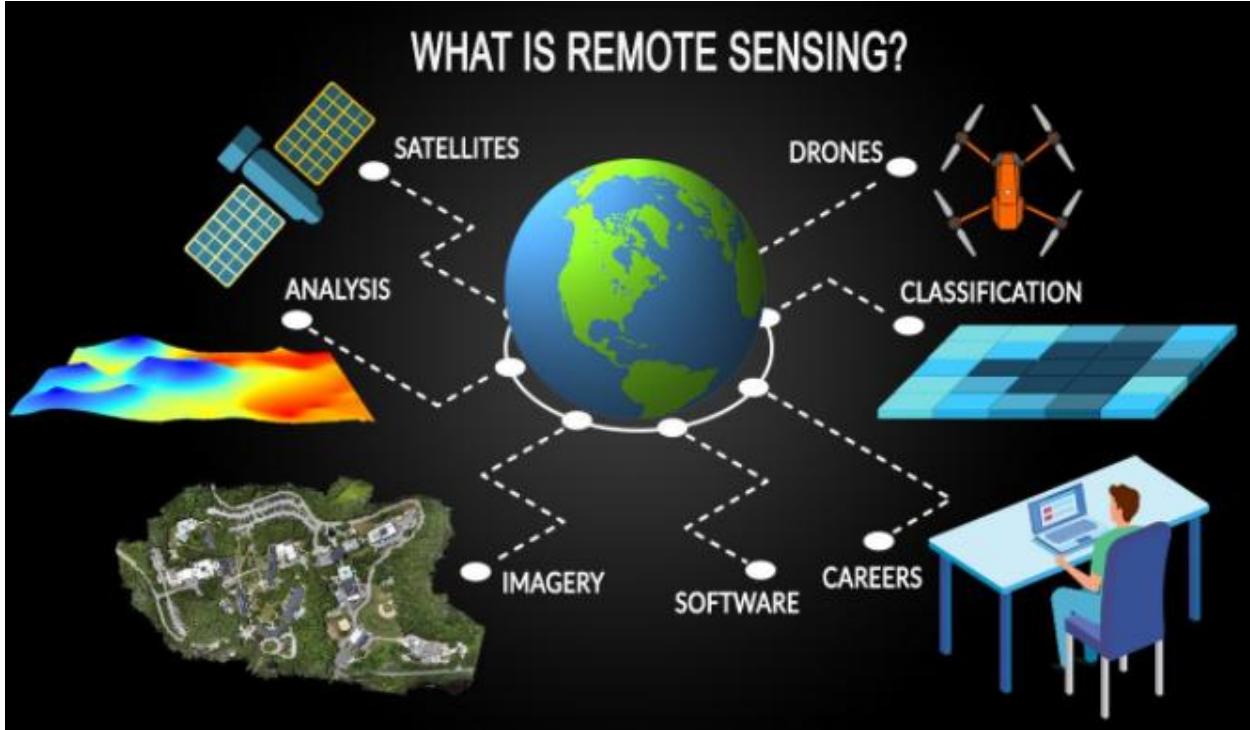


Remote Sensing with Google Earth Engine

César Iván Alvarez Mendoza PhD.
cesarivanalvarezmendoza@gmail.com

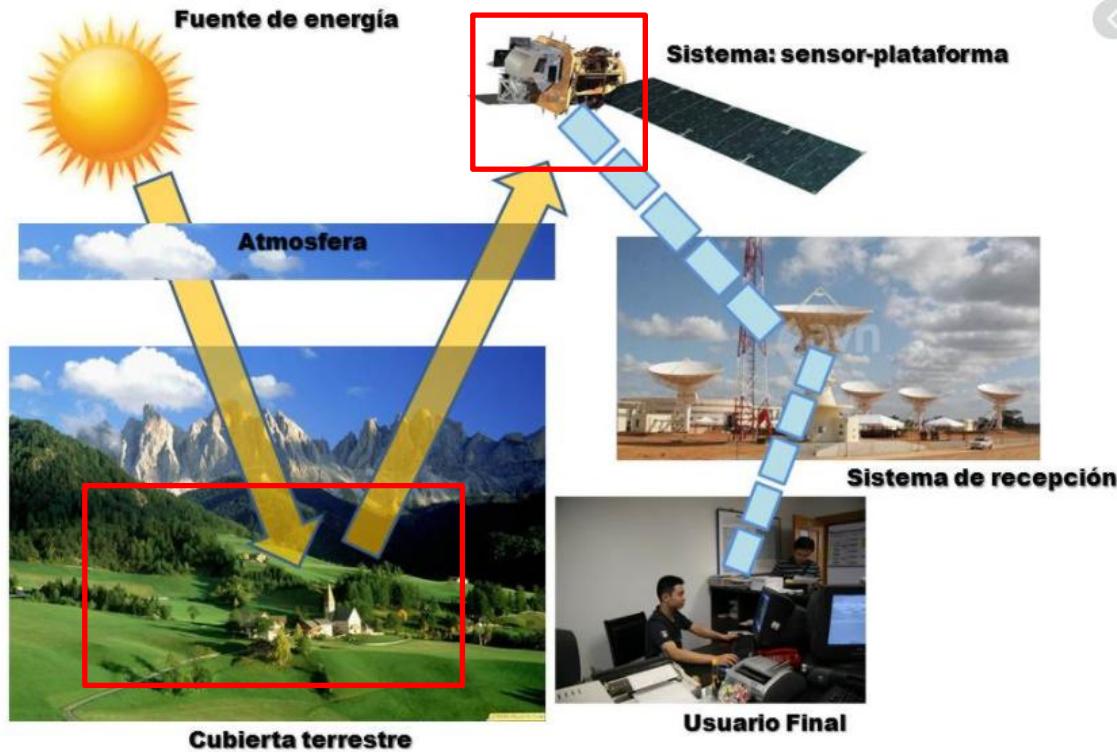


Conceptos de Teledetección



También conocida como Sensores Remotos (**Remote Sensing**) es un método para obtener información sobre las propiedades de un objeto **sin tener contacto físico** con él.

