_(Raw)@1" - "2022-01-23-00:00_2022-01-23-23:59_Sentinel-2_L2A_B04_(Raw)@1" ) / ( + "2022-01-23-00:00_2022-01-23-23:59_Sentinel-2_L2A_B04_(Raw)@1" + 0.5 ) ) * ( 1 + 0.5 )
Expresión válida
Help
Cancel
OK

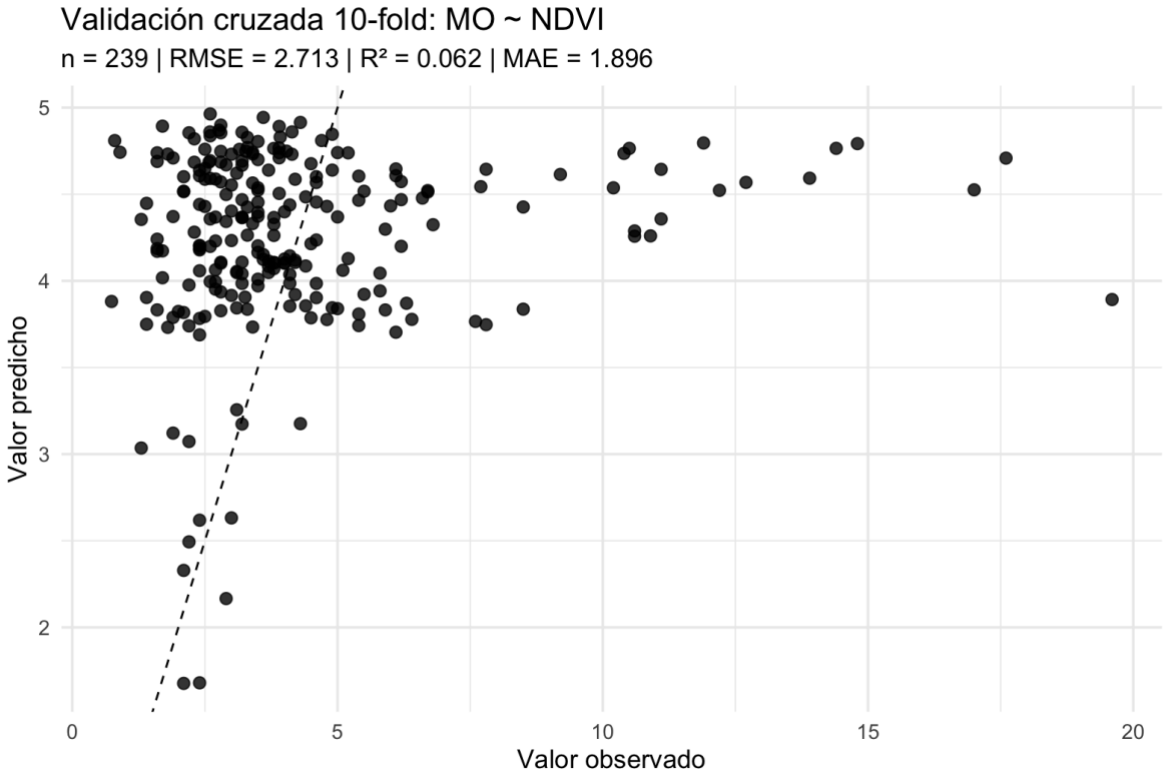
***Gráficos***

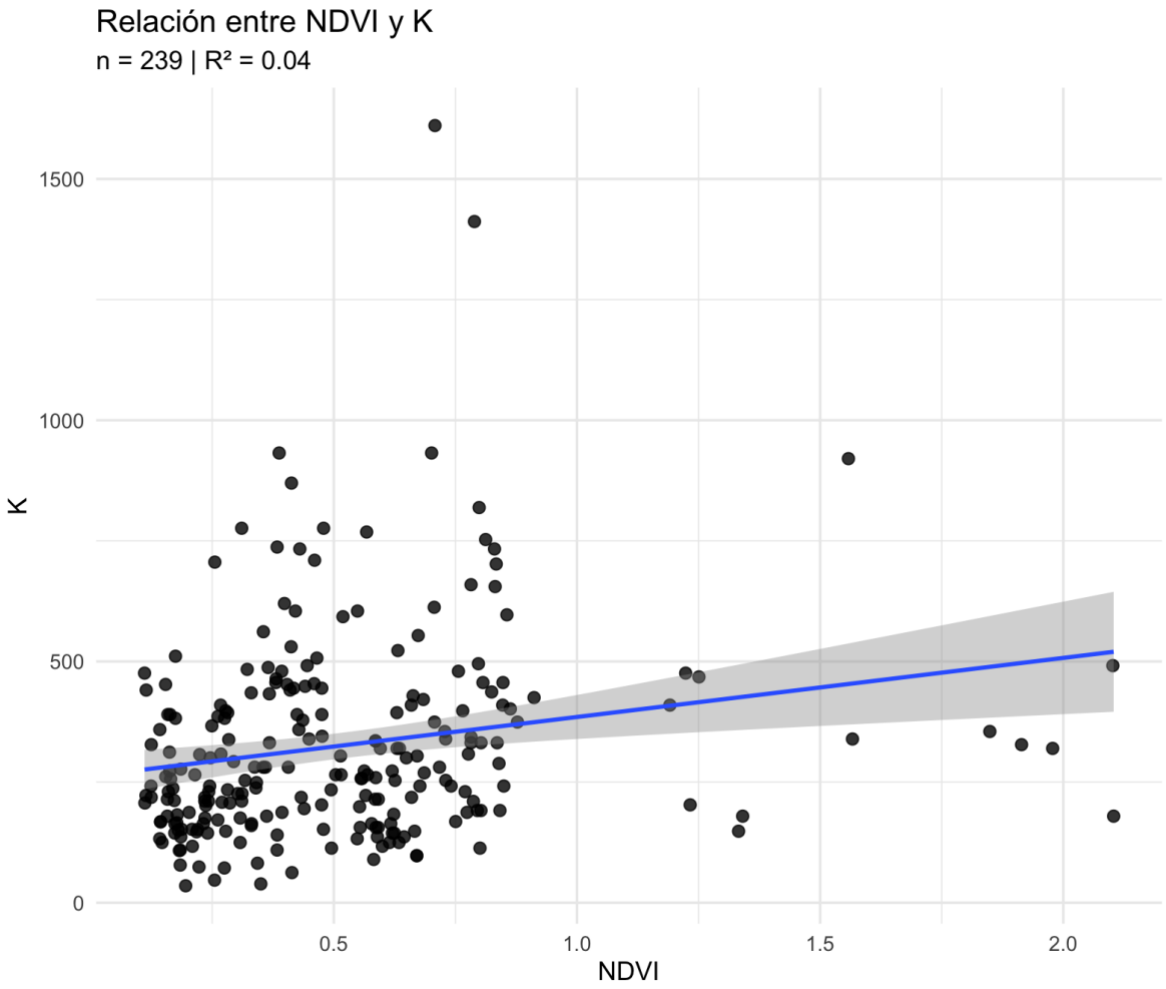

Relación entre NDVI y P
n = 239 | R² = 0.074
P
300
200
100
0
0.5
1.0
1.5
2.0
NDVI

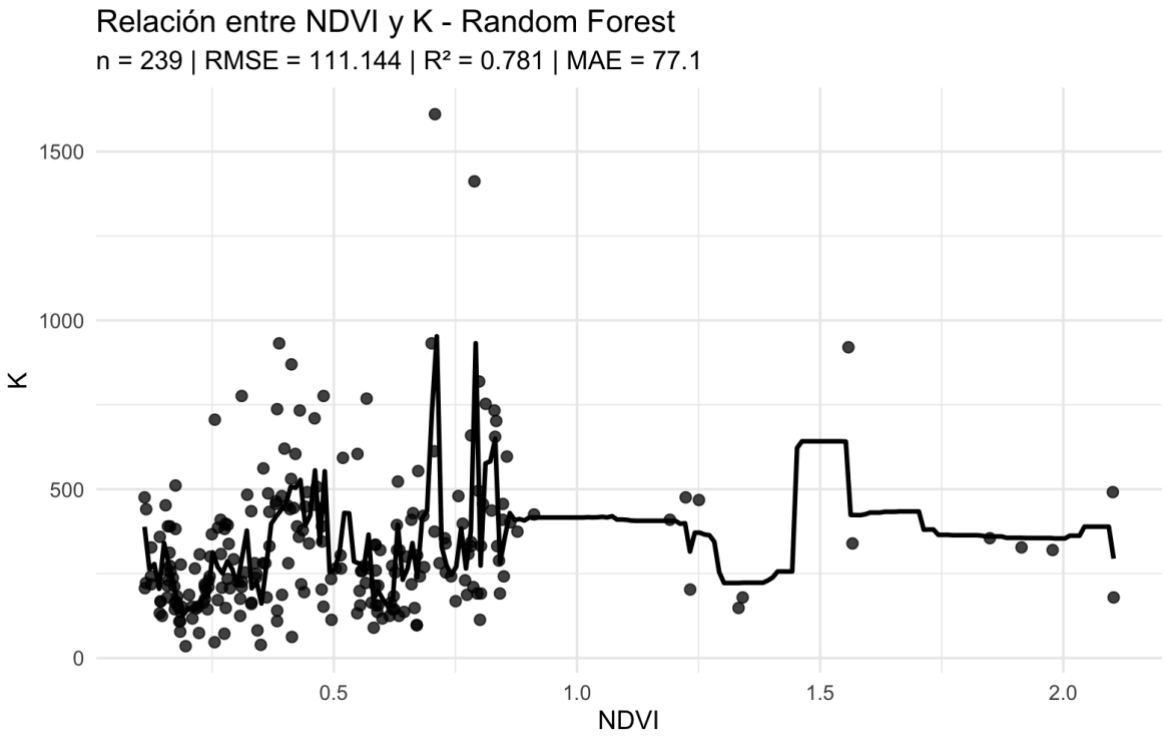

Relación entre NDVI y MO
n = 239 | R² = 0.033
MO
NDVI
0
5
10
15
20
0.5
1.0
1.5
2.0

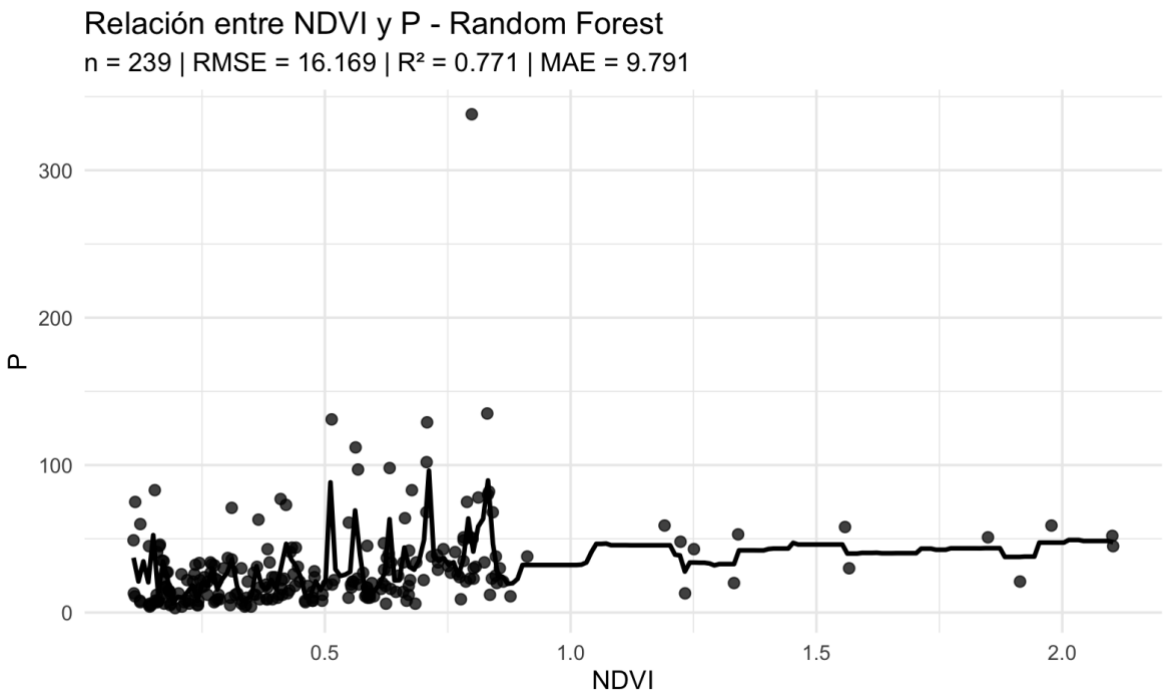

Validación cruzada 10-fold: K ~ NDVI
n = 239 | RMSE = 201.578 | R² = 0.086 | MAE = 149.523
Valor predicho
550
500
450
400
350
300
0
500
1000
1500
Valor observado

