



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Geodésica,  
Cartográfica y Topográfica

Desarrollo de un servicio web (Geoportal AgriService) para  
la monitorización de cultivos y asesoramiento de riego con  
datos de satélite.

Trabajo Fin de Máster

Máster Universitario en Ingeniería Geomática y Geoinformación

AUTOR/A: Amaluisa Peñaherrera, Marco Antonio

Tutor/a: Recio Recio, Jorge Abel

Cotutor/a: Mora Navarro, Joaquín Gaspar

CURSO ACADÉMICO: 2023/2024



Universitat Politècnica de València

Escuela Técnica Superior de Ingeniería Geodésica, Cartográfica y Topográfica ETSIGCT

Máster Universitario en Ingeniería Geomática y Geoinformación

Desarrollo de un servicio web (Geoportal AgriService) para la monitorización de cultivos y asesoramiento de riego con datos de satélite.

Autor:

Marco Antonio Amaluisa Peñaherrera

Tutores UPV:

Jorge Abel Recio Recio

Joaquín Gaspar Mora Navarro

Tutor empresa: Jorge Sánchez Zapero

Curso académico: 2023-2024

## **Agradecimientos**

Agradezco a Dios.

A mis padres, Marco y Myriam.

A mi hermana Antonella.

### Compromiso

"El presente documento ha sido realizado completamente por el firmante; no ha sido entregado como otro trabajo académico previo y todo el material tomado de otras fuentes ha sido convenientemente entrecorillado y citado su origen en el texto, así como referenciado en la bibliografía"

---

Marco Antonio Amaluisa

## Resumen

---

En el contexto de la problemática actual del cambio climático, el cuidado de los recursos hídricos y la necesidad de gestionar de forma eficaz este recurso en la agricultura, se desarrolla la presente investigación que se enmarca en los Objetivos de Desarrollo Sostenible (ODS), principalmente el Objetivo 13 Acción por el clima. Este trabajo pretende crear una herramienta que permita facilitar la toma de decisiones en cuanto a la gestión del recurso hídrico, y, que ofrezca al agricultor una adecuada gestión geoespacial para monitorización de cultivos. El proyecto se relaciona estrechamente con la agricultura sostenible, la seguridad alimentaria, y otros ODS (ODS2, ODS6, ODS9, ODS12).

El trabajo desarrolla como parte de las prácticas en empresa en el marco de trabajo de la empresa The Earth Observation Laboratory (EOLAB), siendo parte del proyecto "Desarrollo de un servicio web (geoportal AgriService) para la monitorización de cultivos y asesoramiento de riego con datos de satélite". La empresa ha desarrollado una cadena de procesado a partir de algoritmos de teledetección obtenidos en proyectos I+D. Hasta el desarrollo del actual proyecto, en la empresa no se ha desarrollado ninguna tecnología o interfaz para que el usuario pueda definir parcelas y acceder a la información de interés de la cadena de procesado de la empresa.

El trabajo de fin de máster tiene el objetivo de desarrollar un Geoportal como un servicio web agrícola "AgriService" de gestión geoespacial para monitorización de cultivos, integrando información obtenida a partir de datos de satélite de observación de la Tierra (Copernicus Sentinel 2A y B) y algoritmos de teledetección para estimar parámetros agronómicos de interés, involucrando herramientas de desarrollo y tratamiento de gran cantidad de información (Big Data) con ayuda de la potencialidad del cálculo en la nube que ofrece Google Earth Engine.

El desarrollo de un algoritmo para el cálculo de variables agronómicas como el índice de vegetación normalizado (NDVI), la temperatura en Superficie (LST), la fracción de cobertura Vegetal (FCOVER), la emisividad (EMI), el albedo (ALB) y la radiación neta (RN) son útiles para obtener la evapotranspiración (ET) equivalente a

la necesidad hídrica total en ausencia de lluvia, aplicando un modelo físico basado en medidas de temperatura de la superficie. Las variables biofísicas se obtienen siguiendo la metodología aplicada por la empresa Earth Observation Laboratory (EOLAB), modelos que se han desarrollado para la cartera de productos y cadena de procesamiento de Sentinel-2 que ofrece la empresa.

Se automatiza la parte de entradas y pre-procesamiento de la cadena de la empresa, mediante la integración de herramientas de interfaz de programación de aplicaciones (APIs) correspondientes para llevar la información al usuario final. En el desarrollo del geoportal se incluye un módulo de suscripción y registro de usuarios en función de la extensión de la parcela. Considerando que el servicio web es proporcionado por la empresa EOLAB, es necesario analizar las limitaciones comerciales de Google Earth Engine.

La automatización de cálculos, la gestión y control de usuarios, se desarrolla en el back-end del geoportal con la ayuda del lenguaje de programación Python, el uso de la API de Google Earth Engine, PostgreSQL y PostGIS para la base de datos y Django como framework de desarrollo web back-end. Para el front-end se utilizan lenguajes de desarrollo web como HTML, CSS y JavaScript, el visor geográfico se construye con la librería Leaflet de JavaScript. Además de esto se utiliza herramientas de desarrollo como Git y Docker y plataformas en la nube como GitHub para producción y despliegue de la aplicación.

El producto final es el geoportal AgriService como servicio web, que permite al usuario, registrarse y acceder a la información de su parcela, en un determinado intervalo de tiempo, visualizar series de tiempo de las variables biofísicas que solicite el usuario y un mapa de su parcela sobre el visor geográfico. La disponibilidad de actualidad de los datos viene marcada por la que ofrecen los conjuntos de datos disponibles en Google Earth Engine.

### **Palabras clave:**

desarrollo web, geoportal, teledetección satelital, agricultura, Google Earth Engine.

## Summary

---

In the current context of climate change issues, the care of water resources and the need to effectively manage this resource in agriculture, this research is framed in the Sustainable Development Goals (SDGs), mainly in Goal 13. Climate action. The aim of this work is creating a tool to facilitate decisions making in managing water resources, This tool offers the farmer an adequate geospatial management for crop monitoring. The project is closely related to sustainable agriculture, food security, and other SDGs (SDG2 - SDG6 - SDG9 - SDG12).

The develop of this work is part of an internships in the company The Earth Observation Laboratory (EOLAB) in the project "Development of a web service (geoportal AgriService) for crop monitoring and irrigation advice with satellite data". The company has developed a processing chain based on remote sensing algorithms obtained in R&D projects. Until the development of the current project, the company has not developed any technology or interface for the user to define plots and select information of interest, to access to the company's processing chain products. The Master's thesis aims to develop a Geoportal as an agricultural web service "AgriService" for geospatial management for crop monitoring and irrigation advice, integrating information obtained from Earth observation satellite data (Copernicus Sentinel 2A and B) and physical models useful for estimating agronomic parameters of interest, involving innovative tools for the management of water resources in agriculture and processing of large amounts of information (Big Data) with the help of the potential of cloud computing offered by Google Earth Engine.

The development of an algorithm for the calculation of agronomic variables such as normalized vegetation index (NDVI), surface temperature (LST), fraction of vegetation cover (FCOVER), emissivity (EMI), albedo (ALB) and net radiation (RN) are useful to obtain the evapotranspiration (ET) equivalent to the total water requirement in the absence of rain, applying a physical model based on surface temperature measurements. The biophysical variables are obtained following the methodology applied by the Earth Observation Laboratory (EOLAB) company, models that have

been developed for the Sentinel-2 product portfolio and processing chain offered by the company.

The input and pre-processing part of the company's chain is automated by integrating the corresponding application programming interface tools (APIs) to bring the information to the end user. In the development of the geoportal, a subscription and user registration module is included depending on the extent of the parcel. Considering that the web service is provided by EOLAB, it is necessary to analyze the commercial limitations of Google Earth Engine.

The automation of calculations, management and user control, is developed in the back-end of the geoportal with the help of the Python programming language, the use of the Google Earth Engine API, Postgres and Postgis for the database and Django as a web development Framework. For the Front-end web development languages such as HTML, CSS and JavaScript are used, the geographic viewer is built with the Leaflet JavaScript library. In addition to this, web development tools such as Git and Docker and cloud platforms such as GitHub are used for production and deployment of the application.

The final product is the AgriService geoportal as a web service, which allows the user to register and access the information of his plot, in a certain time interval, visualize time series of the biophysical variables requested by the user and a map of his plot on the geographic viewer, the availability of current data is marked by that offered by the Datasets available in Google Earth Engine.

**Key words:**

web development, geoportal, satellite remote sensing, agriculture, Google Earth Engine.



## Resumixen

---

En el context de la problemàtica actual del canvi climàtic, la cura dels recursos hídrics i la necessitat de gestionar de manera eficaç este recurs en l'agricultura, es desenvolupa la present investigació que s'emmarca en els Objectius de Desenvolupament Sostenible (ODS), principalment l'Objectiu 13 Acció pel clima. Este treball pretén crear una ferramenta que permeta facilitar la presa de decisions quant a la gestió del recurs hídric, i, que oferisca a l'agricultor una adequada gestió geoespacial per a monitoratge de cultius. El projecte es relaciona estretament amb l'agricultura sostenible, la seguretat alimentària, i altres ODS (ODS2, ODS6, ODS9, ODS12).

El treball desenvolupa com a part de les pràctiques en empresa en el marc de treball de l'empresa The Earth Observation Laboratory (EOLAB), sent part del projecte "Desenvolupament d'un servici web (geoportal AgriService) per al monitoratge de cultius i assessorament de reg amb dades de satèl·lit". L'empresa ha desenvolupat una cadena de processament a partir d'algorismes de teledetecció obtinguts en projectes I+D. Fins al desenvolupament de l'actual projecte, en l'empresa no s'ha desenvolupat cap tecnologia o interfície perquè l'usuari puga definir parcel·les i accedir a la informació d'interés de la cadena de processament de l'empresa.

El treball de fi de màster té l'objectiu de desenvolupar un Geoportal com un servici web agrícola "AgriService" de gestió geoespacial per a monitoratge de cultius, integrant informació obtinguda a partir de dades de satèl·lit d'observació de la Terra (Copernicus Sentinel 2A i B) i algorismes de teledetecció per a estimar paràmetres agronòmics d'interés, involucrant ferramentes de desenvolupament i tractament de gran quantitat d'informació (Big data) amb ajuda de la potencialitat del càlcul en el núvol que oferix Google Earth Engine.

El desenvolupament d'un algorisme per al càlcul de variables agronòmiques com l'índex de vegetació normalitzat (NDVI), la temperatura en Superfície (LST), la fracció de cobertura Vegetal (FCOVER), l'emissivitat (EMI), l'albedo (ALB) i la radiació neta (RN) són útils per a obtindre l'evapotranspiració (ET) equivalent a la necessitat

hídrica total en absència de pluja, aplicant un model físic basat en mesures de temperatura de la superfície. Les variables biofísiques s'obtenen seguint la metodologia aplicada per l'empresa Earth Observation Laboratory (EOLAB), models que s'han desenvolupat per a la cartera de productes i cadena de processament de Sentinel-2 que ofereix l'empresa.

S'automatitza la part d'entrades i pre-processament de la cadena de l'empresa, mitjançant la integració de ferramentes d'interfície de programació d'aplicacions (APIs) corresponents per a portar la informació a l'usuari final. En el desenvolupament del geoportal s'inclou un mòdul de subscripció i registre d'usuaris en funció de l'extensió de la parcel·la. Considerant que el servici web és proporcionat per l'empresa EOLAB, és necessari analitzar les limitacions comercials de Google Earth Engine.

L'automatització de càlculs, la gestió i control d'usuaris, es desenvolupa en el back-end del geoportal amb l'ajuda del llenguatge de programació Python, l'ús de la API de Google Earth Engine, PostgreSQL i PostGIS per a la base de dades i Django com framework de desenvolupament web back-end. Per al front-end s'utilitzen llenguatges de desenvolupament web com a HTML, CSS i JavaScript, el visor geogràfic es construeix amb la llibreria Leaflet de JavaScript. A més d'això s'utilitzen ferramentes de desenvolupament com Git i Docker i plataformes en el núvol com GitHub per a producció i desplegament de l'aplicació.

El producte final és el geoportal AgriService com a servici web, que permet a l'usuari, registrar-se i accedir a la informació de la seua parcel·la, en un determinat interval de temps, visualitzar sèries de temps de les variables biofísiques que sol·licite l'usuari i un mapa de la seua parcel·la sobre el visor geogràfic. La disponibilitat d'actualitat de les dades ve marcada per la qual ofereixen els conjunts de dades disponibles en Google Earth Engine.

#### **Paraules clau:**

desenvolupament web, geoportal, teledetecció satel·litària, agricultura, Google Earth Engine

## Índice de Figuras

Figura 1. Estructura de funcionalidad del geoportal de acuerdo con los objetivos planteados.....	21
Figura 2. Cadena de procesado de EOLAB funcional en el geoportal AgriService.....	26
Figura 3. Contenedores Docker en la VPS para el geoportal AgriService....	36
Figura 4. Esquema de la estructura de funcionamiento del geoportal AgriService.....	39
Figura 5. Configuraciones relevantes para el geoportal en el fichero <i>httpd.conf</i> .....	41
Figura 6. <i>Docker-compose.prod.yml</i> de la base de datos.....	43
Figura 7. <i>Docker-compose.prod.yml</i> de <i>pgAdmin</i> .....	44
Figura 8. Configuración del proxy de <i>pgAdmin</i> en el fichero <i>agriservice.conf</i> .....	44
Figura 9. <i>Dockerfile</i> .....	45
Figura 10. Fichero <i>docker-compose.prod.yml</i> de la aplicación Django .....	46
Figura 11. Organización de los ficheros del <i>front-end</i> .....	48
Figura 12. Formulario de registro de un nuevo usuario.....	49
Figura 13. Bienvenida, botones de registro y botón de cierre de sesión. ....	50
Figura 14. Formulario de autenticación y botón de inicio de sesión.....	51
Figura 15. Ficheros del back-end en el VPS .....	54
Figura 16. Cambio de formato de coordenadas en Python.....	56
Figura 17. JSON de salida generada por el <i>back-end</i> para generar la imagen de NDVI.....	58
Figura 18. Salida del <i>back-end</i> de AgriService .....	60
Figura 19. Configuración del fichero <i>settings.py</i> en el VPS para tener acceso al sitio de administración de Django. ....	63
Figura 20. Configuración de las variables necesarias para enviar e-mail desde el <i>back-end</i> en el fichero <i>settings.py</i> .....	64
Figura 21. Creación y especificación de los campos de la tabla parcelas. ...	65
Figura 22. Esquema los procesos de funcionamiento del geoportal AgriService.....	66

Figura 23. Diagrama de flujo del registro de un nuevo usuario en el geoportal AgriService.....	68
Figura 24. Flujo de obtención del mapa en el geoportal .....	70
Figura 25. Flujo de obtención de serie de tiempo en el geoportal. ....	72
Figura 26. Ficheros en el repositorio GitHub .....	74
Figura 27. EECUs obtenidas del sitio de administración del proyecto Google Cloud creado.....	76

## Índice de Tablas

Tabla 1. Proyectos I+D de EOLAB.....	17
Tabla 2. Productos agronómicos desarrollados por EOLAB.....	18
Tabla 3. Limitaciones identificadas del producto actual de EOLAB y potenciales ventajas de la solución propuesta. ....	19
Tabla 4. Aplicaciones web similares.....	23
Tabla 5. Bandas del conjunto de datos de Sentinel-2 en Google Earth Engine .....	24
Tabla 6. Valores de los factores de ponderación del albedo para Sentinel-2. .....	29
Tabla 7. Información del servidor (VPS).....	40
Tabla 8. Métodos de la clase <i>AgriServiceGEE2</i> utilizados para la generación de los productos de la cadena de procesamiento de EOLAB.....	59
Tabla 9. Configuración de las URLs y vistas en la aplicación <i>Django</i> .....	61
Tabla 10. URLs configuradas para la autenticación y registro de usuarios. ...	62
Tabla 11. Presupuesto parcial del costo del desarrollador.....	75
Tabla 12. Presupuesto de desarrollo del geoportal <i>AgriService</i> .....	77
Tabla 13. Presupuesto mensual del geoportal <i>AgriService</i> .....	78

## Índice

Resumen.....	4
1. Introducción.....	15
1.1. Antecedentes .....	15
1.1.1. Contexto.....	15
1.1.2. Marco de Trabajo: <i>Earth Observation Laboratory</i> (EOLAB).....	16
1.1.3. Motivación.....	18
1.2. Objetivos .....	20
1.3. Relación con Objetivos de Desarrollo Sostenible .....	21
1.4. Estado del Arte.....	23
2. Metodología .....	24
2.1. Productos de Teledetección.....	24
2.1.1. Datos de entrada: Copernicus Misión Sentinel-2.....	24
2.1.2. Algoritmos de teledetección .....	25
2.2. Tecnologías de desarrollo .....	32
2.2.1. Herramientas de desarrollo .....	33
2.2.2. Lenguajes de Programación.....	36
2.2.3. Almacenamiento y Base de datos .....	38
3. Resultados.....	39
3.1. Estructura del geoportal AgriService .....	39
3.1.1. Interfaz/servidor .....	40
3.1.2. Front-end.....	47
3.1.3. Back-end.....	53
3.1.4. Base de datos (usuario/parcela).....	64
3.2. Flujograma de los Procesos.....	65
3.2.1. Registro de usuario/parcela.....	67

3.2.2.	Mapas de las variables biofísicas .....	69
3.2.3.	Series de tiempo de las variables biofísicas.....	71
3.3.	Almacenamiento en la etapa de desarrollo.....	73
4.	Presupuesto.....	74
4.1.	Tiempo de Empleo del Desarrollador.....	74
4.2.	Costo de las Tecnologías de Desarrollo.....	75
4.2.1.	Precios de la Plataforma Google Earth Engine .....	75
4.3.	Presupuesto de desarrollo del proyecto.....	77
4.1.	Presupuesto mensual para continuidad del proyecto.....	78
5.	Conclusiones y discusión .....	79
5.1.	Cumplimiento de Objetivos Iniciales.....	79
5.2.	Conclusiones Técnicas.....	80
5.3.	Fortalezas del Geoportal .....	81
5.4.	Mejoras Futuras.....	82
6.	Bibliografía.....	83
7.	Anexos.....	85
7.1.	Manual del Usuario.....	85
7.2.	Documentación de los métodos para el Cloud Computing (CC) en Google Earth Engine (GEE) .....	85
7.3.	Relación del trabajo con los Objetivos de Desarrollo Sostenible de la Agenda 2030.....	85

# 1. Introducción

---

## 1.1. Antecedentes

En esta sección se describen los antecedentes de este Trabajo Final de máster (TFM), que consisten en el contexto, el marco de trabajo para el desarrollo del proyecto (realizado en la empresa Earth Observation Laboratory, EOLAB) y la motivación de este. Después se describen los objetivos y la relación de este TFM con los objetivos de desarrollo sostenible de la Agenda 2030 de la ONU. Adicionalmente se realiza un estado del arte sobre algunas aplicaciones web similares a la desarrollada en este TFM.

### 1.1.1. Contexto

El cambio climático es uno de los mayores desafíos globales de nuestro tiempo, siendo el incremento de la temperatura (ligado a la sequía y escasez del agua) uno de los principales asuntos de interés a nivel mundial, ya que impacta profundamente en numerosos sectores de las sociedades actuales, como la agricultura.

El cambio climático tiene un impacto significativo en la agricultura a nivel global, afectando tanto la producción de alimentos como la seguridad alimentaria. Los efectos varían según la región y el tipo de cultivo. Los agricultores enfrentan retos como sequías más frecuentes y cambios en los patrones de precipitación, que demandan una gestión más eficiente y adaptativa de los recursos hídricos.

El aumento de temperatura y las alteraciones en los patrones climáticos también tienen un impacto significativo en la agricultura, provocando olas de calor con temperaturas extremas que pueden dañar los cultivos. Por todo ello, es necesaria una adecuada gestión de los recursos hídricos disponibles.

Actualmente, gracias a los satélites de observación de la Tierra, existe una gran cantidad de datos disponibles que permiten monitorizar los cambios producidos en la superficie terrestre y desarrollar multitud de aplicaciones para tomar decisiones acertadas para mitigar los efectos del cambio climático.



Ante estos desafíos, la presente investigación se centra en desarrollar soluciones tecnológicas que ayuden a los agricultores a tomar decisiones sobre el aprovechamiento del agua para adaptarse a las nuevas condiciones climáticas y a gestionar mejor los recursos disponibles.

### 1.1.2. Marco de Trabajo: *Earth Observation Laboratory* (EOLAB).

Este TFM se ha desarrollado a través de un convenio de prácticas con la empresa EOLAB. Esta empresa se establece en 2006 como una empresa de I+D, fundada por un equipo con amplia experiencia en ciencias de la Tierra y teledetección para el estudio del medio ambiente. Desde su creación, ha desarrollado proyectos de investigación, publicaciones científicas y servicios especializados, consolidándose como una empresa en aplicaciones terrestres y productos de observación de la Tierra. El trabajo de EOLAB se centra en la adquisición de datos de campo, la evaluación y validación científica de productos de satélite, y la creación de productos biofísicos mediante el desarrollo e implementación de líneas de procesamiento y algoritmos avanzados. Todo esto enmarcado en la sostenibilidad y la conciencia sobre el cambio climático, integrando estos principios en todos sus proyectos.

EOLAB colabora estrechamente con instituciones académicas, centros de investigación y organismos internacionales, siendo parte del avance del conocimiento en la observación de la Tierra y la respuesta a los desafíos globales del medio ambiente.

La empresa ha trabajado en los últimos años en una serie de proyectos de investigación con el objetivo de desarrollar algoritmos y cadenas de procesamiento que ofrezcan información agronómica mediante tecnologías de observación de la Tierra y teledetección. Dichos proyectos, cofinanciados con fondos del Instituto Valenciano de la Competitividad Empresarial - IVACE (PIDI-CV) y el Centro para el Desarrollo Tecnológico y la Innovación - CDTI se describen en la Tabla 1.

Tabla 1. Proyectos I+D de EOLAB.

Proyecto (financiación)	Periodo	Descripción del proyecto
SECUSEN-2 (IVACE)	2017	Estimación de variables biofísicas para el Seguimiento y monitorización de los Cultivos mediante datos de satélite Sentinel-2 y Landsat-8 de alta resolución espacial.
CULTISAT (IVACE)	2019	Clasificación automática de cultivos a partir de imágenes de satélite de alta resolución espaciotemporal Sentinel-2.
SENYREG (IVACE)	2020	Estimación de las necesidades hídricas de los cultivos de la Comunidad Valenciana a altas resoluciones espaciotemporales con datos satelitales de Sentinel-2.
AGRISEND-123 (CDTI)	2019-2022	Estimación de indicadores agronómicos de alta resolución espacio temporal a partir de observaciones de satélites Sentinel-1-2-3.

En estos proyectos, se diseñaron los algoritmos de teledetección que usan como datos de entrada observaciones de satélites de observación de la Tierra para la estimación de diferentes productos de información agronómica, tanto para zonas piloto de estudio (a nivel de parcelas) como para el área geográfica de la Comunidad Valenciana.

Gracias a estos algoritmos se implementó una cadena de procesado que opera a nivel interno, dentro de la estructura computacional de la empresa, para generar una cartera de productos geo-referenciados que permiten monitorizar el estado de la vegetación y establecer las necesidades de riego.

La Tabla 2 detalla los productos agronómicos desarrollados a nivel interno por EOLAB en los proyectos mencionados.

**Tabla 2.** Productos agronómicos desarrollados por EOLAB.

Familia de productos	Variables proporcionadas
<i>Índices de vegetación</i>	<i>Índice de vegetación de diferencia normalizada (NDVI)</i>
<i>Variables biofísicas:</i>	<i>Índice de superficie de Área Foliar (LAI)</i>
	<i>Fracción de Cobertura Vegetal (FCOVER)</i>
	<i>Fracción de radiación fotosintéticamente activa absorbida por la cubierta vegetal (FAPAR)</i>
<b>Clasificación</b>	<i>Clasificación de tipo de cultivos.</i>
<b>Temperatura</b>	<i>Temperatura de la superficie la superficie (LST)</i>
<b>Necesidad hídrica</b>	Evapotranspiración (ET)

### 1.1.3. Motivación

El uso eficiente del recurso hídrico en la agricultura junto con la cantidad de datos de teledetección ofrecidos por los satélites de observación de la Tierra y herramientas para el manejo de dichos datos, motivan la creación de una plataforma innovadora que permita el acceso a la información de una manera amigable con los usuarios para la adecuada toma de decisiones.

EOLAB ha desarrollado una serie de productos agronómicos (ver Tabla 2) en diversos proyectos de I+D anteriormente descritos, pero no se ha desarrollado ninguna tecnología ni interfaz para que el usuario pueda definir parcelas y acceder a la información de interés, quedando su ámbito de explotación comercial muy limitada o casi nula. Por tanto, es necesario desarrollar una solución innovadora en el ámbito de las tecnologías de la información para poder extender estos desarrollos a diversos usuarios, que puedan ser capaces de interactuar para definir sus necesidades y contratar los servicios agronómicos que pretendemos ofrecer para una mayor eficiencia en la gestión del agua y monitorización del cultivo. El desarrollo de un geoportal daría una solución a dicha limitación.

Es importante mencionar que se ha destacado que la cadena de procesado de la empresa funciona a nivel interno en su infraestructura computacional, lo que genera coste para la empresa tanto de recursos hardware (almacenamiento en discos) como software (procesadores dedicados). La solución desarrollada en este TFM tiene como principal motivación el uso de tecnologías *Cloud Computing* (CC), como *Google Earth Engine* (GEE) para la generación de los productos agronómicos, que serán accesibles a través del geoportal donde el usuario podrá acceder y monitorizar su zona de interés.

El resultado final, por tanto, es un geoportal, que ejecuta los procesos de cálculo en la nube y ofrecerá un servicio preparado para su comercialización. La Tabla 3 identifica las limitaciones que actualmente tienen los productos desarrollados por la empresa y las potenciales ventajas que ofrecerá el desarrollo de la solución propuesta.

**Tabla 3.** Limitaciones identificadas del producto actual de EOLAB y potenciales ventajas de la solución propuesta.

Funcionalidad del producto actual	Limitaciones	Funcionalidad de la nueva solución basada en el geoportal	Ventajas
Procesado local	Demanda de procesadores	Procesado en la nube	Liberación de recursos de procesado Reducción de costes
Almacenamiento local	Necesidad de descargar imágenes	Almacenamiento en la nube	Acceso a catálogos online
	Espacio de almacenamiento en el disco		Reducción de costes Liberación de espacio en disco
No existe interfaz/herramienta	Necesidad de personal técnico dedicado	Acceso al geoportal	Herramienta interactiva donde puede acceder a los productos
	El usuario no tiene ninguna forma efectiva y eficiente de acceder al producto		Permite gestión y control de usuario
	Imposibilidad de comercializar de forma efectiva		La herramienta permite la contratación directa del servicio

## 2. Objetivos

---

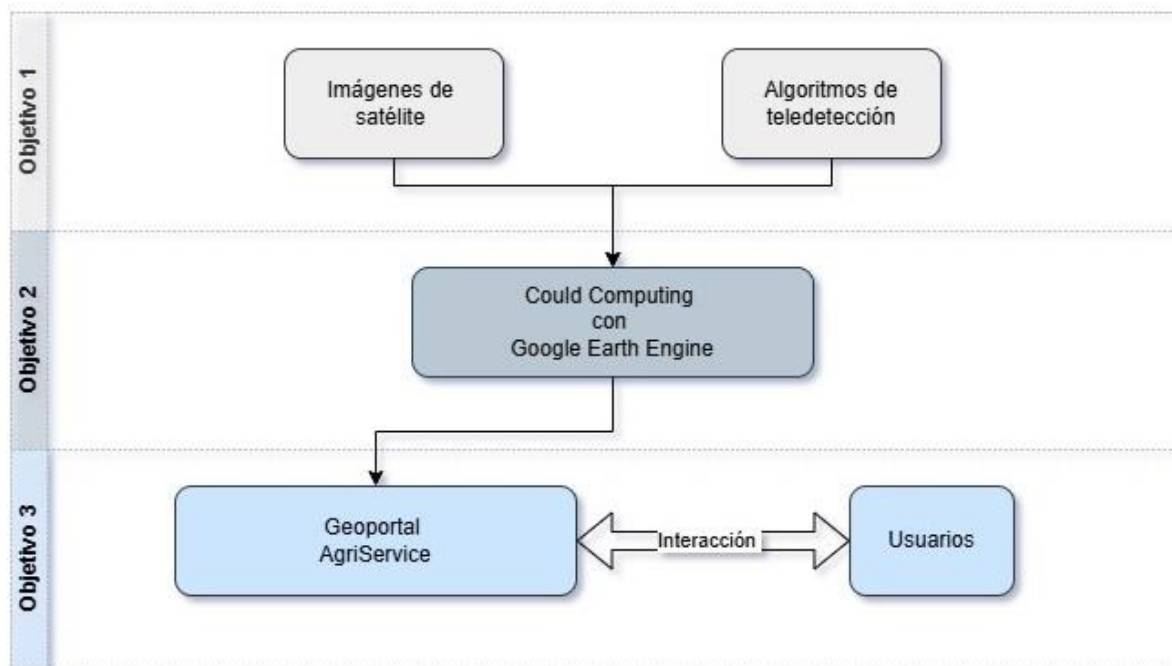
Ante los antecedentes previamente descritos, el objetivo principal de este TFM es proporcionar a los agricultores una herramienta que les permita acceder a la cartera de productos que ha desarrollado la empresa EOLAB para contar con información actualizada sobre el estado de sus cultivos. Los productos se basarán en datos satelitales y modelos de teledetección procesados en la nube. Para esto se definen los objetivos descritos a continuación, que se estructuran en la Figura 1 para componer la funcionalidad del geoportal.

OBJETIVO 1: Ofrecer una serie de productos derivados con datos de satélites de observación de la Tierra desarrollados por la empresa, que permiten monitorizar el estado de la vegetación (NDVI, FCOVER), la temperatura de la superficie (LST) y la evapotranspiración (ET, que es equivalente a la necesidad hídrica del cultivo). Se tendrá acceso tanto a los mapas geográficos de las variables de interés como a las series temporales de los últimos meses.

OBJETIVO 2: Procesar los algoritmos en la nube para ahorrar recursos a la empresa EOLAB a nivel de Software y Hardware a través de la tecnología *Cloud Computing* (CC) que ofrece *Google Earth Engine* (GEE), con técnicas de teledetección y desarrollo web.

OBJETIVO 3: Desarrollar de un servicio web (geoportal AgriService) de gestión geoespacial para la monitorización de cultivos con datos de satélite, como herramienta potencial para un futuro asesoramiento al riego que sea el canal de comunicación con el cliente tanto a nivel de servicio (monitorización de cultivos) como a nivel comercial (sistema de registro, selección de área de interés y suscripción).

Figura 1. Estructura de funcionalidad del geoportal de acuerdo con los objetivos planteados



*Elaboración propia*

## 2.1. Relación con Objetivos de Desarrollo Sostenible

Enmarcado en el desarrollo sostenible de las sociedades, el geoportal *AgriService* aporta directamente a la consecución del objetivo 13 de acción por el clima, además de abarcar temáticas que están relacionadas con otros objetivos, como hambre cero, industria, innovación e infraestructura y ecosistemas terrestres.

A partir de la información tomada del Programa de Programa de las Naciones Unidas para el Desarrollo sobre los Objetivos de Desarrollo Sostenible, (ONU, 2024) se recogen las siguientes relaciones.

Estas relaciones demuestran cómo el Geoportal *AgriService* aborda problemas específicos relacionados con la agricultura y el cambio climático, y además, también tiene el potencial de generar impactos positivos en una variedad de áreas clave para el desarrollo sostenible.

**Objetivo 13: Acción Por El Clima.** El Geoportal AgriService contribuye directamente al Objetivo 13 de la Agenda 2030 para el Desarrollo Sostenible, que busca tomar medidas urgentes para combatir el cambio climático y sus impactos. Al proporcionar herramientas de monitorización de cultivos basado en datos de satélite se puede asegurar un futuro asesoramiento de riego al agricultor, el proyecto ayuda a los agricultores a adaptarse a condiciones climáticas cambiantes y a mitigar los efectos adversos del cambio climático en la producción agrícola. Al optimizar el uso del agua en la agricultura, el Geoportal AgriService también contribuye a la reducción de emisiones de gases de efecto invernadero asociadas con el riego excesivo o ineficiente.

**Otros Objetivos.** Además de los Objetivos 13, el Geoportal AgriService también puede tener impactos positivos en otros Objetivos de Desarrollo Sostenible.

El Objetivo 2 Hambre cero: Lograr la seguridad alimentaria y promover la agricultura sostenible. El Geoportal AgriService apoya este objetivo al proporcionar a los agricultores una herramienta que les permite mejorar la productividad y la eficiencia de sus cultivos. Al monitorizar el estado de los cultivos se fomenta prácticas agrícolas más sostenibles y eficientes en el uso del agua.

Objetivo 6 Agua limpia y saneamiento: Al promover el uso eficiente del agua en la agricultura, el proyecto contribuye a la conservación de recursos hídricos y al acceso equitativo a agua limpia y segura.

Objetivo 9 Industria, innovación e infraestructura: El desarrollo de tecnologías innovadoras basadas en teledetección y sistemas de información geográfica para la agricultura puede promover la innovación y el desarrollo de infraestructuras sostenibles en zonas agrícolas.

Objetivo 15 Vida de ecosistemas terrestres: Al mejorar la gestión de los recursos agrícolas y reducir el impacto ambiental de la agricultura, el Geoportal AgriService contribuye a la conservación y restauración de ecosistemas terrestres y a la biodiversidad.

## 2.2. Estado del Arte

La Tabla 4 muestra, a modo de ejemplo, algunas aplicaciones web similares que se encuentran actualmente disponibles para consulta y cálculo de variables relacionada con agricultura.

Tabla 4. Aplicaciones web similares.