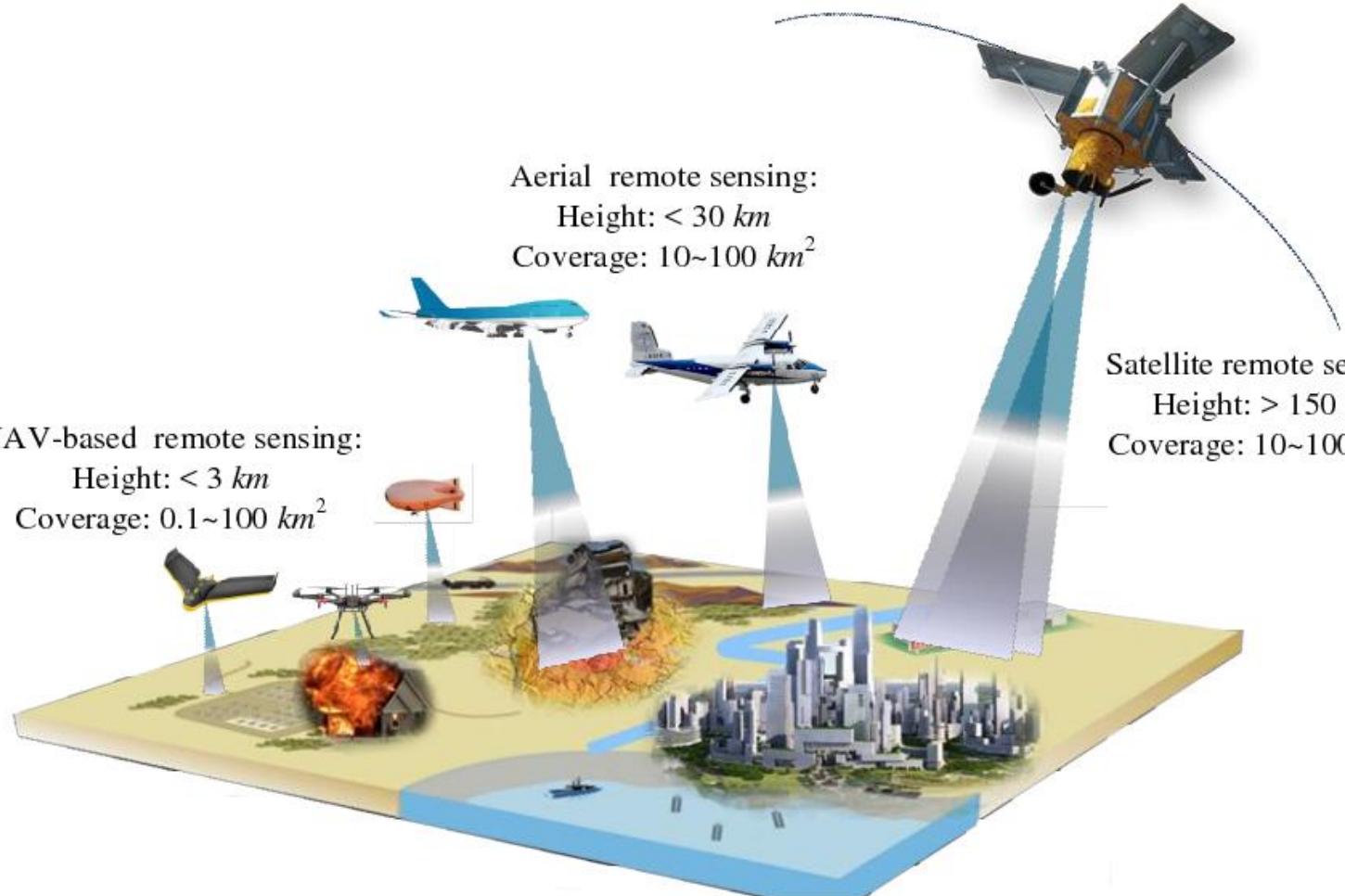
Componentes de la Teledetección



UAV-based remote sensing:
Height: < 3 km
Coverage: 0.1~100 km^2

Aerial remote sensing:
Height: < 30 km
Coverage: 10~100 km^2

Satellite remote sensing:
Height: > 150 km
Coverage: 10~1000 km^2

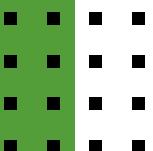


El espectroelectromagnético

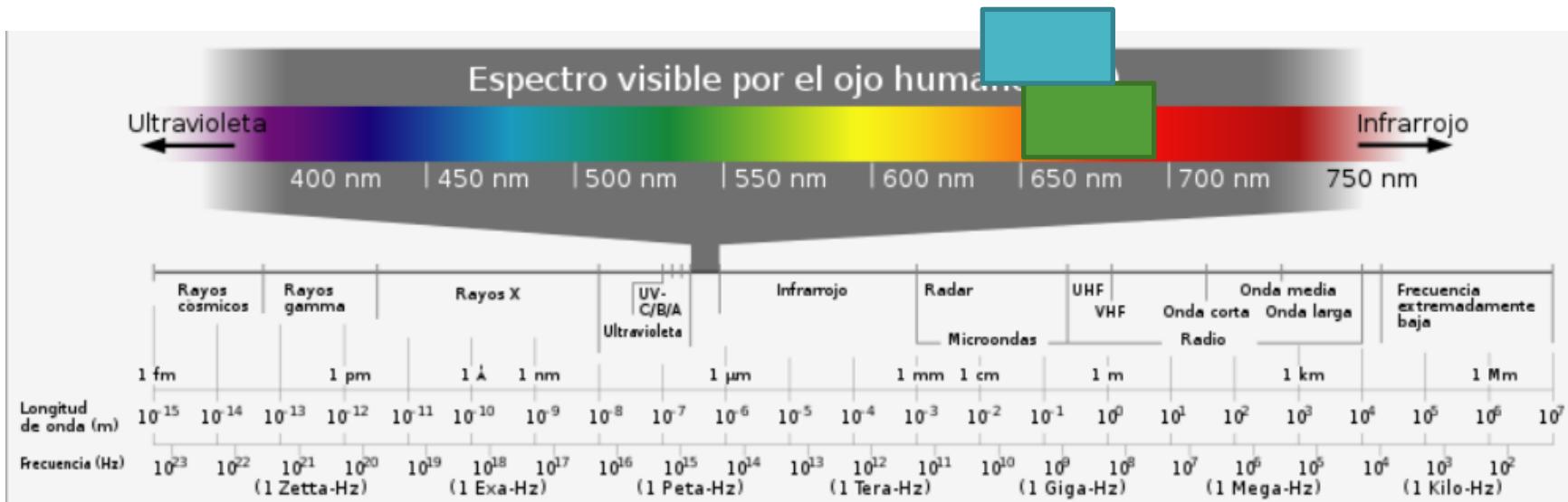
La **radiación electromagnética** comprende una amplia **variedad de** frecuencias o de **longitudes de onda** que abarcan desde los rayos gamma a las ondas de radio. Todas estas emisiones constituyen así el **espectro electromagnético**.