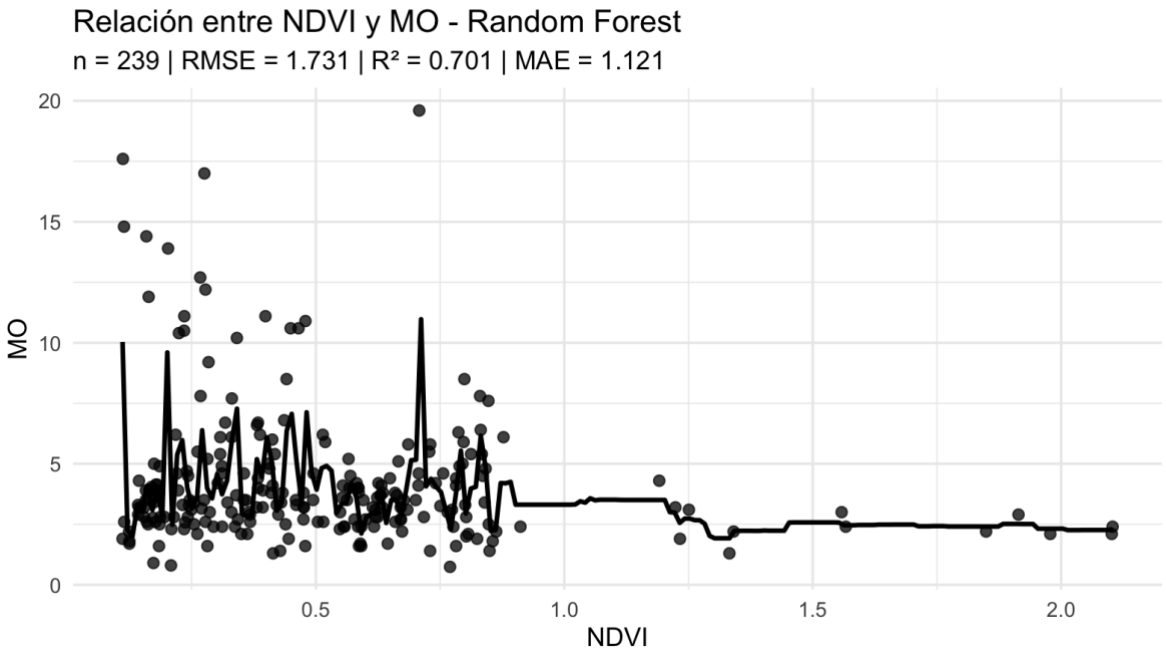

Validación cruzada 10-fold: P ~ NDVI
n = 239 | RMSE = 26.507 | R² = 0.114 | MAE = 17.368
Valor predicho
Valor observado
70
60
50
40
30
20
0
100
200
300

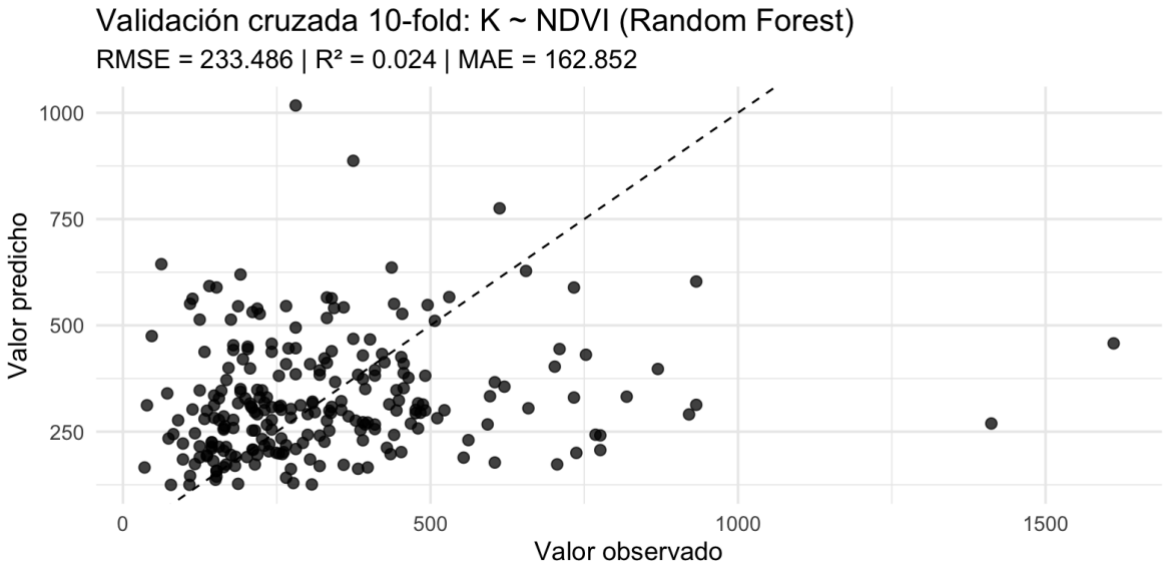

Validación cruzada 10-fold: MO ~ NDVI
n = 239 | RMSE = 2.713 | R² = 0.062 | MAE = 1.896
Valor predicho
Valor observado
5
4
3
2
0
5
10
15
20

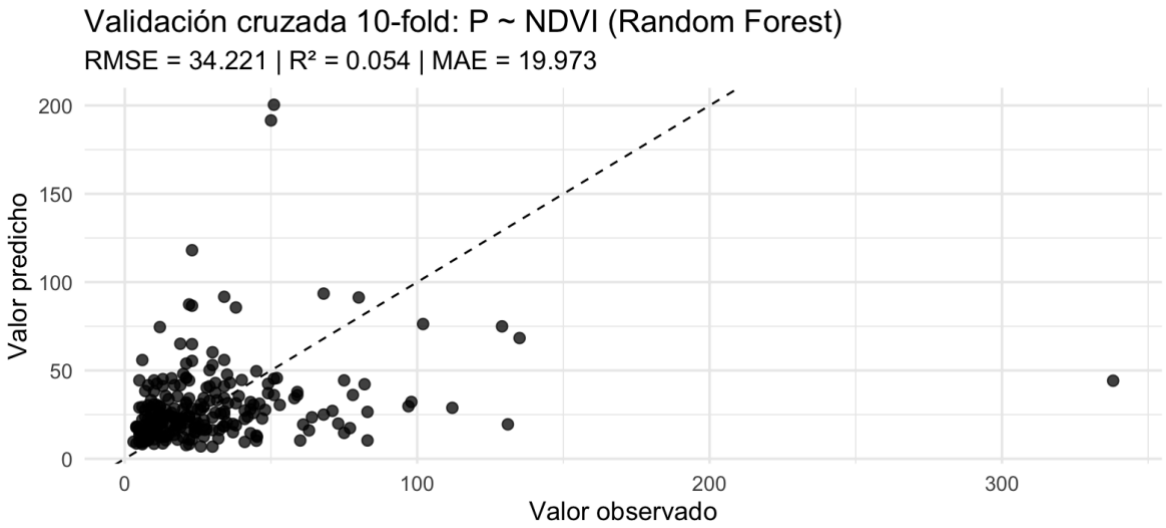

Relación entre NDVI y K
n = 239 | R² = 0.04
1500
1000
500
0
K
0.5
1.0
1.5
2.0
NDVI

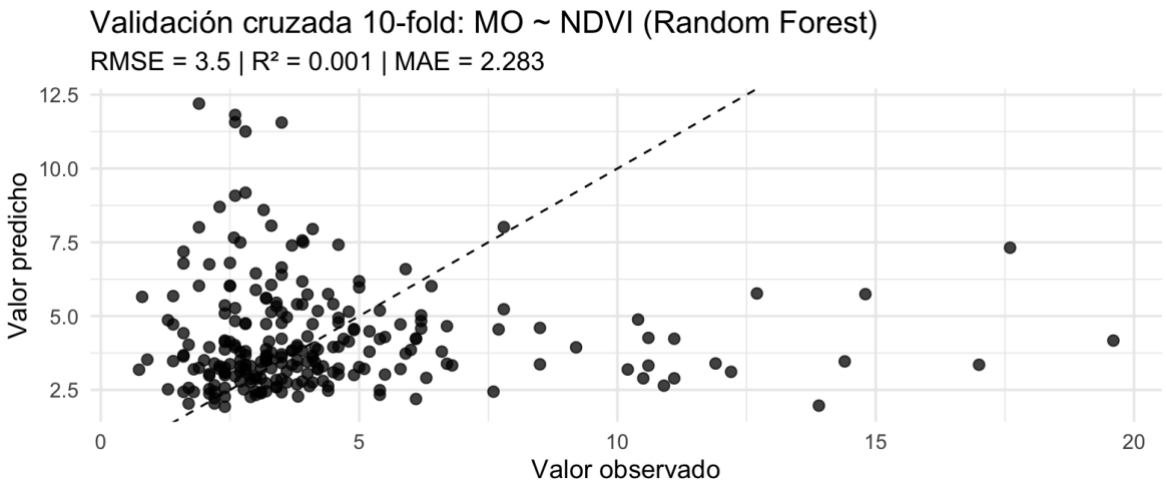

Relación entre NDVI y K - Random Forest
n = 239 | RMSE = 111.144 | R² = 0.781 | MAE = 77.1
1500
1000
500
0
K
0.5
1.0
1.5
2.0
NDVI

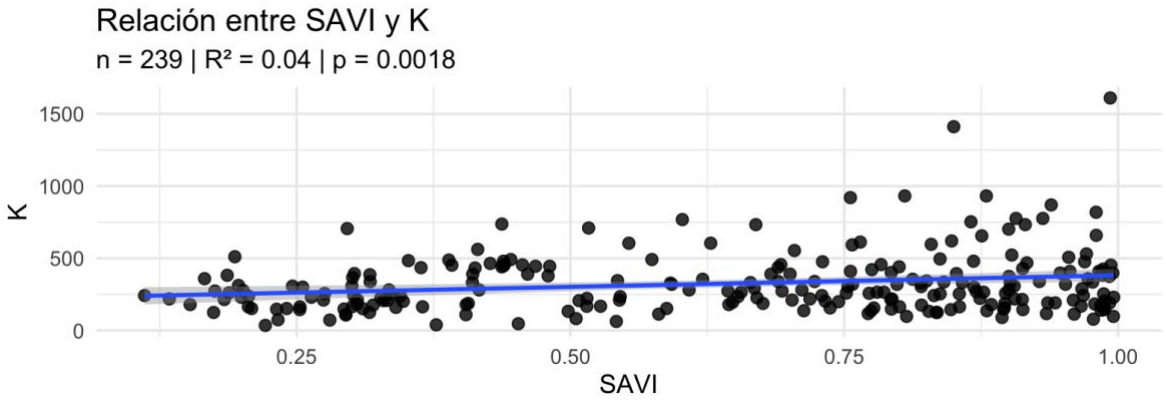

Relación entre NDVI y P - Random Forest
n = 239 | RMSE = 16.169 | R² = 0.771 | MAE = 9.791
300
200
100
0
P
0.5
1.0
1.5
2.0
NDVI

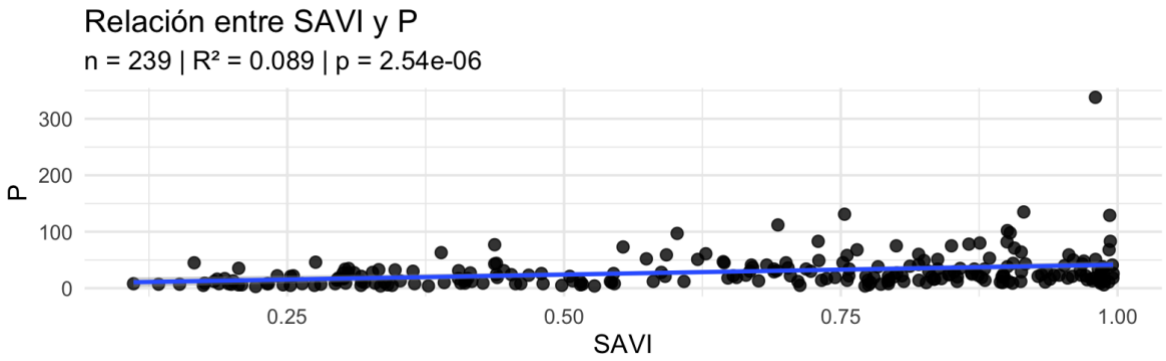

Relación entre NDVI y MO - Random Forest
n = 239 | RMSE = 1.731 | R² = 0.701 | MAE = 1.121
MO
NDVI
0
5
10
15
20
0.5
1.0
1.5
2.0