Aplicación	Descripción	Cálculos Agrícolas	Autor (enlace)
<b>GLAM (Global Agricultural Monitoring)</b>	Monitoreo global de la agricultura mediante el uso de sensores satelitales. Proporciona datos y herramientas para evaluar la salud de los cultivos y la producción agrícola.	Índice de vegetación normalizado (NDVI) Análisis de la salud de los cultivos Monitoreo de la producción agrícola.	USDA/NASA <a href="https://ipad.fas.usda.gov/glam.aspx">https://ipad.fas.usda.gov/glam.aspx</a>
<b>Copernicus Land Monitoring Service (CLMS)</b>	Proporciona productos de monitoreo terrestre a nivel global y local, con datos satelitales de alta resolución.	NDVI LST Fcover monitoreo de la cubierta terrestre.	Copernicus (Programa de la Comisión Europea) <a href="https://land.copernicus.vgt.vito.be/PDF/portal/LegacyApplication.html#Home">https://land.copernicus.vgt.vito.be/PDF/portal/LegacyApplication.html#Home</a>
<b>CropSAT</b>	Plataforma interactiva para monitorear y analizar campos agrícolas utilizando datos satelitales. Permite a los agricultores optimizar el uso de insumos agrícolas.	NDVI Análisis de la variabilidad del campo Gestión de insumos agrícolas.	<i>Swedish University of Agricultural Sciences y partners.</i> <a href="https://cropsat.com/">https://cropsat.com/</a>
<b>SentinelHubEO Browser</b>	Permite la visualización y análisis de datos satelitales de las misiones Sentinel. Incluye una amplia gama de productos derivados.	NDVI LST Fcover ET Análisis de cambios en la cubierta terrestre.	<i>Sinergise</i> <a href="https://apps.sentinel-hub.com/eo-browser/">https://apps.sentinel-hub.com/eo-browser/</a>
<b>Deforestation and Degradation App</b>	Herramienta para el monitoreo de la deforestación y degradación forestal utilizando <i>Google EarthEngine</i> .	NDVI Monitoreo de la cobertura forestal, Análisis de la degradación forestal.	Desarrollado en <i>Google EarthEngine</i> <a href="https://10130092.users.earthengine.app/view/appdeforestationdgdccd">https://10130092.users.earthengine.app/view/appdeforestationdgdccd</a>



### 3. Metodología

---

En este apartado se describe la metodología utilizada en el desarrollo del TFM, que incluye los datos de entradas y algoritmos para obtener los productos de teledetección, y las tecnologías de desarrollo utilizadas para la construcción del geoportal.

#### 3.1. Productos de Teledetección

##### 3.1.1. Datos de entrada: Copernicus Misión Sentinel-2.

Los sensores a bordo de los satélites Sentinel-2 nutren de datos abundantes el programa de Copernicus de la Comisión Europea, estos datos son considerados de gran utilidad para las prácticas agrícolas, aportando una nueva perspectiva de la superficie terrestre, además, la combinación de alta resolución con capacidades espectrales, la cobertura de visión y la frecuencia de paso de los satélites, son características positivas que aportan valor a estos productos (Phiri et al 2020).

Las imágenes utilizadas por el geoportal AgriService se obtienen de los sensores MultiSpectral Instrument (MSI) a bordo de los satélites 2A y 2B de la constelación de Sentinel, que tiene una resolución radiométrica de 13 bandas espectrales desde el visible e infrarrojo cercano al infrarrojo de onda corta, 4 bandas de resolución espacial de 10 m, 6 bandas de 20 m y 3 bandas de 60m, además entre los productos de nivel 2 L2A (corregidos de efectos de atmósfera) se encuentra la clasificación de escena, en GEE se encuentran las bandas y una variedad de productos Tabla 5 (Phiri et al 2020).

La misión Sentinel-2 está diseñada para proporcionar una alta frecuencia de revisita, que es de unos 5 días en el Ecuador.

**Tabla 5.** Bandas del conjunto de datos de Sentinel-2 en Google Earth Engine

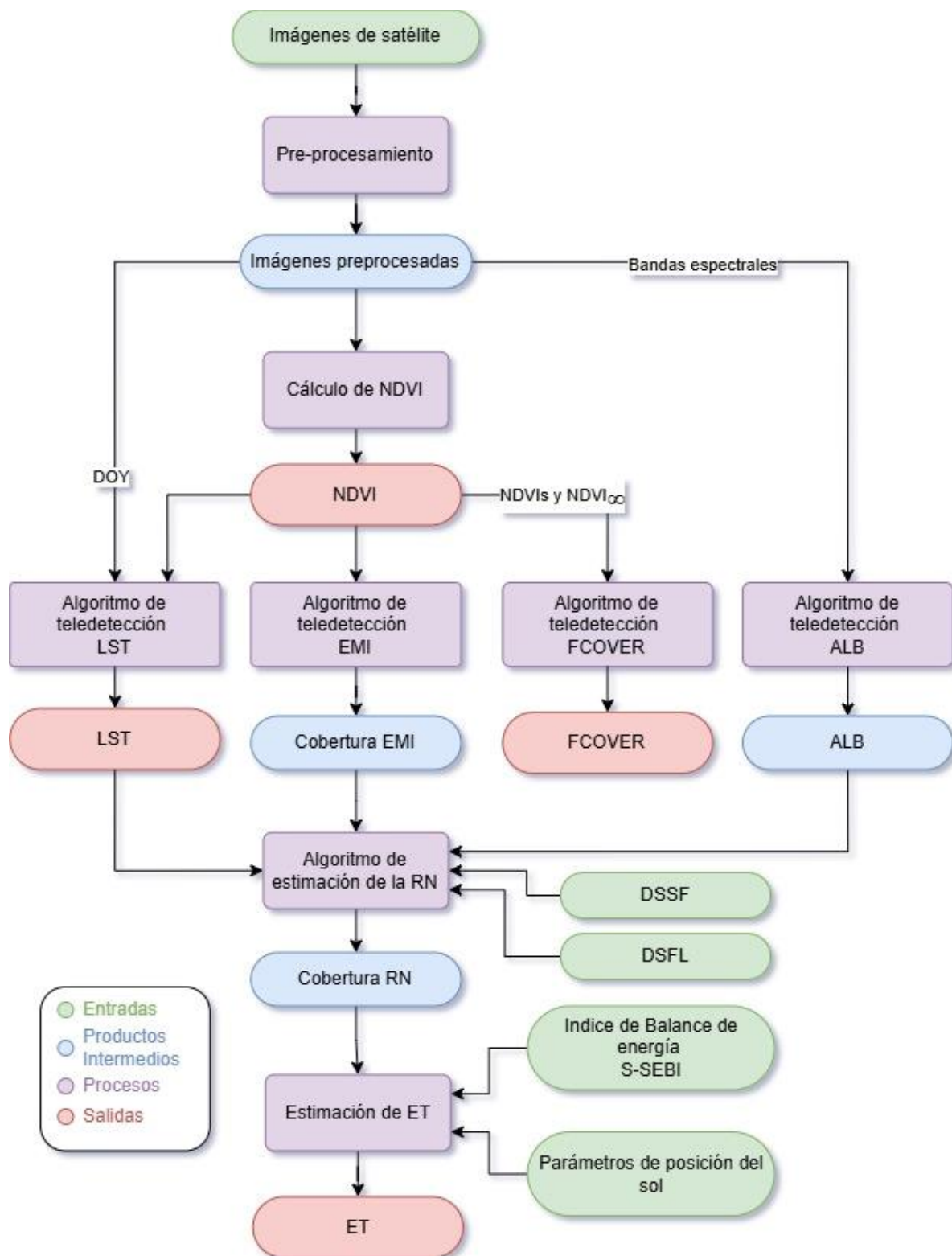
Banda	Resolución Espacial	Longitud de onda	Descripción
B1	60 m	443.9nm (S2A) / 442.3nm (S2B)	Aerosols
B2	10 m	496.6nm (S2A) / 492.1nm (S2B)	Blue
B3	10 m	560nm (S2A) / 559nm (S2B)	Green
B4	10 m	664.5nm (S2A) / 665nm (S2B)	Red

Banda	Resolución Espacial	Longitud de onda	Descripción
B5	20 m	703.9nm (S2A) / 703.8nm (S2B)	Red Edge 1
B6	20 m	740.2nm (S2A) / 739.1nm (S2B)	Red Edge 2
B7	20 m	782.5nm (S2A) / 779.7nm (S2B)	Red Edge 3
B8	10 m	835.1nm (S2A) / 833nm (S2B)	NIR
B8A	20 m	864.8nm (S2A) / 864nm (S2B)	Red Edge 4
B9	60 m	945nm (S2A) / 943.2nm (S2B)	Water vapor
B11	20 m	1613.7nm (S2A) / 1610.4nm (S2B)	SWIR 1
B12	20 m	2202.4nm (S2A) / 2185.7nm (S2B)	SWIR 2
AOT	10 m		Aerosol OpticalThickness
WVP	10 m		Water Vapor Pressure.
SCL	20 m		SceneClassificationMap
TCL_R	10 m		True Color Image, Red channel
TCL_G	10 m		True Color Image, Green channel
TCL_B	10 m		True Color Image, Blue channel
MSK_CLDPRB	20 m		Cloud ProbabilityMap
MSK_SNOWPRB	10 m		Snow ProbabilityMap
QA10	10 m		Alwaysempty
QA20	20 m		Alwaysempty
QA60	60 m		Cloud mask

### 3.1.2. Algoritmos de teledetección

Los algoritmos de teledetección desarrollados por EOLAB se integran con el procesamiento de imágenes de satélite de Copérnicus Sentinel-2 utilizando las bandas espectrales de las imágenes de satélite. De esta manera, es posible estimar el índice de vegetación normalizado, fracción de cobertura vegetal, temperatura de superficie y evapotranspiración mediante algoritmos físicos. La Figura 2 representa la cadena de procesamiento de EOLAB, funcional en el geoportal AgriService, el trabajo recoge los fundamentos matemáticos estudiados y proporcionados por la empresa.

Figura 2. Cadena de procesado de EOLAB funcional en el geoportal AgriService.



Elaboración propia en base a la cadena de procesado de la cartera de productos de EOLAB. En rojo los productos finales que ofrece el Geoportal AgriService.

### 3.1.2.1. Índice de Vegetación Normalizado (NDVI).

El cálculo del NDVI se realiza utilizando las bandas, B4 (rojo - RED) y B8 (infrarrojo cercano - NIR) siguiendo la expresión de normalización de la ecuación (1). El NDVI es una normalización del contraste entre la banda infrarroja con la banda roja (Amiri, et al., 2022).

$$NDVI = \frac{NIR-RED}{NIR+RED} \quad (1)$$

### 3.1.2.2. Fracción de cobertura vegetal (FCOVER).

La estimación de fracción de cobertura vegetal se realiza utilizando la expresión (2). Según Verger i Ten (2008) esta relación semi empírica entre el índice de vegetación normalizado y fracción de cobertura vegetal es la que mejores resultados obtiene, comparándola con otras fórmulas semi empíricas con respecto a muestras tomadas en campo. La ecuación (2) escala el índice de vegetación normalizado estableciendo umbrales de NDVI de suelo ( $NDVI_S$ ) o mínimo y NDVI infinito ( $NDVI_\infty$ ) o máximo. Se utiliza 0.1 como valor de NDVI de suelo y 0.9 para NDVI infinito, estos valores son estimados de los máximos y mínimos obtenidos del sensor en diferentes zonas de estudio como áreas sin vegetación y con máximo desarrollo del estado vegetativo respectivamente (Gutman & Ignatov, 1998).

$$FCV = \frac{NDVI-NDVI_S}{NDVI_\infty-NDVI_S} \quad (2)$$

Según Fensholt & Proud (2012) al ser el FCOVER un valor estimado a partir de escalar el NDVI los valores que puede tomar van de 0 a 1. Siendo 0 los valores donde no existe cobertura de vegetación y 1 los valores donde el terreno está completamente cubierto por vegetación, siendo los rangos:

- 0.1 representa el suelo descubierto, áreas desérticas, superficies urbanas, o tierras agrícolas después de la cosecha;
- 0.3 baja cobertura vegetal, praderas, tierras agrícolas en etapas tempranas de crecimiento, o áreas con vegetación dispersa.
- 0.5 cobertura vegetal moderada, áreas con vegetación significativa, campos de cultivo en crecimiento, praderas densas o matorrales.

- 0.5 - 0.7 alta cobertura vegetal, zonas con vegetación densa, como bosques abiertos o cultivos maduros.
- 0.7 - 1.0: Cobertura vegetal muy alta a completa. Este rango es típico de bosques densos, selvas tropicales y cultivos muy densos.

### 3.1.2.3. *Temperatura en la superficie de la Tierra (LST).*

Esta variable se ha estimado en base al método *split-window* propuesto por Du et al., (2015), el cual en un estudio de validación (García-Santos et al., 2018) con datos de campo de una zona heterogénea, se observó que se determinaba LST con un error de  $\pm 2$  grados Kelvin. Esta formulación calcula LST en grados Kelvin, en función del día del año (DOY) de la imagen de satélite, y del NDVI. La expresión (3) se utiliza para estimar la temperatura en superficie, donde  $a1_{(DOY)}$  representa la temperatura de la superficie para un día específico, se calcula con la ecuación (3.1), sin considerar el efecto de la vegetación, y  $a2_{(DOY)}$  refleja la sensibilidad de LST a la cantidad de vegetación presente y se calcula con la ecuación (3.2) (Li, et al., 2015).

$$LST = a1_{(DOY)} + a2_{(DOY)} * NDVI \quad (3)$$

$$a1_{(DOY)} = -16.07 + 2.668 * \cos(0.02234 * DOY) + 7.235 * \text{sen}(0.02234 * DOY) + 2.59 * \cos(0.04468 * DOY) - 2.272 * \text{sen}(0.04468 * DOY) \quad (3.1)$$

$$a2_{(DOY)} = 307 - (18.24 * \cos(0.01766 * DOY)) - (5.282 * \sin(0.01766 * DOY)) \quad (3.2)$$

### 3.1.2.4. *Evapotranspiración (ET).*

Para estimar esta variable se utiliza el método simplificado del S-SEBI que necesita parámetros de emisividad, albedo y radiación neta o total. Cada uno de estos parámetros tiene sus respectivas consideraciones detalladas a continuación.

- *Emisividad (EMI)*. El valor de emisividad, que es la capacidad de una superficie de emitir radiación térmica (EMI), puede variar por características físicas de la superficie observada. Para calcular este valor se utiliza los valores del NDVI siguiendo la expresión (4) (Sobrino, et al., 2004).

$$\epsilon_s = \begin{cases} 1.0094 + 0.047 \ln(NDVI), & 0 < NDVI \leq 0.727 \\ 0.99, & \text{otros casos} \end{cases} \quad (4)$$

- *Albedo (ALB)*. El albedo es la fracción de radiación incidente que es reflejada y que varía según las propiedades físicas de cada superficie. Se estima con la expresión (5), donde el albedo  $\alpha_s$  es el albedo de superficie,  $\rho_i$  es la reflectividad en cada banda MSI de Sentinel-2, y  $w_i$  es un peso asignado que depende de la contribución de esa banda al albedo total (Bonafoni & Sekertekin, 2020). La Tabla 6 muestra los valores de los pesos  $w_i$  para cada banda de los sensores Sentinel-2 según Bonafoni & Sekertekin (2020).

$$\alpha_s = \sum_i^N \rho_i w_i \quad (5)$$

**Tabla 6.** Valores de los factores de ponderación del albedo para Sentinel-2.

Bandas MSI Sentinel-2	$w_i$
B2	0.2266
B3	0.1236
B4	0.1573
B8	0.3417
B11	0.1170
B12	0.0338

- *Radiación neta o total (RN)*. Otro algoritmo de teledetección necesario es el de la radiación total (RN), que está en función de EMI, ALB y LST. Además, son necesarios, el flujo de onda corta Superficial hacia abajo (*Downward Surface Shortwave Flux - DSSF*) y, el flujo de onda larga Superficial hacia abajo (*Downward Surface Longwave Flux - DSLF*). El DSSF se refiere a la radiación de energía en el intervalo de onda 0.3 a 0.4  $\mu\text{m}$  recogida por la superficie terrestre por unidad de tiempo y área; mientras que, DSLF es la irradiación térmica recogida en superficie en el espectro infrarrojo térmico (Trigo, et al., 2011).

Los conjuntos de datos de GEE que se utilizan para introducir los productos auxiliares DSSF y DSLF al algoritmo de ET son los proporcionados por el Centro Europeo de Predicciones Meteorológicas a Medio Plazo (ECMWF), más específicamente el conjunto de datos de *ERA5-Land* que tiene información disponible de DSSF y DSLF por hora desde 1950 hasta aproximadamente 5 días antes de la fecha actual en una resolución espacial de aproximadamente 9 km de separación de grilla (Muñoz Sabater, 2019).

Para el cálculo de la RN se sigue la expresión (6) en donde  $R_n$  es la radiación neta,  $\alpha$  es ALB, DSSF, DSLF son los parámetros de ERA5-LAND,  $\sigma$  es la constante de Stefan-Boltzmann, LST es la temperatura de la superficie terrestre y  $\epsilon_s$  es EMI.

$$R_n = (1 - \alpha)DSSF + \epsilon_s(DSLF - \sigma LST^4) \quad (6)$$

**Método simplificado del S-SEBI.** La evapotranspiración potencial o total se estima mediante el método simplificado del modelo del S-SEBI propuesto por Roerink et al. (2000). El modelo del S-SEBI se basa en la hipótesis física que la evapotranspiración varía con temperatura para una superficie homogénea. Este modelo indica que, para un determinado valor de albedo en la superficie, existen condiciones meteorológicas específicas, una determinada altura de cultivo (dosel) y un gradiente entre la temperatura de superficie y temperatura del aire. En el valor máximo del gradiente se asume que la ET es nula y en valor mínimo la ET adquiere un valor potencial (Menenti & Choudhury, 1993).

La diferencia entre el método del modelo simplificado y el modelo del S-SEBI es la ventaja de omitir la inclusión de datos meteorológicos. Esto es posible siempre y cuando las condiciones de humedad en superficie sean muy contrastantes, lo que permite observar el cambio significativo de temperaturas correspondiente al espectro de reflectividad. Entonces, para un valor de albedo existirán dos valores extremos de temperatura en superficie (Menenti & Choudhury, 1993).

El método simplificado del S-SEBI consiste en determinar los píxeles secos y húmedos para un valor específico de albedo en una imagen de satélite. Según Roerink et al., 2000 se puede simplificar aún más este método ajustando linealmente la tendencia de valores máximos y mínimos de temperatura en superficie con los valores de albedo.

**Aplicación del método simplificado del S-SEBI.** El método simplificado del S-SEBI se aplica construyendo un vector en pasos de 0.005 desde 0 a 0.995 para valores de albedo en superficie, donde se cumple la condición de la expresión (7), en la que  $k$  representa los valores de albedo en cada paso.

$$k < ALB \leq k + 0.05 \quad (7)$$

En cada paso y para cada posición de ALB se calcula la mediana de ALB y se extrae el valor de LST. Con los valores de LST se obtiene el valor límite mínimo y máximo de temperatura en superficie de la imagen de satélite donde se cumple (7) en cada uno de los pasos.

Una vez construido el vector, se calcula dos regresiones lineales ponderadas de mínimos cuadrados ordinarios (WLS) entre los valores de, mediana de ALB de la imagen y temperatura mínima y máxima. Como valor de los pesos de la regresión se utiliza el número de píxeles que cumplen con (7) en cada paso y se calcula el S-SEBI con la expresión (8). Donde  $a_{max}$  y  $b_{max}$  son el intercepto y la pendiente del ALB con LST máximo, Luego,  $a_{min}$  y  $b_{min}$  corresponden al Intercepto y pendiente para LST mínimo (Roerink et al., 2000).

$$SSEBI = \frac{a_{max} + b_{max} * ALB - LST}{a_{max} - a_{min} + (b_{max} - b_{min}) * ALB} \quad (8)$$



Una vez calculado el S-SEBI se normaliza sus valores utilizando la mediana de los valores que son superiores a 1 y la mediana de los valores que son inferiores a 0. Se recalcula el valor del S-SEBI, y se limita a 1 los valores resultantes superiores a la unidad y a 0 los valores resultantes negativos.

El valor obtenido es equivalente al valor de la evapotranspiración instantánea en litros por segundo, es necesario convertir estos valores en evapotranspiración diaria en milímetros. Se considera los parámetros de, diferencia entre el tiempo de paso del satélite (UTC+0) y la hora que sale el sol para este día en concreto (UTC+0) ( $t$ );  $y$ , el total de horas de sol en ese día completo ( $N$ ), para hacer la conversión de evapotranspiración instantánea a evapotranspiración diaria utilizando la expresión (9) definida por Jackson et al. (1983).

$$J83 = \frac{2N}{\pi * \sin\left(\frac{\pi t}{N}\right)} \quad (9)$$

Finalmente, la evapotranspiración se calcula con la expresión (10) obteniendo el valor de la evapotranspiración en milímetros por cada día. Donde se utiliza el valor de  $RN$ , S-SEB, J83, y  $Gi$  que es el flujo de calor o ganancia de la radiación equivalente al 35% de la radiación total ( $RN$ ) (Norman et al., 1995).

$$ET = SSEBI * (RN - Gi) * J83 \quad (10)$$

### 3.2. Tecnologías de desarrollo

En esta sección se describen las herramientas de desarrollo web y su integración mediante lenguajes de programación, y las herramientas para el almacenamiento en la base de datos.

### 3.2.1. Herramientas de desarrollo

En este apartado se describen las tecnologías de desarrollo y creación de aplicaciones que se utilizan para el funcionamiento del Geoportal AgriService, las tres primeras tecnologías descritas son utilizadas para el CC en GEE mediante el manejo de la API de Python aprovechando los conjuntos de datos disponibles en esta plataforma, las tres siguientes son tecnologías aplicadas para el desarrollo web del geoportal. Se establecen dos etapas para construir la aplicación, la primera de desarrollo, y la segunda de producción y despliegue (publicación), las herramientas descritas a continuación se utilizan en ambas etapas.

**Google Earth Engine.** Según Perilla & Mas (2020) la plataforma de GEE proporciona tres principales características para CC:

- La capacidad de procesamiento de grandes cantidades de datos poniendo a disposición la capacidad de sus servidores, agilizando la velocidad del procesamiento de las imágenes de satélite.
- La potencialidad de actualización de los datos y disponibilidad en la nube, con alrededor de 6000 nuevas escenas de la superficie terrestre al día procedente de diferentes satélites y servicios de observación de la Tierra.
- La interfaz de programación de aplicaciones que permite desarrollar algoritmos de procesamiento.

En este contexto, el geoportal se nutre de los servidores de GEE para hacer CC y realizar los cálculos de los algoritmos de NDVI, FCOVER, LST y ET según la petición que realice el usuario del geoportal. Esto permite ahorrar los recursos necesarios para estos cálculos en un ordenador local (Perilla & Mas 2020).

**API de Python para creación de aplicaciones.** GEE proporciona la capacidad de desarrollar algoritmos de procesamiento utilizando Python como lenguaje de programación. En la actualidad existen varios módulos que permiten utilizar esta API en entornos como *Google Colab* y entornos locales, el módulo de Python se actualiza anualmente (Perilla & Mas 2020).

La documentación de GEE proporciona una clara explicación de la diferencia entre usar la API de Python con la de JavaScript y una completa descripción de cómo tratar los errores y un apartado de buenas prácticas para el desarrollo de aplicaciones, además cuenta con variedad de ejemplos que permiten al desarrollador entender el funcionamiento de las herramientas de programación.

Se aprovecha la documentación de la API de Python de GEE para crear un algoritmo que permite procesar los cálculos en la nube. Para iniciar y finalizar procesamientos mediante la API de Python, se requiere autenticación constante en la plataforma de GEE. Al crear un proyecto en la nube (Cloud Project) y una cuenta de servicio, es posible generar una clave de acceso en formato JSON asociada a dicha cuenta. Esta clave privada facilita la autenticación en Google Earth Engine (GEE) y, por ende, el uso de su API.

**Conjuntos de datos de Google Earth Engine.** GEE ofrece la posibilidad de trabajar con una cantidad de datos actualizados provenientes de varios sensores, constituyendo un enorme catálogo de datos geoespaciales. El geoportal AgriService trabaja con los catálogos de datos de Sentinel-2 disponible en GEE. También, se utiliza el conjunto de datos ERA5\_LAND de ECMWF para el cálculo de la radiación solar (DSSF y DSLF), obteniendo los productos de la colección como entrada para el cálculo a la hora de paso aproximada del satélite (11:00:00 UTC+0).

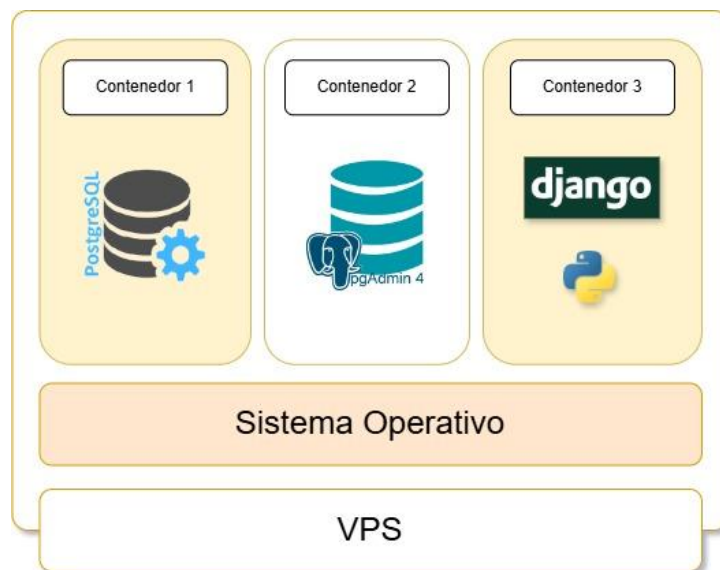
**Apache HTTP.** Apache HTTP es un proyecto de desarrollo y mantenimiento abierto que provee un eficiente y extensible servidor con los estándares HTTP desde 1995 (Apache Software Foundation, 2024). Apache HTTP se utiliza para configurar los puertos de internet y certificados para tener acceso a la página web de manera segura, en la etapa de publicación del geoportal, la empresa proporciona un Servidor Privado Virtual (VPS) con Apache HTTP instalado.

**Django.** Es un marco de desarrollo de aplicaciones Python web con un diseño programático de código abierto (Django Software Foundation, 2024). El geoportal AgriService aprovecha la potencialidad para obtener un *back-end* capaz de conectar el CC del *back-end* con el *front-end* y servirlo mediante Apache HTTP. Además, permite manejar el registro de los usuarios del geoportal y envío de notificaciones mediante correo electrónico. El marco de desarrollo Django facilita la interacción con la base de datos a los usuarios del geoportal y el manejo para activación o desactivación de usuarios a los administradores del geoportal, además, permite transferir las peticiones del usuario del geoportal a GEE y las respuestas del CC a la interfaz del usuario del geoportal.

**Docker Compose.** Es una herramienta de código abierto que facilita el desarrollo eficiente y experiencia de despliegue de aplicaciones de una forma rápida y organizada a través de contenedores que almacenan la aplicación de manera segura y más liviana que una máquina virtual. Es una manera de distribuir los componentes de una aplicación de manera modular con cierta flexibilidad en el desarrollo (Docker Inc., 2024).

Esta herramienta se utiliza en la fase de publicación del geoportal en un VPS contratado previamente por la empresa para albergar su página web y un sitio como servicio de validación de Albedo (SALVAL). Al usar contenedores para AgriService, se evita la suspensión de los servicios web actualmente funcionales, además, se evade conflictos detectados entre las versiones de Django y la base de datos MySQL de la empresa. La organización del geoportal en la VPS mediante contenedores se muestra en la Figura 3.

Figura 3. Contenedores Docker en la VPS para el geoportal AgriService.



Elaboración propia

**GitHub.** Es una plataforma de desarrollo colaborativo basada en la nube que utiliza el sistema de control de versiones Git. Proporciona herramientas para la gestión de proyectos y el seguimiento de cambios en el código, facilitando la colaboración entre desarrolladores. GitHub es utilizado para documentar el desarrollo del geoportal AgriService a través de la creación de un repositorio, donde se realizan seguimientos de problemas en la programación y revisar cambios en las versiones del código.

### 3.2.2. Lenguajes de Programación

En esta sección se describe la manera en la que se utilizan los lenguajes de programación para desarrollar el geoportal y los principales módulos utilizados.

**Python.** Este es el lenguaje de programación utilizado en el *back-end* del geoportal AgriService. Los módulos principales para el desarrollo de los algoritmos son el de la API de GEE, el del marco de desarrollo Django, *SklearnPySolar* para la estimación de la evapotranspiración, *Pillow* para construcción de imágenes para las leyendas de las variables calculadas mostradas en el mapa, *Datetime* para el manejo de fechas y *PgOperations* para la interacción desde el código con la base de datos de usuarios. Además, para el uso de este lenguaje en el *back-end* se crea un entorno virtual con la versión 3.11.4.

**HTML.** (*Hyper Text Markup Language*) es el lenguaje de marcado estándar utilizado para crear y estructurar el contenido del geoportal en la web. Se emplea una serie de etiquetas para definir y organizar encabezados, párrafos, enlaces, imágenes, formularios y otros tipos de contenido multimedia. HTML es el lenguaje base de la página web de registro de usuarios y del geoportal. Es esencial para la construcción de la estructura que permite la integración de CSS y JavaScript para mejorar la interfaz del usuario y funcionalidad.

**CSS.** (*Cascading Style Sheets*) es un lenguaje de diseño utilizado para describir la presentación de los documentos en HTML de, registro de usuarios y geoportal. CSS permite controlar la apariencia del *front-end* de la aplicación, diseño, colores, fuentes, márgenes, espacios y disposición de los elementos. Mediante el uso de hojas de estilo externas. Esta herramienta facilita la separación del contenido y el diseño, permitiendo mantener un código organizado, claro, fácil de mantener y actualizar, para obtener un geoportal atractivo y coherente.

**JavaScript.** Este lenguaje de programación se utiliza para probar el funcionamiento de los métodos de la API de GEE en el *code editor*, para visualizar imágenes en la propia interfaz de GEE. Esto es útil para evitar errores al programar en editor de código Python, como paso previo a su inclusión en el algoritmo del *back-end*. A la vez, conlleva la constante traducción entre lenguajes de programación en el contexto de la API de GEE.

También, este lenguaje se utiliza para desarrollo del funcionamiento del *front-end* para, establecer la interacción entre los objetos instanciados en el HTML, hacer solicitudes a las URLs establecidas con Django, generar mapas en la interfaz del usuario y crear gráficos de series de tiempo de las variables biofísicas. Los módulos utilizados con este lenguaje de programación son: el módulo Ajax para conectar el *front-end* con el *back-end*, *rSlider* para interacción con los gráficos de las series de tiempo, *ChartJS* para la construcción de los gráficos de series de tiempo, y *Leaflet* para creación e interacción con el mapa.

*Leaflet* es un módulo gratuito de uso sencillo de JavaScript para la creación de mapas interactivos. Si bien la potencialidad de esta librería no es comparable con la de otras librerías, es útil para conseguir el objetivo de este Proyecto. El geoportal AgriService trabaja con esta librería para permitir al usuario final visualizar e interactuar con los mapas de las variables biofísicas y datos geográficos de su parcela agrícola.

### 3.2.3. Almacenamiento y Base de datos

Para el almacenamiento en la base de datos de usuarios se utiliza PostgreSQL y PostGIS por sus ventajas potenciales y su código abierto.

**PostgreSQL.** Es un sistema de gestión de bases de datos relacional y objeto-relacional de código abierto. Se utiliza PostgreSQL para almacenar los datos de los usuarios de manera robusta, escalable, extensible, y estandarizada. PostgreSQL ofrece una amplia gama de características avanzadas, y soporte para procedimientos almacenados y tipos de datos avanzados, además es ampliamente utilizado en aplicaciones que requieren el manejo de un gran volumen de datos. El geoportal AgriService hace uso de la extensión espacial PostGIS para almacenar los datos geográficos de las parcelas de cada usuario.

**PostGIS.** Es una extensión de PostgreSQL que añade soporte para datos geoespaciales y funciones de análisis espacial. Se utiliza esta extensión para almacenar, consultar y manipular datos espaciales directamente en la base de datos de la que se sirve AgriService, al ser la base de los sistemas de información geográfica (SIG) es ampliamente utilizado en aplicaciones de planificación urbana, gestión de recursos naturales, y servicios basados en la localización.

## 4. Resultados

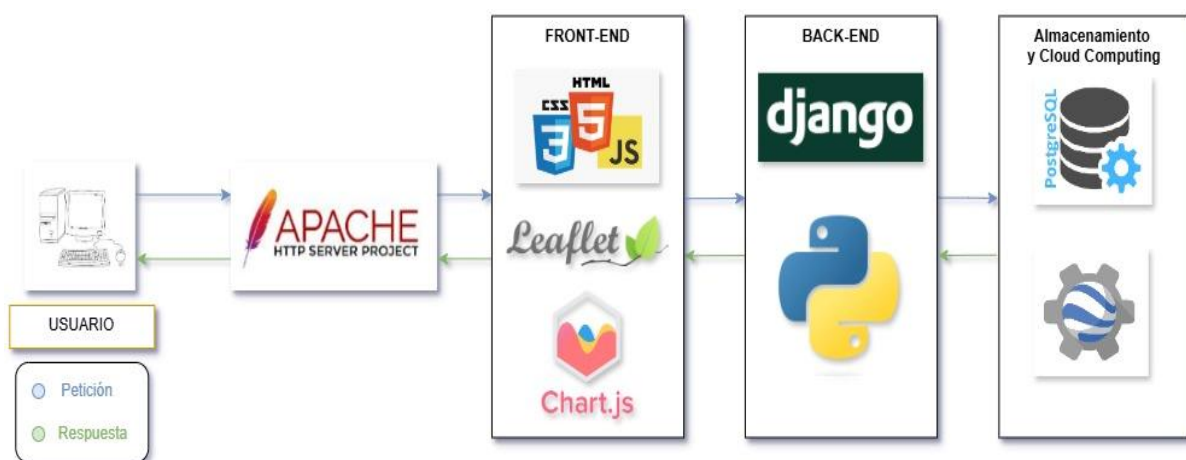
En esta sección se detalla la estructura en cuanto a la interfaz, *front-end* y *back-end*, y base de datos, se describen los flujogramas que siguen los procesos del geoportal para el registro de parcelas agrícolas asociadas a un usuario, visualizar mapas de coberturas de las variables biofísicas y el cálculo de las series de tiempo.