Las radiaciones más utilizadas en la teledetección son:

- Las **microondas** se usan en los sensores de radar.
- La **radiación infrarroja** radiación emitida por los cuerpos calientes.
- El **espectro visible (RGB)**
- La **radiación ultravioleta** es la componente principal de la radiación solar.



El espectroelectromagnético

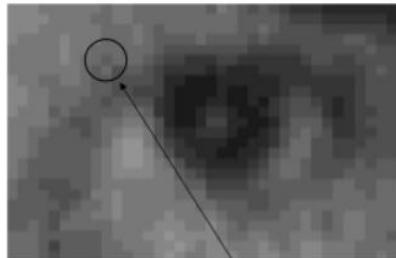


- ■ ■
- ■ ■
- ■ ■
- ■ ■
- ■ ■
- ■ ■
- ■ ■

Tipos de Resoluciones

- Resolución Espacial - Geográfica
- Resolución Espectral
- Resolución Radiométrica
- Resolución Temporal

Resolución Espacial - Geográfica



RESOLUCION

Geográfica



TAMAÑO
DEL PIXEL

Detalle mínimo
TAMAÑO
DEL GRANO



IMAGEN
DIGITAL



FOTOGRAFIA

ERNESTO G. ABRIL (2013)

Resolución

Geográfica

aerofotografía



Imagen satelital

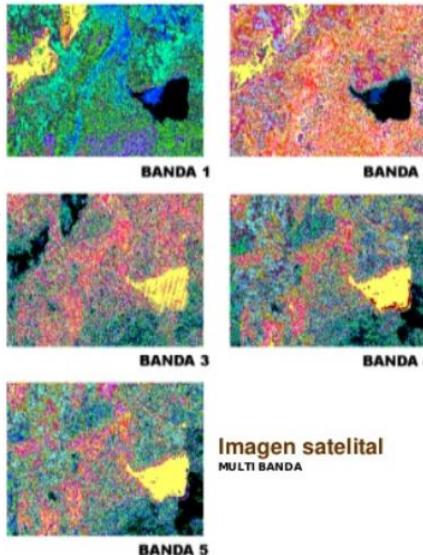
TAMAÑO
DEL PIXEL
Detalle mínimo

ERNESTO G. ABRIL (2013)

Resolución Espacial - Geográfica



Resolución Espectral



Resolución

Espectral

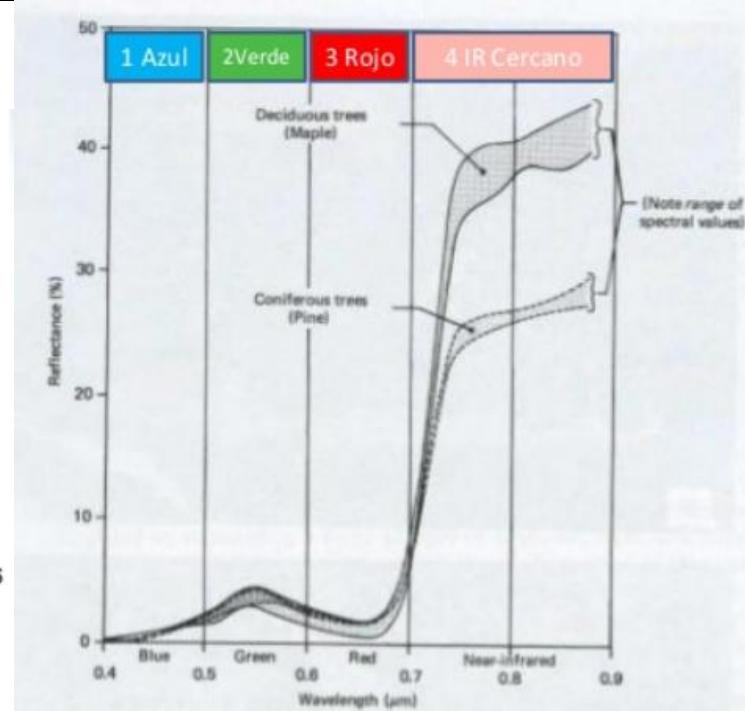
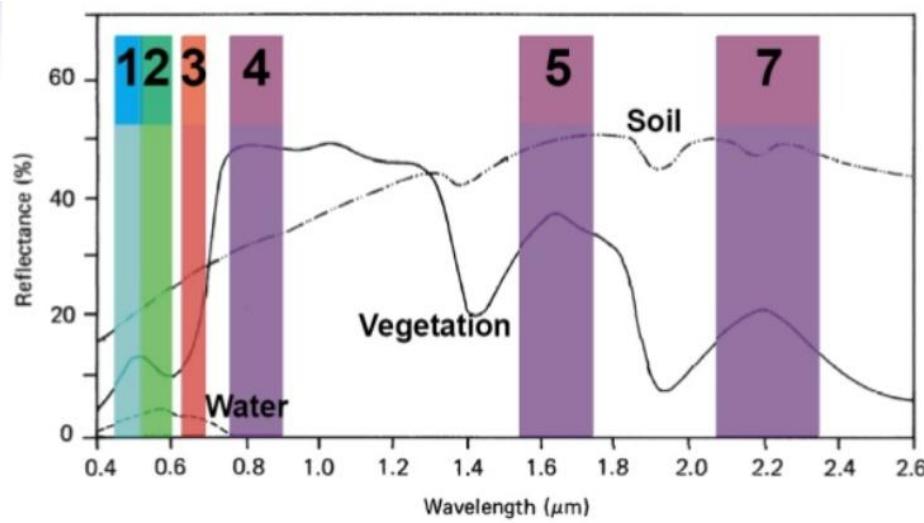
Cantidad de
Bandas del sensor