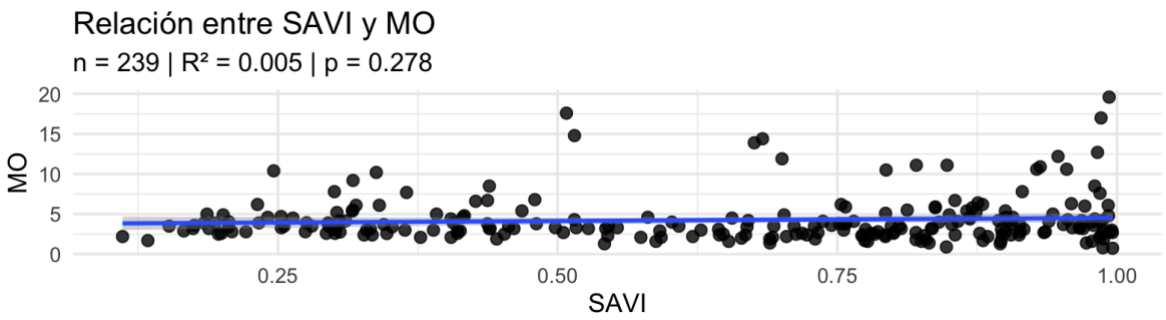

Validación cruzada 10-fold: K ~ NDVI (Random Forest)
RMSE = 233.486 | R² = 0.024 | MAE = 162.852
Valor predicho
Valor observado
250
500
750
1000
0
500
1000
1500

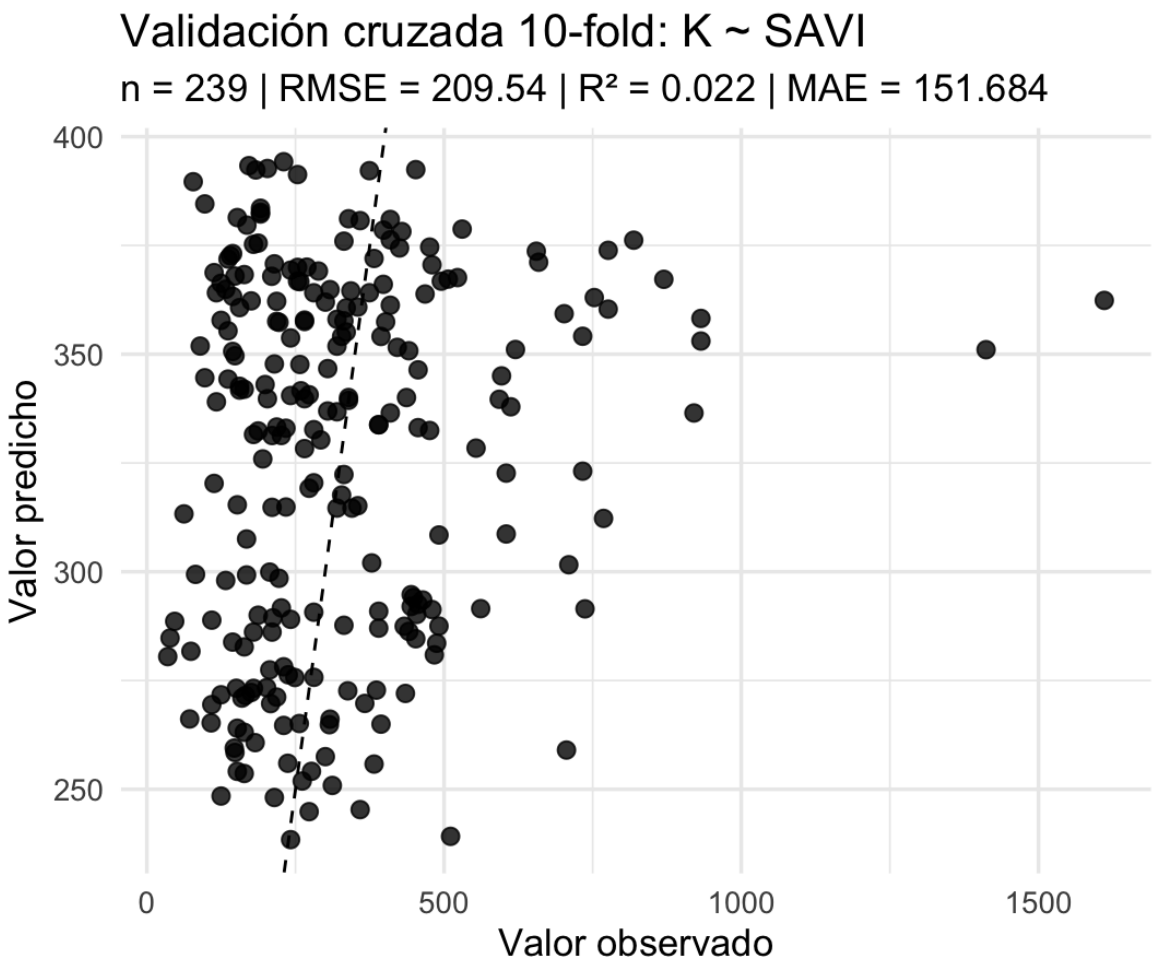

Validación cruzada 10-fold: P ~ NDVI (Random Forest)
RMSE = 34.221 | R² = 0.054 | MAE = 19.973
Valor predicho
Valor observado
0
50
100
150
200
0
100
200
300

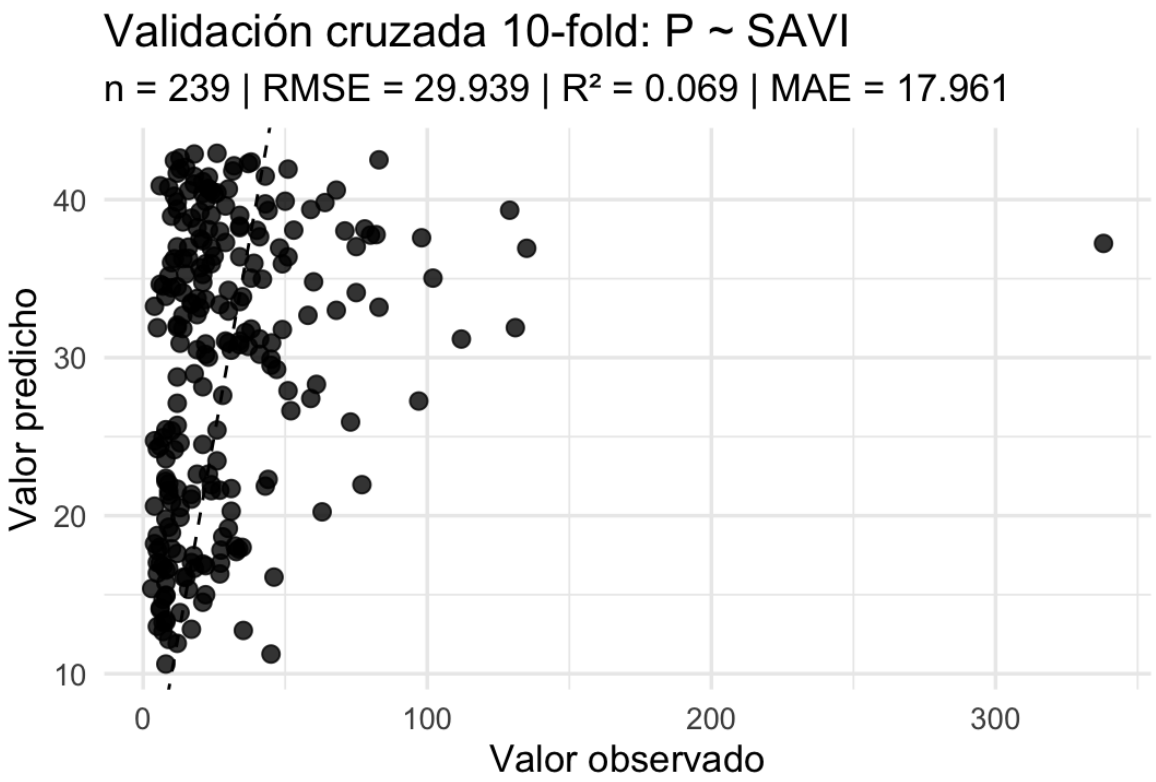

Validación cruzada 10-fold: MO ~ NDVI (Random Forest)
RMSE = 3.5 | R² = 0.001 | MAE = 2.283
Valor predicho
Valor observado
2.5
5.0
7.5
10.0
12.5
0
5
10
15
20

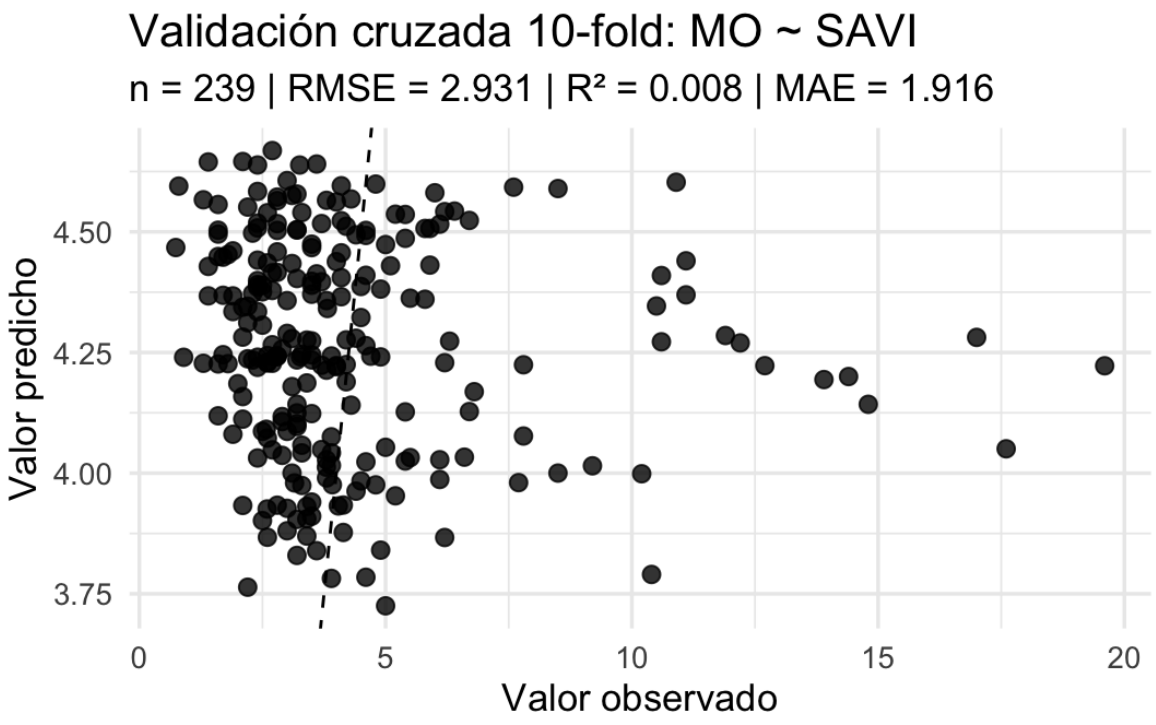

Relación entre SAVI y K
n = 239 | R² = 0.04 | p = 0.0018
K
SAVI
0
500
1000
1500
0.25
0.50
0.75
1.00