### 4.1. Estructura del geoportal AgriService

El geoportal está compuesto por cuatro principales componentes esquematizados en La Figura 4.

- La interfaz del usuario, accesible gracias a la configuración del servidor apache HTTP, que se realiza en el momento de la publicación del geoportal.
- El *front-end*, desarrollado mediante HTML, CSS y JavaScript, siendo primordial el módulo *Leaflet* para la interacción del usuario con el mapa, y *ChartJS* para los gráficos de las series de tiempo.
- El *back-end*, desarrollo basado en el marco Django para aplicaciones Python.
- La base de datos para el almacenamiento de los datos de usuario y las parcelas en el geoportal.

Figura 4. Esquema de la estructura de funcionamiento del geoportal AgriService.



Elaboración propia



### 4.1.1. Interfaz/servidor

En la etapa de publicación del geoportal son necesarias las credenciales de acceso a la VPS de EOLAB, así como el subdominio (*agriservice.eolab.es* con el alias *www.agriservice.eolab.es*) donde va a ser publicado. En esta etapa se instala *Docker* en el VPS de acuerdo con las características del distribuidor del sistema operativo que se muestra en la Tabla 7. Se crean los tres contenedores especificados en la Figura 3.

Tabla 7. Información del servidor (VPS)

Artículo	Detalle
Sistema Operativo	Linux
Versión Apache	2.4.59
Arquitectura	X86_64
Distribución	CentOS 7

Tras la instalación de Docker en el sistema operativo, se debe, configurar el Apache del VPS, crear los contenedores (de la base de datos, del administrador de la base de datos y de la aplicación) y habilitar el administrador de usuarios.

#### 4.1.1.1. Configuración Apache.

Se configura el *Apache HTTP* del VPS editando el fichero *httpd.conf*, donde se activa la inclusión de modificaciones en los *hosts virtuales* a través de los puertos de internet (80 y 443). El Apache del VPS está configurado de manera modular, es decir, está dividido en varias carpetas dentro del VPS, existe un fichero global que integra toda la configuración, siendo esta una manera ordenada y eficiente de mantener la configuración de *Apache HTTP*.

Cuando se crea un subdominio, automáticamente se generan dos *hosts virtuales* en el fichero global *httpd.conf*, uno para el puerto 80 y otro para el puerto 443. Dentro de la configuración de ambos *hosts* se encuentran dos directorios (uno para cada *host*) de las carpetas donde se pueden crear ficheros de extensión *.conf* para incluir nuevas configuraciones en cada *host*. La Figura 5 muestra algunas de las configuraciones del fichero global generado automáticamente con el certificado SSL para realizar peticiones por el puerto 443 de manera segura y los directorios para incluir las configuraciones.

Figura 5. Configuraciones relevantes para el geoportal en el fichero *httpd.conf*.

```
File: httpd.conf

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
DO NOT EDIT. AUTOMATICALLY GENERATED.  USE INCLUDE FILES IF YOU NEED TO MAKE A CHANGE
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
***
<VirtualHost [REDACTED]>
  ServerName agriservice.eolab.es
  ServerAlias www.agriservice.eolab.es
  ***
  # To customize this VirtualHost use an include file at the following location
  # Include "/etc/apache2/conf.d/userdata/std/2_4/eolab3/agriservice.eolab.es/*.conf"
  ***
<VirtualHost [REDACTED]>
  ServerName agriservice.eolab.es
  ServerAlias www.agriservice.eolab.es
  ***
  SSLCertificateFile [REDACTED]/agriservice.eolab.es/combined
  ***
  # To customize this VirtualHost use an include file at the following location
  Include "/etc/apache2/conf.d/userdata/ssl/2_4/eolab3/agriservice.eolab.es/*.conf"
  ***
```

*Elaboración propia*

Se activan estos directorios borrando la almohadilla (#) de las líneas correspondientes a los directorios y creando el fichero *agriservice.conf* en ambos directorios. En este fichero se incluyen todos los proxys y cambios realizados para el funcionamiento del sitio web de AgriService.

#### 4.1.1.2. Creación de contenedores.

Para la composición de cada contenedor, se necesita crear tres ficheros en cada contenedor:

- Fichero *.env*: Fichero de variables de entorno de los puertos utilizados por cada contenedor.
- Fichero *.env.prod*: Fichero con las variables de entorno específicas para cada contenedor.
- Fichero *docker-compose.prod.yml*: Fichero de creación del contenedor.

**Contenedor Base de datos.** El primer fichero, *.env*, tiene la configuración de los puertos utilizados para crear el túnel entre el contenedor de la base de datos y la VPS. El segundo, *.env.prod*, contiene las claves de acceso y configuraciones de la base de datos *PostgreSQL* dentro del contenedor. El tercer fichero es *docker-compose.prod.yml* para *PostgreSQL*, donde se especifica las características del contenedor para *Docker compose*. La versión de *PostgreSQL* configurada es la de la imagen *postgis/postgis:15-3.4*, este fichero toma las configuraciones de los dos anteriores, y crea el contenedor.

En este contenedor se crea una red llamada *postgis* a la que se podrá acceder desde otros contenedores mediante un puente. Esto es útil para acceder al contenedor de la base de datos desde otros contenedores como el de *pgAdmin* y el de la aplicación. La Figura 6 muestra el fichero *docker-compose.prod.yml* con las configuraciones de la creación de la base de datos.

Figura 6. Docker-compose.prod.yml de la base de datos

```
services:
  postgres:
    image: postgres/postgis:15-3.4
    restart: unless-stopped
    env_file:
      - .env
      - .env.prod
    ports:
      - IP de la VPN : {PRODUCTION_FORWARDED_PORT}:5432
    volumes:
      - postgres-data-prod:/var/lib/postgresql/data
    healthcheck:
      test: [ "CMD", "pg_isready", "-q", "-d", "postgres", "-U", "postgres" ]
      interval: 10s
      timeout: 5s
      retries: 5
    networks:
      - postgres
volumes:
  postgres-data-prod:
    driver: local
networks:
  postgres:
    driver: bridge
```

*Elaboración propia*

Contenedor de administración de la base de datos (PgAdmin). En el fichero `.env` se escribe el puerto de producción en el VPS y el puerto del contenedor, el `.env.prod` contiene las credenciales de acceso al sitio de administración de la base de datos (pgAdmin). En el fichero `docker-compose.prod.yml` que se muestra en la Figura 7 se configura la dependencia del contenedor a la red `postgres` para poder conectar con el contenedor de la base de datos.

Este contenedor necesita la creación de dos carpetas adicionales, una para los archivos de datos (`pgadmin_prod/data`) y otra para los archivos de servidores (`pgadmin_prod/servers`). *Linux* asigna un propietario por defecto a las carpetas creadas, por esto, se necesita cambiar la configuración del propietario de manera recursiva, para poder acceder a estas carpetas del VPS desde el contenedor. El nuevo propietario tiene como ID el puerto de producción en el VPS configurado en el fichero `.env`. También, se asigna un grupo propietario de las carpetas que tiene como ID el puerto del contenedor configurado en el fichero `.env`.

Figura 7. Docker-compose.prod.yml de pgAdmin

```
services:
  pgadmin4:
    image: dpage/pgadmin4:8.3
    env_file:
      - .env
      - .env.prod
    volumes:
      - ./pgadmin4_prod/data:/var/lib/pgadmin
      - ./pgadmin4_prod/servers/servers.json:/pgadmin4/servers.json
    ports:
      - IP de la VPN :{PRODUCTION_FORWARDED_PORT}:80
    restart: unless-stopped
    networks:
      - postgis_postgis
networks:
  postgis_postgis:
    external: true
```

*Elaboración propia*

Los directorios de datos y servidores se comparten mediante los volúmenes de Docker entre el contenedor y la carpeta *pgadmin\_prod* del VPS. Luego, para poder acceder a este contenedor desde el dominio, se configura un proxy en el fichero *agriservice.conf* que permite acceder directamente al puerto designado para el sitio de administración de la base de datos desde el navegador. Para esto se añade la configuración de la Figura 8 al directorio.

Figura 8. Configuración del proxy de pgAdmin en el fichero *agriservice.conf*

```
</Proxy>
<Location /pgadmin/>
  ProxyPass http://[REDACTED]/
  ProxyPassReverse http://[REDACTED]/
  Header add myheader X-Script-Name
  RequestHeader set X-Script-Name /pgadmin
</Location>
```

*Elaboración propia*

**Contenedor de la Aplicación Django.** Se crea un contenedor de la aplicación para incluir el desarrollo en Python realizado en el *back-end*. Esto tiene la ventaja de poder actualizar los códigos ante futuras nuevas versiones de los algoritmos.

Se crea el fichero *Dockerfile* donde se configura la versión de Python a utilizar, en este caso la 3.12-alpine. Adicionalmente se configura las variables de entorno y dependencias de compiladores necesarias para la aplicación Django. Se escribe la sentencia de instalación del fichero *requirements.txt* donde se encuentran las librerías de los módulos Python que utiliza el *back-end*. Finalmente se ejecutan las migraciones de la configuración de Django. La Figura 9 tiene el Fichero *Dockerfile* creado para la aplicación Django.

Figura 9. Dockerfile

```
# pull official base image
FROM python:3.12-alpine

# set work directory
WORKDIR /usr/src/app

# set environment variables
ENV PYTHONDONTWRITEBYTECODE 1
ENV PYTHONUNBUFFERED 1

# install psycogp2 dependencies
#pango es una dependencia de weasyprint
RUN apk update && apk add postgresql-dev gcc musl-dev libpq-dev wget pango \
    libffi-dev python3-dev
RUN apk add --update --no-cache --virtual .tmp-build-deps \
    libc-dev linux-headers

RUN apk add g++
|

# install dependencies
RUN pip install --upgrade pip

COPY ./requirements.txt .

RUN pip3 install -r requirements.txt
```

*Elaboración propia*

Es necesario crear el fichero *.env* que tiene el puerto por el que el VPS hace el túnel con el contenedor *Docker* para acceder a la aplicación. El fichero *.env.prod* contiene la configuración del super usuario, el *DEBUG* en falso para la etapa de producción, la configuración de la base de datos y las variables de entorno para el envío de email de registro (*settings.py*).

Para el envío de emails se utiliza la configuración proporcionada por el proveedor para utilizar el *Webmail* contratado por la empresa. Finalmente se configura el fichero *docker-compose.prod.yml* donde se configura la imagen del servicio, el servidor *gunicorn*, en este fichero también se configura el tiempo de espera con el parámetro *- timeout* porque la respuesta del *cloud computing* toma cierto tiempo en realizarse dependiendo de la cantidad de imágenes que se haya introducido, un *- timeout* de 200, siendo el cálculo de la evapotranspiración la que más tiempo conlleva, aproximadamente 120 segundos aproximadamente.

También se configura las carpetas asociadas entre el contenedor y el VPS con los volúmenes de *Docker*, donde se establece una carpeta para la aplicación y otra en el front-end para guardar las imágenes generadas al calcular los mapas y llevarla a la leyenda, se configura los puertos para el túnel, y se lo conecta con la red externa. La Figura 10 muestra la configuración de este contenedor.

Figura 10. Fichero *docker-compose.prod.yml* de la aplicación Django

```
services:
  geoportaleolab:
    restart: unless-stopped
    build: ./geoportaleolab
    command: gunicorn geoportaleolab.wsgi:application --timeout 200 --bind 0.0.0.0:8000
    volumes:
      - ./geoportaleolab:/usr/src/app
      - /usr/src/app/public_html/AgriService/img:/usr/src/img_html
    ports:
      - IP de la VPS :${PRODUCTION_DOCKER_DJANGO_API_FORWARDED_PORT}:8000
    env_file:
      - .env
      - .env.prod
    networks:
      - postgres_postgis

#to check if postgres is already ready. This service
#will not start up to the postgres service is ready
#depends_on:
#  postgres_postgis:
#    condition: service_healthy

networks:
  postgres_postgis:
    driver: bridge
    external: true
```

*Elaboración propia*

#### 4.1.1.3. **Habilitación de administrador de usuarios (Django Administrador).**

El sitio de administración de usuarios de *Django* permite la gestión de usuarios. Dicha administración se puede hacer directamente con el administrador propio de la base de datos (*pgAdmin*), pero el administrador de *Django* permite la gestión de usuarios del geoportal de una manera mucho más visual. Para esto se crea un proxy en el fichero *agriservice.conf* del sitio de administración y se mueven los ficheros *static* del sitio de administración al directorio del *front-end*.

#### 4.1.2. **Front-end**

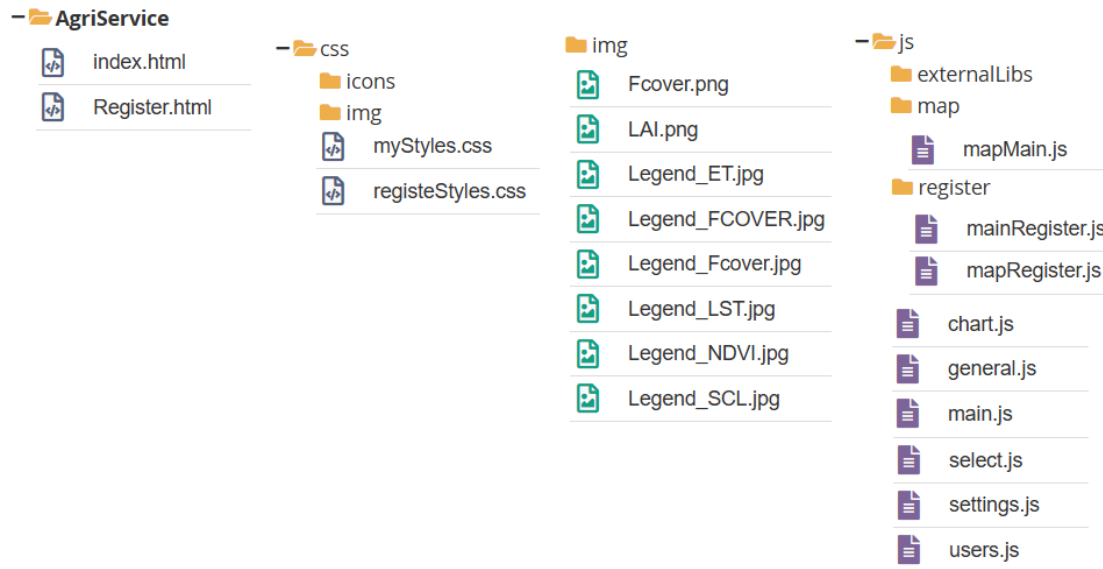
Los ficheros del *front-end* se organizan como se muestra en la Figura 11 para almacenar de manera ordenada los estilos, funciones y documentos del geoportal. A esta organización se le añade la carpeta *static* del sitio de administración de Django. La organización del *front-end* es:

- Dos documentos HTML uno principal y otro de la página de registro. En el Anexo 1 sección 2.2, se muestra la página principal del geoportal, y, en la sección 2.4, la página de registro.
- Un directorio de estilos (*css*) que almacena los iconos e imágenes que utiliza la interfaz del geoportal y los ficheros de programación de los estilos.
- En el directorio *img* se almacenan las imágenes utilizadas para las leyendas del geoportal generadas desde el *back-end*.
- El directorio *js* almacena toda la programación de interacción del geoportal. Dentro de este directorio:
  - La carpeta *map* contiene las interacciones en el mapa
  - Los ficheros *select.js* y *chart.js* de interacción con las series de tiempo
  - Un directorio *register* que almacena todo lo referente a las interacciones en el documento de registro.
  - Finalmente, los módulos utilizados para la programación de la interacción se almacenan en el directorio *externalLibs*.



- Adicionalmente, se crea un fichero *settings.js* con las configuraciones para la etapa de desarrollo y producción, y, otro fichero *general.js* para inicializar variables globales de JavaScript.

Figura 11. Organización de los ficheros del *front-end*



Elaboración propia

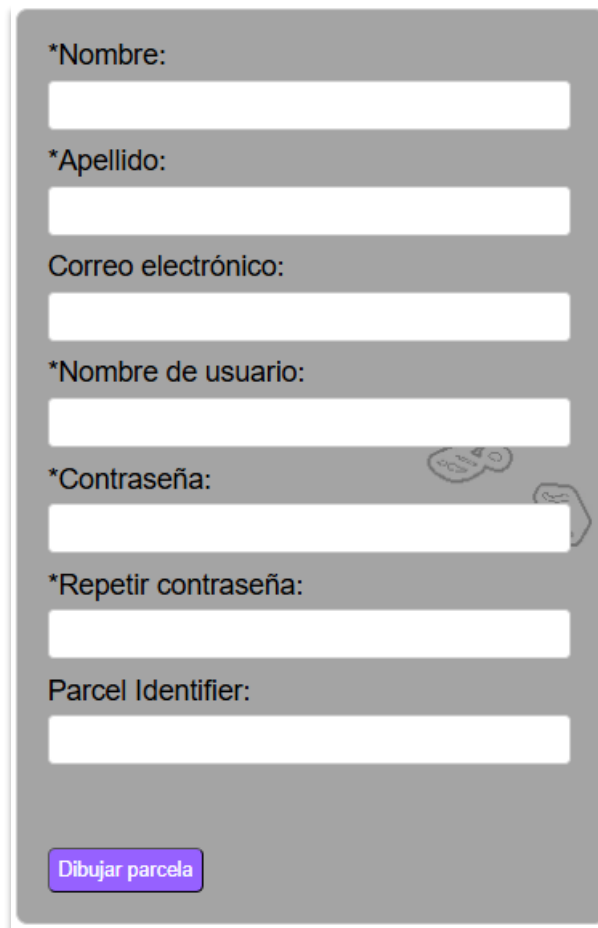
El desarrollo del *front-end* comienza creando los documentos HTML con lenguaje HTML5. La estructura estándar de HTML5 consta de cabecera (*head*) y cuerpo (*body*) de la página web. En la cabecera se importan los ficheros JS y CSS necesarios para agregar los estilos y la funcionalidad del sitio web. En el cuerpo se instancian todos los elementos HTML del geoportal. En el cuerpo también se añade la propiedad "*onload*" y se le asigna una función que se ejecuta una vez cargados todos los elementos HTML de la página web. La función inicia la configuración de funcionalidad asignada a cada elemento HTML. Para ambos documentos en el cuerpo se instancia un pie de página y un encabezado.

#### 4.1.2.1. Sitio web de Registro.

En el HTML de registro se instancia un elemento HTML de tipo formulario con los datos de registro de un usuario en la base de datos, otro elemento HTML de tipo botón para activar el dibujo de la parcela y un elemento HTML para contener el

mapa. Se les aplica los estilos a los elementos. La funcionalidad de estos elementos sirve para recoger la información de registro que se traslada a la base de datos de AgriService. La Figura 12 muestra el formulario en la página de registro.

**Figura 12.** Formulario de registro de un nuevo usuario.

El formulario de registro está presentado en un recuadro gris con un fondo blanco. Incluye los siguientes campos de texto: un campo para el nombre con un asterisco (\*), un campo para el apellido con un asterisco (\*), un campo para el correo electrónico, un campo para el nombre de usuario con un asterisco (\*), un campo para la contraseña con un asterisco (\*), un campo para repetir la contraseña con un asterisco (\*), y un campo para el Parcel Identifier. En la parte inferior izquierda del recuadro hay un botón rectangular con el texto "Dibujar parcela".

*Elaboración propia*

El mapa se inicializa haciendo clic en el botón de dibujar parcela. En este elemento se utiliza la librería *Leaflet* para mostrar dos mapas base, uno de *Open Street Maps* (OSM) y otro con las imágenes de satélite de *Google* y se configura la interacción con el plugin *Leaflet Draw* para crear, editar o borrar polígonos. En el Anexo 1 sección 2.8 se muestra una explicación detallada del uso de cada ícono de *Leaflet Draw* configurado. También se crea un botón de registro que aparece en pantalla al activar el dibujo de la parcela, este botón envía los datos del formulario y la geometría creada al back-end mediante *Ajax*.

#### 4.1.2.2. Sitio web principal AgriService.

En el documento HTML principal *index.html* se crean varios elementos para garantizar una experiencia de usuario funcional, de tal manera que se pueda actualizar en futuras versiones con una experiencia de usuario más intuitiva. Para esto se instancia los siguientes elementos HTML, cada uno con sus estilos asignados en el fichero CSS:

**Bienvenida, Registro y cierre de sesión.** Se instancia un contenedor de bienvenida con un mensaje en una etiqueta de párrafo, donde el texto del mensaje cambia de acuerdo con el nombre de usuario, una vez que este se ha autenticado; la página principal muestra por defecto un mensaje de bienvenida en esta etiqueta. Junto al elemento descrito, se instancian dos botones, uno de Registro, que redirige al usuario a la página de registro, y, otro de cierre de sesión, para cuando un usuario se encuentra autenticado. Se configura estos botones de tal manera que, en la página principal aparezca el de Registro y una vez autenticado aparezca el de cierre de sesión, para esto se utiliza JS. La figura 13 muestra estos elementos.

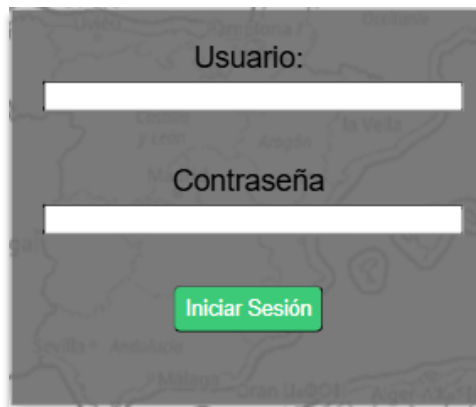
**Figura 13.** Bienvenida, botones de registro y botón de cierre de sesión.



*Elaboración propia*

**Formulario de autenticación.** Este formulario recoge la información de autenticación que será enviada por *Ajax* al *back-end*, también se instancia un botón que activa la función de envío del formulario. Al recibir una respuesta *back-end* se reconfiguran los estilos del encabezado, se oculta el botón de registro, se muestra el botón de cierre de sesión, cambia el mensaje en el párrafo de bienvenida, y muestra el mapa con todos sus íconos de interacción. En caso de recibir una respuesta de no autenticación muestra un mensaje de ingreso incorrecto de credenciales en el en el párrafo de bienvenida. La Figura 14 muestra el formulario de autenticación.

Figura 14. Formulario de autenticación y botón de inicio de sesión.

El formulario de autenticación tiene un fondo gris con un patrón de líneas blancas. En la parte superior, el texto "Usuario:" está centrado sobre un campo de entrada de texto blanco. Debajo, el texto "Contraseña" está centrado sobre otro campo de entrada de texto blanco. En la parte inferior del formulario, hay un botón rectangular con un fondo verde y el texto "Iniciar Sesión" en blanco.

*Elaboración propia*

**Contenedor de toast.** Este elemento se instancia para mostrar mensajes del geoportal y controlar los tiempos de procesamiento e interacciones del usuario. El Anexo 1 sección 2.12 muestra este elemento. Se configura una función que muestra un mensaje y congela los clics del usuario durante la ejecución de una petición al *back-end*.

**Contenedor de fechas.** Se instancia un contenedor con dos etiquetas de entrada de tipo fecha, de esas entradas se toma la fecha de inicio y final de los procesamientos en el geoportal, la visualización de este contenedor se desactiva una vez activado el procesamiento de obtención del mapa de índice de vegetación mediante *Ajax*. En el Anexo 1 sección 3.1 se detalla como ingresar el periodo de tiempo.

**Contenedor del mapa.** Se instancia un contenedor de mapa utilizando la librería *Leaflet*, que en la pantalla principal del geoportal se lo configura para que se visualice una vez autenticado. En este contenedor se han configurado dos mapas base, uno con Open Street Maps OSM y otro con las imágenes de satélite de Google (Google), como capa de teselas en el geoportal. También, se crea una capa con la información de la parcela del usuario que llega del *back-end* y utilizando *Leaflet* se construye un polígono. Además, se crea un control de capas donde se muestran los mapas base, el polígono de la parcela y cada uno de los productos solicitados con su respectiva leyenda.

Cada vez que se activa el cálculo de algún producto se genera una capa de teselas con la URL proporcionada por el *back-end* y se la añade al mapa y al control de capas. Utilizando los marcadores de *Leaflet* se configura un marcador para identificar visualmente el punto de coordenadas donde se encuentra el punto de coordenadas sobre el que se calcula la gráfica de serie de tiempo. El funcionamiento de este contenedor se detalla gráficamente en el Anexo 1 sección 4.

**Iconos y Ventanas.** Los iconos de interacción que se crean para desplegar las ventanas correspondientes. El funcionamiento de estos iconos y despliegue de ventanas se detalla en el Anexo 1. Los contenedores instanciados son:

- imágenes de satélite
- series de tiempo
- información de CC
- mensajes del geoportal

- *Contenedor de imágenes de satélite.* Permite interactuar con las imágenes de satélite mediante un control de desplazamiento, visualizar la fecha de la imagen actual en el mapa en un mensaje instanciado en una etiqueta párrafo y también, tiene un botón para ejecutar la petición de la imagen al *back-end*. En el Anexo 1 sección 4.3 se detalla el uso del control de deslizamiento.

- *Contenedor de información de CC.* Permite visualizar la información de procesamiento mediante mensajes en etiquetas de párrafo instanciados dentro del contenedor y que se actualiza con cada petición de una nueva serie de tiempo. La Figura 17 muestra el contenedor de información CC.

- *Contenedor de gráficos de series de tiempo.* Permite visualizar las series de tiempo generadas, se activa en cada respuesta positiva del *back-end* ante una nueva petición de datos de valores para construir las series de tiempo mediante un clic en la parcela. Se crea también un elemento de entrada de fechas de inicio y fin en un deslizador para interactuar con los gráficos. La Figura 18 muestra esos elementos.

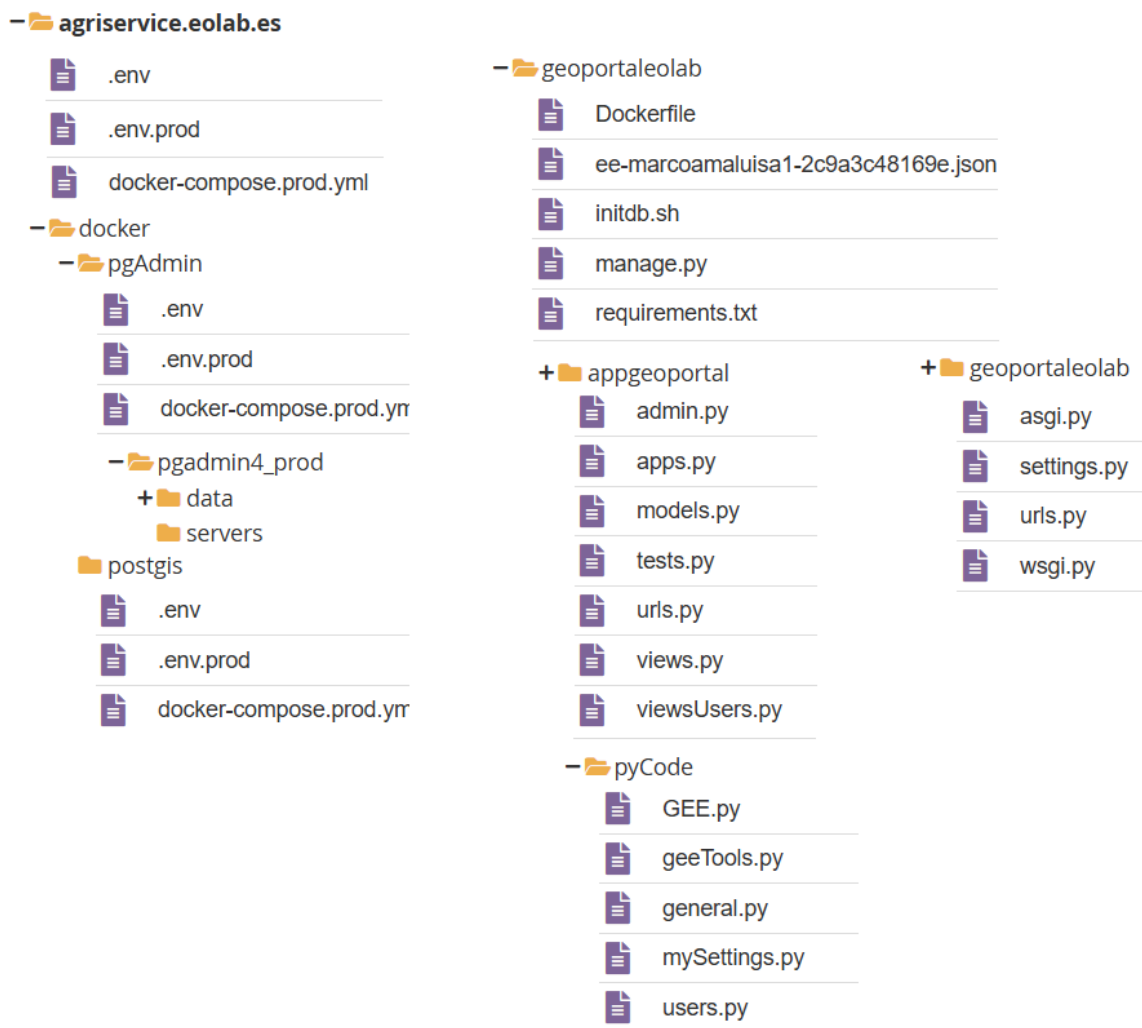
- *Contenedor de mensajes del geoportal.* Dentro de este contenedor se instancian tres etiquetas de tipo párrafo en los que se va apilando mensajes del geoportal respecto a la petición de mapas, y series de tiempo, además, también muestra los errores y advertencias de cada procesamiento. La Figura 19 muestra el contenedor de mensajes.

**Interacción con los productos.** Se crean cuatro iconos con un párrafo informativo cada uno, para cada producto calculado en el geoportal, se puede visualizar en el Anexo 1 sección 3. Al activar el ícono de NDVI se desactiva el contenedor de fechas y se activan los iconos de temperatura de superficie, fracción de cobertura vegetal y evapotranspiración, también se activa la interacción con las imágenes de satélite y envía la petición al *back-end* para procesar el mapa de NDVI, además activa la captura de clics sobre el mapa para la generación de series de tiempo y se configura los estilos para darle una opacidad a los productos inactivos. En el fichero CSS se configura los estilos de los elementos para que los párrafos aparezcan cuando el ratón se coloca arriba de cada ícono.

#### 4.1.3. Back-end

En esta sección se describe la integración de los algoritmos de teledetección con las imágenes de satélite mediante la programación de la cadena de procesado en *Python* utilizando la API de GEE. La organización de los ficheros en el *back-end* se muestra en la figura 15.

Figura 15. Ficheros del back-end en el VPS



*Elaboración propia*

Para poder acceder a la Api de *Python* de manera adecuada y comenzar la programación de los algoritmos de teledetección, se crea un proyecto en la nube (Cloud Project) para albergar cuentas de servicio, y se crea una cuenta de servicio siguiendo la documentación para crear aplicaciones de GEE. Luego se crea una llave privada para esta cuenta de servicio, que es un fichero JSON que permite acceder a la API de GEE para el desarrollo de aplicaciones.

Para comenzar a desarrollar se crean tres ficheros de *Python*, el primero *MySettings.py* sirve para almacenar el id de la cuenta de servicio y la clave asociada. Además, en la etapa de desarrollo del geoportal este fichero es útil para colocar directorios de almacenamiento e información de la base de datos.

El segundo fichero, es *GEE.py* donde se programan siete funciones;

- Una función que interactúa con la base de datos para seleccionar la parcela de un usuario ya registrado.
- Otra función para llevar la parcela seleccionada al *front-end* del geoportal en el formato adecuado.
- Cuatro funciones para calcular las series de tiempo de cada variable biofísica en base al diagrama de flujo respectivo y utilizando los métodos de la clase *AgriServiceGEE2*, desarrollada con la API de *Python* de GEE.
- Una última función, que sirve para obtener la imagen recortada respecto de la parcela del usuario, aplicar una paleta de colores para la simbología y generar una imagen de leyenda siguiendo el diagrama de flujo de la generación de mapas.

El tercer fichero es *geeTools.py*, que tiene una clase de *Python* de nombre *AgriServiceGEE2* donde se desarrollan 28 métodos útiles para hacer CC y extraer los datos de las series de tiempo y generar mapas de las variables biofísicas, utilizando los algoritmos de teledetección en GEE. La clase *AgriServiceGEE2* recibe dos argumentos de entrada, un id de una cuenta de servicio de GEE, y una llave asociada a dicha cuenta de servicio.

En la etapa de desarrollo del geoportal se utiliza un editor de código para programar los 28 métodos de la clase *AgriServiceGEE2*. La API de *Python* tiene limitaciones de visualización en comparación con el *Code Editor* de la propia plataforma de GEE por lo que se desarrolló cada método en el entorno de GEE en el *Code Editor* y luego se traduce el código a *Python* para integrarlos en el *back-end*.

El tiempo de cómputo en la obtención de las series de tiempo y mapas de las variables biofísicas dependen del adecuado desarrollo de la interacción entre las funciones de *GEE.py* y los métodos de la clase *AgriServiceGEE2* de *geeTools.py*. Las funciones de fichero *GEE.py* y los métodos de la clase *AgriServiceGEE2* se describen con más detalle en el Anexo 2.



Es necesario tomar en cuenta que *PostGIS* y *GEE* trabajan con el formato estándar *GeoJSON* de orden de las coordenadas, es decir, primero longitud y luego latitud para un punto de coordenadas en las geometrías. *Leaflet* trabaja con coordenadas no estandarizadas y ordena un punto de coordenadas colocando primero la latitud, y luego el valor de longitud en las geometrías. La Figura 16 muestra el cambio realizado en la programación del código de *Python* para controlar este aspecto, tanto para la selección de la parcela, utilizando el método *polygon2Leaflet* de la clase *AgriServiceGEE2*, como para las funciones de generación de mapas y creación de las series de tiempo.

Figura 16. Cambio de formato de coordenadas en Python.