aerofotografía

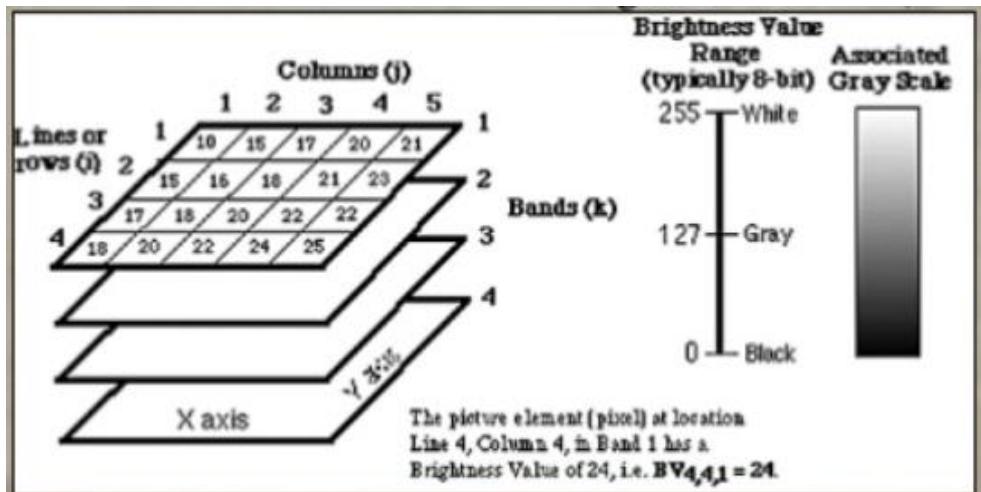


Resolución Espectral - Firmas espectrales

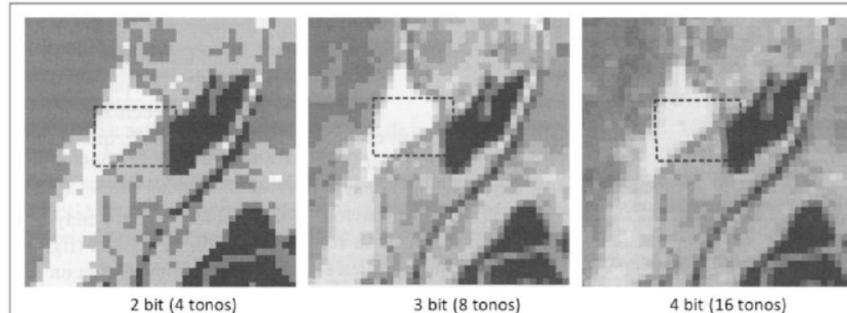


Resolución Radiométrica

A mayor **número de niveles** que pueden ser registrados mayor es la resolución radiométrica del sistema sensor 8, 12, 16, 24 entre otros

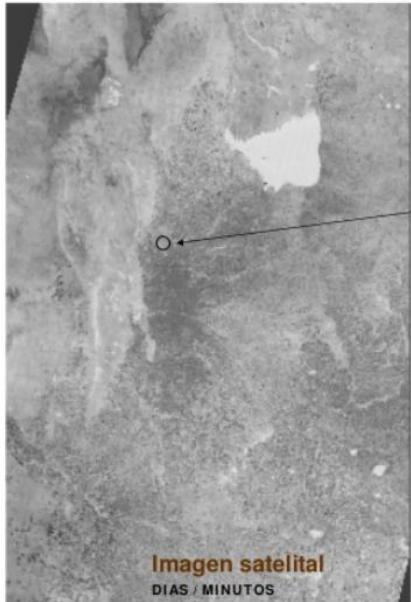


Resolución Radiométrica



Fuente: Adaptado de Chuvieco , 2002

Resolución Temporal



Resolución Temporal

Frecuencia de toma



1965



1978

aerofotografía

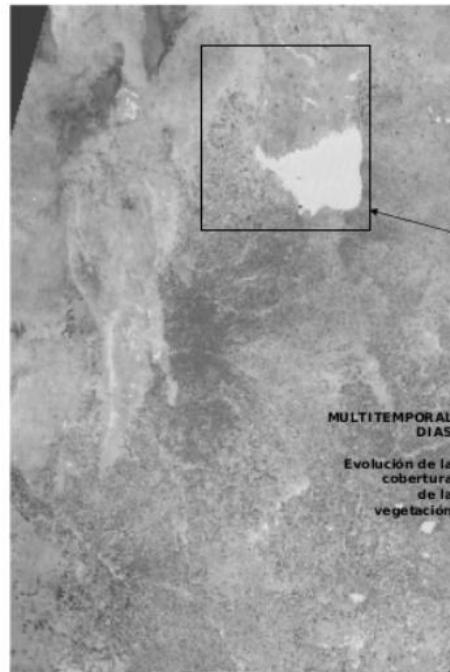
PLURIANUAL
(Catastro urbano)



1980

AEROFOTOGRAFIA PANORAMICA

ABRIL, E. G., 2013. UNC.



NOAA-AVHRR
Imágenes satelitales

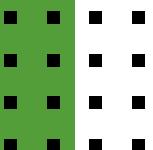


Corrección imágenes satelitales

La corrección de la imagen tiende a eliminar anomalías en su localización o en los Números Digitales (ND) para dar una adecuada posición de los píxeles y/o de su valor de radiancia.

Muchas veces no se corrige defectos de la imagen, sino que se la modifica para un mejor uso o interpretación. Las correcciones presentes para imágenes satelitales se detallan a continuación:

- **Movimiento de la plataforma, proyección central de los sensores, terrenos muy quebrados (corrección geométrica)**
- **Pixeles defectuosos por fallas en los sensores (corrección radimétrica)**
- **Alteración del valor de pixel por Interferencia de la atmósfera, (corrección atmosférica).**

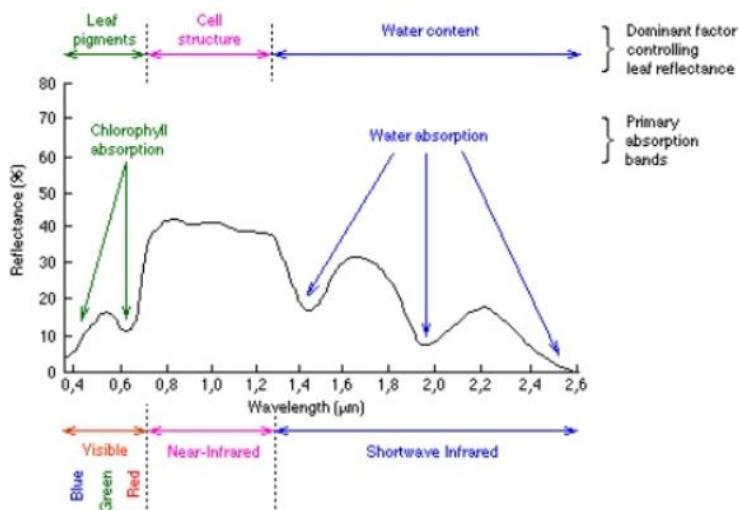


Índices multiespectrales

Existe un gran número de índices espectrales que pueden analizar diversos aspectos como la vegetación, los recursos hídricos, la nieve, el suelo, el fuego, entre otros. Es decir damos un **realce** al elemento de interés sobre la superficie terrestre.

Radiancia es el flujo radiante emitido, reflejado, transmitido o recibido por una superficie dada, por unidad de ángulo sólido por unidad de área proyectada.