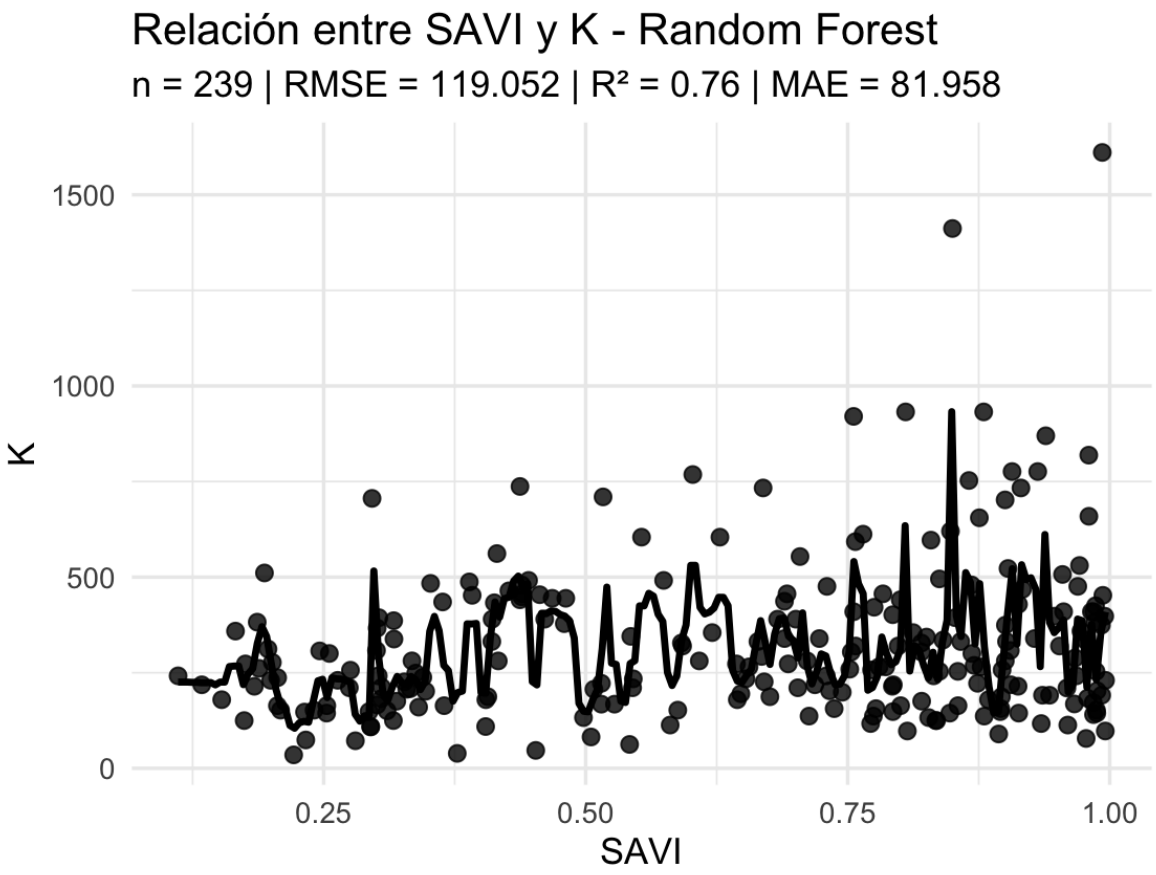

Relación entre SAVI y P
n = 239 | R² = 0.089 | p = 2.54e-06
P
300
200
100
0
0.25
0.50
0.75
1.00
SAVI

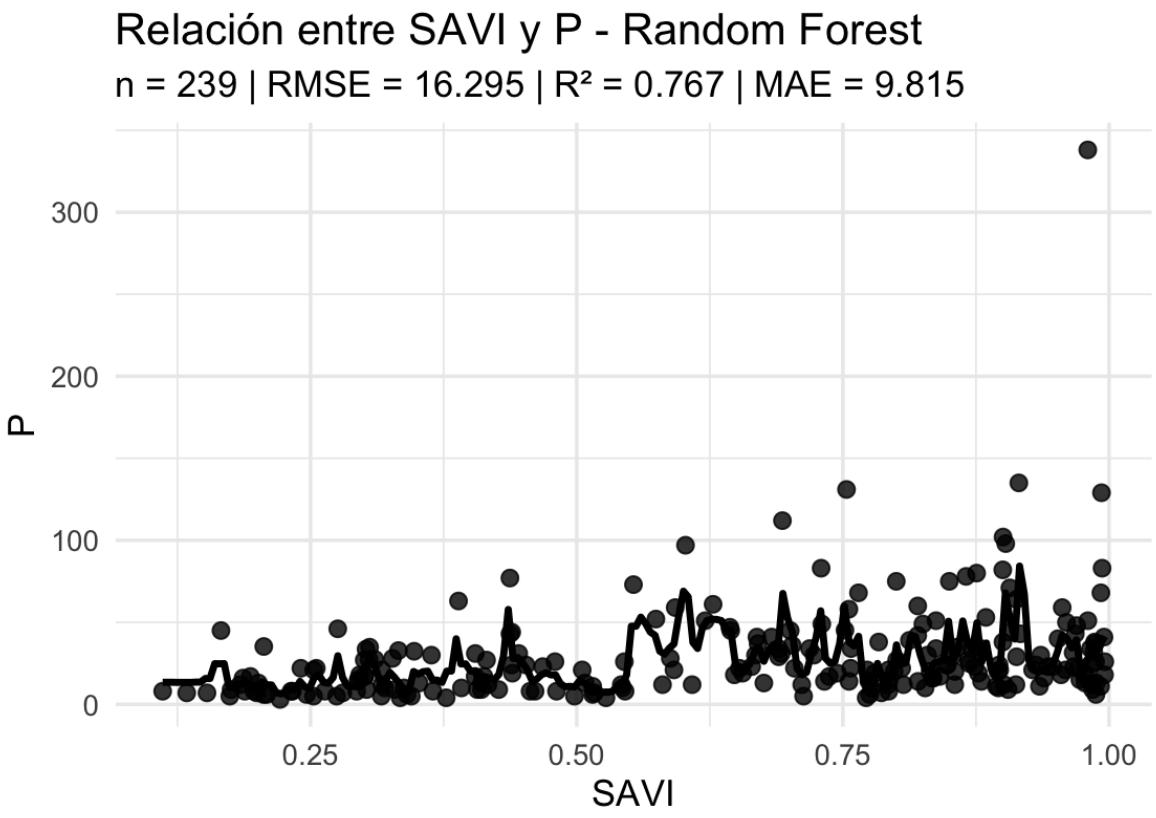

Relación entre SAVI y MO
n = 239 | R² = 0.005 | p = 0.278
MO
20
15
10
5
0
0.25
0.50
0.75
1.00
SAVI

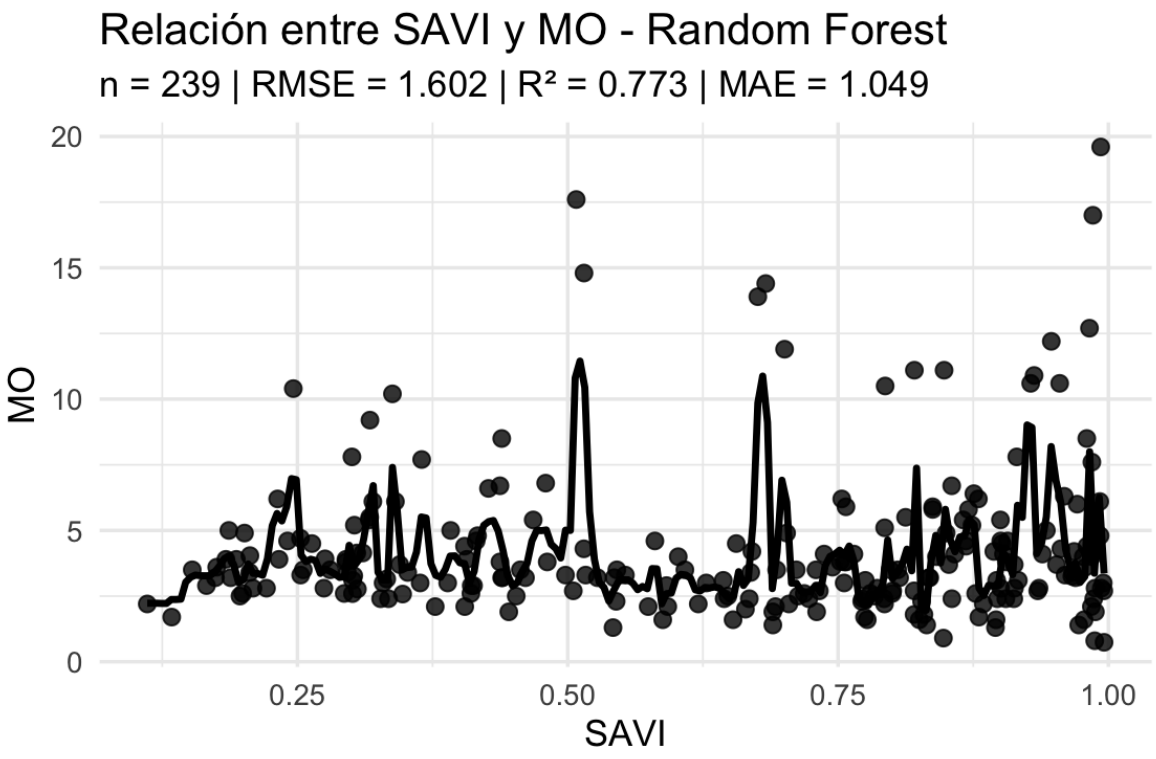

Validación cruzada 10-fold: K ~ SAVI
n = 239 | RMSE = 209.54 | R² = 0.022 | MAE = 151.684
Valor predicho
400
350
300
250
0
500
1000
1500
Valor observado

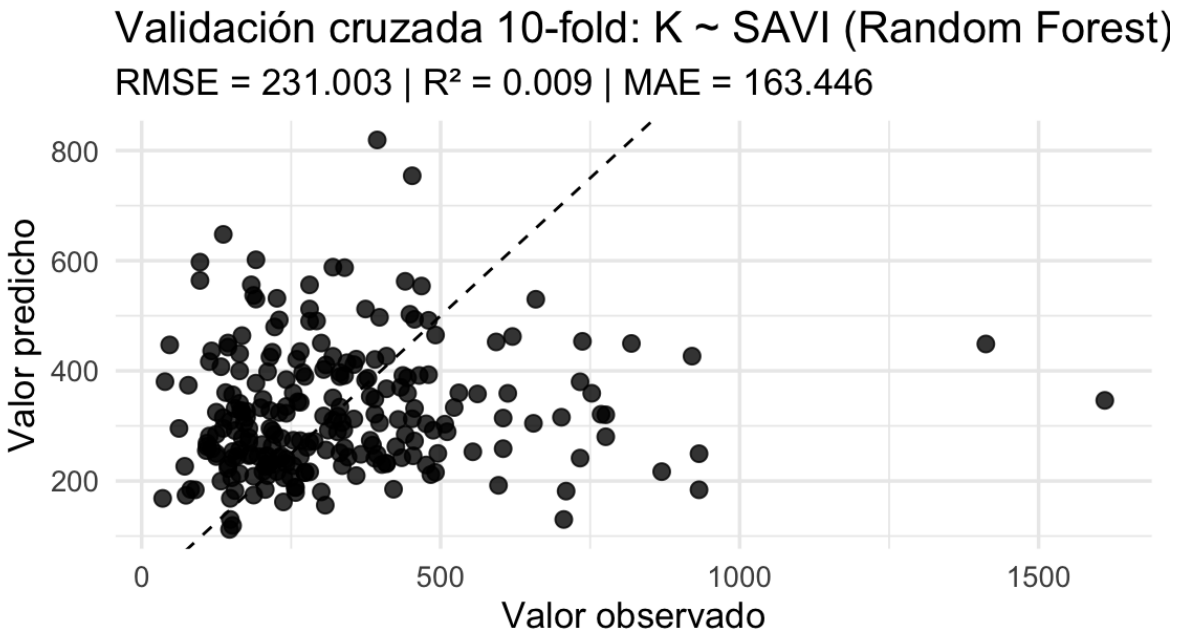

Validación cruzada 10-fold: P ~ SAVI
n = 239 | RMSE = 29.939 | R² = 0.069 | MAE = 17.961
Valor predicho
40
30
20
10
0
100
200
300
Valor observado