ESTÁNDAR A LEAFLET

```
#Extension
coordinates_to_Leaflet = aGEE.polygon2Leaflet(user_data['extension'])

coordinates_to_Leaflet = []

for data in polygonGoeJson['coordinates']:
    for coordinate in data:
        coordinates_to_Leaflet.append(coordinate[::-1])
```

LEAFLET A ESTANDAR

## Mapas

```
polygon = [coord[::-1] for coord in ext]

polygon_leaf = aGEE.polygon2Leaflet(aGEE.Polygon(polygon))
params_mapID['extension'] = polygon_leaf
```

## Series de tiempo

```
#leaflet a GEE
coords = coords[::-1]
point = aGEE.Point(coords)
ext = [coord[::-1] for coord in ext]
polygon = aGEE.Polygon(ext)
```

*Elaboración propia*

**Programación de la Obtención de Mapas.** En el caso de la obtención de mapas, se crea una función que recibe las coordenadas de la extensión de la parcela, un rango de fechas, el nombre de un conjunto de datos de GEE y el índice GEE de una imagen, que por defecto tiene valor *None*. La función verifica que las fechas sean coherentes, es decir, que la fecha de inicio del rango no sea futura a la fecha de final del rango. GEE recibe coordenadas en formato estándar, se transforma el formato recibido de *Leaflet* al estándar, en el caso de que no se especifique ningún id de imagen de GEE, llama al método *imagesList*, que recibe como atributos de entrada, un polígono, un rango de fechas, y el nombre de un conjunto de datos de GEE, y proporciona una colección de imágenes preprocesadas en un determinado rango de fechas. En el caso de que se proporciona el id de una imagen no calcula ninguna colección y trabaja directamente con el id.

Estos datos se organizan como parámetros de entrada para enviarlo al método *mapID* que recibe un indicador entre los parámetros que indica cuál de las cuatro variables biofísicas se obtendrá el mapa y devuelve una *URL* con el mapa procesado, además genera una imagen con la librería *Pillow* de la paleta de colores y los valores calculados en la leyenda y lo almacena en un determinado directorio. Este método puede recibir el id de GEE de una imagen o una lista de ids de GEE y tratarlos para generar el mapa. El JSON generado se pasa al *front-end* que mediante funciones de *JavaScript* genera la imagen, un ejemplo de la salida generada se muestran en la Figura 17.

Figura 17. JSON de salida generada por el *back-end* para generar la imagen de NDVI.

```

{
  "ok": "true",
  "message": "Imagen de Índice de vegetación en el mapa. Haz clic sobre la imagen",
  "data": {
    "urlImageEE": {
      "url": "https://earthengine.googleapis.com/v1alpha/projects/earthengine-legacy/maps/3784fbc13b0216718d4cc615be161b3f-9320afe3b19667d0892b35e06ecc70a7/tiles/{z}/{x}/{y}",
      "id": "20211216T073219_20211216T074502_T37MET",
      "product": "S2B_MSIL2A_20211216T073219_N0301_R049_T37MET_20211216T100529",
      "message": "Imagen de Índice de vegetación en el mapa. Haz clic sobre la imagen"
    },
    "params_map": {
      "index": "20211216T073219_20211216T074502_T37MET",
      "datetime": [
        "2023-01-01",
        "2024-01-25"
      ],
      "extension": [
        [39.061853,-2.103321],
        [39.060207,-2.102849],
        [39.05669,-2.1016040],
        [39.051957,-2.099762],
        [39.046193,-2.097657],
        [39.042768,-2.096239],
        [39.043868,-2.09100],
        [39.045389,-2.091553],
        [39.045523,-2.091500],
        [39.045689,-2.091371],
        [39.045806,-2.091135],
        [39.045981,-2.089847],
        [39.047524,-2.088899],
        [39.047957,-2.088041],
        [39.054119,-2.081689],
        [39.054619,-2.081345],
        [39.055152,-2.082075],
        [39.059149,-2.080959],
        [39.060049,-2.083108],
        [39.060498,-2.084331],
        [39.060632,-2.085962],
        [39.060624,-2.087572],
        [39.060441,-2.088216],
        [39.061853,-2.103321]
      ],
      "tile": "NDVI",
      "collection": "COPERNICUS/S2_SR"
    }
  }
}

```

Elaboración propia

**Programación de la obtención de series de tiempo.** Para la creación de series de tiempo se sigue el diagrama de flujo correspondiente y se desarrollan cuatro funciones, que reciben las mismas entradas cada una, y devuelven los resultados del procesamiento en un mismo formato de salida. Las entradas para el índice de vegetación cambian por el hecho de que calcular el índice de vegetación es obligatorio para todos los algoritmos de teledetección desarrollados en el geoportal. Para el cálculo de FCOVER, LST y ET se incluye un parámetro de entrada más que es una cadena de caracteres con el nombre de la variable a calcular.

Las funciones verifican que uno de los parámetros proporcionados sea el índice de vegetación, transforma las coordenadas de formato *Leaflet* a formato estándar, utilizan el método *checkInsideOutside* de la clase *AgriServiceGEE2*, verifica que el punto de coordenadas se encuentre dentro de un polígono. Si el punto está fuera de la parcela devuelve un mensaje con esta advertencia. Luego con el método *ImageCollection* selecciona una colección de imágenes de satélite de *Sentinel 2*, de acuerdo con el conjunto de datos especificado en el *front-end*.

Luego se hace un preprocesamiento a las imágenes de satélite con el método *preProcessingSentinel* de la clase *AgriServiceGEE2* que recorta la imagen con respecto a una determinada geometría, en este caso el polígono de la parcela, aplica una máscara de nubes en las imágenes de la colección utilizando la banda *SceneClassificationSCL*, donde los valores de los píxeles estén clasificados como vegetación (4) y suelo desnudo (5), y, crea una colección con las imágenes donde en un determinado punto de coordenadas los valores son vegetación y suelo desnudo.

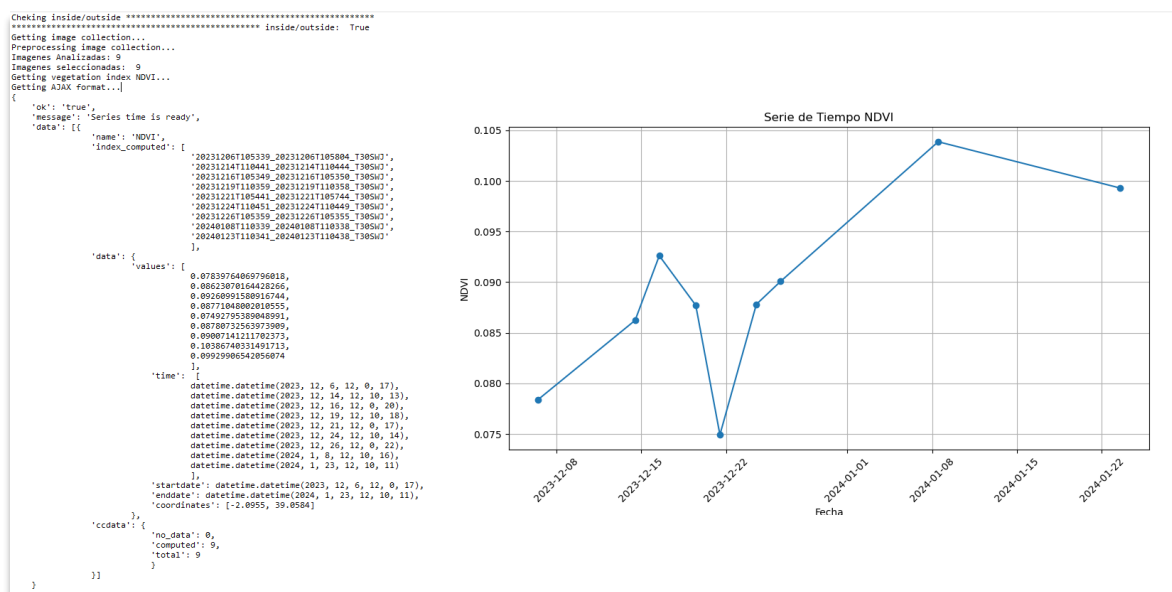
Para cada una de las variables biofísicas se utiliza los métodos de la clase *AgriServiceGEE2*, desarrollados en base al flujograma de la cadena de procesado de EOLAB. La Tabla 8 muestra los métodos aplicados para la obtención de cada producto después del preprocesamiento. La descripción de estas funciones se encuentra detallada en el Anexo 2 tabla 1.

**Tabla 8.** Métodos de la clase *AgriServiceGEE2* utilizados para la generación de los productos de la cadena de procesamiento de EOLAB.

Producto	Método de la clase <i>AgriServiceGEE</i>
Índice de Vegetación (NDVI)	<ul style="list-style-type: none"> <li>• <i>calcIVfromCollection</i></li> </ul>
Temperatura de superficie (LST)	<ul style="list-style-type: none"> <li>• <i>calcIVfromCollection</i></li> <li>• <i>calcLSTfromCollection</i></li> </ul>
Fracción de Cobertura Vegetal (FCOVER)	<ul style="list-style-type: none"> <li>• <i>calcIVfromCollection</i></li> <li>• <i>calcLSTfromCollection</i></li> <li>• <i>calcFCOVERfromCollection</i></li> </ul>
Evapotranspiración (Necesidad hídrica total - ET)	<ul style="list-style-type: none"> <li>• <i>calcIVfromCollection</i></li> <li>• <i>calcLSTfromCollection</i></li> <li>• <i>calcALBFromCollection</i></li> <li>• <i>calcRNFromCollection</i></li> <li>• <i>calcVectorsFromCollection</i></li> <li>• <i>calcRegressionFeomCollection</i></li> <li>• <i>CalcSSEBIFromCollection</i></li> <li>• <i>removeOutliersFromCollection</i></li> <li>• <i>calcSSEBINormalizedFromCollection</i></li> <li>• <i>calcETFromCollection</i></li> </ul>

Una vez calculado el producto se traslada los datos calculados de GEE al local utilizando el método *valuesInPoint* que extrae los valores de una colección de imágenes, especificando las propiedades que se requiere de cada imagen y la banda de la que se requiere extraer los valores, finalmente, se organizan estos valores en un formato para enviarlo a través de Django al *front-end*. La Figura 18 muestra un ejemplo del resultado arrojado por el *back-end* para generar la serie de tiempo.

Figura 18. Salida del *back-end* de AgriService



Elaboración propia

**Configuración del Proyecto Django.** En la etapa de desarrollo, en el fichero *Settings.py* del proyecto Django creado se edita la configuración de *ALLOWED\_HOST*. Para permitir cualquier *host* acceder a la aplicación, se configura la base de datos especificando el nombre, el puerto y el host que en este caso es *localhost*, se añade a las aplicaciones instaladas la aplicación creada en la sección *INSTALLED\_APPS*, y se hace la migración para aplicar las configuraciones. Además, se crea el súper usuario *eolab* que es el administrador del sitio de Django, del que posteriormente se cambia la contraseña al pasar del modo desarrollo al modo despliegue.

Una vez realizadas las configuraciones, se crean dos ficheros en el directorio de la aplicación, *url.py*, donde se programan las URLs de la aplicación que se vinculan una con cada vista en el fichero *views.py*, y, en este fichero se crean las funciones para cada fichero, la Tabla 9 muestra las URLs configuradas y la función del fichero *views.py* asociada.

**Tabla 9.** Configuración de las URLs y vistas en la aplicación *Django*.

URL	VISTA	Tipo de petición	Auth	Descripción
helloWorld/	helloWorld (View)	GET	NO	Sirve para probar el correcto funcionamiento de la aplicación en la etapa de desarrollo producción y despliegue de la aplicación
selectParcela/	selectParcela (LoginRequire dMixin, View)	POST	SI	Selección de la parcela de un usuario almacenado en la base de datos.
tileParcelaNDVI/	tileParcela (LoginRequire dMixin,View)	POST	SI	Obtener URL del mapa de NDVI de GEE
tileParcelaLST/	tileParcela (LoginRequire dMixin,View)	POST	SI	Obtener URL del mapa de LST de GEE
tileParcelaFCOVER/	tileParcela (LoginRequire dMixin,View)	POST	SI	Obtener URL del mapa de FCOVER de GEE
tileParcelaET/	tileParcela (LoginRequire dMixin,View)	POST	SI	Obtener URL del mapa de ET de GEE
chart/	Chart (LoginRequire dMixin,View)	POST	SI	Obtener los valores de las series de tiempo de las variables biofísicas de GEE

Para los mapas se desarrolla una URL para cada variable, de manera que al solicitar varias URLs de la imagen se ejecute la misma vista y el *back-end* organice la salida de información requerida. Para las series de tiempo se crea una sola URL con una sola vista donde la información para ingresar al *Back-end* se organiza en la vista *Chart*. Finalmente se conecta las vistas con las funciones del Fichero *GEE.py* del *back-end* y se obtiene una aplicación con URLs funcionales.

**Desarrollo del registro y gestión de usuarios.** Una vez configurado y conectado el *back-end* con las URLs y vistas correspondientes se crea otro fichero *users.py* donde se programa todo lo referente a la autenticación y registro de los usuarios. Para esto se desarrollan cuatro URLs en el fichero *URLs.py*. Las funciones se detallan en la Tabla 10, con esto el usuario puede iniciar y cerrar sesión en el geoportal y también registrar nuevos usuarios en la base de datos. Además, se crea un fichero adicional *ViewsUsers.py* para configurar las vistas correspondientes a cada URL.

**Tabla 10.** URLs configuradas para la autenticación y registro de usuarios.

URL	VISTA	Tipo de petición	Descripción
app_login/	AppLogin	POST	Verifica que el usuario este registrado y lo autenticado, en caso de no estar registrado devuelve un mensaje de "Credenciales incorrectas".
app_logout/	AppLogout	GET	Cierra la sesión de un determinado usuario.
app_register/	AppRegister	POST	Registra un nuevo usuario en la base de datos y notifica al administrador del geoportal.

En cuanto al registro de usuarios, Django por defecto coloca activo al registrar un nuevo usuario en la base de datos. Hay que tener en cuenta que, para poder autenticarse es necesario verificar el pago, para esto se cambia la configuración para que el registro de un usuario nuevo por defecto sea inactivo, esto implica que, una vez registrado el usuario, deberá ser activado manualmente por el administrador del geoportal. La activación de usuarios de manera interactiva se hace a través del acceso la página de administración de Django, por lo que, se configura el acceso al sitio web de administración de Django en la etapa de despliegue del geoportal. Se configura el fichero *settings.py* donde se añaden las configuraciones de la Figura 19 para la activación del sitio de administración Django a través del proxy asignado.

**Figura 19.** Configuración del fichero *settings.py* en el VPS para tener acceso al sitio de administración de Django.

```
STATIC_ROOT = '/[REDACTED]/public_html/AgriService/agriservice_static/'
STATIC_URL = 'agriservice_static/'
CSRF_TRUSTED_ORIGINS=['https://agriservice.eolab.es']
```

También, es necesario configurar el envío de emails, para notificar al administrador de AgriService sobre el registro de un nuevo usuario y para enviar los datos de pago al usuario registrado. En la etapa de producción del geoportal, se edita el fichero *settings.py* con las configuraciones que se muestran en la Figura 20. En esta figura se observa el método *getenv* de la librería *os*, que es un método útil para extraer variables de entorno, en este caso utilizada para extraer la variable de entorno del contenedor Docker sobre las configuraciones de e-mail, de esta manera, es posible cambiar configuración de credenciales de email sin necesidad de configurar el fichero *settings.py*.



Figura 20. Configuración de las variables necesarias para enviar e-mail desde el *back-end* en el fichero *settings.py*.

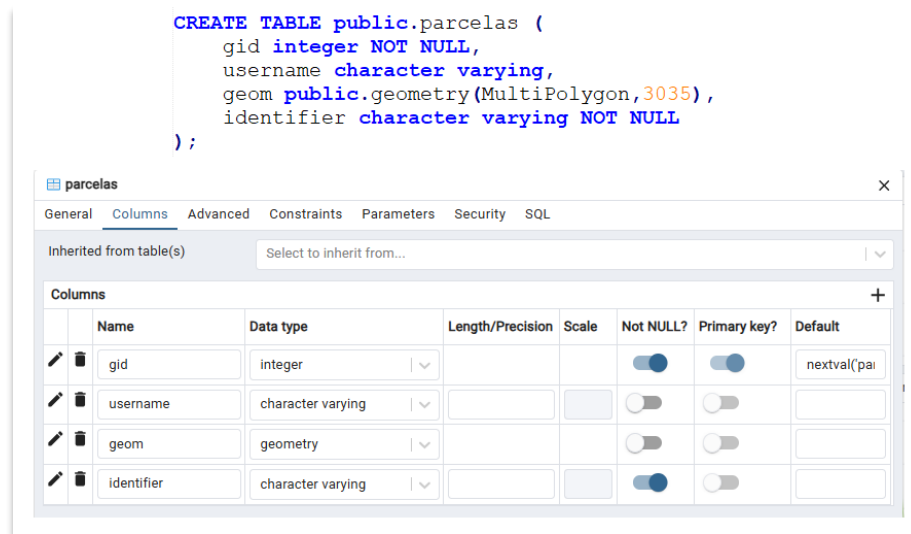
```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'  
EMAIL_HOST = os.getenv("EMAIL_HOST")  
EMAIL_PORT = os.getenv("EMAIL_PORT")  
EMAIL_USE_TLS = True  
EMAIL_HOST_USER = os.getenv("EMAIL_HOST_USER")  
EMAIL_HOST_PASSWORD = os.getenv("EMAIL_HOST_PW")
```

#### 4.1.4. Base de datos (usuario/parcela)

La base de datos servirá para almacenar los datos del usuario y las parcelas asignadas a cada uno, por lo que, para el modo desarrollo se instala *PostgreSQL*, la extensión *PostGIS* a nivel local en un ordenador, y se crea una base de datos que alberga las Tablas de Django en el esquema *public*. Además, se crea la tabla *parcelas*, con los campos especificados en la Figura 21. Donde:

- El campo *gid* es un número entero que se utiliza como identificador único para cada usuario, este valor no puede ser nulo.
- El campo *username* contiene el nombre de todos los usuarios registrados como cadenas de texto.
- El campo *geom* almacena las geometrías de tipo multipolígono en el sistema de coordenadas ETR89-extendido para el área de Europa con código EPSG 3035
- El campo *identifier* es una cadena de texto con el identificador de la parcela asignada a un usuario.

Figura 21. Creación y especificación de los campos de la tabla parcelas.



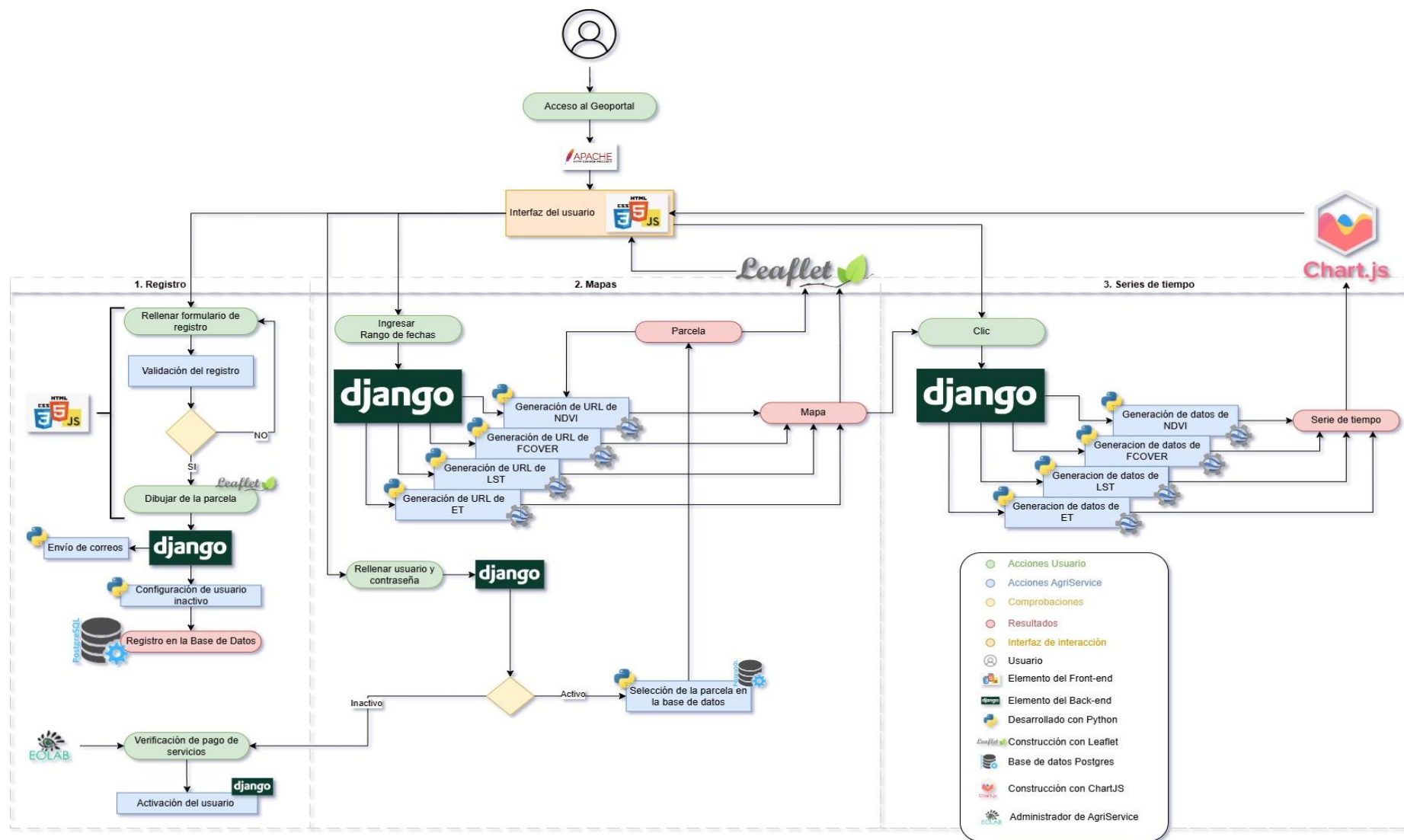
## 4.2. Flujograma de los Procesos

La Figura 22 describe los procesos creados para el funcionamiento del geoportal, aplicando las tecnologías de desarrollo mencionadas en la metodología. Todo lo referente al *back-end* se hace después de pasar por las URLs configuradas con Django. Y todo lo referente al *front-end* está construido utilizando HTML (para instanciar elementos), CSS (para darle estilo a dichos elementos) y JS (para la funcionalidad).

Los procesos comienzan con la interacción del usuario con la interfaz, siendo 3 principales procesos los que se ejecutan: registro, creación de mapas y generación de series de tiempo. La creación de mapa termina en la construcción de las *tileLayers* con la librería *Leaflet*, un *tileLayer* que es un tipo de capa de *Leaflet* que almacena ráster también conocidos como mosaicos o arreglos de pixeles, y es útil para mostrar los mapas en la Interfaz del usuario.

Lo mismo sucede con los gráficos de series de tiempo que solo pueden activarse una vez mostrados los mapas en la interfaz, las series de tiempo pasan a la interfaz del usuario, construyendo un gráfico con *ChartJS*.

Figura 22. Esquema los procesos de funcionamiento del geoportal AgriService



#### 4.2.1. Registro de usuario/parcela

Se establece que para poder interactuar con los productos del geoportal AgriService es necesario asociar un usuario a una determinada extensión de terreno o parcela de interés, que deberán estar registrados en la base de datos de usuarios.

En la Figura 23 se describe la estructura de funcionalidad del registro de usuarios del geoportal AgriService con las tecnologías de desarrollo web utilizadas. Es necesario que la interacción del usuario con el geoportal comience con el registro de un área geográfica de interés, por lo que, mediante el acceso a la página web por el servidor Apache HTTP, en el *front-end*. *El usuario debe:*

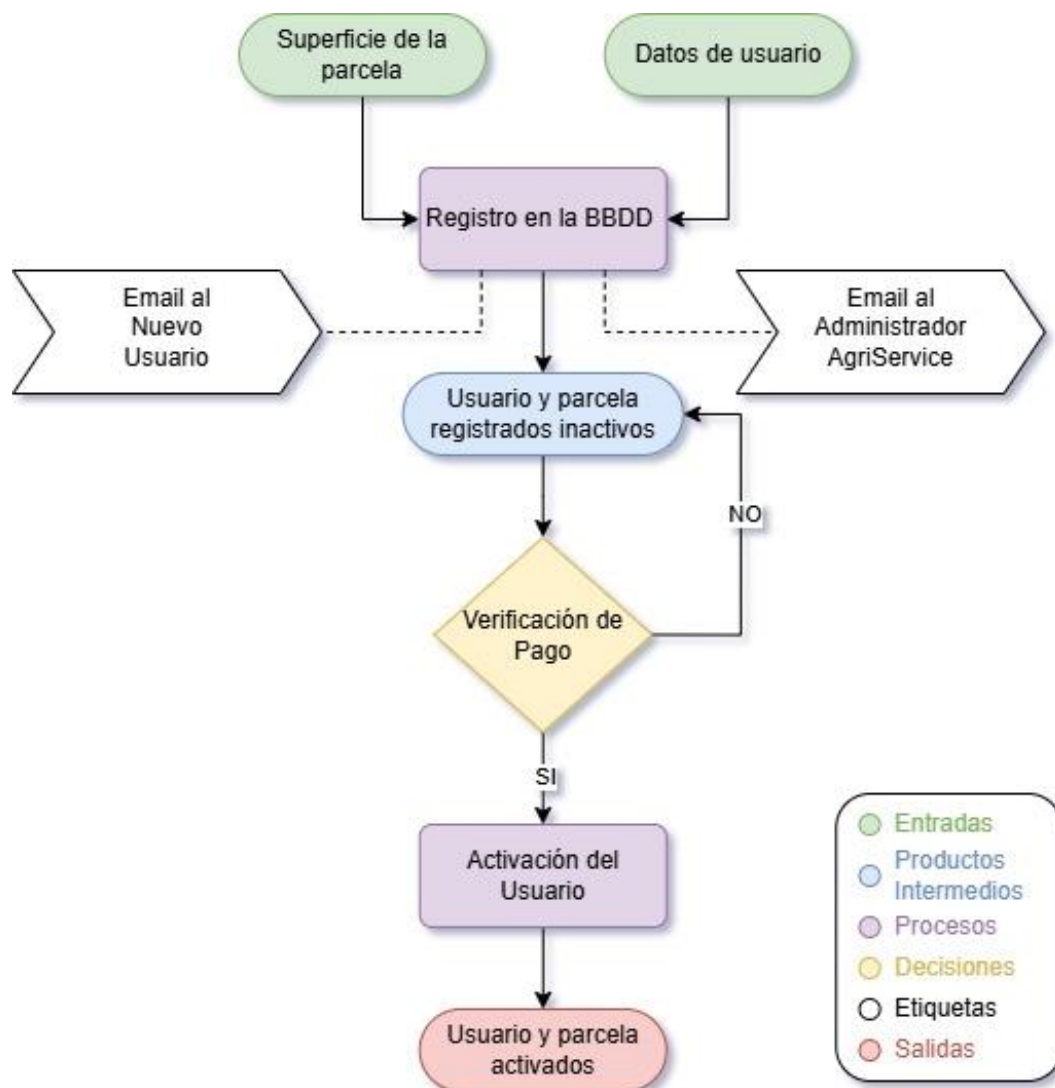
- Rellenar el formulario de registro
- Dibujar un área de interés donde desea realizar los cálculos que se recogen en un JSON.

Al aceptar se hace una petición de tipo *POST* al *back-end*, utilizando el módulo *Ajax* de JavaScript, con la ayuda del marco de desarrollo web Django:

- Se recogen los datos de la petición *Post*.
- Se tratan los datos del usuario y la información gráfica de la parcela, para que el formato se pueda ingresar en la Tabla de usuarios y parcelas en la base de datos.
- Se registra el usuario y la parcela asociada.
- Se informa de este registro al administrador del geoportal AgriService, y al usuario, mediante correo electrónico

El usuario queda registrado, temporalmente inactivo, hasta que el administrador del geoportal cambie su estado de usuario registrado inactivo a usuario registrado activo.

Figura 23. Diagrama de flujo del registro de un nuevo usuario en el geoportal AgriService.



*Elaboración propia*

Cada usuario accede a un formulario de registro para ingresar sus datos. Estos datos se pueden asociar a una parcela dibujada en el mapa y se estructuran para almacenarlos en la base de datos de usuarios del geoportal AgriService. Una vez completado el registro, se enviará una notificación por correo electrónico tanto al usuario como al administrador del geoportal.

Los usuarios nuevos quedan registrados en la base de datos como usuarios inactivos, mientras se realiza el pago de los servicios a la empresa, una vez verificado el pago realizado, se activa al usuario, dándole acceso a los productos generados mediante los algoritmos de la empresa EOLAB.

#### 4.2.2. Mapas de las variables biofísicas

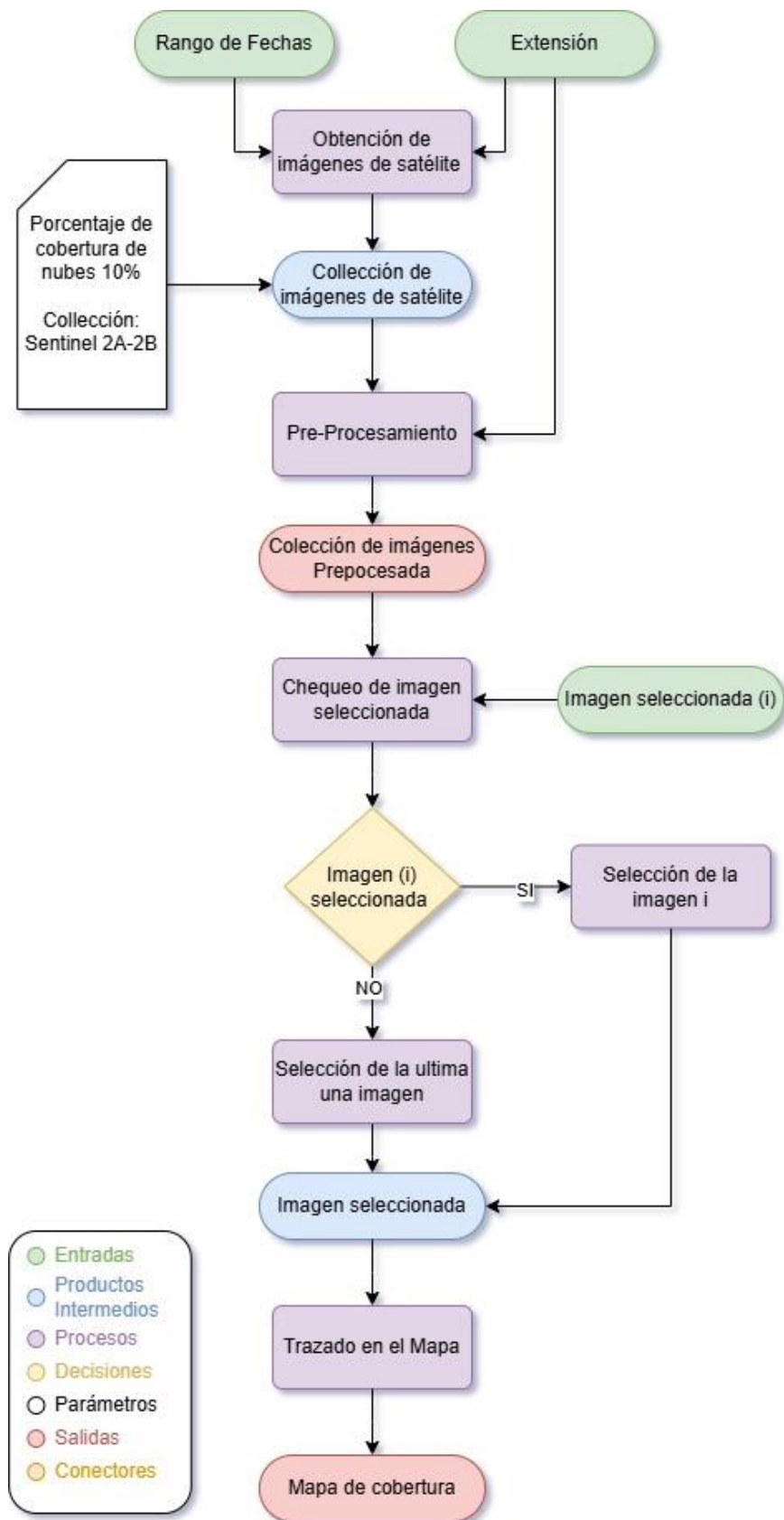
Una vez registrada la parcela, es necesario autenticarse con el nombre de usuario y contraseña registrado, cabe recalcar que, mientras no se encuentre autenticado un usuario no tiene acceso a la visualización de coberturas de las variables biofísicas en el mapa.

El usuario ingresa como datos entrada, un intervalo determinado de fechas para calcular las imágenes de satélite de Sentinel-2A y 2B, y activa el cálculo del NDVI, este producto es obligatorio para los algoritmos de las otras tres variables biofísicas, por lo que, es necesario que se active siempre que se quiera realizar una estimación de cualquier variable en el geoportal. Una vez activado el NDVI, es posible activar cualquiera de los otros productos del geoportal, luego, se toma la extensión de la parcela del usuario y el intervalo de fechas ingresado, y se extrae una colección de imágenes de satélite en la zona y fechas determinadas, además se le añade parámetros de nubosidad. Por defecto, se utilizan imágenes donde el porcentaje de píxeles clasificados como nubes sea menor al 10% de la escena esté cubierta por nubes.

Se realiza un preprocesamiento a las imágenes obtenidas, donde se descartan píxeles que no estén clasificados como vegetación o no-vegetación dentro de la escena y se recorta la imagen de satélite a la parcela del usuario en cuestión. De este procesamiento se obtiene una colección con todas las imágenes disponibles para el cálculo del NDVI, esta colección está disponible para que el usuario seleccione cualquier imagen que requiera visualizar. La Figura 23 describe el flujograma de obtención de imágenes de satélite para el geoportal.

El cálculo de los mapas de cobertura de FCOVER, LST y ET llevan el flujo de la Figura 24, aumentando las entradas y productos intermedios que se generan al incluir los algoritmos de teledetección de cada variable en específico.