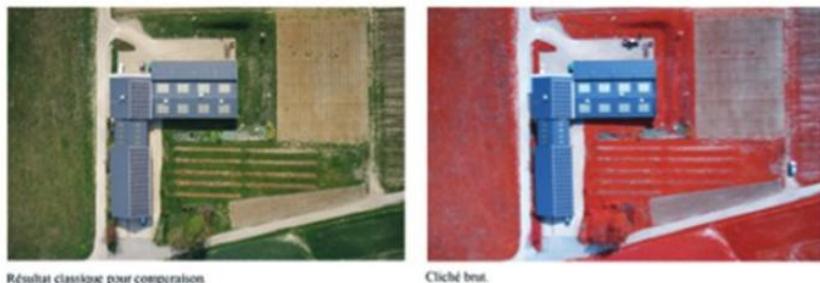
Reflectancia es la relación entre la radiancia reflejada contra la potencia total de energía.



NDVI

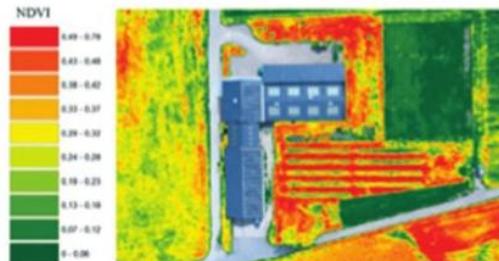
El Índice de Vegetación de Diferencia Normalizada (NDVI), es un índice de vegetación que se utiliza para estimar la cantidad, calidad y desarrollo de la vegetación con base a la medición de la intensidad de la radiación de ciertas bandas del espectro electromagnético

$$NDVI = \frac{NIR-RED}{NIR+RED}$$

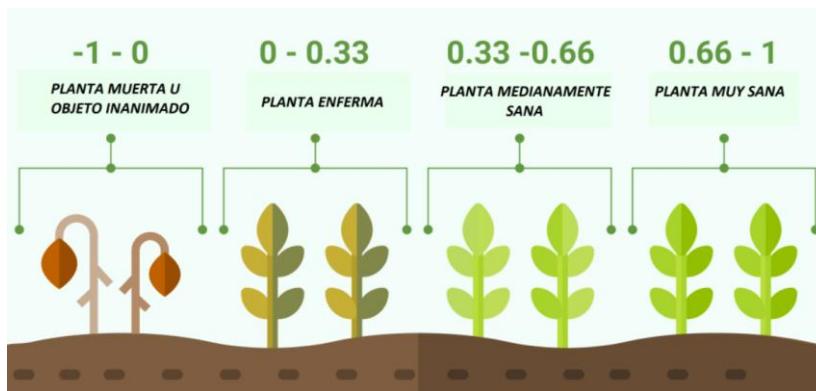
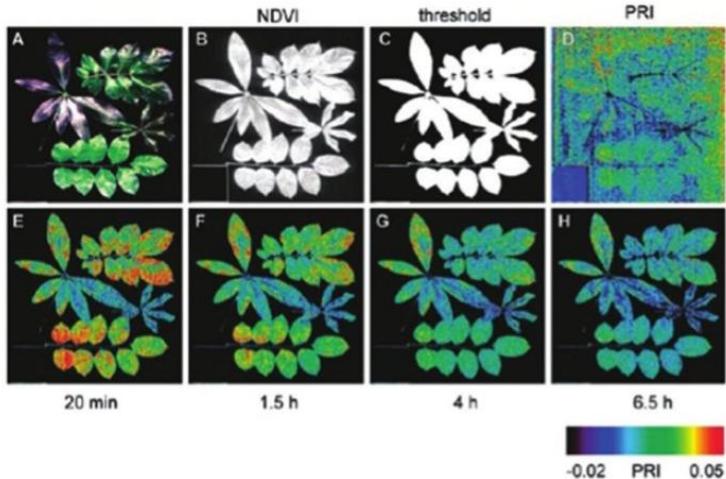
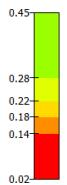
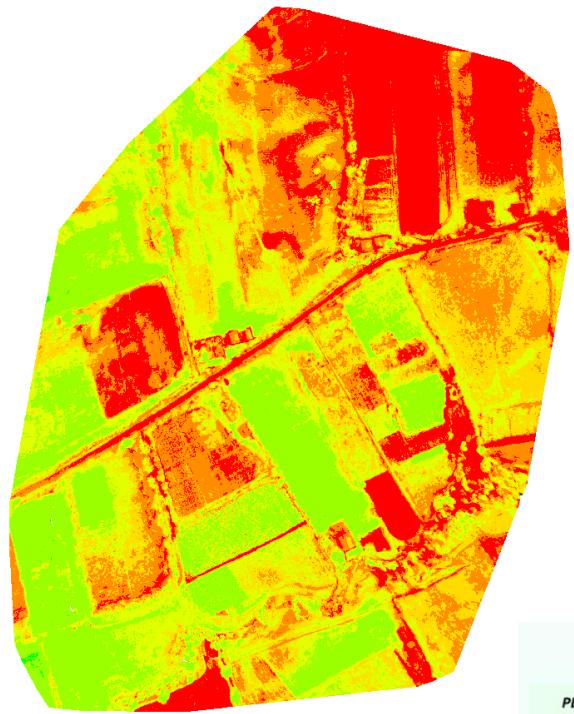