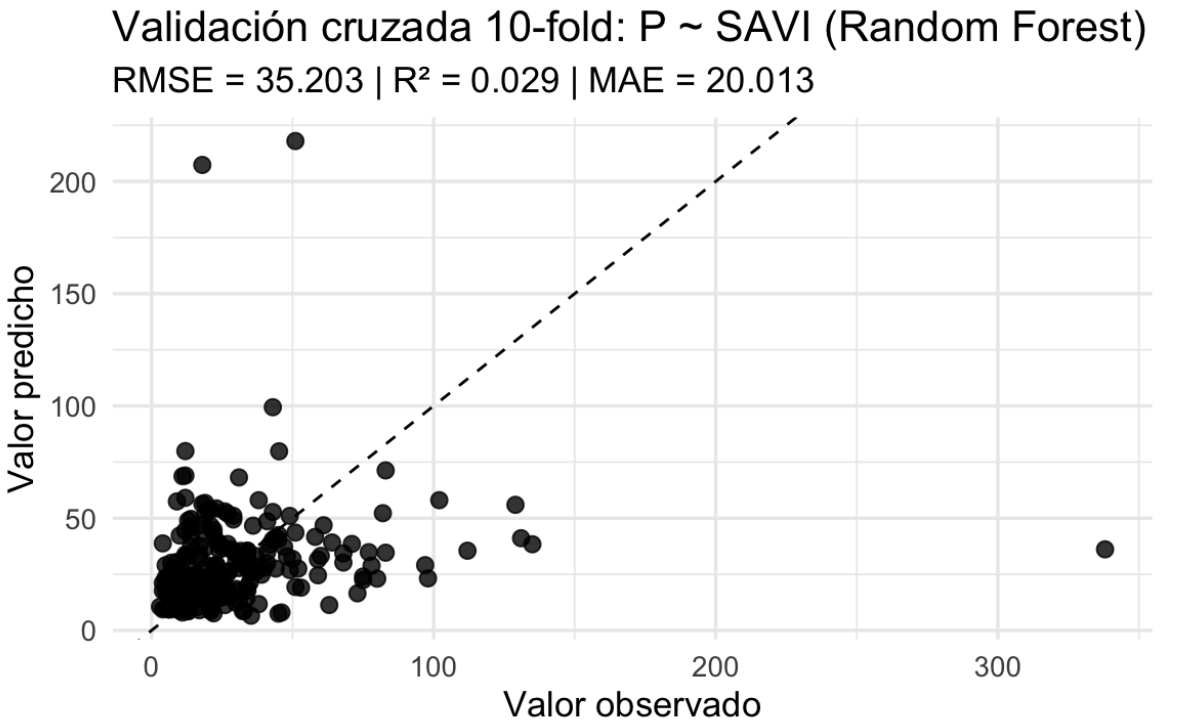

Validación cruzada 10-fold: MO ~ SAVI
n = 239 | RMSE = 2.931 | R² = 0.008 | MAE = 1.916
Valor predicho
4.50
4.25
4.00
3.75
0
5
10
15
20
Valor observado

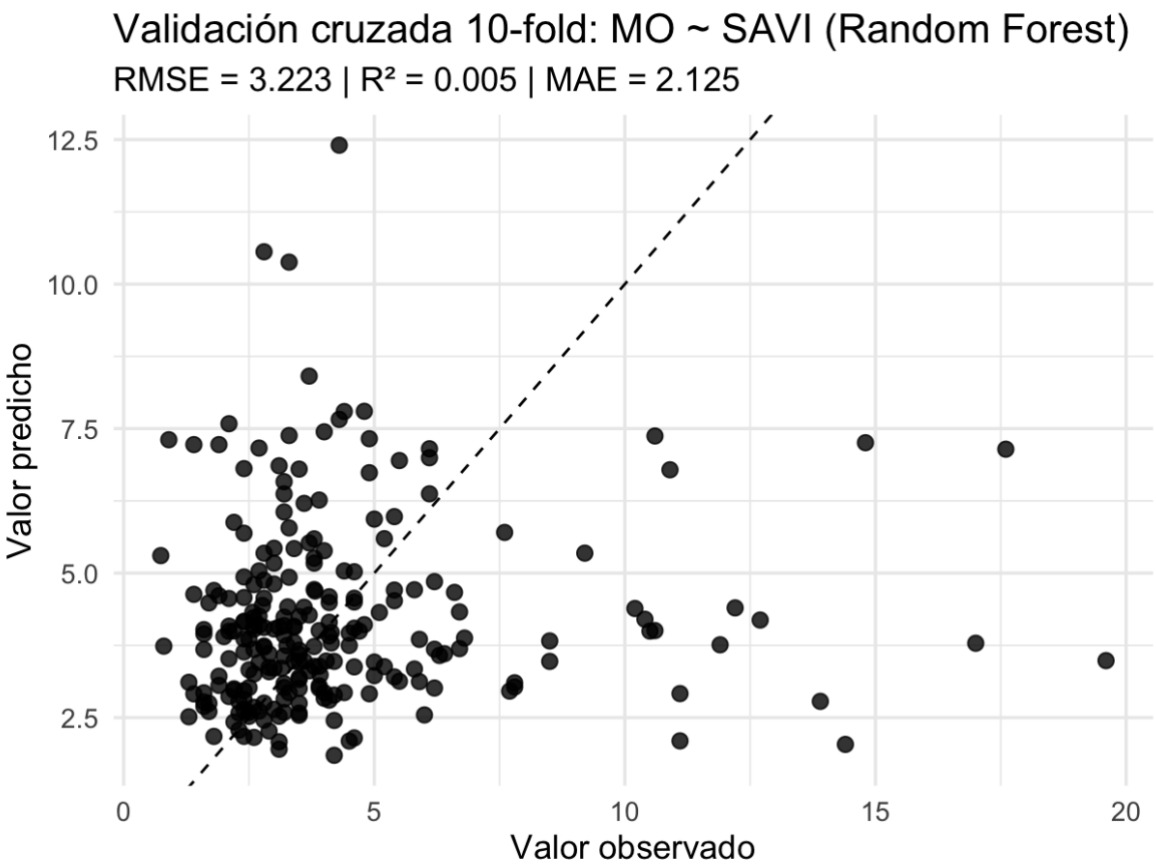

Relación entre SAVI y K - Random Forest
n = 239 | RMSE = 119.052 | R² = 0.76 | MAE = 81.958
1500
1000
500
0
K
0.25
0.50
0.75
1.00
SAVI

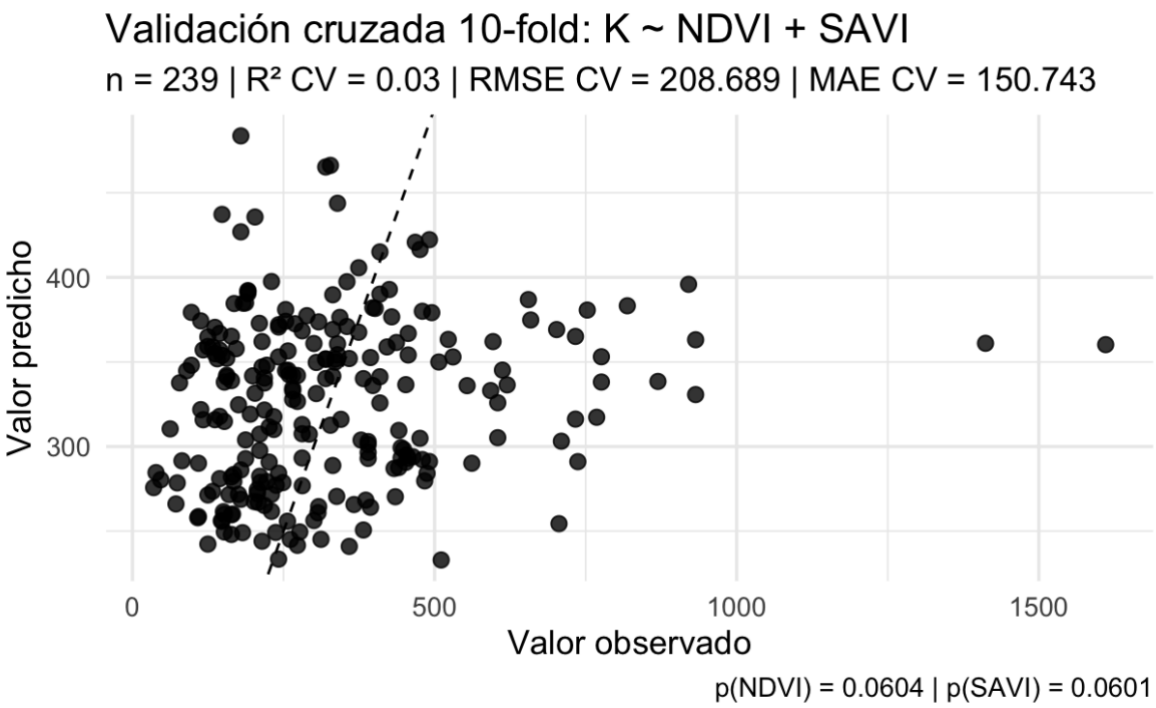

Relación entre SAVI y P - Random Forest
n = 239 | RMSE = 16.295 | R² = 0.767 | MAE = 9.815
P
SAVI
0
100
200
300
0.25
0.50
0.75
1.00

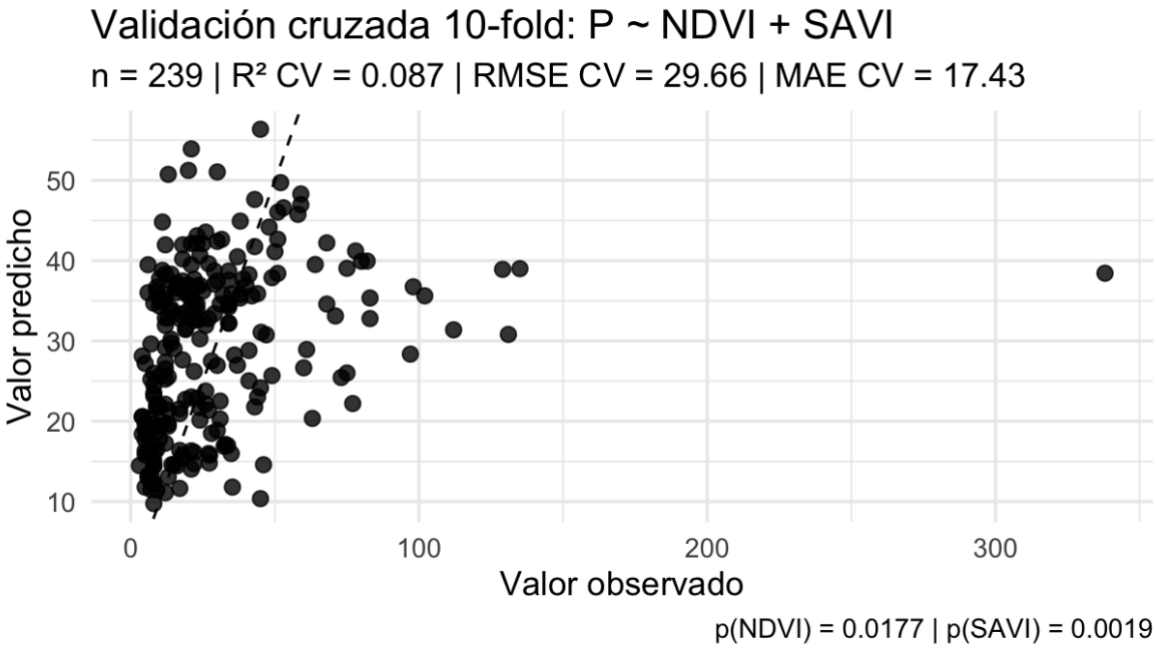

Relación entre SAVI y MO - Random Forest
n = 239 | RMSE = 1.602 | R² = 0.773 | MAE = 1.049
MO
SAVI
0
5
10
15
20
0.25
0.50
0.75
1.00

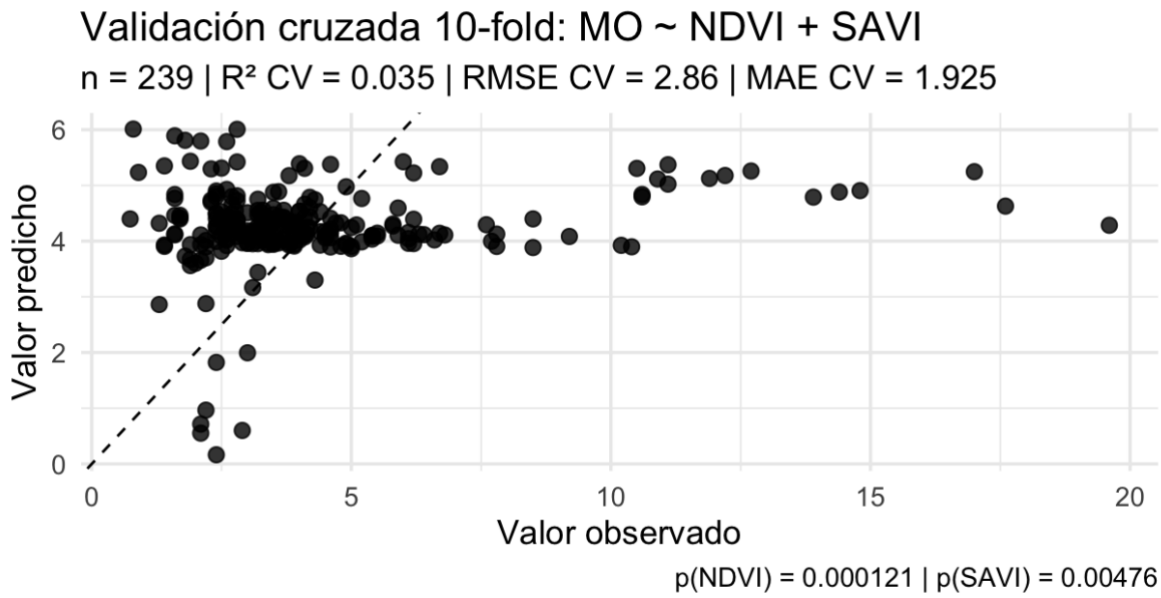

Validación cruzada 10-fold: K ~ SAVI (Random Forest)
RMSE = 231.003 | R² = 0.009 | MAE = 163.446
Valor predicho
800
600
400
200
0
500
1000
1500
Valor observado