Figura 24. Flujo de obtención del mapa en el geoportal



Elaboración propia

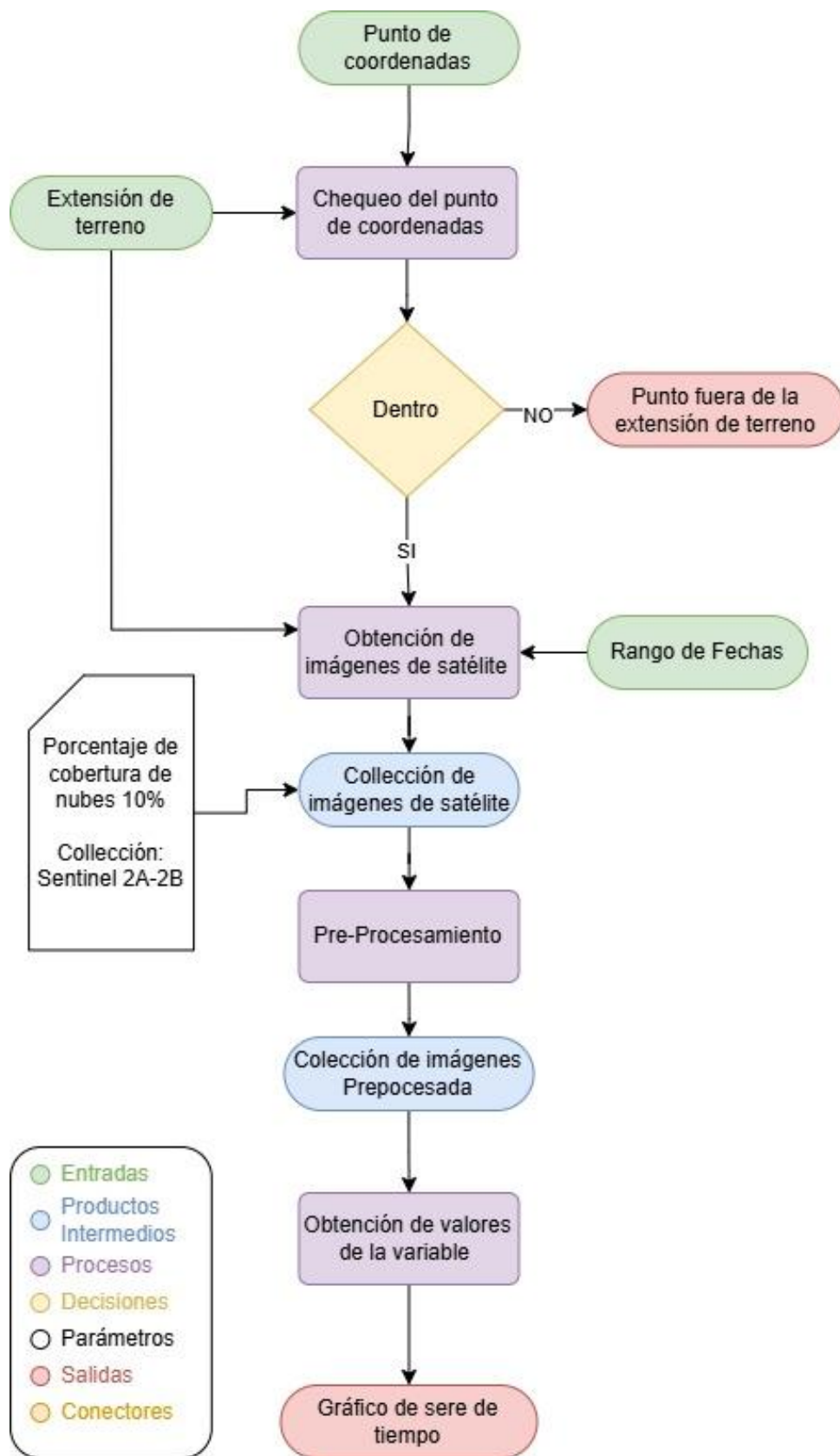
### 4.2.3. Series de tiempo de las variables biofísicas

Una vez establecido el intervalo de tiempo y activada la visualización en el mapa de las variables, es posible calcular una serie de tiempo sobre una coordenada en específico, cabe recalcar que si no se encuentra activada la visualización en el mapa no será posible la visualización de las series de tiempo sobre una coordenada.

La metodología utilizada para el cálculo de las series de tiempo es similar a la de la visualización de las coberturas, sin embargo, cambia luego de generar la colección de imágenes preprocesadas, luego de esto, depende de cada algoritmo de teledetección, y de sus entradas, y productos intermedios, la Figura 25 pretende representar de manera general para todas las variables la metodología de obtención de las series de tiempo.



Figura 25. Flujo de obtención de serie de tiempo en el geoportal.



Elaboración propia

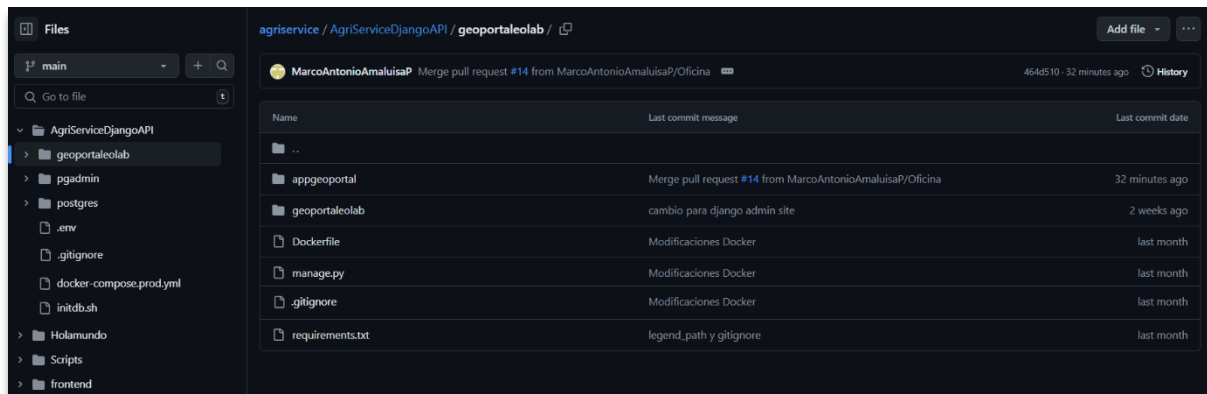
### 4.3. Almacenamiento en la etapa de desarrollo

El almacenamiento en un repositorio GitHub aportó significativamente a la velocidad de desarrollo, producción y despliegue del geoportal, especialmente en términos de control de versiones y automatización. Los primeros pasos en la etapa de desarrollo del geoportal, se realizaron utilizando ficheros comprimidos, ficheros de texto, hojas de cálculo y documentos de Microsoft Office para documentar el desarrollo, lo que limita en tiempo, control de versiones en la construcción del geoportal. Con el repositorio de GitHub el seguimiento a versiones anteriores el momento de cambiar algoritmos en los métodos en la clase creada con las herramientas de la API de GEE se realiza de una manera eficaz.

Una vez instalado los requerimientos Git para el uso del repositorio se crea un proyecto de tipo privado para no comprometer los algoritmos de la empresa y se añaden los ficheros desarrollados hasta el momento a excepción de las credenciales de la cuenta de servicio de GEE. Esto permite el desarrollo a nivel local en la oficina de EOLAB y en ordenadores fuera de la empresa siempre y cuando se disponga de una llave y una cuenta de servicio. La etapa de desarrollo del Geoportal se realiza en un ordenador local en las oficinas de EOLAB, y por teletrabajo en un ordenador fuera de la oficina de EOLAB.

Los ficheros colocados en el repositorio privado son los de creación de contenedores de *Docker* para la construcción de la base de datos *PostgreSQL* y su cliente *pgAdmin*. Al utilizar *Docker Compose* se facilita el manejo de las bases de datos en ficheros que se pueden subir al repositorio GitHub, por lo que se coloca en el repositorio los ficheros de extensión *yml* y se ignora la clonación de ficheros *prod* para evitar comprometer variables de entorno de puertos y configuraciones del VPS. La figura 26 muestra los ficheros subidos al repositorio GitHub.

Figura 26. Ficheros en el repositorio GitHub



*Elaboración propia*

## 5. Presupuesto

Se desglosa el presupuesto necesario para el desarrollo del geoportal AgriService. Se abordarán los siguientes aspectos:

1. Tiempo de empleo del desarrollador.
2. Costo de las tecnologías de desarrollo.
3. Presupuesto de desarrollo del proyecto.
4. Presupuesto mensual para continuidad del proyecto.

### 5.1. Tiempo de Empleo del Desarrollador.

El desarrollo del Geoportal AgriService requiere la contratación de un desarrollador con experiencia en teledetección y desarrollo de aplicaciones web, es decir, un ingeniero geomático. Valor medio de un geomático en España de 24 000 a 30 000 euros anuales, por lo tanto, se considera un valor de 15 euros/hora (este valor puede variar según la experiencia y la ubicación específica del profesional). El trabajo se estima para alcanzar un total de 751 horas de trabajo, con un horario de 8 horas diarias durante 5 días a la semana. El cálculo de este recurso se muestra en la Tabla 10.

Tabla 11. Presupuesto parcial del costo del desarrollador

Tiempo estimado	Valor (euros/hora)	Total
Total de horas de trabajo: 751 horas		
Horas de trabajo por día: 8 horas		
Días de trabajo por semana: 5 días	15.00 €	751 horas * 15 euros/hora
Horas de trabajo por semana: 40 horas/semana		<b>11 265 euros</b>
Total de semanas necesarias: 19 semanas		

## 5.2. Costo de las Tecnologías de Desarrollo

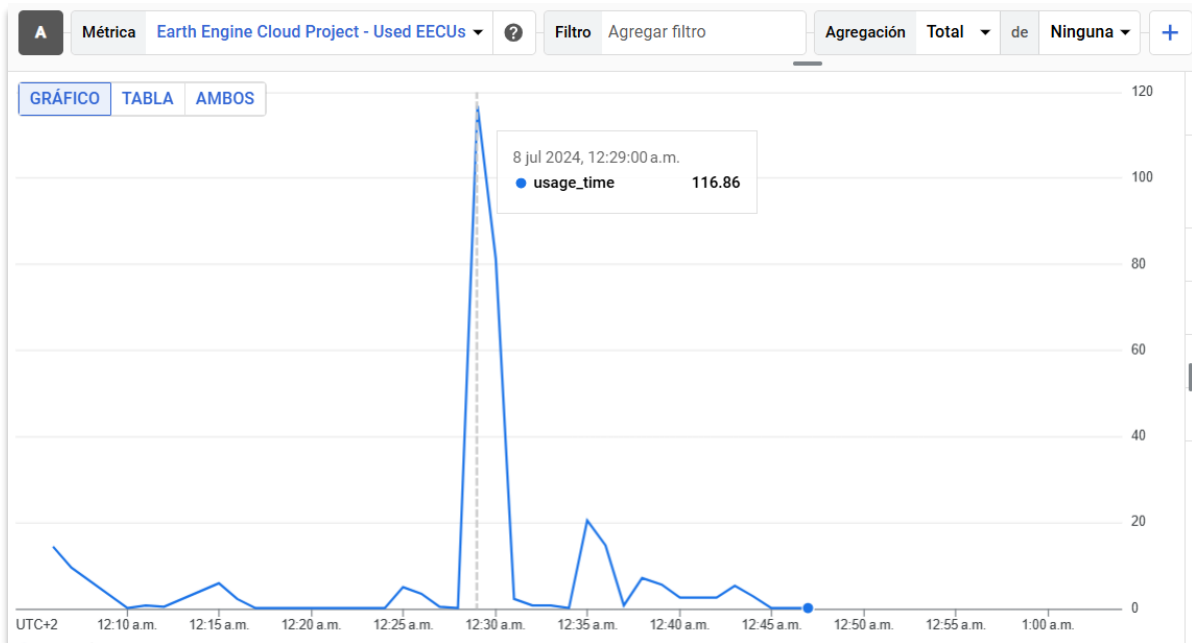
Para el desarrollo del geoportal se utilizarán diversas tecnologías, todas ellas de acceso libre y gratuito, o semi gratuito que no implican ningún costo para el desarrollo y comercialización, a excepción de Google Earth Engine (GEE), por lo que se le da un apartado en específico en esta sección.

### 5.2.1. Precios de la Plataforma Google Earth Engine

Google Earth Engine (GEE) ofrece un nivel gratuito de uso para actividades no comerciales que principalmente quedan englobadas en el rubro de investigación y aprendizaje. El desarrollo del Geoportal se ha realizado con una cuenta no comercial destinada a la investigación, sin embargo, por la intención de comercializar los productos de la empresa EOLAB, el proyecto cae sobre el rubro de uso comercial.

GEE utiliza Unidades Computacionales de Earth Engine (EECU) que es una unidad de medida utilizada por Google Earth Engine para calcular y limitar el uso computacional de los recursos en la plataforma. Se ha proyectado un presupuesto en base a las cantidades procesadas actualmente con la cuenta no comercial donde los EECUs se obtiene de un procesamiento en el geoportal con los cuatro productos activos con 59 imágenes de satélite analizadas y 58 procesadas, la Figura 27 muestra el gráfico de EECUs. También se considera el parámetro de *timeout* establecido en la configuración del contenedor de la aplicación.

Figura 27. EECUs obtenidas del sitio de administración del proyecto Google Cloud creado.



### Parámetros del cálculo

- EECUs utilizadas: 120 EECUs
- Tiempo máximo de cálculo: 200 segundos (configuración para suspender el cálculo después de este tiempo)
- Costo por EECU-hora: \$1.33

Cálculo del tiempo de procesamiento por solicitud:

$$200 \text{ segundos} = 200/3600 \text{ horas} \approx 0.0556 \text{ horas}$$

Cálculo del costo por solicitud:

$$120 \text{ EECUs} * 0.0556 \text{ horas} * \$1.33/\text{EECU-hora} = \$8.88 \text{ por solicitud}$$

**Presupuesto Estimado Mensual.** Suponiendo que el Geoportal procese datos para 100 solicitudes por mes:

Costo mensual:

$$100 \text{ solicitudes/mes} * \$8.88/\text{solicitud} = \$888/\text{mes}$$

### 5.3. Presupuesto de desarrollo del proyecto.

Asumiendo los valores desglosados anteriormente, a excepción de los precios de la plataforma de GEE, se presenta un presupuesto general del desarrollo del geoportal AgriService en la Tabla 12.

**Tabla 12.** Presupuesto de desarrollo del geoportal AgriService

PRESUPUESTO DE DESARROLLO DEL PROYECTO (5 meses)				
UNIDAD	CONCEPTO	VALOR	CANTIDAD	TOTAL
<b>Costos directos</b>				
<b>Servicios profesionales (desarrollador)</b>				
horas	Ingeniero en Geomática	€ 15.00	751	€ 11,265.00
<b>Equipos</b>				
horas	Alquiler de laptop gama media	€ 10.00	751	€ 7,510.00
global	Materiales			€ 200.00
<b>Tecnología de desarrollo</b>				
mes	Web hosting	€ 6.95	2	€ 13.90
<b>Costos Indirectos</b>				
mes	Arriendo de oficina	€ 1,000.00	5	€ 5,000.00
mes	Servicios	€ 150.00	5	€ 750.00
global	Imprevistos			€ 500.00
<b>Subtotal</b>				€ 25,238.90
<b>Costos adicionales</b>				
Beneficio Industrial			10%	€ 2,523.89
Impuesto sobre el valor añadido (IVA)			21%	€ 5,300.17
<b>TOTAL DESARROLLO</b>				<b>€ 33,062.96</b>

## 5.1. Presupuesto mensual para continuidad del proyecto.

En caso de que se requiera pasar a una versión comercial del geoportal AgriService será necesario adquirir la versión comercial del GEE. Por lo que se considera los precios de la plataforma para el presupuesto mensual (Tabla 13) y dar continuidad al proyecto.

**Tabla 13.** Presupuesto mensual del geoportal AgriService

PRESUPUESTO MENSUAL PARA CONTINUIDAD DEL PROYECTO				
UNIDAD	CONCEPTO	VALOR	CANTIDAD	TOTAL
<b>Costos directos</b>				
<b>Servicios profesionales (desarrollador)</b>				
horas	Ingeniero en Geomática	€ 15.00	160	€ 2,400.00
<b>Equipos y materiales</b>				
horas	Alquiler de laptop gama media	€ 10.00	160	€ 1,600.00
global	Materiales			€ 80.00
<b>Tecnología de desarrollo</b>				
mes	Google Earth Engine	€ 799.20	1	€ 799.20
mes	Web hosting	€ 6.95	1	€ 6.95
<b>Costos Indirectos</b>				
mes	Arriendo de oficina	€ 1 000.00	1	€ 1,000.00
mes	Servicios	€ 150.00	1	€ 150.00
global	Imprevistos			€ 100.00
<b>Subtotal</b>				€ 6,136.15
<b>Costos adicionales</b>				
Beneficio Industrial			10%	€ 613.62
Impuesto sobre el valor añadido (IVA)			21%	€ 1,288.59
<b>TOTAL MENSUAL</b>				<b>€ 8,038.36</b>

## 6. Conclusiones y discusión

---

En el presente proyecto se ha desarrollado un geoportal orientado a agricultores, utilizando tecnologías de desarrollo web. Se proporciona una herramienta que facilita la visualización y análisis de datos satelitales en tiempo real, así como el acceso a mapas y series de tiempo de cuatro variables biofísicas clave: índice de vegetación, fracción de cobertura vegetal, temperatura en superficie y evapotranspiración. Estas funcionalidades se han implementado con Google Earth Engine (GEE) para realizar el computo en la nube siguiendo la cadena de procesamiento basada en algoritmos de teledetección desarrollados por EOLAB.

### 6.1. Cumplimiento de Objetivos Iniciales.

Los objetivos propuestos del proyecto (Figura 1) se han cumplido obteniendo un geoportal con estructura actualizable y procesamientos funcionales. En esta sección se concluye sobre cada uno de los objetivos planteados.

**Ofrecer productos derivados de la integración de imágenes de satélite con algoritmos de teledetección para monitorizar el estado de la vegetación:** Se ha desarrollado una clase con 28 métodos, además 4 funciones en Python utilizando las herramientas de programación de aplicaciones (API) de Python para Google Earth Engine para integrar la automatización de procesamiento de imágenes de satélite con algoritmos de teledetección y que actualmente están disponibles como productos en una plataforma accesible y funcional para los agricultores.

**Procesamiento de algoritmos en la nube para ahorrar recursos a la empresa, utilizando Google Earth Engine:** Se ha creado un proyecto en la nube y una cuenta de servicio que permite utilizar las herramientas de programación de aplicaciones (API) de Python para Google Earth Engine, con las que, se ha desarrollado procesamientos de cálculo de grandes cantidades de datos de satélite aplicando técnicas de teledetección. Se ha integrado este desarrollo en el *back-end* del geoportal para desplegarlo en internet utilizando herramientas de desarrollo web.



Desarrollar una herramienta de gestión geoespacial para la monitorización de cultivos con datos de satélite como canal de comunicación entre el cliente agricultor y la empresa a nivel de servicio y a nivel comercial. Se ha desarrollado una herramienta que a nivel de servicio provee a cada usuario la capacidad de monitorizar sus cultivos para una adecuada toma de decisiones usando los productos de EOLAB. Se configura el sitio de administración de Django y el cliente pgAdmin de PostgreSQL en internet para la suscripción y gestión de usuarios de AgriService a través de la página de registro del geoportal. A nivel comercial, el geoportal AgriService le da a la empresa la posibilidad de brindar la comercialización de sus productos y de un potencial asesoramiento al riego.

## 6.2. Conclusiones Técnicas

**Trabajo con Git y GitHub.** El uso de Git y GitHub ha sido fundamental para el control de versiones y las actualizaciones de código. Estas herramientas han permitido un seguimiento preciso de los cambios en el código, facilitando la integración continua.

**Trabajo con Docker Compose.** Esta herramienta ha sido esencial para la gestión de contenedores y el despliegue del entorno de desarrollo. Ha permitido la creación de un entorno consistente y reproducible, facilitando el desarrollo y la prueba de la aplicación en el entorno del VPS.

**Trabajo con el sistema operativo del VPS.** La utilización del sistema operativo Linux del VPS ha sido crucial para el despliegue del geoportal. La configuración modular del servidor Apache en la distribución CentOS 7 ha permitido gestionar los recursos de manera eficiente y garantizar la disponibilidad y rendimiento de la aplicación. Sin embargo, la configuración modular de Apache, ha sido un desafío, requiriendo un manejo cuidadoso y detallado para asegurar que todos los componentes funcionen correctamente. A pesar de estos desafíos, el sistema operativo y la arquitectura X86\_64 del VPS han proporcionado una base robusta y estable para la etapa de producción y despliegue del geoportal.

**Trabajo con GEE.** Google Earth Engine ha sido una herramienta poderosa para el procesamiento y análisis de datos satelitales en la nube durante todo el

desarrollo del geoportal. Su capacidad para manejar grandes volúmenes de datos y realizar cálculos complejos ha sido vital para el éxito del proyecto. Sin embargo, el trabajar con la API de Python conlleva la limitación de acceso a ciertas herramientas de visualización que dentro del entorno de Google Earth Engine están disponibles. Además, en la etapa de desarrollo resulta aún más limitada cuando se trabaja directamente en un editor de código y no en plataformas como Google Colab, principalmente en la generación y visualización de imágenes procesadas sobre el mapa.

### 6.3. Fortalezas del Geoportal

**Front-end y Back-end actualizable:** Actualmente las funciones desarrolladas del *back-end* para el cálculo de las series de tiempo tiene una organización que permite añadir más variables biofísicas, además el *front-end* también cuenta con una estructura replicable para añadir variables.

**Procesamiento en la nube:** La automatización y el procesamiento en tiempo real en la nube eliminan la necesidad de preprocesamiento manual, ahorrando recursos de hardware y software a la empresa EOLAB.

**Acceso a datos históricos y en tiempo real:** La capacidad de acceder a un historial de imágenes y datos actuales facilita una toma de decisiones más informada.

#### Inconvenientes del Geoportal

**Dependencia de datos externos:** La disponibilidad y fiabilidad de los datos satelitales externos pueden afectar el funcionamiento del geoportal. Por ejemplo, si hay cambios en los conjuntos de datos de GEE o se actualizan las colecciones, los procesamientos del geoportal se pueden ver comprometidos.

**Retraso en la actualización de datos:** Los datos auxiliares para el cálculo de la evapotranspiración tienen un retraso, que para los casos más extremos puede llegar hasta tres meses. Esto limita la capacidad de respuesta inmediata de los agricultores frente a las recomendaciones de riego. Particularmente, estos datos se corresponden con los productos de radiación solar DSSF y DSLF.

## 6.4. Mejoras Futuras

Los inconvenientes que el proyecto presenta deberán abordarse en el futuro como la dependencia de datos externos, algunas propuestas para mejorar la funcionalidad del geoportal y crear un plan de negocio basado en el presupuesto.

**Dependencia de datos externos.** Actualmente, el producto de evapotranspiración (ET) depende de datos de DSSF y DSLF obtenidos de ERA5-Land, que pueden tener un retraso de hasta tres meses. La empresa está actualmente investigando soluciones para abordar esta limitación, como la obtención de estos productos de otros servicios (Organización Europea para la Explotación de Satélites Meteorológicos - EUMETSAT), la implementación de algoritmos propios que calculen estos productos auxiliares o el desarrollo de nuevos métodos basados en inteligencia artificial que no sean dependientes de estos productos.

**Propuestas de Mejora para el Geoportal:** Trabajar en mejoras de funcionalidad como lista de parcelas por usuario, confirmación por email de usuario y generación de informes en PDF experiencia del usuario, así como también la inclusión de otros productos de satélite generados por la empresa.

**Presupuesto y cuestiones comerciales:** Las restricciones presupuestarias inherentes a la comercialización del servicio son una limitación significativa. Es crucial desarrollar un plan de negocio robusto que contemple estas restricciones y permita la sostenibilidad del proyecto.

## 7. Bibliografía

---

- Aleixandre Verger i Ten (2008). [Tesis Doctoral]. *En Anàlisi comparativa d'algorismes operacionals d'estimació de parametres biofísics de la coberta vegetal amb teledetecció*. Universitat de Valencia.
- Amiri M., Pourghasemi H. (2022). *Chapter 8 - Mapping the NDVI and monitoring of its changes using Google Earth Engine and Sentinel-2 image*. Computers in Earth and Environmental Sciences. Hamid Reza Pourghasemi. Elsevier. 127-136. <https://doi.org/10.1016/B978-0-323-89861-4.00044-0>.
- Bonafoni S., Sekertekin, A. (2020) *Albedo Retrieval From Sentinel-2 by New Narrow-to-Broadband Conversion Coefficients*. IEEE Geoscience and Remote Sensing Letters. 17(9). 1618- 1622 <https://doi.org/10.1109/LGRS.2020.2967085>.
- Django Software Foundation. (2024). Django. Django Project. <https://www.djangoproject.com/>
- Docker Inc. (2024). Docker Compose overview. Docker Documentation. <https://docs.docker.com/compose/>
- Du, C., Ren, H., Qin, Q., Meng, J., Zhao, S. (2015). *A Practical Split-Window Algorithm for Estimating Land Surface Temperature from Landsat 8 Data*. Remote Sens. 7, 647-665.
- Fensholt, R., & Proud, S.R. (2012). *Evaluation of the consistency of global land-cover products and their suitability for climate modelling*. Remote Sensing of Environment, 124, 640-653. <https://doi.org/10.1016/j.rse.2011.11.004>
- García-Santos, V., Cuxart, J., Martínez-Villagrasa, D., Jiménez, M., Simó, G. (2018). *Comparison of Three Methods for Estimating Land Surface Temperature from Landsat 8-TIRS Sensor Data*. Remote Sensing, 10(9), 1450. <https://doi.org/10.3390/rs10091450>
- Gutman, G., Ignatov, A. (1998). *The Derivation of the Green Vegetation Fraction from NOAA/AVHRR Data for Use in Numerical Weather Prediction Models*. International Journal of Remote Sensing. 19. <https://doi.org/10.1080/014311698215333>.
- Jackson, R.D., Hatfield, J., Reginato, R., Idso, S., Pinter, P. (1983) *Estimation of daily evapotranspiration from one time of day measurements*. Agri. Water Manage. 3(7), 351-362. [https://doi.org/10.1016/0378-3774\(83\)90095-1](https://doi.org/10.1016/0378-3774(83)90095-1).
- Li, X., Zhang, Y., & Tang, K. (2015). *Estimation of Land Surface Temperature from MODIS Data Using Vegetation Indices and Empirical Models*. Remote Sensing of Environment, 165, 137-149. <https://doi.org/10.1016/j.rse.2015.06.013>

- Menenti, M.; Choudhury, B. *Parameterization of land surface evaporation by means of location dependent potential evaporation and surface temperature range*. Proceedings of IAHS conference on Land Surface Processes. IAHS Publ. 1993, 212, 561-568.
- Muñoz Sabater, J. (2019) *ERA5-Land monthly averaged data from 1981 to present*. Copernicus Climate Change Service (C3S) Climate Data Store (CDS). <https://doi.org/10.24381/cds.68d2bb30>
- Myneni, R.B., Ramakrishna, R., Nemani, R., Running, S.W. (1997). *Albedo estimation techniques for land surfaces: A review of methods*. Remote Sensing of Environment, 59(3), 523-536. [https://doi.org/10.1016/S0034-4257\(97\)00042-4](https://doi.org/10.1016/S0034-4257(97)00042-4)
- Norman, J., Kustas, W., Humes, K. (1995) *Source approach for estimating soil and vegetation energy fluxes in observations of directional radiometric surface temperature*. Agricultural and Forest Meteorology, 77(3-4). 263-293. [https://doi.org/10.1016/0168-1923\(95\)02265-Y](https://doi.org/10.1016/0168-1923(95)02265-Y)
- Perilla, G., Mas, J. (2020). *Google EarthEngine (GEE): una poderosa herramienta que vincula el potencial de los datos masivos y la eficacia del procesamiento en la nube*. Investigaciones geográficas. 101(e59929). <https://doi.org/10.14350/rig.59929>
- Phiri, D., Simwanda, M., Salekin, S., Nyirenda, V., Murayama Y., Ranagalage, M. (2020). *Sentinel-2 Data for Land Cover/Use Mapping: A Review*. Remote Sensing. 12,14- 2291. <https://doi.org/10.3390/rs12142291>
- Roerink, G., Su, Z., & Menenti, M. (2000). *S-SEBI: A simple remote sensing algorithm to estimate the surface energy balance*. Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere, 25(2), 147- [https://doi.org/10.1016/S1464-1909\(99\)00128-8](https://doi.org/10.1016/S1464-1909(99)00128-8).
- Sobrino, J.A., Jiménez-Muñoz, J.C., & Paolini, L. (2004). *Estimation of Land Surface Emissivity and Temperature from Radiometric Temperature Measurements*. Remote Sensing of Environment, 90(4), 507-522. <https://doi.org/10.1016/j.rse.2004.09.003>
- The Apache Software Foundation. (2024). Apache HTTP server project. Apache.org. <https://httpd.apache.org/>

Trigo, I., Dacamara, C., Viterbo, P., Roujean, J., Olesen, F., Barroso, C., Camacho-de Coca, F., Carrer, D., Freitas, C., García-Haro, J., Geiger, B., Gellens-Meulenberghs, F., Ghilain, N., Meliá, J., Pessanha, L., Siljamo N., Arboleda, A. (2011). *The Satellite Application Facility for Land Surface Analysis*. *International Journal of Remote Sensing*, 32(10), 2725-2744. <https://doi.org/10.1080/01431161003743199>

## 8. Anexos

---

- 8.1. Manual del Usuario
- 8.2. Documentación de los métodos para el Cloud Computing (CC) en Google Earth Engine (GEE)
- 8.3. Relación del trabajo con los Objetivos de Desarrollo Sostenible de la Agenda 2030.

## ANEXO 9.1 Manual del Usuario



*Geoportal*  
**AgriService**

# MANUAL DEL USUARIO

Servicio web de agricultura basado en imágenes de satélite

Version: 1.0

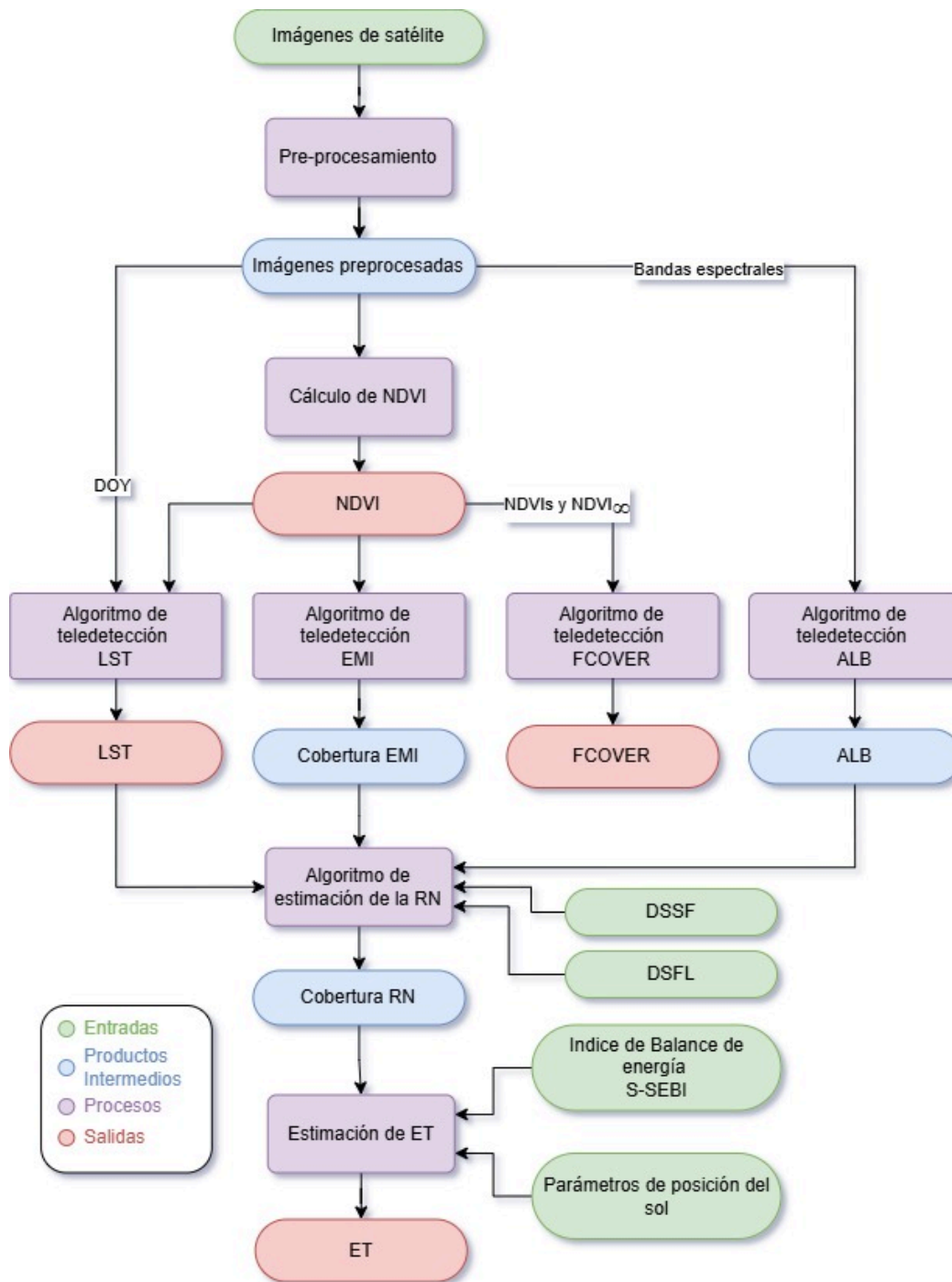


# INTRODUCCIÓN

Bienvenido al Geoportal AgriService, un servicio web diseñado para la monitorización de cultivos y el asesoramiento de riego utilizando datos satelitales. Este manual de usuario le guiará a través de las funciones clave del sistema, incluyendo el acceso seguro, la consulta de información histórica sobre temperatura, índice de vegetación y evapotranspiración, y la visualización de gráficos secuenciales.