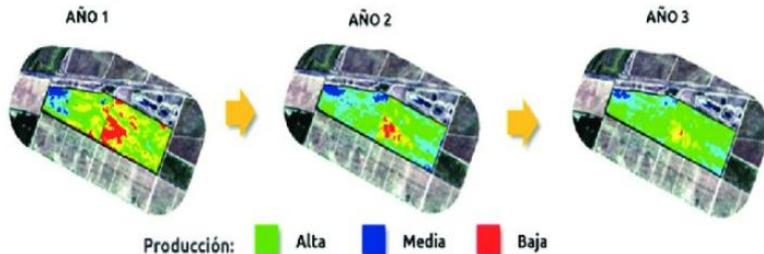
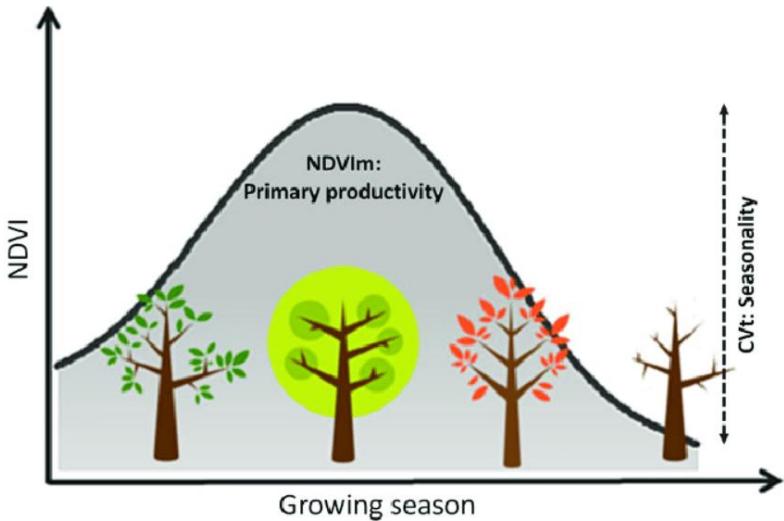
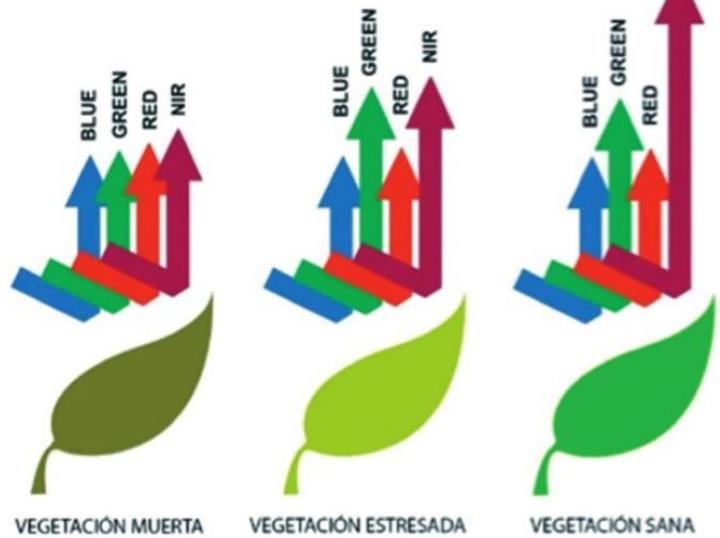
Résultat classique pour comparaison.

Cliché brut.



Carte d'activité végétale NDVI obtenue à partir du cliché brut.

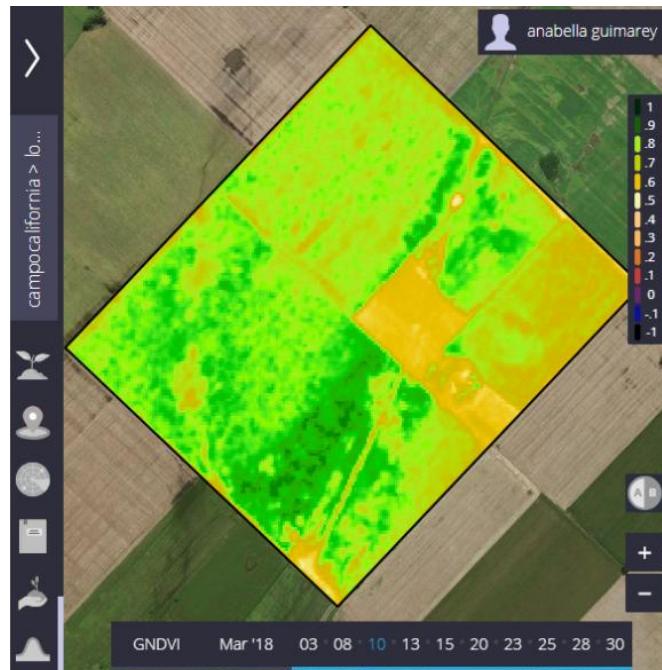




GNDVI

El Índice GNDVI (Vegetación de Diferencia Normalizada Verde) es un índice del "verdor" de la planta o actividad fotosintética. Es uno de los índices de vegetación más utilizados para determinar la captación de agua y nitrógeno en el dosel del cultivo.

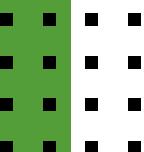
$$GNDVI = \frac{R_{NIR} - R_{Green}}{R_{NIR} + R_{Green}}$$



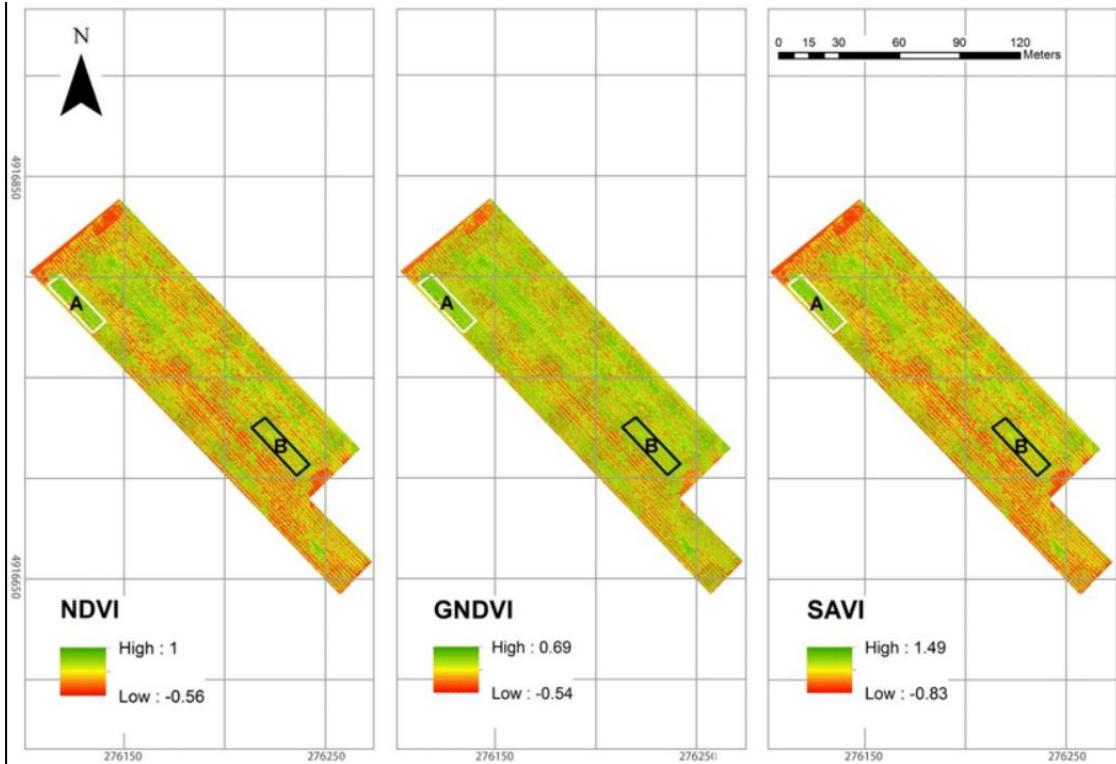
SAVI

$$\text{SAVI} = \frac{(1 + L)(\text{NIR} - \text{Red})}{(\text{NIR} + \text{Red} + L)}$$

El SAVI (Índice de Vegetación Ajustado al Suelo), muestra una ligera variante respecto a la fórmula tradicional del NDVI para evitar distorsiones en los valores de análisis cuando la vegetación se encuentra sobre suelos expuestos. Condiciones como la temperatura o la humedad pueden ejercer influencia en las bandas de trabajo analizadas y, por tanto, en los resultados ofrecidos por el indicador. En este caso, el índice de vegetación SAVI tratará de evitar esta influencia del suelo sobre los resultados añadiendo un factor adicional (L) en la ecuación del NDVI que permitirá trabajar en escenarios donde el desarrollo vegetal sea incipiente.