Validación cruzada 10-fold: P ~ SAVI (Random Forest)
RMSE = 35.203 | R² = 0.029 | MAE = 20.013
Valor predicho
200
150
100
50
0
0
100
200
300
Valor observado


Validación cruzada 10-fold: MO ~ SAVI (Random Forest)
RMSE = 3.223 | R² = 0.005 | MAE = 2.125
Valor predicho
Valor observado
12.5
10.0
7.5
5.0
2.5
0
5
10
15
20


Validación cruzada 10-fold: K ~ NDVI + SAVI
n = 239 | R² CV = 0.03 | RMSE CV = 208.689 | MAE CV = 150.743
Valor predicho
400
300
0
500
1000
1500
Valor observado
p(NDVI) = 0.0604 | p(SAVI) = 0.0601


Validación cruzada 10-fold: P ~ NDVI + SAVI
n = 239 | R² CV = 0.087 | RMSE CV = 29.66 | MAE CV = 17.43
Valor predicho
50
40
30
20
10
0
100
200
300
Valor observado
p(NDVI) = 0.0177 | p(SAVI) = 0.0019




*Códigos*

**NDVI**

```
# ============================================================
# NDVI: GRAFICOS GLOBALES POR VARIABLE
# ============================================================

# install.packages(c("readxl", "dplyr", "purrr", "stringr", "ggplot2"))

library(readxl)
library(dplyr)
library(purrr)
library(stringr)
library(ggplot2)

# 1. Seleccionar archivo Excel
archivo <- file.choose()

# 2. Leer todas las hojas y consolidarlas
hojas <- excel_sheets(archivo)

datos <- map_dfr(hojas, function(h) {
  df <- read_excel(archivo, sheet = h)

  names(df) <- str_trim(names(df))
```

```
  # Corregir NVDI a NDVI
  if ("NVDI" %in% names(df)) {
    names(df)[names(df) == "NVDI"] <- "NDVI"
  }

  df$Sitio <- h
  df
})

# 3. Limpiar y asegurar tipos numéricos
datos <- datos %>%
  select(Sitio, ZONA, NDVI, pH, K, P, MO) %>%
  mutate(
    Sitio = as.factor(Sitio),
    ZONA  = as.factor(ZONA),
    NDVI  = as.numeric(NDVI),
    pH    = as.numeric(pH),
    K     = as.numeric(K),
    P     = as.numeric(P),
    MO    = as.numeric(MO)
  )

# 4. Carpeta donde se guardarán los gráficos
carpeta_salida <- "~/Desktop/graficos_estudio"
dir.create(carpeta_salida, showWarnings = FALSE)

# 5. Variables a graficar
variables <- c("pH", "K", "P", "MO")

# 6. Función para crear un gráfico por variable
crear_grafico <- function(variable, base) {

  df <- base %>%
    filter(!is.na(NDVI), !is.na(.data[[variable]]))

  mod <- lm(as.formula(paste(variable, "~ NDVI")), data = df)
  r2  <- round(summary(mod)$r.squared, 3)
  n   <- nrow(df)

  titulo_auto <- paste("Relación entre NDVI y", variable)
  subtitulo_auto <- paste("n =", n, "| R² =", r2)

  g <- ggplot(df, aes(x = NDVI, y = .data[[variable]])) +
    geom_point(size = 2.5, alpha = 0.8) +
    geom_smooth(method = "lm", se = TRUE, linewidth = 1) +
    labs(
```

```
      title = titulo_auto,
      subtitle = subtitulo_auto,
      x = "NDVI",
      y = variable
    ) +
    theme_minimal(base_size = 13)

  return(g)
}

# 7. Crear, mostrar y guardar los 4 gráficos
for (v in variables) {
  grafico <- crear_grafico(v, datos)

  print(grafico)

  ggsave(
    filename = file.path(carpeta_salida, paste0("grafico_", v, "_vs_NDVI.png")),
    plot = grafico,
    width = 9,
    height = 6,
    dpi = 300
  )
}

# 8. Mensaje final
cat("Listo.\n")
cat("Se generaron 4 gráficos en:\n")
cat(carpeta_salida, "\n")


# ============================================================
# NDVI: REGRESION LINEAL SIMPLE + VALIDACION CRUZADA 10-FOLD
# ============================================================

# install.packages(c("readxl", "dplyr", "purrr", "stringr", "caret", "ggplot2", "writexl"))

library(readxl)
library(dplyr)
library(purrr)
library(stringr)
library(caret)
library(ggplot2)
library(writexl)

# 1. Seleccionar archivo Excel
archivo <- file.choose()
```

```
# 2. Leer todas las hojas
hojas <- excel_sheets(archivo)

datos <- map_dfr(hojas, function(h) {
  df <- read_excel(archivo, sheet = h)
  names(df) <- str_trim(names(df))

  if ("NVDI" %in% names(df)) {
    names(df)[names(df) == "NVDI"] <- "NDVI"
  }

  df$Sitio <- h
  df
})

# 3. Revisar columnas esperadas
columnas_esperadas <- c("Sitio", "ZONA", "NDVI", "pH", "K", "P", "MO")
faltantes <- setdiff(columnas_esperadas, names(datos))

if (length(faltantes) > 0) {
  stop(
    paste(
      "Faltan estas columnas en el archivo:",
      paste(faltantes, collapse = ", ")
    )
  )
}

# 4. Limpiar y transformar datos
datos <- datos %>%
  select(Sitio, ZONA, NDVI, pH, K, P, MO) %>%
  mutate(
    Sitio = as.factor(Sitio),
    ZONA  = as.factor(ZONA),
    NDVI  = as.numeric(NDVI),
    pH    = as.numeric(pH),
    K     = as.numeric(K),
    P     = as.numeric(P),
    MO    = as.numeric(MO)
  )

# Si quieres filtrar NDVI fuera de rango teórico [-1, 1], descomenta esto:
# datos <- datos %>%
#   filter(!is.na(NDVI), NDVI >= -1, NDVI <= 1)

# 5. Configurar validación cruzada 10-fold
```

```
set.seed(123)

ctrl <- trainControl(
  method = "cv",
  number = 10,
  savePredictions = "final"
)

# 6. Variables respuesta
variables <- c("pH", "K", "P", "MO")

# 7. Función para ajustar y validar cada modelo
validar_modelo_cv <- function(variable, base) {

  df <- base %>%
    select(Sitio, ZONA, NDVI, all_of(variable)) %>%
    filter(!is.na(NDVI), !is.na(.data[[variable]]))

  if (nrow(df) < 10) {
    stop(paste("Muy pocos datos para la variable:", variable))
  }

  formula_modelo <- as.formula(paste(variable, "~ NDVI"))

  modelo_cv <- train(
    formula_modelo,
    data = df,
    method = "lm",
    trControl = ctrl,
    metric = "RMSE"
  )

  resumen <- data.frame(
    Variable = variable,
    n = nrow(df),
    RMSE_cv = modelo_cv$results$RMSE,
    R2_cv = modelo_cv$results$Rsquared,
    MAE_cv = modelo_cv$results$MAE
  )

  pred <- modelo_cv$pred
  pred$Variable <- variable

  grafico <- ggplot(pred, aes(x = obs, y = pred)) +
    geom_point(size = 2.5, alpha = 0.8) +
    geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
    labs(
```

```
      title = paste("Validación cruzada 10-fold:", variable, "~ NDVI"),
      subtitle = paste(
        "n =", nrow(df),
        "| RMSE =", round(modelo_cv$results$RMSE, 3),
        "| R² =", round(modelo_cv$results$Rsquared, 3),
        "| MAE =", round(modelo_cv$results$MAE, 3)
      ),
      x = "Valor observado",
      y = "Valor predicho"
    ) +
    theme_minimal(base_size = 13)

  list(
    modelo = modelo_cv,
    resumen = resumen,
    predicciones = pred,
    grafico = grafico
  )
}

# 8. Ejecutar validación para cada variable
resultados <- map(variables, ~ validar_modelo_cv(.x, datos))
names(resultados) <- variables

# 9. Crear tabla resumen
tabla_resumen <- bind_rows(map(resultados, "resumen"))
print(tabla_resumen)

# 10. Unir todas las predicciones
predicciones_todas <- bind_rows(map(resultados, "predicciones"))

# 11. Crear carpeta de salida en el escritorio
carpeta_salida <- "~/Desktop/validacion_cruzada_10fold"
dir.create(carpeta_salida, showWarnings = FALSE)

# 12. Guardar gráficos
for (v in variables) {
  ggsave(
    filename = file.path(carpeta_salida, paste0("CV10_", v,
"_observado_vs_predicho.png")),
    plot = resultados[[v]]$grafico,
    width = 8,
    height = 6,
    dpi = 300
  )
}
```

```
# 13. Guardar resultados en Excel y CSV
write_xlsx(
  list(
    Resumen_CV10 = tabla_resumen,
    Predicciones_CV10 = predicciones_todas
  ),
  path = file.path(carpeta_salida, "resultados_validacion_cruzada_10fold.xlsx")
)

write.csv(
  tabla_resumen,
  file.path(carpeta_salida, "resumen_validacion_cruzada_10fold.csv"),
  row.names = FALSE
)

write.csv(
  predicciones_todas,
  file.path(carpeta_salida, "predicciones_validacion_cruzada_10fold.csv"),
  row.names = FALSE
)

# 14. Mostrar gráficos en pantalla
for (v in variables) {
  print(resultados[[v]]$grafico)
}

# 15. Mensaje final
cat("\n==========================================\n")
cat("Proceso terminado correctamente.\n")
cat("Los resultados se guardaron en:\n")
cat(carpeta_salida, "\n")
cat("==========================================\n")


# ============================================================
# NDVI: RANDOM FOREST + VALIDACION CRUZADA 10-FOLD
# ============================================================

# install.packages(c("readxl", "dplyr", "purrr", "stringr", "ggplot2", "randomForest",
"rsample", "writexl"))

library(readxl)
library(dplyr)
library(purrr)
library(stringr)
library(ggplot2)
library(randomForest)
```

```
library(rsample)
library(writexl)

# 1. Seleccionar archivo Excel
archivo <- file.choose()

# 2. Leer todas las hojas y consolidar
hojas <- excel_sheets(archivo)

datos <- map_dfr(hojas, function(h) {
  df <- read_excel(archivo, sheet = h)
  names(df) <- str_trim(names(df))

  if ("NVDI" %in% names(df)) {
    names(df)[names(df) == "NVDI"] <- "NDVI"
  }

  df$Sitio <- h
  df
})

# 3. Revisar columnas esperadas
columnas_esperadas <- c("Sitio", "ZONA", "NDVI", "pH", "K", "P", "MO")
faltantes <- setdiff(columnas_esperadas, names(datos))

if (length(faltantes) > 0) {
  stop(
    paste(
      "Faltan estas columnas en el archivo:",
      paste(faltantes, collapse = ", ")
    )
  )
}

# 4. Limpiar base
datos <- datos %>%
  select(Sitio, ZONA, NDVI, pH, K, P, MO) %>%
  mutate(
    Sitio = as.factor(Sitio),
    ZONA  = as.factor(ZONA),
    NDVI  = as.numeric(NDVI),
    pH    = as.numeric(pH),
    K     = as.numeric(K),
    P     = as.numeric(P),
    MO    = as.numeric(MO)
  )
```

```
# Si quieres filtrar NDVI fuera de rango teórico [-1, 1], descomenta esto:
# datos <- datos %>%
#   filter(!is.na(NDVI), NDVI >= -1, NDVI <= 1)

# 5. Variables respuesta
variables <- c("pH", "K", "P", "MO")

# 6. Crear carpetas de salida
carpeta_salida <- "~/Desktop/random_forest_ndvi"
dir.create(carpeta_salida, showWarnings = FALSE)

carpeta_relacion <- file.path(carpeta_salida, "graficos_relacion")
carpeta_cv <- file.path(carpeta_salida, "graficos_validacion")

dir.create(carpeta_relacion, showWarnings = FALSE)
dir.create(carpeta_cv, showWarnings = FALSE)

# 7. Funciones de métricas
rmse_fun <- function(obs, pred) {
  sqrt(mean((obs - pred)^2, na.rm = TRUE))
}

mae_fun <- function(obs, pred) {
  mean(abs(obs - pred), na.rm = TRUE)
}

r2_fun <- function(obs, pred) {
  if (length(obs) < 2) return(NA_real_)
  cor(obs, pred, use = "complete.obs")^2
}

# 8. Ajuste RF + gráfico de relación
ajustar_rf_relacion <- function(variable, base) {

  df <- base %>%
    select(NDVI, all_of(variable)) %>%
    filter(!is.na(NDVI), !is.na(.data[[variable]]))

  set.seed(123)

  modelo <- randomForest(
    formula = as.formula(paste(variable, "~ NDVI")),
    data = df,
    ntree = 500,
    importance = TRUE
  )
```

```
  ndvi_seq <- seq(min(df$NDVI), max(df$NDVI), length.out = 200)
  grid_pred <- data.frame(NDVI = ndvi_seq)
  grid_pred$Prediccion_RF <- predict(modelo, newdata = grid_pred)

  df$Predicho_RF <- predict(modelo, newdata = df)

  rmse <- rmse_fun(df[[variable]], df$Predicho_RF)
  mae  <- mae_fun(df[[variable]], df$Predicho_RF)
  r2   <- r2_fun(df[[variable]], df$Predicho_RF)

  grafico <- ggplot(df, aes(x = NDVI, y = .data[[variable]])) +
    geom_point(size = 2.3, alpha = 0.75) +
    geom_line(
      data = grid_pred,
      aes(x = NDVI, y = Prediccion_RF),
      linewidth = 1.1,
      inherit.aes = FALSE
    ) +
    labs(
      title = paste("Relación entre NDVI y", variable, "- Random Forest"),
      subtitle = paste(
        "n =", nrow(df),
        "| RMSE =", round(rmse, 3),
        "| R² =", round(r2, 3),
        "| MAE =", round(mae, 3)
      ),
      x = "NDVI",
      y = variable
    ) +
    theme_minimal(base_size = 13)

  list(
    modelo = modelo,
    resumen = data.frame(
      Variable = variable,
      n = nrow(df),
      RMSE_ajuste = rmse,
      R2_ajuste = r2,
      MAE_ajuste = mae
    ),
    grafico = grafico
  )
}

# 9. Validación cruzada 10-fold
validar_rf_cv10 <- function(variable, base) {
```

```
df <- base %>%
  select(NDVI, all_of(variable)) %>%
  filter(!is.na(NDVI), !is.na(.data[[variable]]))

if (nrow(df) < 10) {
  stop(paste("No hay suficientes datos para 10-fold en:", variable))
}

set.seed(123)
folds <- vfold_cv(df, v = 10)

predicciones_cv <- map_dfr(seq_len(nrow(folds)), function(i) {
  split_i <- folds$splits[[i]]

  train_data <- analysis(split_i)
  test_data  <- assessment(split_i)

  set.seed(123)
  modelo_fold <- randomForest(
    formula = as.formula(paste(variable, "~ NDVI")),
    data = train_data,
    ntree = 500,
    importance = FALSE
  )

  preds <- predict(modelo_fold, newdata = test_data)

  data.frame(
    Fold = paste0("Fold_", i),
    Observado = test_data[[variable]],
    Predicho = preds
  )
})

rmse_cv <- rmse_fun(predicciones_cv$Observado, predicciones_cv$Predicho)
mae_cv  <- mae_fun(predicciones_cv$Observado, predicciones_cv$Predicho)
r2_cv   <- r2_fun(predicciones_cv$Observado, predicciones_cv$Predicho)

grafico_cv <- ggplot(predicciones_cv, aes(x = Observado, y = Predicho)) +
  geom_point(size = 2.3, alpha = 0.75) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
  labs(
    title = paste("Validación cruzada 10-fold:", variable, "~ NDVI (Random Forest)"),
    subtitle = paste(
      "RMSE =", round(rmse_cv, 3),
      "| R² =", round(r2_cv, 3),
      "| MAE =", round(mae_cv, 3)
```

```
      ),
      x = "Valor observado",
      y = "Valor predicho"
    ) +
    theme_minimal(base_size = 13)

  list(
    resumen = data.frame(
      Variable = variable,
      RMSE_cv10 = rmse_cv,
      R2_cv10 = r2_cv,
      MAE_cv10 = mae_cv
    ),
    predicciones = predicciones_cv,
    grafico = grafico_cv
  )
}

# 10. Ajuste de relación
resultados_relacion <- map(variables, ~ ajustar_rf_relacion(.x, datos))
names(resultados_relacion) <- variables

tabla_relacion <- bind_rows(map(resultados_relacion, "resumen"))
print(tabla_relacion)

for (v in variables) {
  print(resultados_relacion[[v]]$grafico)

  ggsave(
    filename = file.path(carpeta_relacion, paste0("RF_relacion_", v, "_vs_NDVI.png")),
    plot = resultados_relacion[[v]]$grafico,
    width = 9,
    height = 6,
    dpi = 300
  )
}

# 11. Validación cruzada
resultados_cv <- map(variables, ~ validar_rf_cv10(.x, datos))
names(resultados_cv) <- variables

tabla_cv <- bind_rows(map(resultados_cv, "resumen"))
print(tabla_cv)

predicciones_cv_todas <- bind_rows(
  lapply(seq_along(variables), function(i) {
    x <- resultados_cv[[i]]$predicciones
```

```
    x$Variable <- variables[i]
    x
  })
)

for (v in variables) {
  print(resultados_cv[[v]]$grafico)

  ggsave(
    filename = file.path(carpeta_cv, paste0("RF_CV10_", v,
"_observado_vs_predicho.png")),
    plot = resultados_cv[[v]]$grafico,
    width = 8,
    height = 6,
    dpi = 300
  )
}

# 12. Guardar resultados
write_xlsx(
  list(
    Resumen_relacion_RF = tabla_relacion,
    Resumen_validacion_RF_CV10 = tabla_cv,
    Predicciones_CV10 = predicciones_cv_todas
  ),
  path = file.path(carpeta_salida, "resultados_random_forest_ndvi.xlsx")
)

write.csv(
  tabla_relacion,
  file.path(carpeta_salida, "resumen_relacion_random_forest_ndvi.csv"),
  row.names = FALSE
)

write.csv(
  tabla_cv,
  file.path(carpeta_salida, "resumen_validacion_random_forest_ndvi_cv10.csv"),
  row.names = FALSE
)

write.csv(
  predicciones_cv_todas,
  file.path(carpeta_salida, "predicciones_random_forest_ndvi_cv10.csv"),
  row.names = FALSE
)

cat("\n==========================================\n")
```

```
cat("Proceso terminado correctamente.\n")
cat("Resultados guardados en:\n")
cat(carpeta_salida, "\n")
cat("==========================================\n")
```