## FLUJOGRAMA DE LOS PRODUCTOS DISPONIBLES EN AGRISERVICE



# ÍNDICE

1. Requisitos del sistema
2. Acceso al Sistema
3. Interfaz del usuario
4. Generación de mapas
5. Generación de serie de tiempo
6. Consejos y Buenas Prácticas
7. Soporte y Contacto



# 1. REQUISITOS DEL SISTEMA

Para utilizar el Sistema se requiere:

- Un navegador web actualizado (Google Chrome, Mozilla Firefox, etc.)
- Conexión a internet
- Credenciales de acceso (usuario y contraseña)

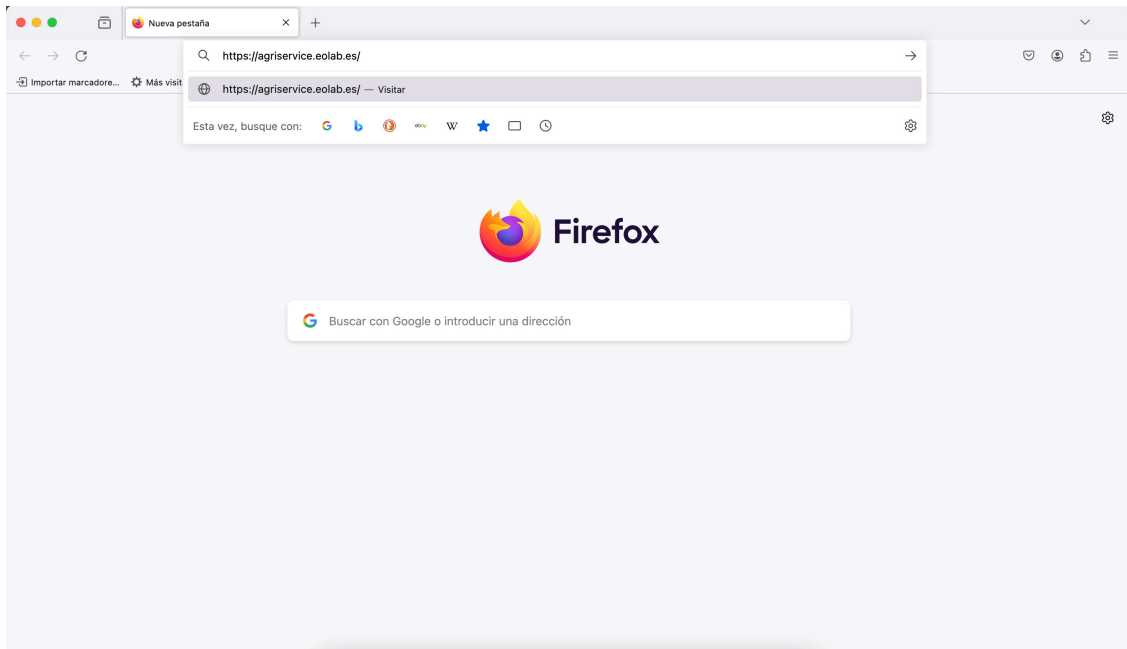


## 2. ACCESO AL SISTEMA

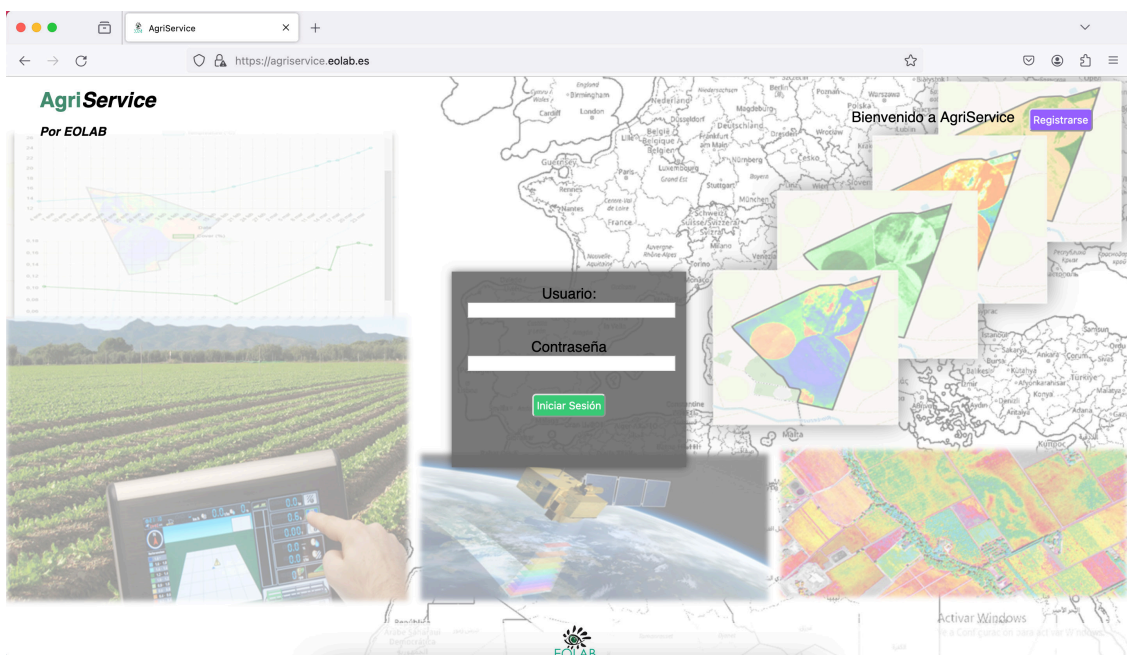


1. Navegue a la página de inicio: Abra su navegador web y vaya a

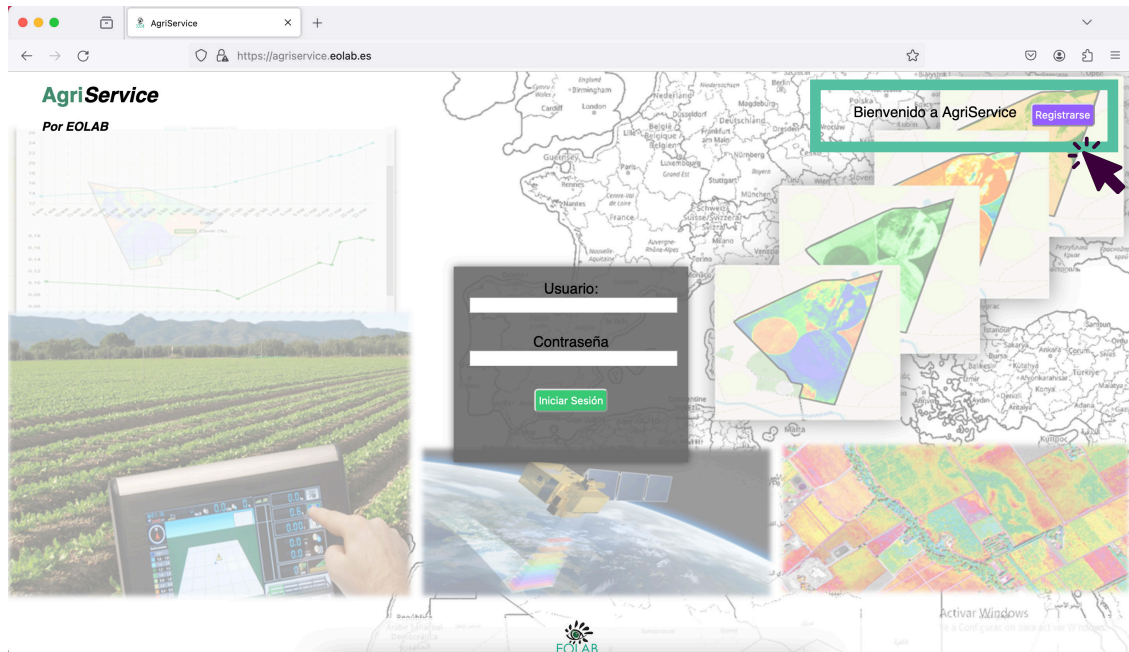
<https://agriservice.eolab.es/>



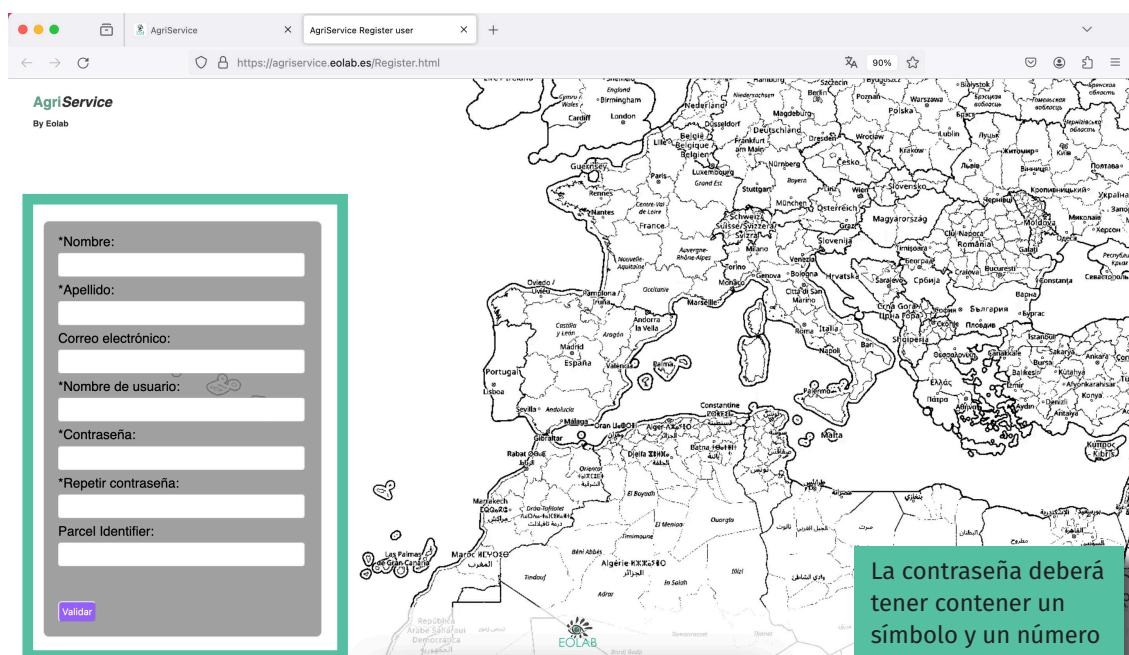
2. Ha ingresado en el Geoportal AgriService.



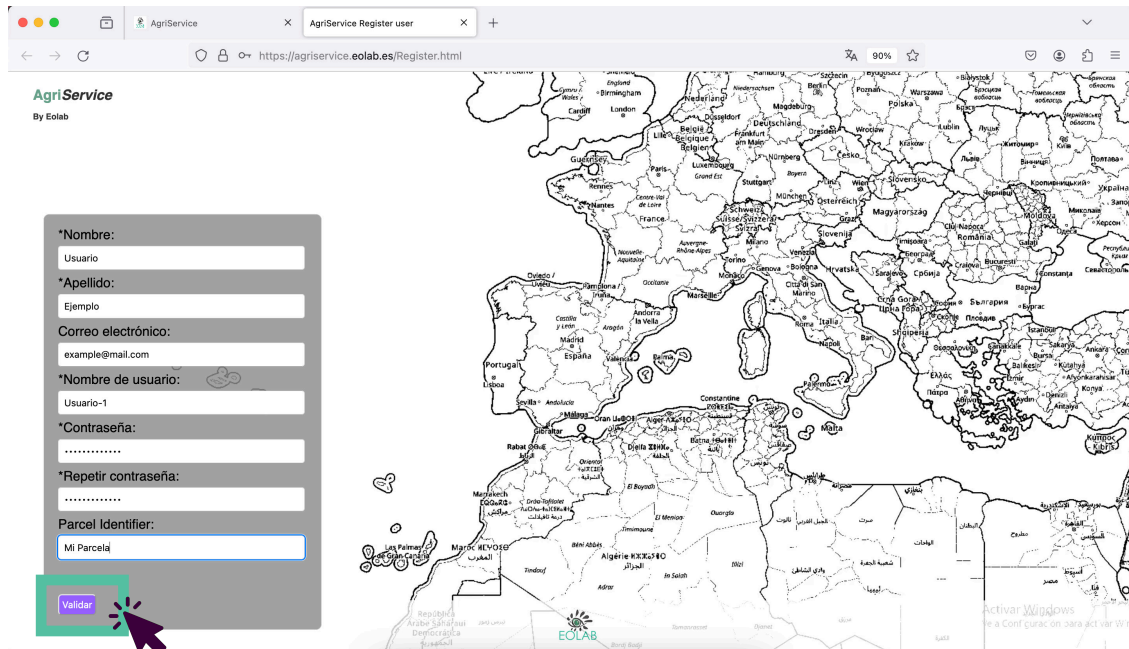
3. **Realice su registro**, su cuenta se registra ingresando en el botón Registro de la parte superior derecha de la pantalla.



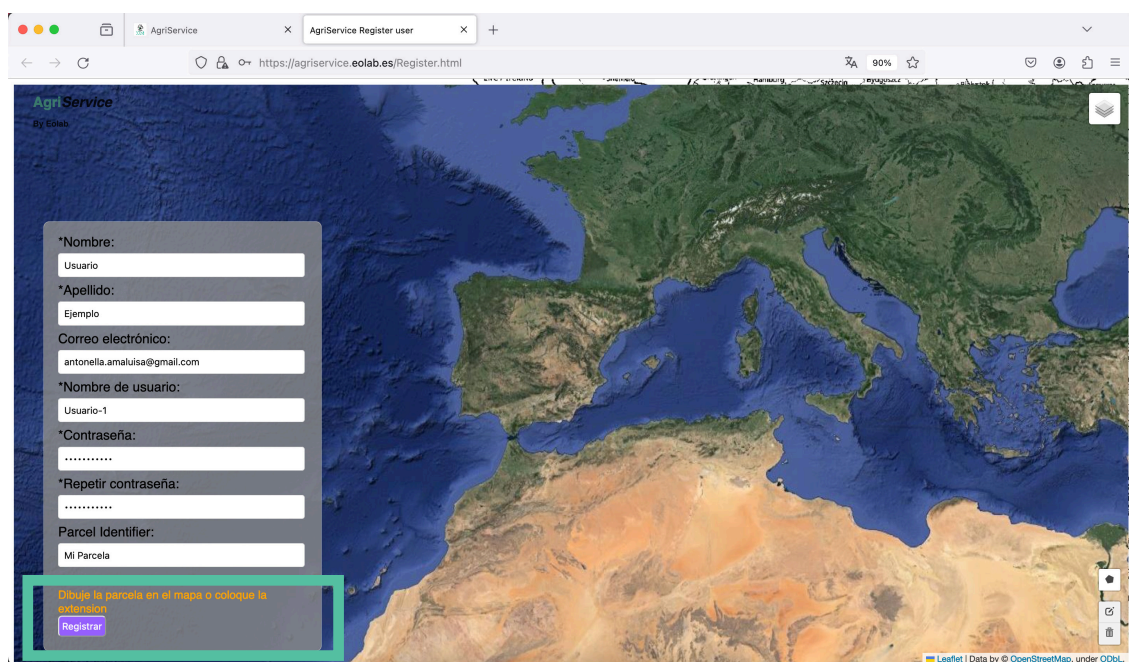
4. Complete todos los campos con la información solicitada.



5. Al finalizar haga pulse el botón **Validar**



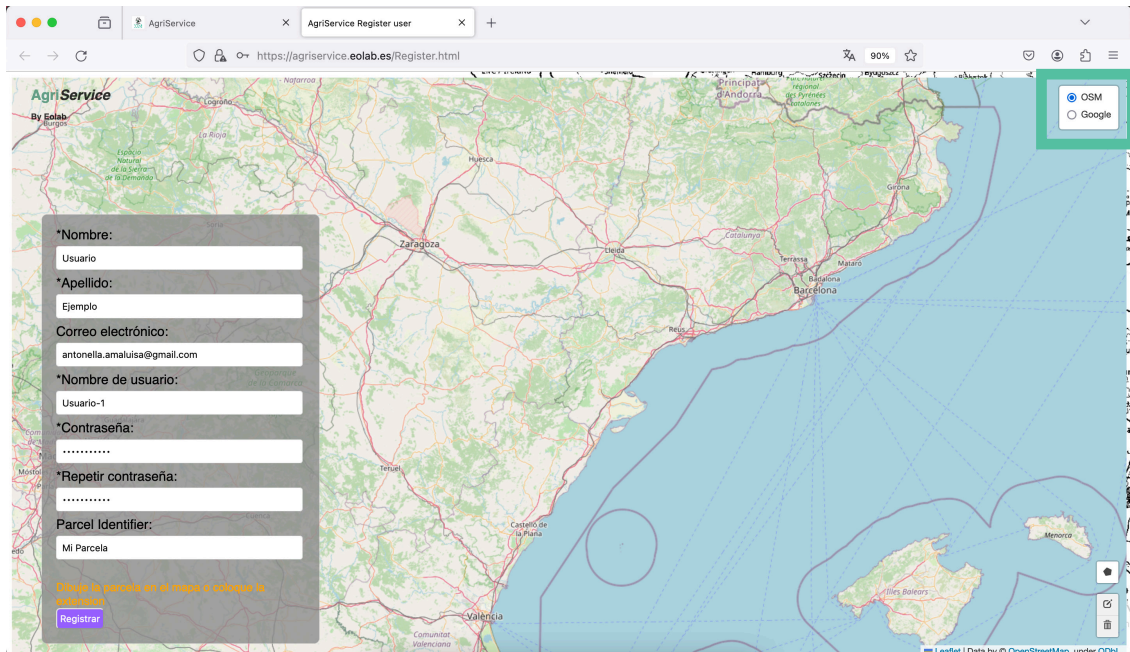
6. A cambiar la pantalla se solicitará que **dibuje la parcela en el mapa**. Utilice el scroll del ratón para acercar o alejar el mapa.




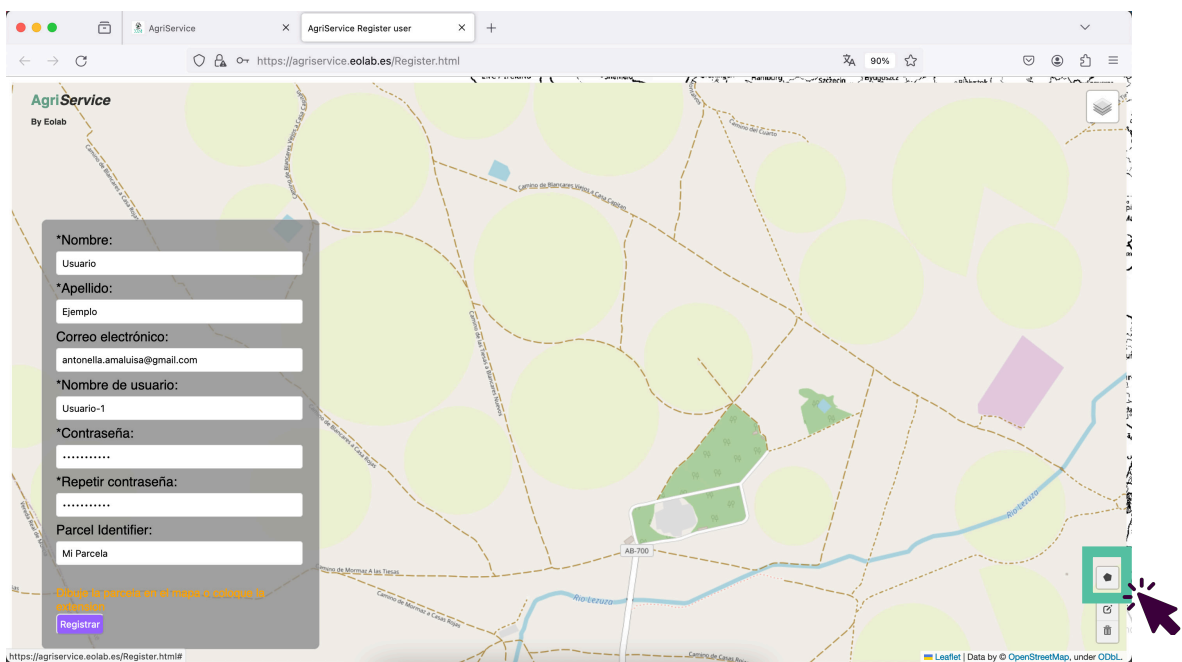


7. En caso de requerirlo, se puede cambiar el mapa base para una mejor identificación.

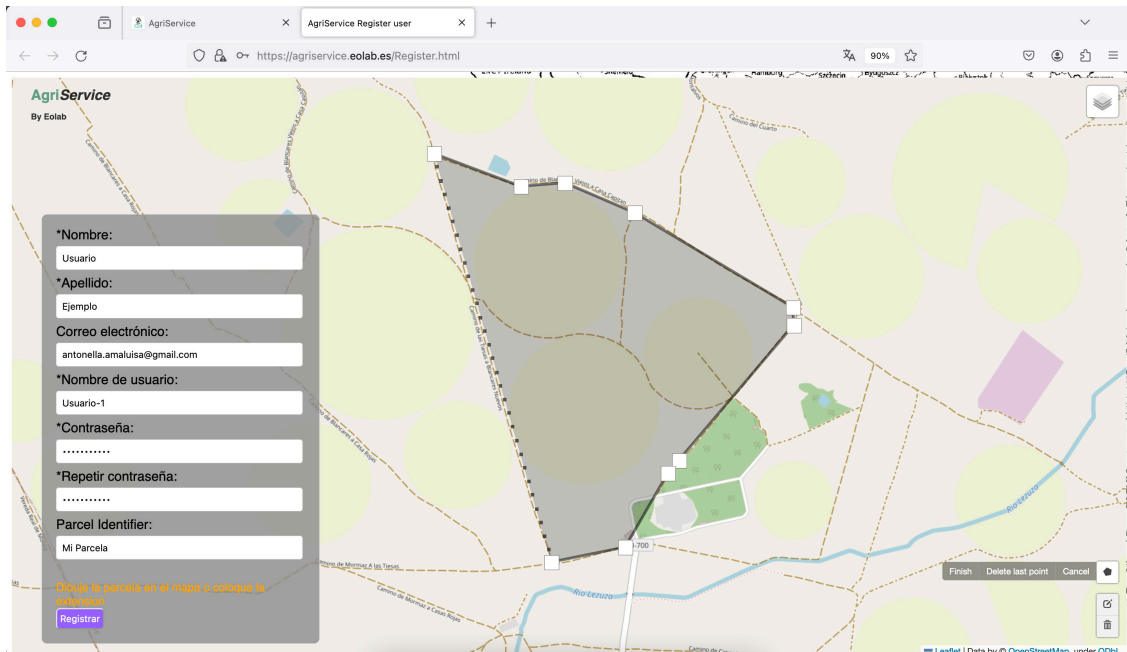
Haciendo clic en el botón de **control de capas**  de la esquina superior derecha.




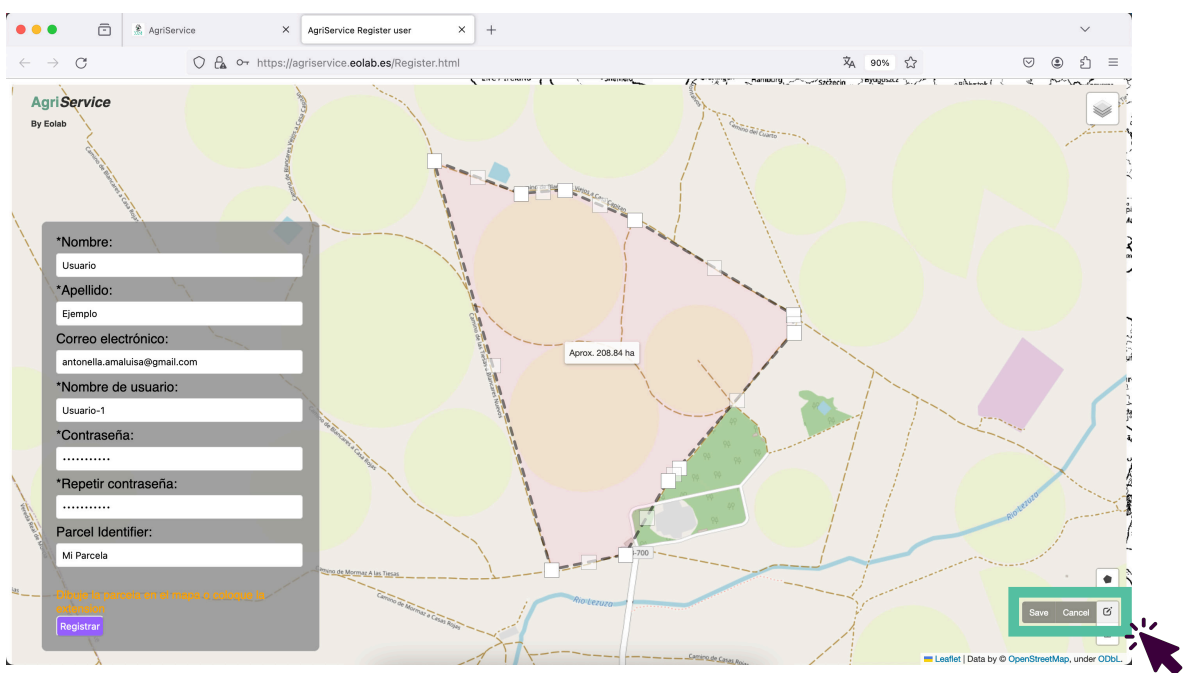
8. Dibujar el polígono con ayuda de la herramienta  ubicada en el lado derecho inferior



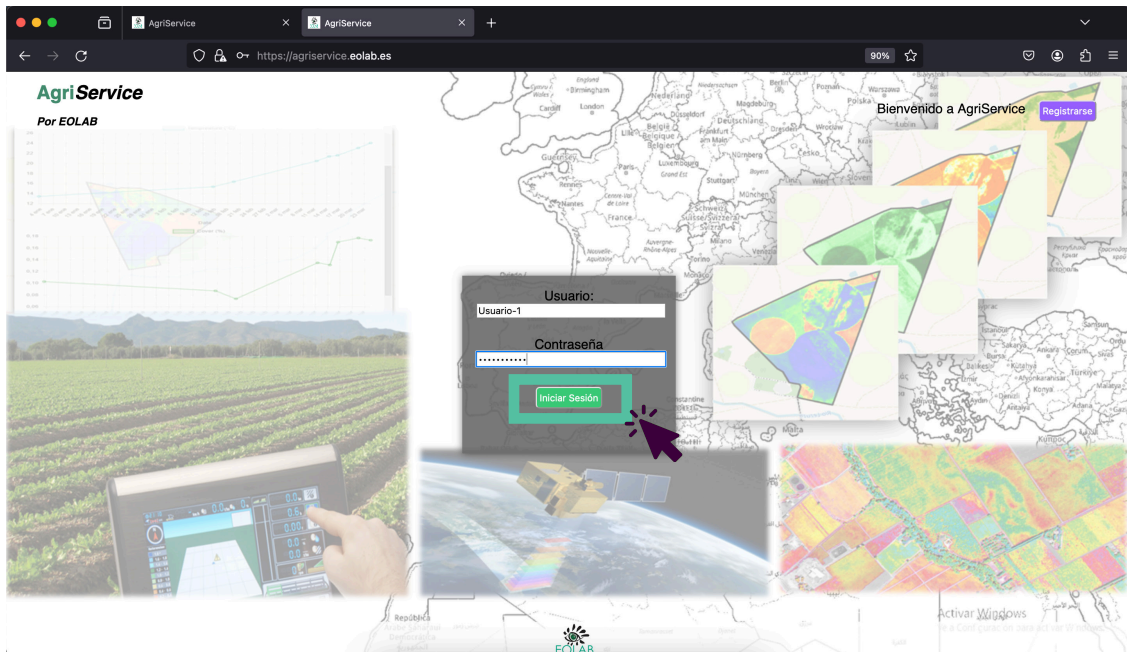
9. Dar clic en cada vertice de la parcela hasta cerrar el polígono



10. Una vez dibujado el polígono, se puede editar sus vertices o eliminar, con la ayuda de la herramienta de edición,  ubicada en el lateral izquierdo inferior



11. Una vez registrado, recibirá un **mail al correo electrónico registrado** con la confirmación de la cuenta con el nombre de la parcela, el nombre del usuario y los datos de pago.



12. Una vez activada la cuenta, ingrese sus credenciales en la pantalla de **inicio de sesión**, introduzca su nombre de usuario y contraseña. Si son correctas, será redirigido a la interfaz principal del Geoportal.



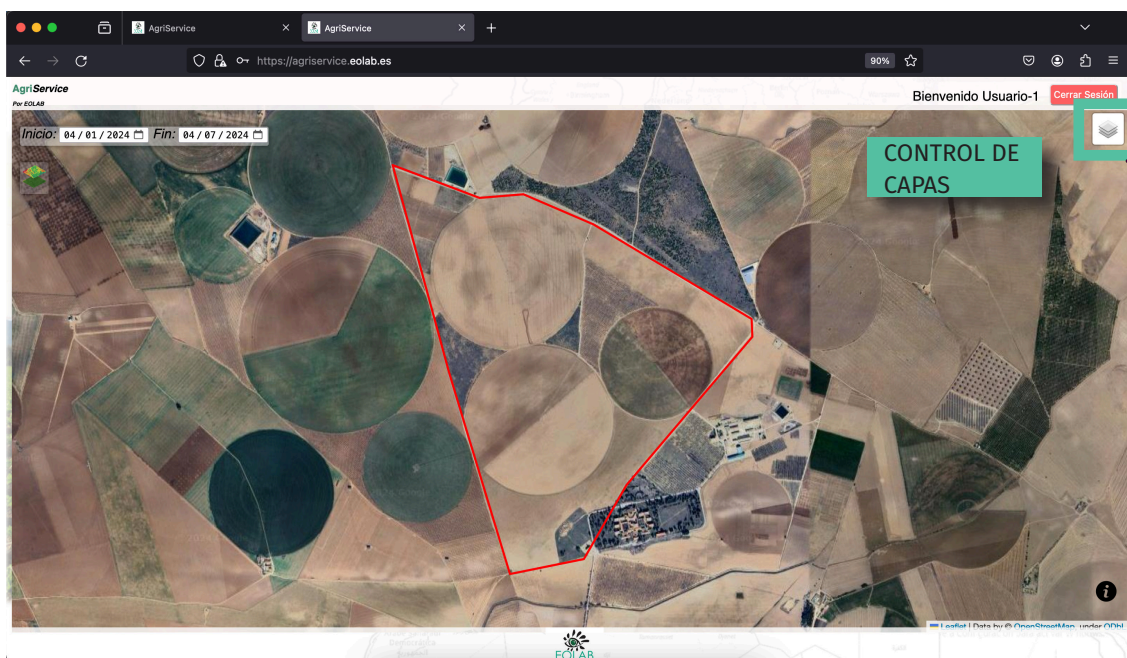
# 3. INTERFAZ DEL USUARIO



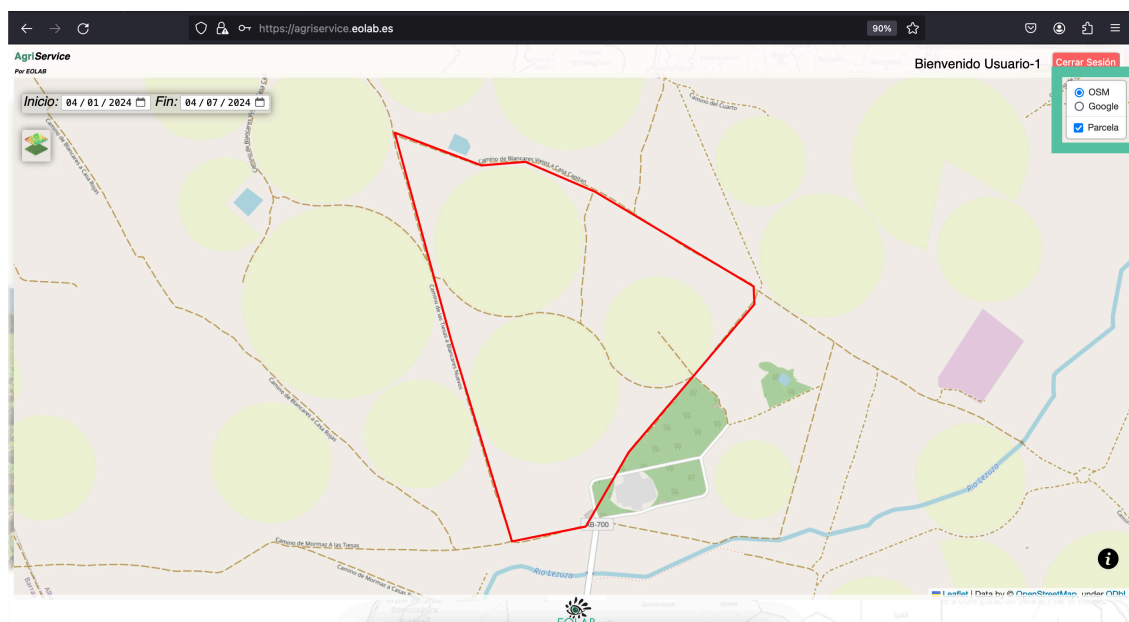
1. Una vez iniciado sesión, la **interfaz principal** se presenta con varios componentes:
  - La **parcela registrada**
  - **Periodo de tiempo** con el que se va a trabajar, en el que se fijaran la fecha de inicio y fecha de fin, del periodo de interés que por defecto es seis meses antes a la fecha actual



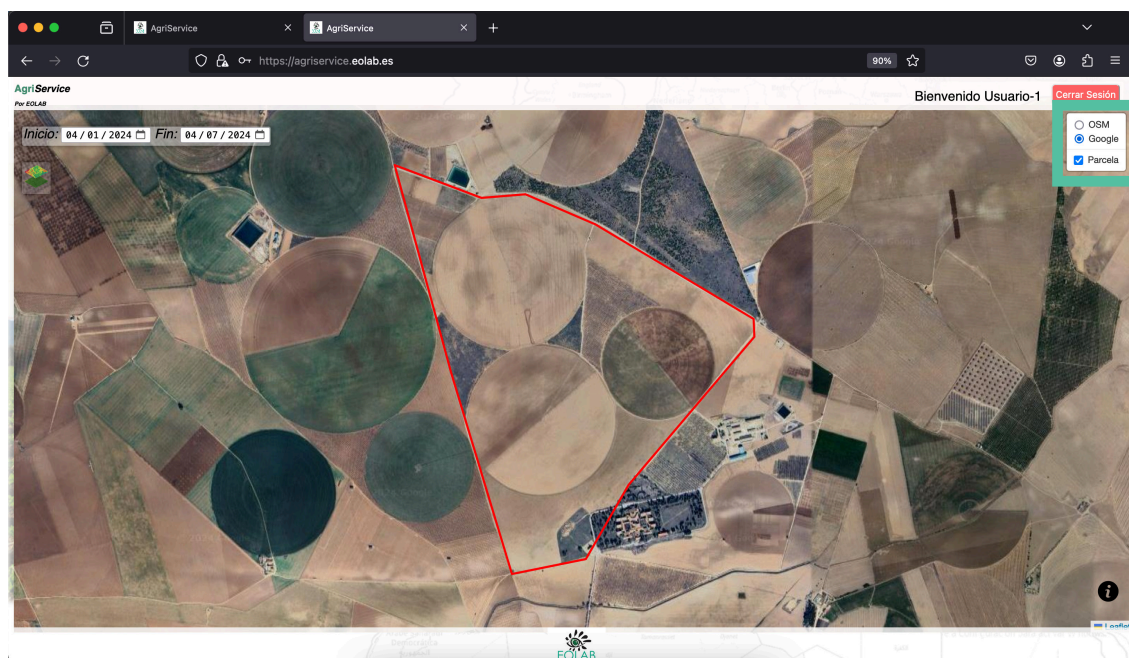
- En el borde derecho superior se encuentra el botón de **control de capas**



#### MAPA BASE OSM




#### MAPA BASE GOOGLE

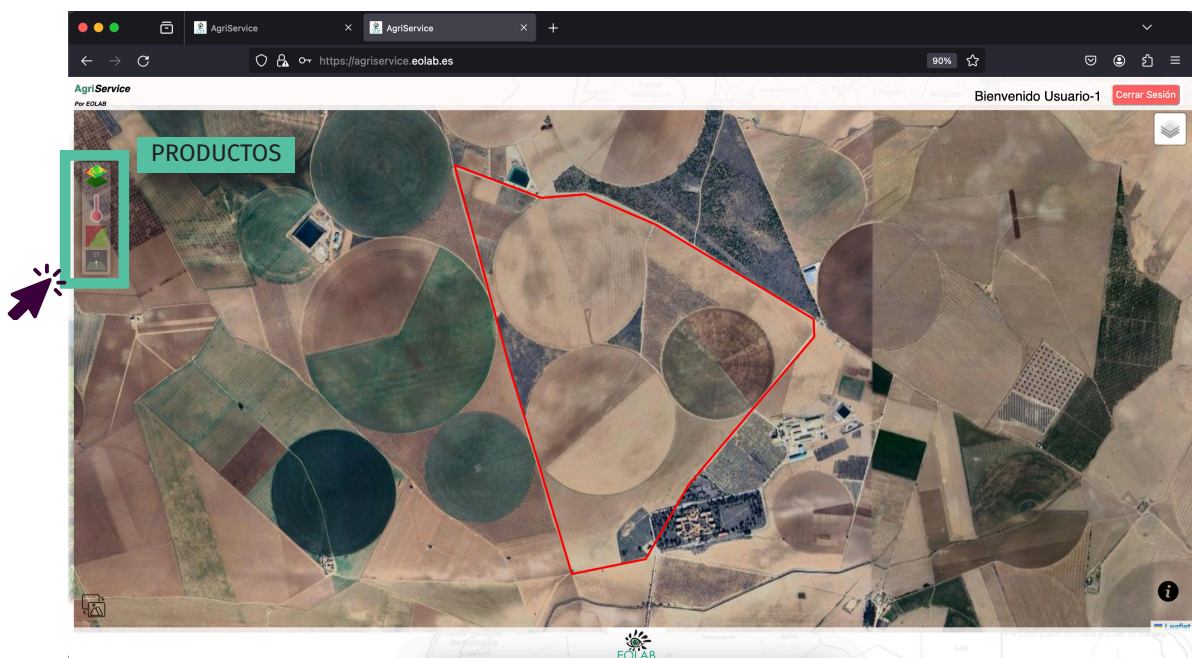


### 3. INTERFAZ DEL USUARIO

- En el borde derecho inferior se encuentra el **ícono de información**  que despliega una ventana con los mensajes del geoportal luego de cada interacción.