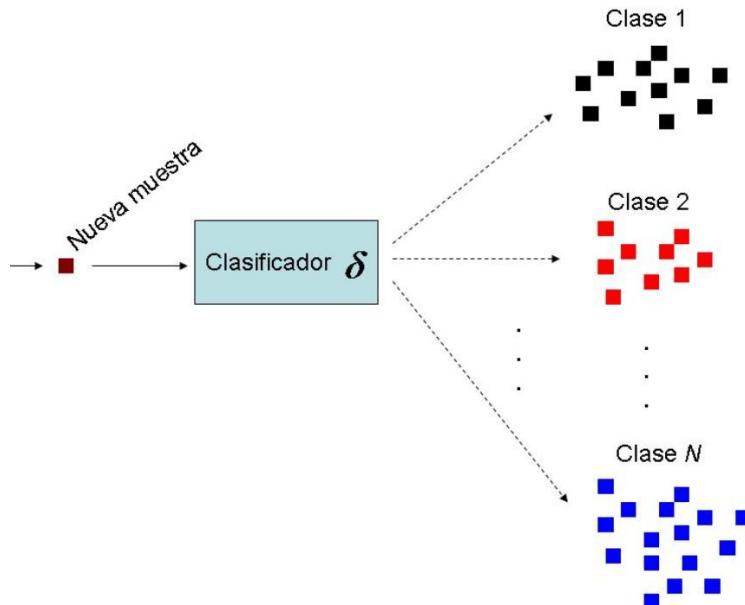


Índices multiespectrales

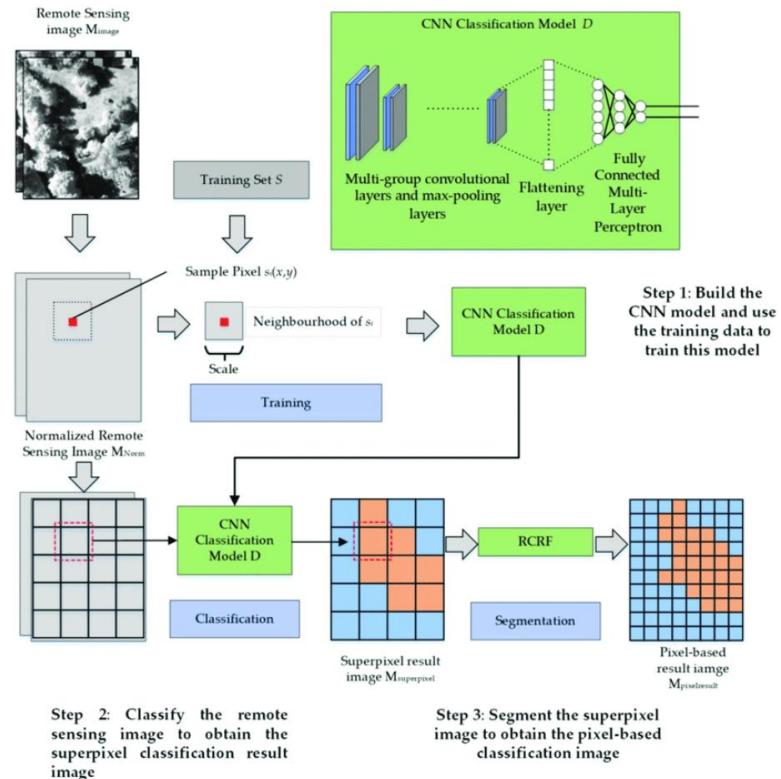
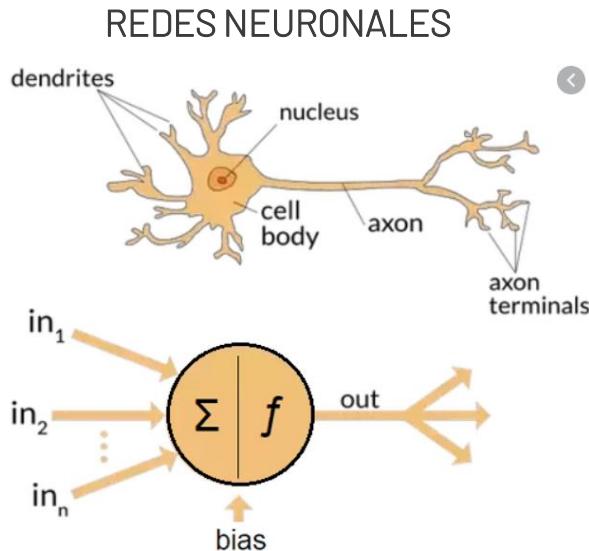


Clasificación Supervisada

La clasificación supervisada requiere que se ingrese regiones de interés (ROIs) para cada clase de cobertura del suelo, estos son polígonos que se trazan sobre áreas homogéneas que se sobreponen a pixeles pertenecientes a la misma clase de la cobertura del suelo.