**SAVI**

```
# ============================================================
# SAVI: REGRESION LINEAL SIMPLE
# ============================================================

# install.packages(c("readxl", "dplyr", "purrr", "stringr", "ggplot2", "writexl"))

library(readxl)
library(dplyr)
library(purrr)
library(stringr)
library(ggplot2)
library(writexl)

# 1. Seleccionar archivo Excel
archivo <- file.choose()

# 2. Leer todas las hojas y consolidar
hojas <- excel_sheets(archivo)

datos <- map_dfr(hojas, function(h) {
  df <- read_excel(archivo, sheet = h)
  names(df) <- str_trim(names(df))
  df$Sitio <- h
  df
})

# 3. Revisar columnas esperadas
columnas_esperadas <- c("Sitio", "ZONA", "SAVI", "pH", "K", "P", "MO")
faltantes <- setdiff(columnas_esperadas, names(datos))

if (length(faltantes) > 0) {
  stop(
    paste(
      "Faltan estas columnas en el archivo:",
      paste(faltantes, collapse = ", ")
    )
  )
}
```

```
# 4. Limpiar y ordenar base
datos <- datos %>%
  select(Sitio, ZONA, SAVI, pH, K, P, MO) %>%
  mutate(
    Sitio = as.factor(Sitio),
    ZONA  = as.factor(ZONA),
    SAVI  = as.numeric(SAVI),
    pH    = as.numeric(pH),
    K     = as.numeric(K),
    P     = as.numeric(P),
    MO    = as.numeric(MO)
  )

# 5. Variables respuesta
variables <- c("pH", "K", "P", "MO")

# 6. Crear carpeta de salida
carpeta_salida <- "~/Desktop/regresion_lineal_savi"
dir.create(carpeta_salida, showWarnings = FALSE)

# 7. Funciones de métricas
rmse_fun <- function(obs, pred) {
  sqrt(mean((obs - pred)^2, na.rm = TRUE))
}

mae_fun <- function(obs, pred) {
  mean(abs(obs - pred), na.rm = TRUE)
}

# 8. Función para ajustar modelo y extraer resultados
ajustar_modelo_lineal <- function(variable, base) {

  df <- base %>%
    select(SAVI, all_of(variable), Sitio, ZONA) %>%
    filter(!is.na(SAVI), !is.na(.data[[variable]]))

  formula_modelo <- as.formula(paste(variable, "~ SAVI"))
  modelo <- lm(formula_modelo, data = df)

  resumen_modelo <- summary(modelo)
  coeficientes <- coef(resumen_modelo)

  intercepto <- coeficientes["(Intercept)", "Estimate"]
  pendiente  <- coeficientes["SAVI", "Estimate"]
  p_intercepto <- coeficientes["(Intercept)", "Pr(>|t|)"]
  p_savi <- coeficientes["SAVI", "Pr(>|t|)"]
```

```
pred <- predict(modelo, newdata = df)

r2 <- resumen_modelo$r.squared
r2_aj <- resumen_modelo$adj.r.squared
rmse <- rmse_fun(df[[variable]], pred)
mae <- mae_fun(df[[variable]], pred)
n <- nrow(df)

tabla_resumen <- data.frame(
  Variable = variable,
  n = n,
  Intercepto = intercepto,
  Pendiente_SAVI = pendiente,
  p_Intercepto = p_intercepto,
  p_SAVI = p_savi,
  R2 = r2,
  R2_ajustado = r2_aj,
  RMSE = rmse,
  MAE = mae
)

df_pred <- data.frame(
  Observado = df[[variable]],
  Predicho = pred,
  SAVI = df$SAVI
)

titulo_auto <- paste("Relación entre SAVI y", variable)
subtitulo_auto <- paste(
  "n =", n,
  "| R² =", round(r2, 3),
  "| p =", signif(p_savi, 3)
)

grafico_relacion <- ggplot(df, aes(x = SAVI, y = .data[[variable]])) +
  geom_point(size = 2.5, alpha = 0.8) +
  geom_smooth(method = "lm", se = TRUE, linewidth = 1) +
  labs(
    title = titulo_auto,
    subtitle = subtitulo_auto,
    x = "SAVI",
    y = variable
  ) +
  theme_minimal(base_size = 13)

grafico_pred <- ggplot(df_pred, aes(x = Observado, y = Predicho)) +
  geom_point(size = 2.5, alpha = 0.8) +
```

```
    geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
    labs(
      title = paste("Observado vs Predicho -", variable),
      subtitle = paste(
        "RMSE =", round(rmse, 3),
        "| MAE =", round(mae, 3)
      ),
      x = "Valor observado",
      y = "Valor predicho"
    ) +
    theme_minimal(base_size = 13)

  list(
    modelo = modelo,
    resumen = tabla_resumen,
    predicciones = df_pred,
    grafico_relacion = grafico_relacion,
    grafico_pred = grafico_pred
  )
}

# 9. Ajustar modelos para las 4 variables
resultados <- map(variables, ~ ajustar_modelo_lineal(.x, datos))
names(resultados) <- variables

# 10. Unir resultados
tabla_resumen_final <- bind_rows(map(resultados, "resumen"))
predicciones_final <- bind_rows(
  lapply(seq_along(variables), function(i) {
    x <- resultados[[i]]$predicciones
    x$Variable <- variables[i]
    x
  })
)

print(tabla_resumen_final)

# 11. Guardar gráficos
for (v in variables) {
  ggsave(
    filename = file.path(carpeta_salida, paste0("relacion_SAVI_", v, ".png")),
    plot = resultados[[v]]$grafico_relacion,
    width = 9,
    height = 6,
    dpi = 300
  )
```

```
  ggsave(
    filename = file.path(carpeta_salida, paste0("observado_vs_predicho_", v, ".png")),
    plot = resultados[[v]]$grafico_pred,
    width = 8,
    height = 6,
    dpi = 300
  )
}

# 12. Guardar tablas
write_xlsx(
  list(
    Resumen_modelos = tabla_resumen_final,
    Predicciones = predicciones_final
  ),
  path = file.path(carpeta_salida, "resultados_regresion_lineal_savi.xlsx")
)

write.csv(
  tabla_resumen_final,
  file.path(carpeta_salida, "resumen_modelos_savi.csv"),
  row.names = FALSE
)

write.csv(
  predicciones_final,
  file.path(carpeta_salida, "predicciones_modelos_savi.csv"),
  row.names = FALSE
)

# 13. Mostrar gráficos en pantalla
for (v in variables) {
  print(resultados[[v]]$grafico_relacion)
  print(resultados[[v]]$grafico_pred)
}

# 14. Mensaje final
cat("\n==========================================\n")
cat("Proceso terminado correctamente.\n")
cat("Los resultados se guardaron en:\n")
cat(carpeta_salida, "\n")
cat("==========================================\n")
```

```
# ============================================================
# RANDOM FOREST CON SAVI
# Código completo
# ============================================================

# 1) Instalar paquetes si no los tienes
# install.packages("randomForest")
# install.packages("ggplot2")

# 2) Cargar librerías
library(randomForest)
library(ggplot2)

# 3) Revisar nombres de columnas del data frame
names(datos)

# 4) Definir variable respuesta
variable_respuesta <- "pH, K, P, MO"

# 5) Crear una base solo con las variables necesarias
datos_savi <- datos[, c(variable_respuesta, "SAVI")]

# 6) Eliminar filas con valores faltantes
datos_savi <- na.omit(datos_savi)

# 7) Ajustar modelo Random Forest
set.seed(123)

formula_rf_savi <- as.formula(paste(variable_respuesta, "~ SAVI"))

modelo_rf_savi <- randomForest(
  formula_rf_savi,
  data = datos_savi,
  ntree = 500,
  importance = TRUE
)

# 8) Ver resumen del modelo
print(modelo_rf_savi)

# 9) Generar predicciones
datos_savi$pred_rf_savi <- predict(modelo_rf_savi, newdata = datos_savi)

# 10) Calcular métricas
observado <- datos_savi[[variable_respuesta]]
predicho  <- datos_savi$pred_rf_savi
```

```
r2_rf_savi <- cor(observado, predicho)^2
rmse_rf_savi <- sqrt(mean((observado - predicho)^2))
mae_rf_savi <- mean(abs(observado - predicho))

cat("=====================================\n")
cat("MÉTRICAS RANDOM FOREST CON SAVI\n")
cat("=====================================\n")
cat("R2   =", round(r2_rf_savi, 4), "\n")
cat("RMSE =", round(rmse_rf_savi, 4), "\n")
cat("MAE  =", round(mae_rf_savi, 4), "\n")

# 11) Tabla con observado y predicho
tabla_predicciones_rf_savi <- data.frame(
  Observado = observado,
  Predicho = predicho
)

head(tabla_predicciones_rf_savi)

# 12) Gráfico observado vs predicho
grafico_rf_savi <- ggplot(datos_savi, aes(x = observado, y = predicho)) +
  geom_point(size = 3, alpha = 0.7, color = "darkgreen") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Observado vs Predicho - Random Forest con SAVI",
    x = "Valores observados",
    y = "Valores predichos"
  ) +
  annotate(
    "text",
    x = min(observado, na.rm = TRUE),
    y = max(predicho, na.rm = TRUE),
    label = paste0(
      "R2 = ", round(r2_rf_savi, 3),
      "\nRMSE = ", round(rmse_rf_savi, 3),
      "\nMAE = ", round(mae_rf_savi, 3)
    ),
    hjust = 0,
    vjust = 1,
    size = 5
  ) +
  theme_minimal()

print(grafico_rf_savi)

# 13) Importancia de variable
print(importance(modelo_rf_savi))
```

```
varImpPlot(modelo_rf_savi)

# 14) Guardar resultados en una tabla resumen
resultados_rf_savi <- data.frame(
  Modelo = "Random Forest con SAVI",
  Variable = variable_respuesta,
  R2 = r2_rf_savi,
  RMSE = rmse_rf_savi,
  MAE = mae_rf_savi
)

print(resultados_rf_savi)

# 15) Exportar resultados a CSV si quieres
# write.csv(resultados_rf_savi, "resultados_rf_savi.csv", row.names = FALSE)
# write.csv(tabla_predicciones_rf_savi, "predicciones_rf_savi.csv", row.names = FALSE)

# ============================================================
# VALIDACIÓN CRUZADA CON SAVI
# Código completo
# ============================================================

# 1) Instalar paquetes si no los tienes
# install.packages("caret")
# install.packages("ggplot2")
# install.packages("dplyr")

# 2) Cargar librerías
library(caret)
library(ggplot2)
library(dplyr)

# 3) Revisar nombres de columnas del data frame
names(datos)

# 4) Definir variable respuesta
variable_respuesta <- "pH, K, P, MO "

# 5) Crear una base solo con las variables necesarias
datos_savi <- datos[, c(variable_respuesta, "SAVI")]

# 6) Eliminar filas con valores faltantes
datos_savi <- na.omit(datos_savi)

# 7) Definir fórmula
formula_savi <- as.formula(paste(variable_respuesta, "~ SAVI"))
```

```
# 8) Configurar validación cruzada
set.seed(123)

control_cv <- trainControl(
  method = "cv",
  number = 10,
  savePredictions = "final"
)

# 9) Entrenar modelo lineal con validación cruzada
modelo_cv_savi <- train(
  formula_savi,
  data = datos_savi,
  method = "lm",
  trControl = control_cv
)

# 10) Ver resumen del modelo
print(modelo_cv_savi)

# 11) Extraer predicciones de validación cruzada
pred_savi <- modelo_cv_savi$pred

# 12) Calcular métricas
observado <- pred_savi$obs
predicho  <- pred_savi$pred

r2_savi <- cor(predicho, observado)^2
rmse_savi <- sqrt(mean((observado - predicho)^2))
mae_savi <- mean(abs(observado - predicho))

cat("=====================================\n")
cat("MÉTRICAS VALIDACIÓN CRUZADA CON SAVI\n")
cat("=====================================\n")
cat("R2   =", round(r2_savi, 4), "\n")
cat("RMSE =", round(rmse_savi, 4), "\n")
cat("MAE  =", round(mae_savi, 4), "\n")

# 13) Tabla con observado y predicho
tabla_predicciones_cv_savi <- data.frame(
  Observado = observado,
  Predicho = predicho
)

head(tabla_predicciones_cv_savi)

# 14) Gráfico observado vs predicho
```

```
grafico_cv_savi <- ggplot(pred_savi, aes(x = obs, y = pred)) +
  geom_point(size = 3, alpha = 0.7, color = "darkgreen") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Observado vs Predicho - Validación cruzada con SAVI",
    x = "Valores observados",
    y = "Valores predichos"
  ) +
  annotate(
    "text",
    x = min(pred_savi$obs, na.rm = TRUE),
    y = max(pred_savi$pred, na.rm = TRUE),
    label = paste0(
      "R2 = ", round(r2_savi, 3),
      "\nRMSE = ", round(rmse_savi, 3),
      "\nMAE = ", round(mae_savi, 3)
    ),
    hjust = 0,
    vjust = 1,
    size = 5
  ) +
  theme_minimal()

print(grafico_cv_savi)

# 15) Guardar resultados en una tabla resumen
resultados_cv_savi <- data.frame(
  Modelo = "Validación cruzada con SAVI",
  Variable = variable_respuesta,
  R2 = r2_savi,
  RMSE = rmse_savi,
  MAE = mae_savi
)

print(resultados_cv_savi)

# 16) Exportar resultados a CSV si quieres
# write.csv(resultados_cv_savi, "resultados_cv_savi.csv", row.names = FALSE)
# write.csv(tabla_predicciones_cv_savi, "predicciones_cv_savi.csv", row.names = FALSE)
```