- En el borde izquierdo superior se encuentra el menú desplegable  que aglutina los **productos del geoportal**

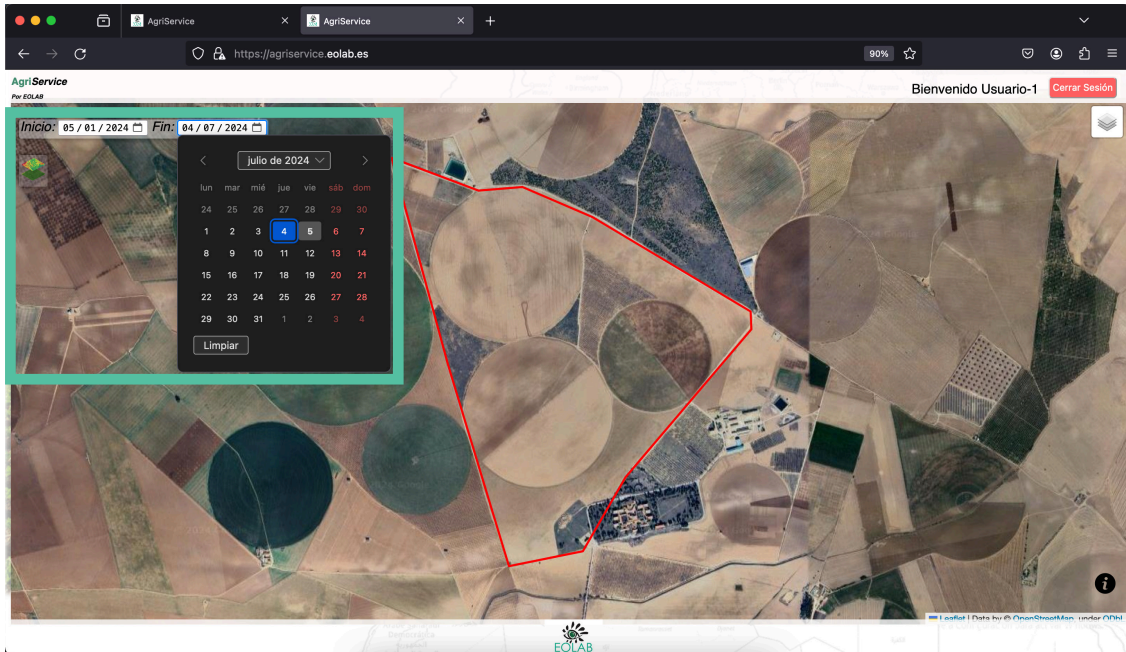


# 4. GENERACIÓN DE MAPAS

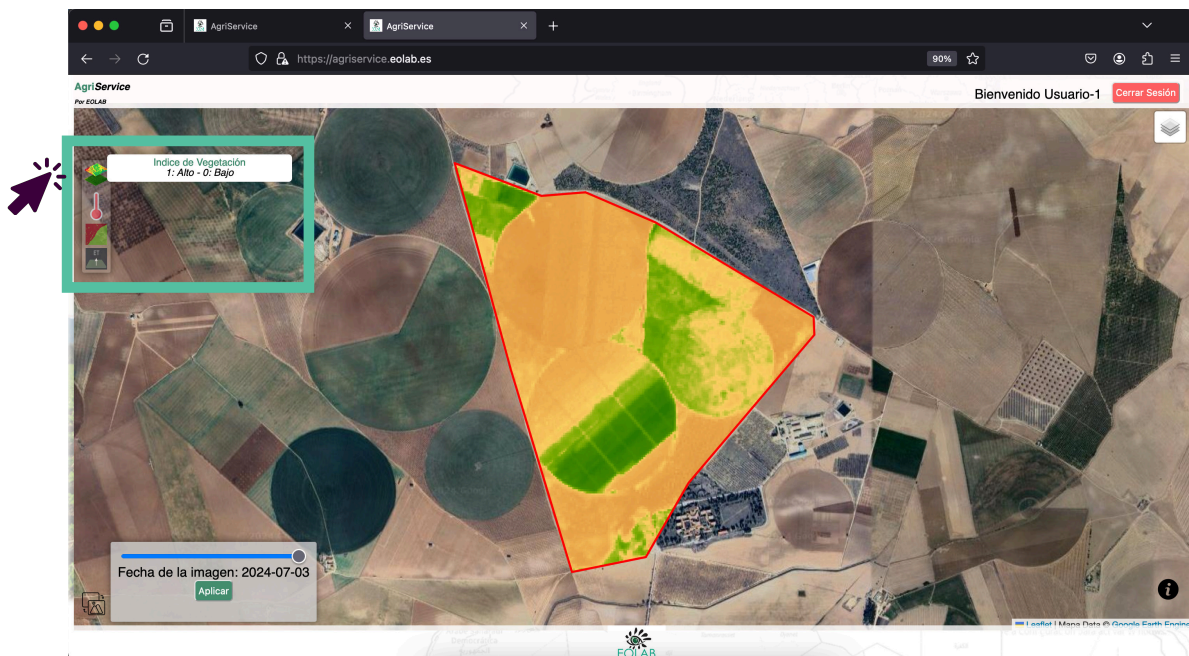




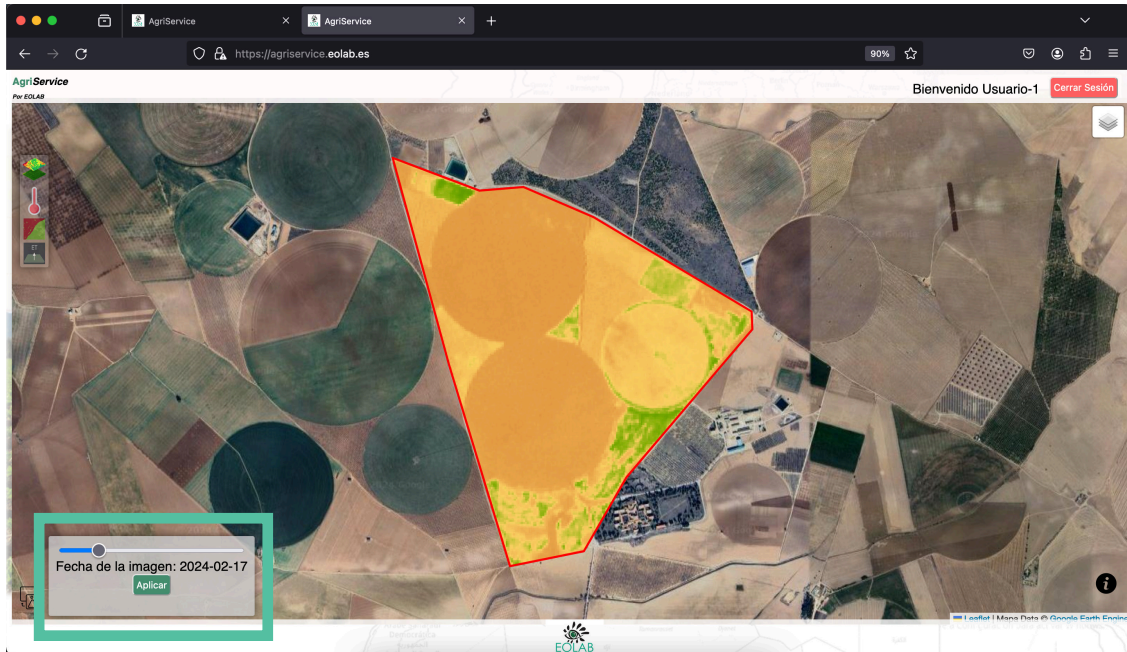
1. Se establece un período de tiempo para el cálculo de los productos



2. **Interacción sobre la Parcela seleccionada.** Active el *índice de vegetación* para obtener la imagen, series de tiempo, advertencias y errores generados en los cálculos.



3. Dentro del rango de fechas establecidos se podrá modificar la fecha exacta para la visualización del *índice de vegetación* utilizando el control de deslizamiento.



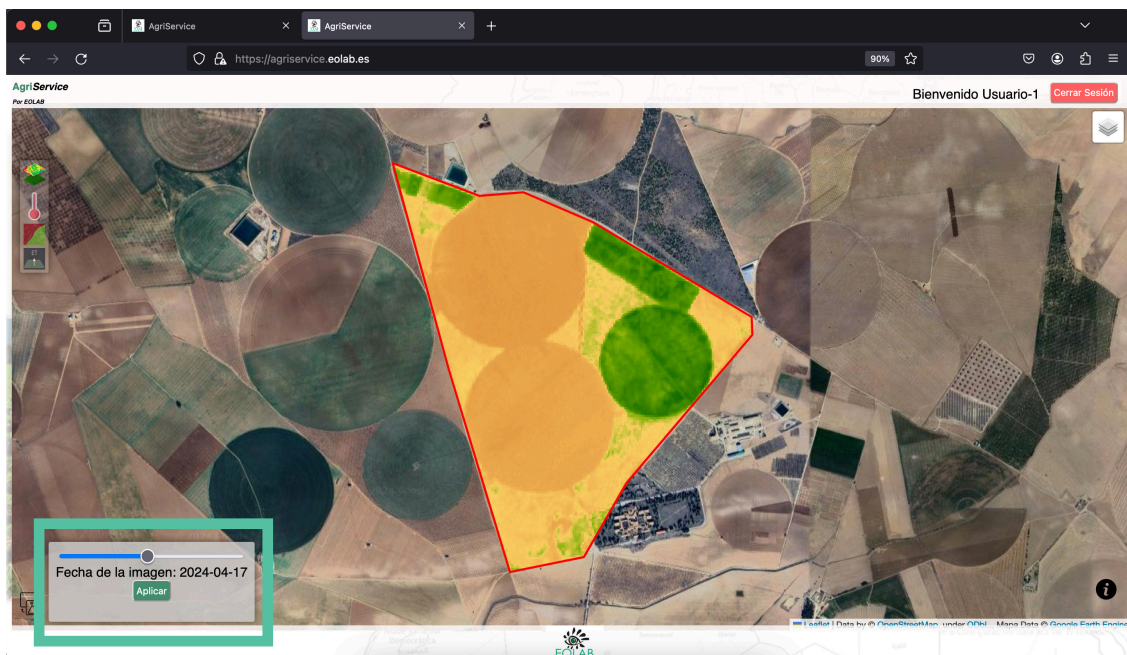
Tome en cuenta que mientras mayor sea el rango de fechas mayor será el número de imágenes de satélite a analizar y el cálculo puede tomar algunos minutos



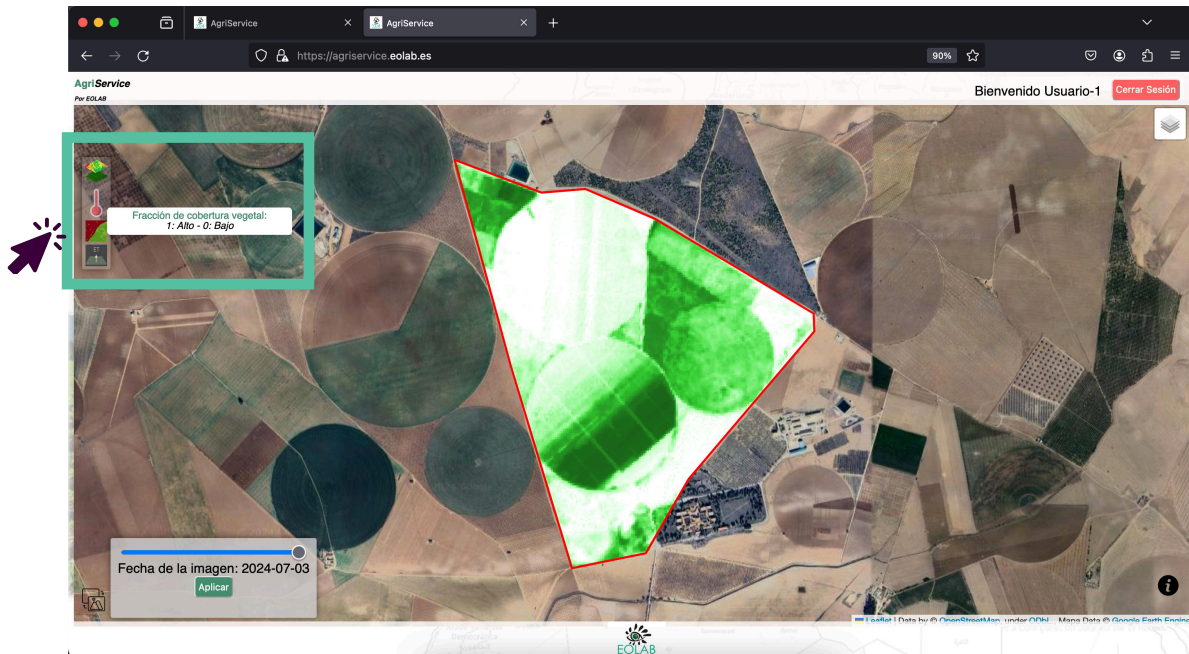
4. Active la *temperatura de superficie (°C)* para obtener la imagen, series de tiempo, advertencias y errores generados en los cálculos.



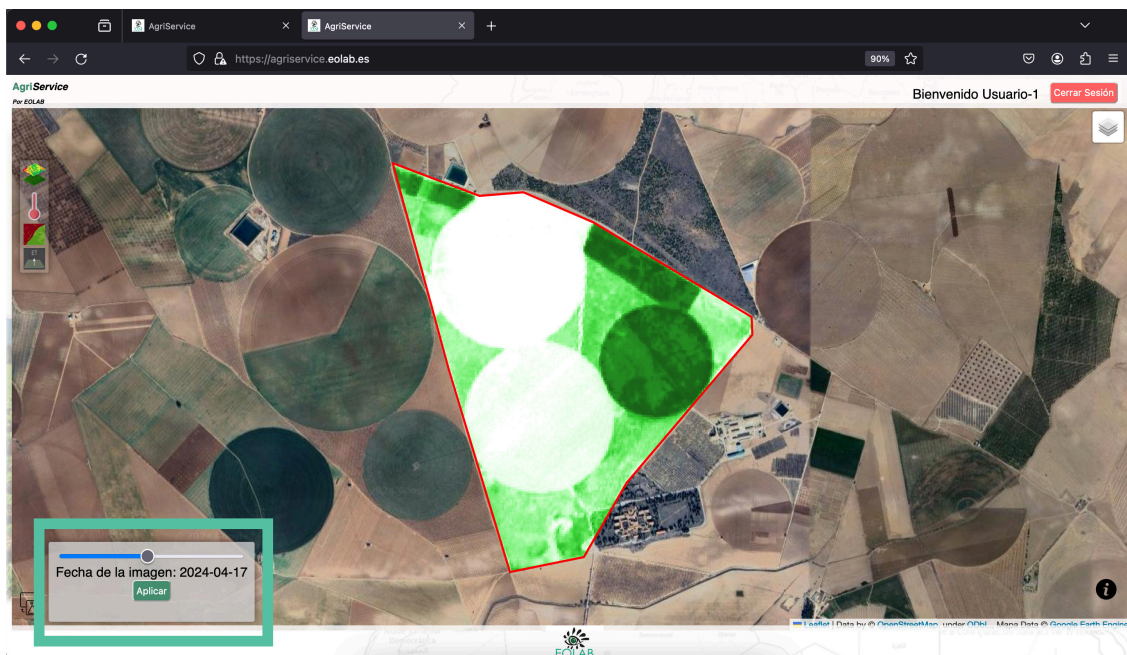
5. Dentro del rango de fechas establecidos se podrá modificar la fecha exacta para la visualización de la *temperatura de superficie (°C)* utilizando el control de deslizamiento.



6. Active la *fracción de cobertura vegetal* para obtener la imagen, series de tiempo, advertencias y errores generados en los cálculos.



7. Dentro del rango de fechas establecidos se podrá modificar la fecha exacta para la visualización de la *fracción de cobertura vegetal* utilizando el control de deslizamiento.



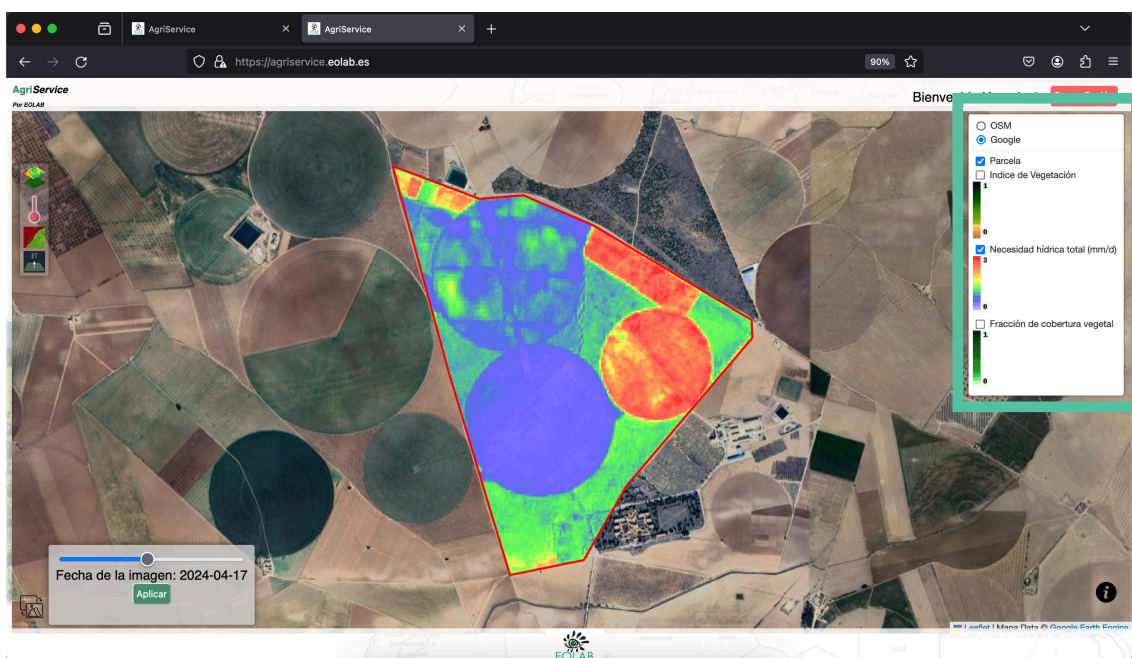
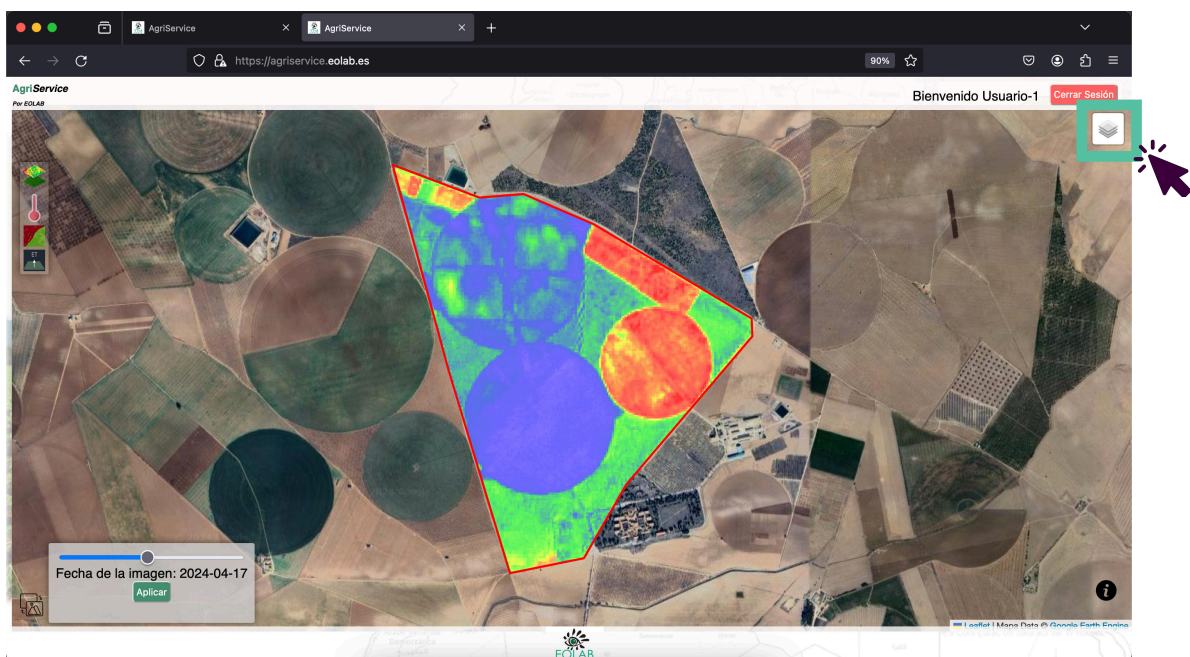
8. Active la *necesidad hídrica (mm/d)* para obtener la imagen, series de tiempo, advertencias y errores generados en los cálculos.



9. Dentro del rango de fechas establecidos se podrá modificar la fecha exacta para la visualización de la *necesidad hídrica (mm/d)* utilizando el control de deslizamiento.



10. Con el botón **control de capas**, puede administrar la visualización de los productos cargados



# 5. GENERACIÓN DE SERIE DE TIEMPO

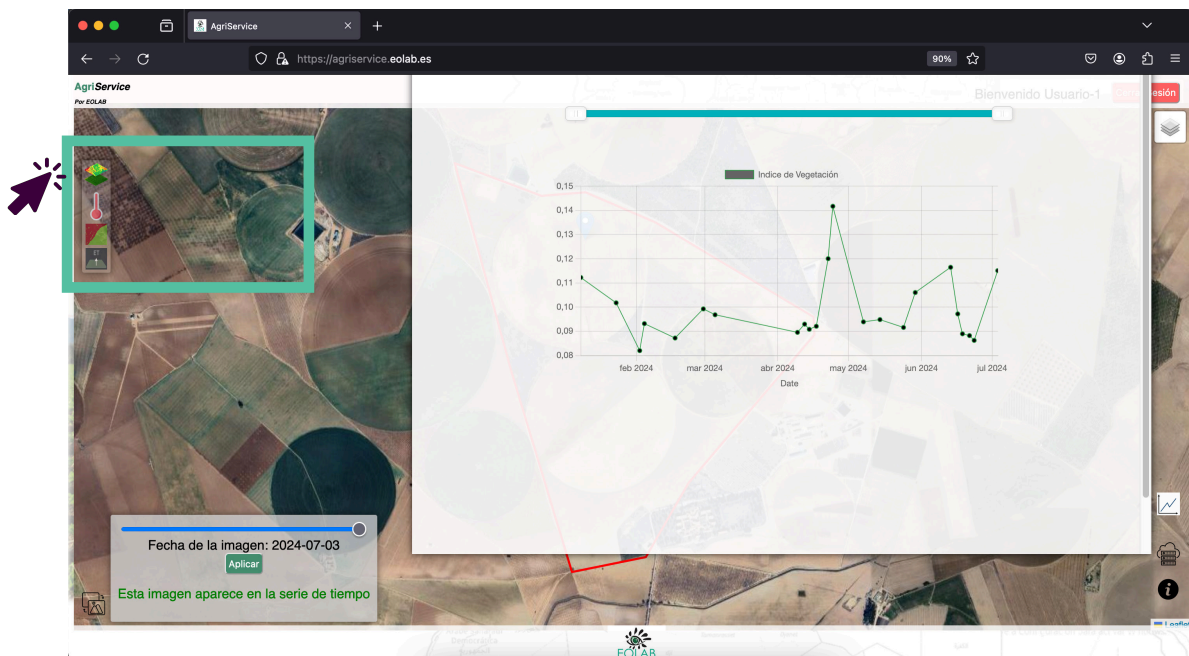


## 5. GENERACIÓN DE SERIE DE TIEMPO

1. Para calcular la **serie de tiempo**, debe dar clic sobre las coordenadas específicas del punto que desea calcular.



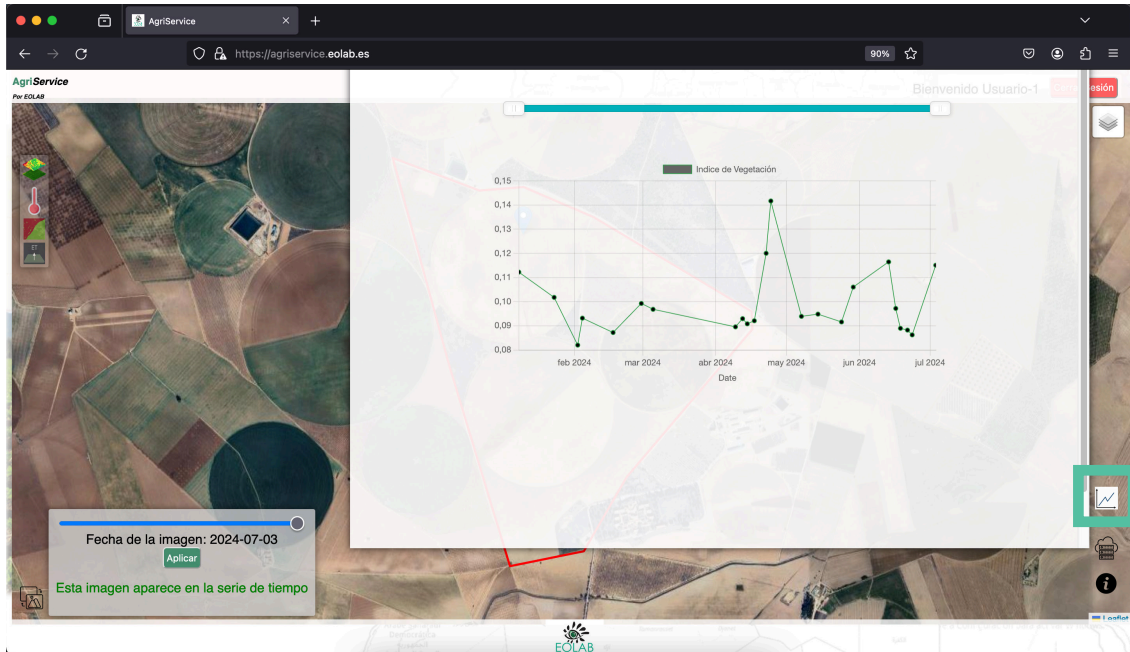
2. Se despliega una ventana flotante que muestra la serie de tiempo del producto seleccionado. En este caso se visualiza el *índice de vegetación*



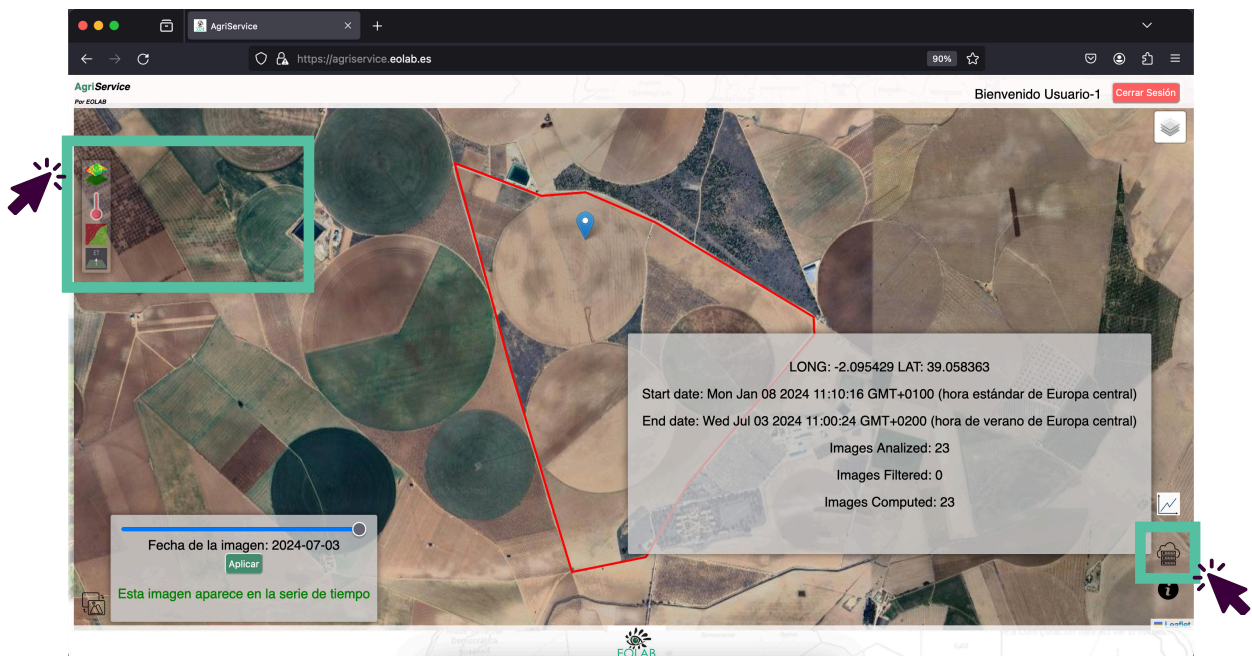


## 5. GENERACIÓN DE SERIE DE TIEMPO

3. Para **minimizar la ventana**, de clic en el botón del lateral izquierdo que aparece tras realizar el cálculo de serie de tiempo

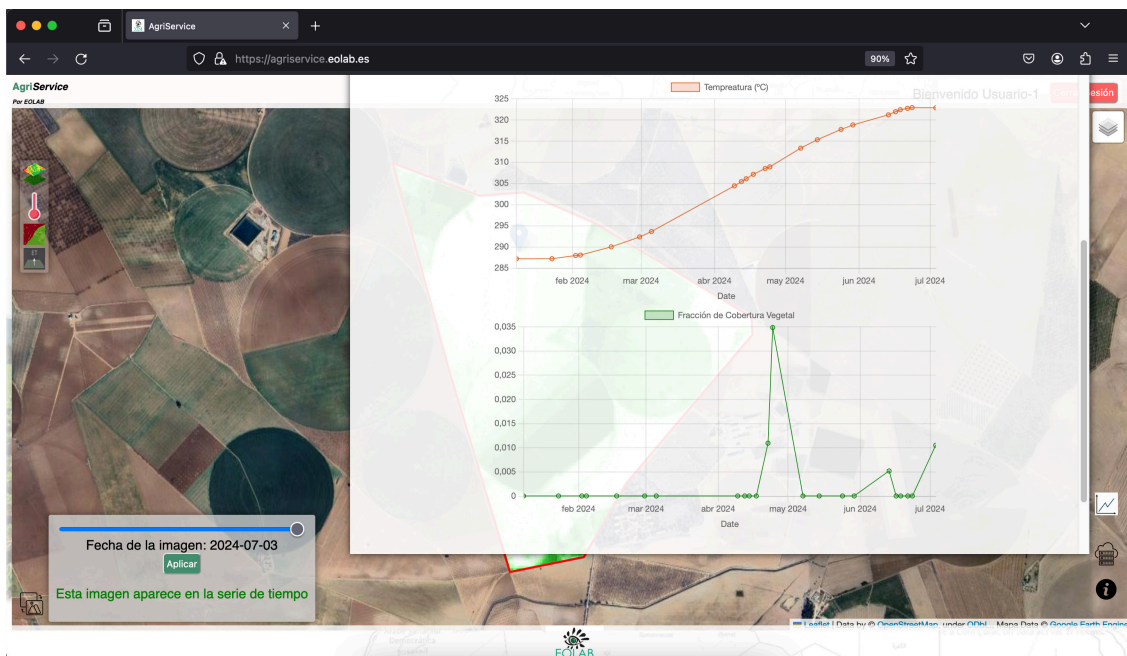
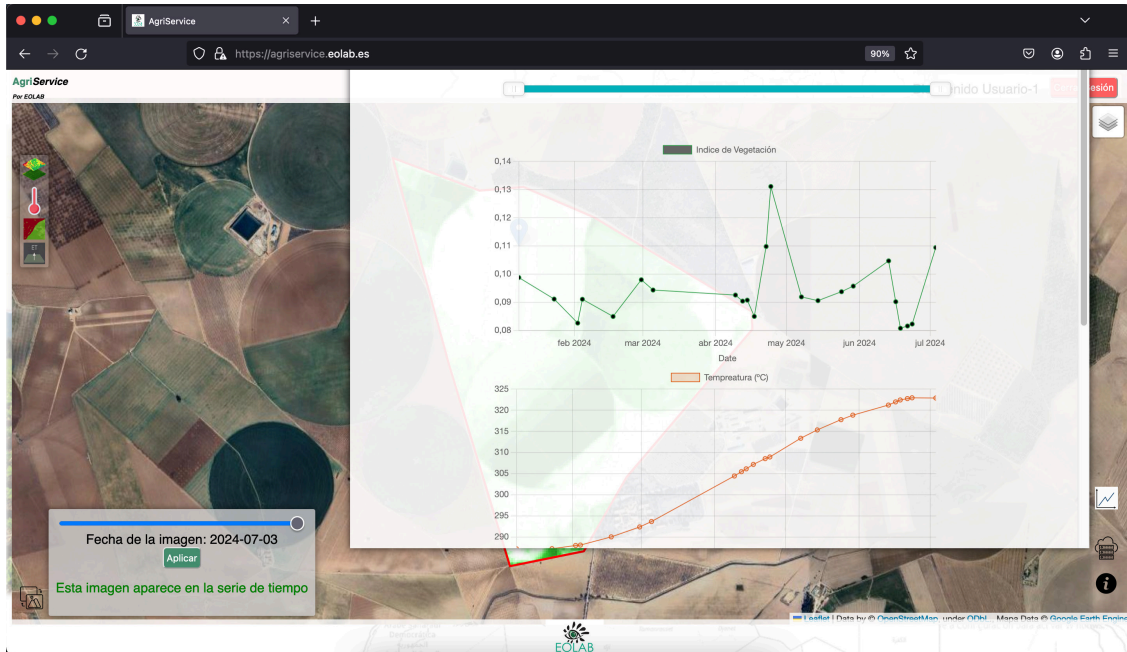


4. El segundo botón que aparece con el producto de serie de tiempo, permite **visualizar los datos** relacionados con el cálculo.