**NDVI y SAVI en conjunto**

```
# ============================================================
# VALIDACIÓN CRUZADA CON NDVI + SAVI
# Código completo
# ============================================================

# 1) Instalar paquetes si no los tienes
# install.packages("caret")
# install.packages("ggplot2")
# install.packages("dplyr")

# 2) Cargar librerías
library(caret)
library(ggplot2)
library(dplyr)

# 3) Revisar nombres de columnas del data frame
names(datos)

# 4) Definir variable respuesta
variable_respuesta <- "pH, K, P, MO "

# 5) Crear una base solo con las variables necesarias
datos_ndvi_savi <- datos[, c(variable_respuesta, "NDVI", "SAVI")]

# 6) Eliminar filas con valores faltantes
datos_ndvi_savi <- na.omit(datos_ndvi_savi)

# 7) Definir fórmula
formula_ndvi_savi <- as.formula(paste(variable_respuesta, "~ NDVI + SAVI"))

# 8) Configurar validación cruzada
set.seed(123)

control_cv <- trainControl(
  method = "cv",
  number = 10,
  savePredictions = "final"
)

# 9) Entrenar modelo lineal con validación cruzada
modelo_cv_ndvi_savi <- train(
  formula_ndvi_savi,
  data = datos_ndvi_savi,
  method = "lm",
  trControl = control_cv
```

```
)

# 10) Ver resumen del modelo
print(modelo_cv_ndvi_savi)

# 11) Extraer predicciones de validación cruzada
pred_ndvi_savi <- modelo_cv_ndvi_savi$pred

# 12) Calcular métricas
observado <- pred_ndvi_savi$obs
predicho  <- pred_ndvi_savi$pred

r2_ndvi_savi <- cor(predicho, observado)^2
rmse_ndvi_savi <- sqrt(mean((observado - predicho)^2))
mae_ndvi_savi <- mean(abs(observado - predicho))

cat("================================================\n")
cat("MÉTRICAS VALIDACIÓN CRUZADA CON NDVI + SAVI\n")
cat("================================================\n")
cat("R2   =", round(r2_ndvi_savi, 4), "\n")
cat("RMSE =", round(rmse_ndvi_savi, 4), "\n")
cat("MAE  =", round(mae_ndvi_savi, 4), "\n")

# 13) Tabla con observado y predicho
tabla_predicciones_cv_ndvi_savi <- data.frame(
  Observado = observado,
  Predicho = predicho
)

head(tabla_predicciones_cv_ndvi_savi)

# 14) Gráfico observado vs predicho
grafico_cv_ndvi_savi <- ggplot(pred_ndvi_savi, aes(x = obs, y = pred)) +
  geom_point(size = 3, alpha = 0.7, color = "purple") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", color = "red") +
  labs(
    title = "Observado vs Predicho - Validación cruzada con NDVI + SAVI",
    x = "Valores observados",
    y = "Valores predichos"
  ) +
  annotate(
    "text",
    x = min(pred_ndvi_savi$obs, na.rm = TRUE),
    y = max(pred_ndvi_savi$pred, na.rm = TRUE),
    label = paste0(
      "R2 = ", round(r2_ndvi_savi, 3),
      "\nRMSE = ", round(rmse_ndvi_savi, 3),
```

```
    "\nMAE = ", round(mae_ndvi_savi, 3)
   ),
   hjust = 0,
   vjust = 1,
   size = 5
 ) +
 theme_minimal()

print(grafico_cv_ndvi_savi)

# 15) Guardar resultados en una tabla resumen
resultados_cv_ndvi_savi <- data.frame(
 Modelo = "Validación cruzada con NDVI + SAVI",
 Variable = variable_respuesta,
 R2 = r2_ndvi_savi,
 RMSE = rmse_ndvi_savi,
 MAE = mae_ndvi_savi
)

print(resultados_cv_ndvi_savi)

# 16) Exportar resultados a CSV si quieres
# write.csv(resultados_cv_ndvi_savi, "resultados_cv_ndvi_savi.csv", row.names = FALSE)
# write.csv(tabla_predicciones_cv_ndvi_savi, "predicciones_cv_ndvi_savi.csv", row.names
= FALSE)
```