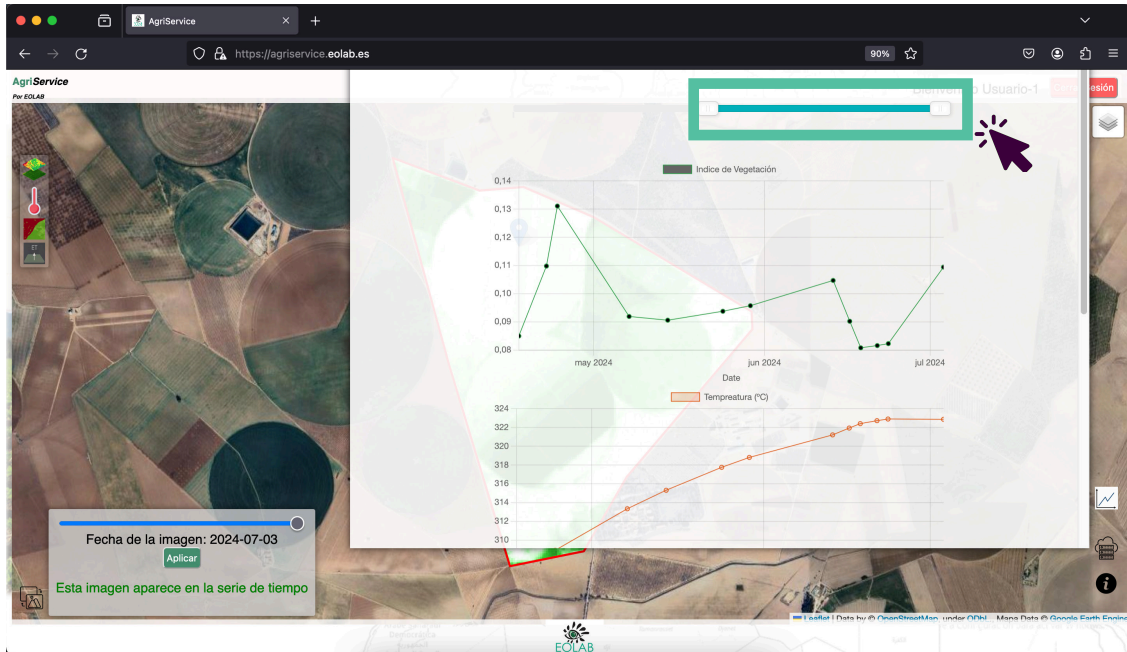
## 5. GENERACIÓN DE SERIE DE TIEMPO

5. Se pueden **calcular todos los productos**, y visualizar las series de tiempo en una misma ventana flotante



## 5. GENERACIÓN DE SERIE DE TIEMPO

6. Dentro del rango de fechas establecidos se podrá modificar las fechas e visualización de la serie de tiempo utilizando el control de deslizamiento.

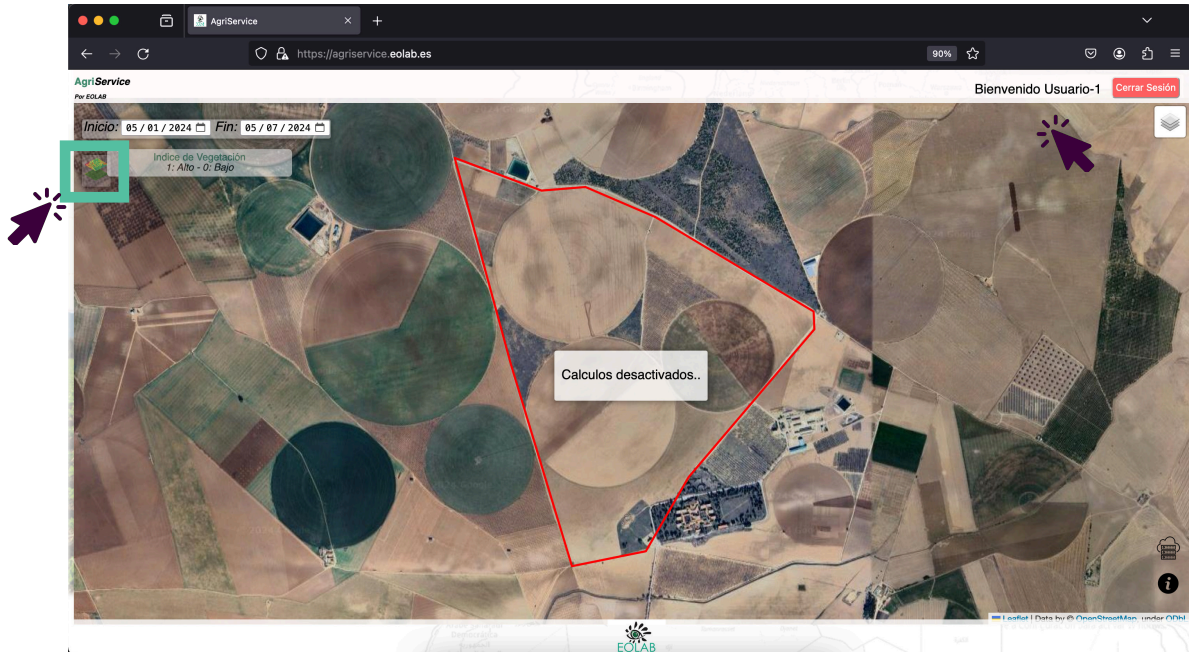


7. Para **calcular un punto distinto**, se minimiza la ventana, y se da clic en otro punto. Si el clic se da fuera de la parcela, aparecerá un mensaje de error

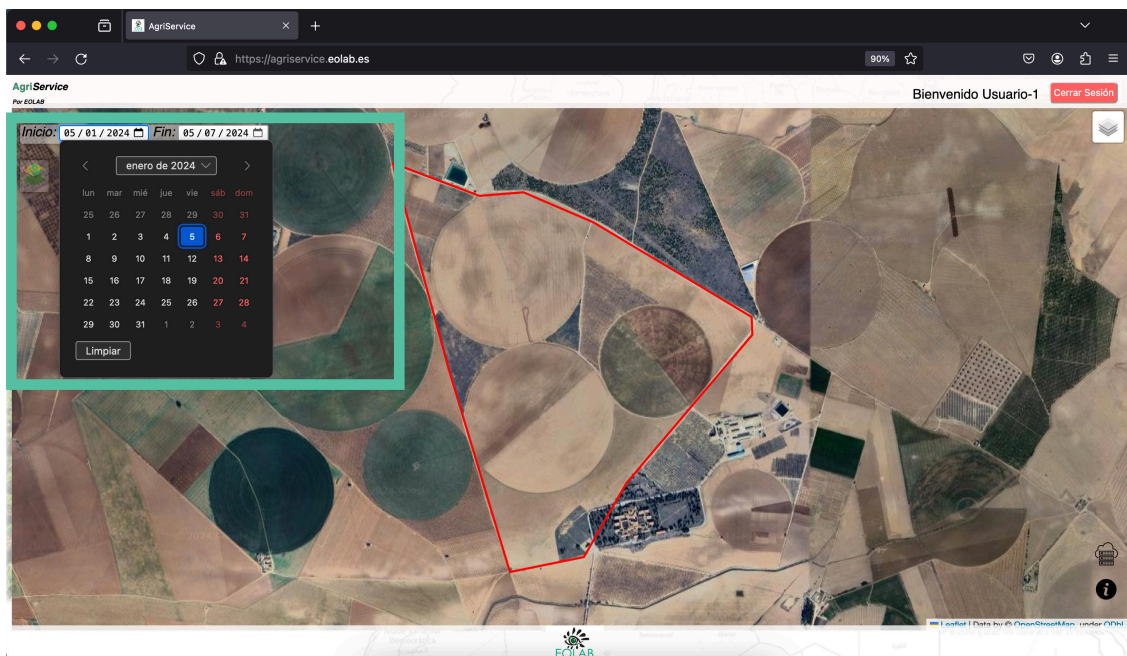


## 5. GENERACIÓN DE SERIE DE TIEMPO

8. Para calcular un **rango distinto de fechas**, se desactivan todos los productos calculados haciendo clic en el botón de los desplegable de los productos.



9. Se establece un nuevo período de tiempo para volver a calcular los productos



# 6. CONSEJOS Y BUENAS PRÁCTICAS

- **Actualice regularmente los datos:** Asegúrese de que los datos satelitales estén actualizados para obtener análisis precisos.
- **Verifique la calidad de los datos:** Revise los datos para detectar posibles anomalías o errores.
- **Utilice comparaciones temporales:** Compare los datos de diferentes períodos para identificar tendencias y patrones.
- **Aproveche las funciones avanzadas:** Explore las herramientas avanzadas del Geoportal para optimizar la monitorización y el asesoramiento.



# 7. SOPORTE Y CONTACTO

Si necesita asistencia adicional, puede ponerse en contacto con el equipo de soporte del Geoportal AgriService:

- Correo electrónico: [marco.amaluisa@eolab.es](mailto:marco.amaluisa@eolab.es)
- Horario de atención: Lunes a Viernes, 9:00 AM - 6:00 PM

Gracias por utilizar Geoportal AgriService para sus necesidades de monitorización y asesoramiento agrícola.



# **Geoportal** **AgriService**

**MARCO A. AMALUISA PEÑAHERRERA**

Universitat Politècnica de València  
Escuela Técnica Superior de Ingeniería Geodésica, Cartográfica y Topográfica  
Máster Universitario en Ingeniería Geomática y Geoinformación

## ANEXO 9.2. Documentación de los métodos para el Cloud Computing (CC) en Google Earth Engine (GEE)



# Documentación de los métodos programados para el Cloud Computing (CC) en Google Earth Engine (GEE)

# ÍNDICE

1. Funciones del Back-end
2. Clase AgriServiceGEE2

### Funciones del Back-end

Estas funciones utilizan la clase *AgriServiceGEE2* para ejecutar los procesos descritos.

Función	Entrada	Descripción	Salida
<i>parcelUserData</i>	Username: Cadena de caracteres que identifican a un usuario registrado en una base de datos.	Verifica la existencia de un usuario en la base de datos y extrae la parcela asociada a dicho usuario como geometría de tipo polígono de EE.	Devuelve el nombre del usuario en la llave 'user' y un objeto que representa la geometría de la parcela asociada a dicho usuario como polígono EE en la llave 'extension'.
<i>selectParcela</i>	User: Cadena de caracteres que identifica a un usuario registrado en una base de datos.	Verifica que el usuario no sea anónimo y si está asociado a una parcela utilizando <i>parcelUserData</i> y genera una lista de coordenadas (latitud, longitud) de la extensión de la parcela	Diccionario de la forma:  <pre>{   "ok": "true",   "message": "Series time is ready",   "data": [{}]} </pre> <p>En la clave "data" devuelve las coordenadas de los vértices de la parcela (Latitud, Longitud) y añade un mensaje de Bienvenida al usuario utilizando la llave "user"</p>
<i>ccDataNDVI</i>	geom: una lista de listas con las coordenadas de los vértices (Longitud, Latitud).  Coords: Lista de coordenadas (Longitud,Latitud)	Transforma coordenadas de las geometrías de (latitud, longitud) a (longitud, latitud), verifica si un punto está dentro de un polígono, y calcula los valores de NDVI en el punto de coordenadas durante un rango de	Output: que es un diccionario con el formato de salida establecido en el método <i>formatOutput</i> de la clase <i>AgriServiceGEE2</i> .

Función	Entrada	Descripción	Salida
	<p>Dateline: Lista con fechas de procesamiento (inicio, final).</p> <p>IV: cadena de texto con el identificador del índice de vegetación.</p> <p>Name_collection: Cadena de texto con el nombre de la colección Sentinel-2 (COPERNICUS/S2_SR_HARMONIZED - COPERNICUS/S2_SR).</p>	<p>fechas, a partir de las imágenes de satélite disponibles y sin nubes.</p>	
ccDataLST	<p>geom: una lista de listas con las coordenadas de los vértices (Longitud, Latitud).</p> <p>Coords: Lista de coordenadas (Longitud, Latitud)</p> <p>Dateline: Lista con fechas de procesamiento (inicio, final).</p> <p>IV: cadena de texto con el identificador del índice de vegetación.</p> <p>Bioph_var: cadena de texto con el identificador de la temperatura de superficie ('LST').</p> <p>Name_collection: Cadena de texto con el nombre de la colección Sentinel-2 (COPERNICUS/S2_SR_HARMONIZED - COPERNICUS/S2_SR).</p>	<p>Transforma coordenadas de las geometrías de (latitud, longitud) a (longitud, latitud), verifica si un punto está dentro de un polígono, y calcula los valores de LST en el punto de coordenadas durante un rango de fechas, a partir de las imágenes de satélite disponibles y sin nubes.</p>	<p>Output: que es un diccionario con el formato de salida establecido en el método <i>formatOutput</i> de la clase <i>AgriServiceGEE2</i>.</p>
ccDataFCOVER	<p>geom: una lista de listas con las coordenadas de los vértices (Longitud, Latitud).</p>	<p>Transforma coordenadas de las geometrías de (latitud, longitud) a (longitud, latitud), verifica si un punto</p>	<p>Output: que es un diccionario con el formato de salida establecido en el</p>

Función	Entrada	Descripción	Salida
	<p>Coords: Lista de coordenadas (Longitud, Latitud)</p> <p>Dateline: Lista con fechas de procesamiento (inicio, final).</p> <p>IV: cadena de texto con el identificador del índice de vegetación.</p> <p>Bioph_var: cadena de texto con el identificador de la fracción de cobertura vegetal ('FCOVER').</p> <p>Name_collection: Cadena de texto con el nombre de la colección Sentinel-2 (COPERNICUS/S2_SR_HARMONIZED - COPERNICUS/S2_SR).</p>	<p>está dentro de un polígono, y calcula los valores de FCOVER en el punto de coordenadas durante un rango de fechas, a partir de las imágenes de satélite disponibles y sin nubes.</p>	<p>método <i>formatOutput</i> de la clase <i>AgriServiceGEE2</i>.</p>
ccDataET	<p>geom: una lista de listas con las coordenadas de los vértices (Longitud, Latitud).</p> <p>Coords: Lista de coordenadas (Longitud, Latitud)</p> <p>Dateline: Lista con fechas de procesamiento (inicio, final).</p> <p>IV: cadena de texto con el identificador del índice de vegetación.</p> <p>Bioph_var: cadena de texto con el identificador de la evapotranspiración ('ET').</p>	<p>Transforma coordenadas de las geometrías de (latitud, longitud) a (longitud, latitud), verifica si un punto está dentro de un polígono, y calcula los valores de ET en el punto de coordenadas durante un rango de fechas, a partir de las imágenes de satélite disponibles y sin nubes.</p>	<p>Output: que es un diccionario con el formato de salida establecido en el método <i>formatOutput</i> de la clase <i>AgriServiceGEE2</i>.</p>

Función	Entrada	Descripción	Salida
	<p>Name_collection: Cadena de texto con el nombre de la colección Sentinel-2 (COPERNICUS/S2_SR_HARMONIZED - COPERNICUS/S2_SR).</p>		
<p>tileParcela</p>	<p>geom: una lista de listas con las coordenadas de los vértices (Longitud, Latitud).</p> <p>Tile: cadena de texto con el identificador del producto deseado sobre el mapa: 'NDVI' 'Temperature', 'fcover', 'et'</p> <p>Dateline: Lista con fechas de procesamiento (inicio, final).</p> <p>Name_collection: Cadena de texto con el nombre de la colección Sentinel-2 (COPERNICUS/S2_SR_HARMONIZED - COPERNICUS/S2_SR).</p> <p>Image: (Opcional, Default: None) cadena de texto que representa un id de EE de una imagen de satélite.</p>	<p>Si no se especifica el parámetro <i>Image</i>, obtiene una lista de ids de EE de imágenes de satélite y escoge el ultimo id, a partir de esto establece un diccionario de parámetros para el método mapId de la clase AgriServiceGEE2 y obtiene una url del mapa con los parámetros de visualización preestablecidos para cada producto.</p>	<p>Diccionario de la forma:</p> <pre>{   "ok": "true",   "message": "Series time is ready",   "data": [{}]} </pre> <p>Donde la llave data tiene la url generada para construir la capa tile y los parámetros de construcción.</p>

## Clase AgriServiceGEE2

La clase recibe dos parámetros de inicialización:

1. El ID de una cuenta de Servicio de Google Earth Engine
2. El directorio de la localización del fichero JSON con la llave de acceso a la cuenta de servicio.

Método	Entrada	Descripción	Salida
ImageEE	Indeximage: cadena de texto que representa un id de EE de una imagen de satélite.	Transforma un id de GEE en un objeto que representa una imagen de EE.	ImageEE: objeto que representa una imagen de EE.
Point	geom: una lista de Coordenadas (Longitud, Latitud).	Construye un objeto que representa una geometría EE de tipo punto.	PointEE: objeto que representa un punto de EE.
Polygon	geom: una lista de listas con las coordenadas de los vértices (Longitud, Latitud).	Construye un objeto que representa una geometría EE de tipo polígono.	PolygonEE: objeto que representa un polígono EE.
polygon2Leaflet	PolygonEE: objeto que representa un polígono EE.	Transforma una geometría EE de tipo polígono en una lista de coordenadas (Latitud, Longitud).	Lista: Lista de listas con las coordenadas de los vértices (Latitud, Longitud).
checkInsideOutside	PolygonEE: objeto que representa un polígono de EE.  PointEE: objeto que representa un punto de EE.	Verifica si una geometría EE de tipo punto está espacialmente dentro de otra geometría EE de tipo polígono.	Booleano
ImageCollection	Polygon: objeto que representa un polígono de EE.  Dateline: Lista con fechas de procesamiento (inicio, final).  Name_collection: Cadena de texto con el nombre de la colección Sentinel-2	Computa una colección de imágenes EE.	Diccionario con el total de imágenes procesadas en la clave 'total_images', y el objeto que representa la colección de imágenes EE, en la clave 'collection'.

Método	Entrada	Descripción	Salida
	<p>(COPERNICUS/S2_SR_HARMONIZED - COPERNICUS/S2_SR).</p> <p>N: (Opcional, Default: None) Valor de porcentaje para filtro de nubosidad.</p>		
harmonized	<p>CollectionEE: objeto que representa una colección de imágenes EE armonizadas de Sentinel-2.</p> <p>Dateline: Lista con fechas de procesamiento (inicio, final).</p>	<p>Transforma las imágenes de la colección armonizada cuya fecha sea posterior al 25/01/2022 (PROCESSING_BASELINE '04.00').</p>	<p>Diccionario con el total de imágenes procesadas en la clave 'total_images', y CollectionEE que es un objeto que representa la colección de imágenes EE, en la clave 'collection'.</p>
preProcessingSentinel	<p>Polygon: objeto que representa un polígono de EE.</p> <p>CollectionEE: objeto que representa una colección de imágenes EE armonizadas de Sentinel-2.</p> <p>Name_collection: Cadena de texto con el nombre de la colección Sentinel-2 (COPERNICUS/S2_SR_HARMONIZED - COPERNICUS/S2_SR).</p> <p>Point: (Opcional, Default: None) objeto que representa un punto de EE.</p>	<p>Recorta cada imagen de una colección de imágenes EE con respecto a un polígono EE, en el caso de especificar un Punto EE calcula una colección de imágenes EE donde la clasificación de uso de suelo en cada imagen es vegetación y suelo desnudo sobre dicho punto.</p>	<p>Diccionario con el total de imágenes procesadas en la clave 'total_images', y CollectionEE que es un objeto que representa la colección preprocesada de imágenes EE, en la clave 'collection'.</p>
calcIVfromCollection	<p>CollectionEE: objeto que representa una colección de imágenes EE de Sentinel-2.</p>	<p>Calcula y añade una banda de índice de vegetación para todas las</p>	<p>CollectionEE que es un objeto que representa la colección de</p>



Método	Entrada	Descripción	Salida
	<p>IV: cadena de texto con el identificador del índice de vegetación.</p> <p>Bands: Lista con las bandas utilizadas para el cálculo del índice de vegetación.</p>	<p>imágenes de una colección de imágenes EE.</p>	<p>imágenes EE con la banda del índice de vegetación.</p>
calcLSTFromCollection	<p>CollectionEE: objeto que representa una colección de imágenes EE de Sentinel-2 con la banda de índice de vegetación.</p> <p>IV: cadena de texto con el identificador del índice de vegetación.</p>	<p>Calcula y añade una banda de la temperatura en superficie (°K) para todas las imágenes de una colección de imágenes EE.</p>	<p>CollectionEE que es un objeto que representa la colección de imágenes EE con la banda de temperatura en superficie.</p>
calcFCOVERFromCollection	<p>CollectionEE: objeto que representa una colección de imágenes EE de Sentinel-2 con la banda de índice de vegetación.</p> <p>IV: cadena de texto con el identificador del índice de vegetación.</p> <p>Ndvis: valor flotante que representa el índice de vegetación de suelo (mínimo).</p>	<p>Calcula y añade una banda de la fracción de cobertura vegetal para todas las imágenes de una colección de imágenes EE.</p>	<p>CollectionEE que es un objeto que representa la colección de imágenes EE con la banda de fracción de cobertura vegetal.</p>

Método	Entrada	Descripción	Salida
	Ndvi8: valor flotante que representa el índice de vegetación infinito (máximo).		
calcEMIFromCollection	CollectionEE: objeto que representa una colección de imágenes EE de Sentinel-2 con la banda de índice de vegetación.  IV: cadena de texto con el identificador del índice de vegetación.	Calcula y añade una banda de emisividad para todas las imágenes de una colección de imágenes EE.	CollectionEE que es un objeto que representa la colección de imágenes EE con la banda de emisividad.
calcALBFromCollection	CollectionEE: objeto que representa una colección de imágenes EE de Sentinel-2 con la banda de índice de vegetación.	Calcula y añade una banda de albedo para todas las imágenes de una colección de imágenes EE.	CollectionEE que es un objeto que representa la colección de imágenes EE con la banda de albedo.
calcRNFromCollection	Polygon: objeto que representa un polígono de EE  CollectionEE: objeto que representa una colección de imágenes EE de Sentinel-2 con la banda de índice de vegetación.  IV: cadena de texto con el identificador del índice de vegetación.  Dateline: Lista con fechas de procesamiento (inicio, final).	Calcula y añade una banda de radiación total para todas las imágenes de una colección de imágenes EE utilizando las imágenes del conjunto de datos ERA5-Land para obtener el DSSF y DSLF.	CollectionEE que es un objeto que representa la colección de imágenes EE con la banda de radiación total.
WLS	t1: lista de valores máximos de la variable independiente de la regresión	Calcula una regresión lineal de mínimos cuadrados ponderados de valores mínimos y máximos	Diccionario con los coeficientes de la regresión.

Método	Entrada	Descripción	Salida
	<p>t9: lista de valores mínimos de la variable independiente de la regresión</p> <p>a: lista de valores de la variable dependiente a estimar</p> <p>n: lista de valores de los pesos de la regresión lineal.</p>		
calcVectorsFromCollection	CollectionEE: objeto que representa una colección de imágenes EE de Sentinel-2 con la banda de temperatura de superficie y albedo.	Calcula un vector de regresión en base a las bandas de albedo y temperatura de superficie para todas las imágenes de una colección de imágenes EE y la coloca entre las propiedades de la imagen.	CollectionEE que es un objeto que representa la colección de imágenes EE con la propiedad del vector de regresión.
calcRegressionFromCollection	<p>CollectionEE: objeto que representa una colección de imágenes EE de Sentinel-2 con un vector de regresión en sus propiedades.</p> <p>Coords: Lista de coordenadas (Longitud, Latitud)</p>	Aplica el método WLS y calcula un feature EE para todas las imágenes de una colección de imágenes EE con las propiedades de los coeficientes de la regresión y los parámetros solares en un punto de coordenadas.	FeatureCollectionEE que es un objeto que representa una colección de features EE con los coeficientes de la regresión y parámetros solares.
calcSSEBIFromCollection	<p>FeatureCollectionEE: objeto que representa una colección de features EE con los coeficientes de la regresión y parámetros solares.</p> <p>CollectionEE: objeto que representa una colección de imágenes EE de Sentinel-2 con la banda de radiación total.</p>	Calcula y añade una banda de S-SEBI para todas las imágenes de una colección de imágenes EE juntando una colección de features EE con una colección de imágenes EE a través del id de EE.	CollectionEE que es un objeto que representa la colección de imágenes EE con la banda S-SEBI.

Método	Entrada	Descripción	Salida
removeOutliersFromCollection	CollectionEE: objeto que representa la colección de imágenes EE con la banda S-SEBI.	Calcula y añade una banda de S-SSEBI filtrado para todas las imágenes de una colección de imágenes EE construyendo una ventana de valores menores a 1.1 y mayores a -0.1 de S-SEBI	CollectionEE que es un objeto que representa la colección de imágenes EE con la banda S-SEBI filtrado.
calcSSEBINormalizedCollection	CollectionEE: un objeto que representa la colección de imágenes EE con la banda S-SEBI filtrados.	Calcula y añade una banda de S-SSEBI normalizado y saturado a valores entre 1 y 0 para todas las imágenes de una colección de imágenes EE	CollectionEE que es un objeto que representa la colección de imágenes EE con la banda S-SEBI normalizado.
calcETFromCollection	CollectionEE: un objeto que representa la colección de imágenes EE con las bandas de S-SEBI normalizado, radiación total y parámetros solares.	Calcula y añade una banda de evapotranspiración para todas las imágenes de una colección de imágenes EE	CollectionEE que es un objeto que representa la colección de imágenes EE con la banda de evapotranspiración.
valuesInPoint	CollectionEE: un objeto que representa la colección de imágenes EE.  Point: objeto que representa un punto de EE.  Prop: Lista de propiedades que debe contener el identificador de una de las bandas de la colección a graficar como último elemento.	Calcula los valores de una banda a lo largo de todas las imágenes de una colección de imágenes EE	Una lista de listas donde cada lista contiene las propiedades y el valor de la banda en un punto EE. El valor de la banda será <i>null</i> si en una imagen no existe la banda proporcionada en el parámetro <i>Prop</i> .
formatOutput	chart_data: diccionario con los parámetros del CC en GEE tiene la forma:	Organiza los datos del CC para a serie de tiempo.	CCdata que es un diccionario organizado con estructura  {

Método	Entrada	Descripción	Salida
	<pre> chart_data = {   'totalcc': no. de imágenes sin     procesar,   'computedcc': no. de imágenes preprocesadas,   'variable': variable biofísica,   'dates': fechas de     procesamiento,   'coords': lista de coordenadas     (longitud, latitud),   'value_list': lista de valores en     el punto de     coordenadas,   'time_list': lista con las fechas     de los valores,   'index': lista de id de EE de     cada imagen }                     </pre>		<pre> "ok": "true", "message": "Series time is ready", "data": {} }                     </pre> <p>Donde data contiene la información para construir un gráfico de serie de tiempo y los datos del computo en la nube.</p>
imagesList	<p>Polygon: objeto que representa un polígono de EE</p> <p>Tile: Cadena de caracteres que identifica al producto deseado</p> <p>Dateline: Lista con fechas de procesamiento (inicio, final).</p> <p>Name_collection: Cadena de texto con el nombre de la colección Sentinel-2 (COPERNICUS/S2_SR_HARMONIZED - COPERNICUS/S2_SR).</p>	<p>Computa una colección de imágenes EE, la preprocesa y la convierte en una lista de id de EE.</p>	<p>ImgesListID que es un diccionario con la lista de id de EE de las imágenes filtradas en la llave 'filtered' y todos los id de EE de las imágenes sin filtrar en la llave 'no_filtered'</p>

Método	Entrada	Descripción	Salida
mapID	<p>Params: diccionario con los parámetros para obtener una imagen de EE</p> <pre>Params = {   'index': id de EE de la imagen,   'dateline': lista de fechas,   'extension': polígono de EE,   'tile': identificador del producto,   'collection':   COPERNICUS/S2_SR_HARMONIZED - COPERNICUS/S2_SR }</pre>	<p>Genera una URL y una imagen de leyenda de acuerdo con los parámetros ingresados</p>	<p>URL que es una cadena de caracteres de la url que permite construir los tiles generados</p> <p>Imagen de leyenda</p>

## ANEXO 9.3. Relación del trabajo con los Objetivos de Desarrollo Sostenible de la Agenda 2030.

**ANEXO 9.3. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030**

**Anexo al Trabajo de Fin de Grado y Trabajo de Fin de Máster: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030**

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

<b>Objetivos de Desarrollo Sostenibles</b>	<b>Alto</b>	<b>Medio</b>	<b>Bajo</b>	<b>No Procede</b>
ODS 1. <b>Fin de la pobreza.</b>				<b>X</b>
ODS 2. <b>Hambre cero.</b>		<b>X</b>		
ODS 3. <b>Salud y bienestar.</b>				<b>X</b>
ODS 4. <b>Educación de calidad.</b>				<b>X</b>
ODS 5. <b>Igualdad de género.</b>				<b>X</b>
ODS 6. <b>Agua limpia y saneamiento.</b>		<b>X</b>		
ODS 7. <b>Energía asequible y no contaminante.</b>				<b>X</b>
ODS 8. <b>Trabajo decente y crecimiento económico.</b>				<b>X</b>
ODS 9. <b>Industria, innovación e infraestructuras.</b>		<b>X</b>		
ODS 10. <b>Reducción de las desigualdades.</b>				<b>X</b>
ODS 11. <b>Ciudades y comunidades sostenibles.</b>				<b>X</b>
ODS 12. <b>Producción y consumo responsables.</b>				<b>X</b>
ODS 13. <b>Acción por el clima.</b>	<b>X</b>			
ODS 14. <b>Vida submarina.</b>				<b>X</b>
ODS 15. <b>Vida de ecosistemas terrestres.</b>		<b>X</b>		
ODS 16. <b>Paz, justicia e instituciones sólidas.</b>				<b>X</b>
ODS 17. <b>Alianzas para lograr objetivos.</b>				<b>X</b>



**Descripción de la alineación del TFG/TFM con los ODS con un grado de relación más alto.**

El Geoportal AgriService contribuye directamente al Objetivo 13 de la Agenda 2030 para el Desarrollo Sostenible, que busca tomar medidas urgentes para combatir el cambio climático y sus impactos. Al proporcionar herramientas de monitorización de cultivos basadas en datos de satélite se puede asegurar un futuro asesoramiento de riego al agricultor, el proyecto ayuda a los agricultores a adaptarse a condiciones climáticas cambiantes y a mitigar los efectos adversos del cambio climático en la producción agrícola. Al optimizar el uso del agua en la agricultura, el Geoportal AgriService también contribuye a la reducción de emisiones de gases de efecto invernadero asociadas con el riego excesivo o ineficiente.