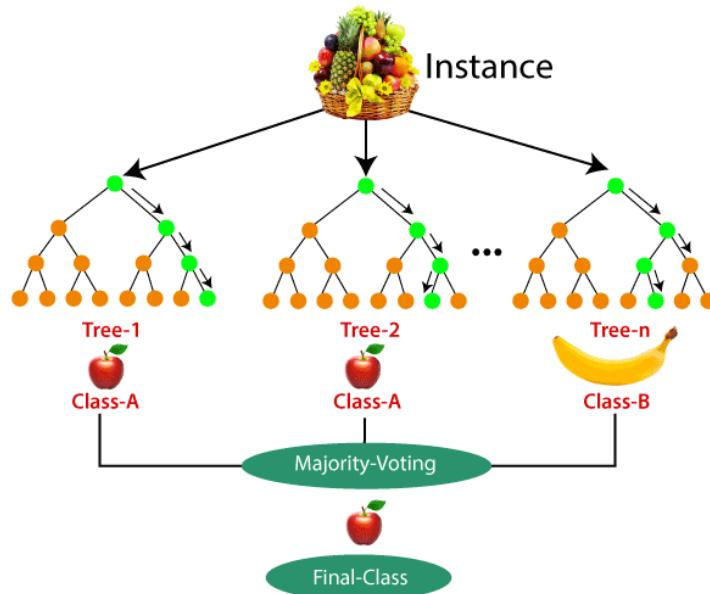
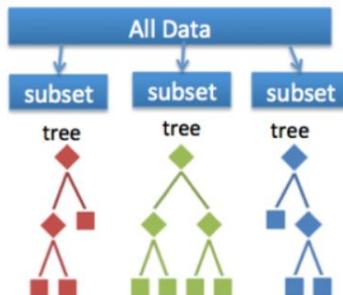


Clasificación Supervisada - Métodos

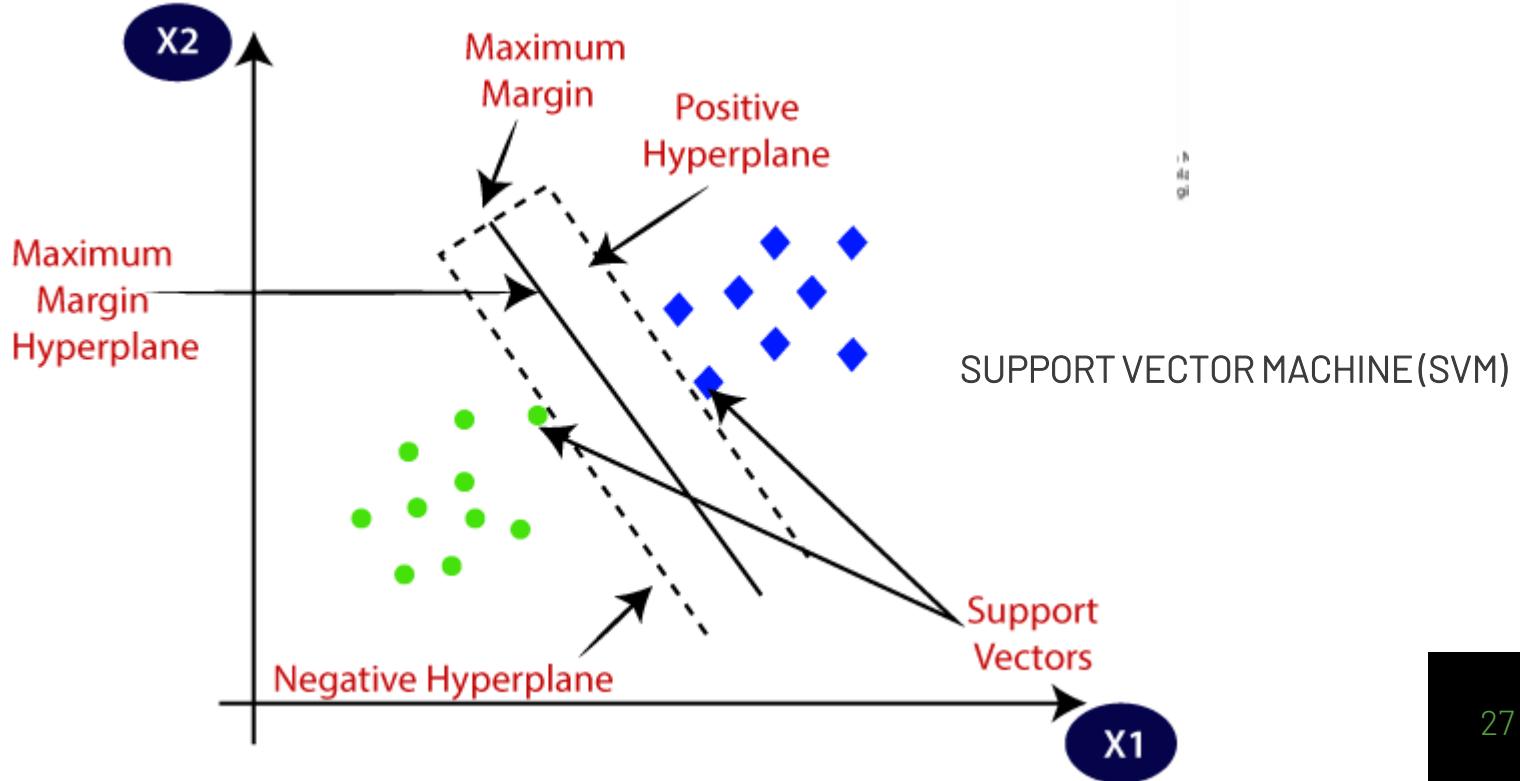


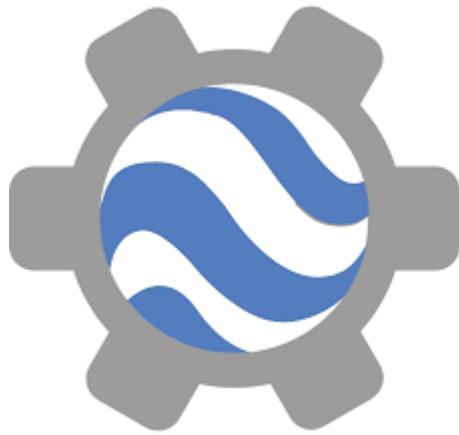
Clasificación Supervisada - Métodos

RANDOM FOREST



Clasificación Supervisada - Métodos





Google Earth Engine (GEE)

Google Earth Engine (o simplemente Earth Engine o **GEE**), es una **plataforma en la nube** para realizar análisis científicos y visualización de datos geoespaciales. Google lo define como «¡La plataforma de **procesamiento geoespacial** basado en la nube más avanzada del mundo!»



Ventajas de usar GEE

- Cargar datos ráster y vectoriales propios (por ejemplo, archivos GeoTIFF o Shapefile) para realizar análisis.
- Acceso a un catálogo de datos, que incluye todo el catálogo Landsat , MODIS, Sentinel, datos de temperatura, entre otros.
- Se puede mostrar los resultados de los análisis en Google Maps o en cualquier otra plataforma de mapas, como ArcGIS o QGIS.
- Earth Engine es gratuito para investigación, educación y para usos sin ánimo de lucro. Para aplicaciones comerciales, se permite la evaluación de Earth Engine.
- **No es necesario descargar las imágenes para hacer el análisis, disminuyendo de esta manera los tiempos de procesamiento.**

Donde podemos usar Google Earth Engine (GEE)?

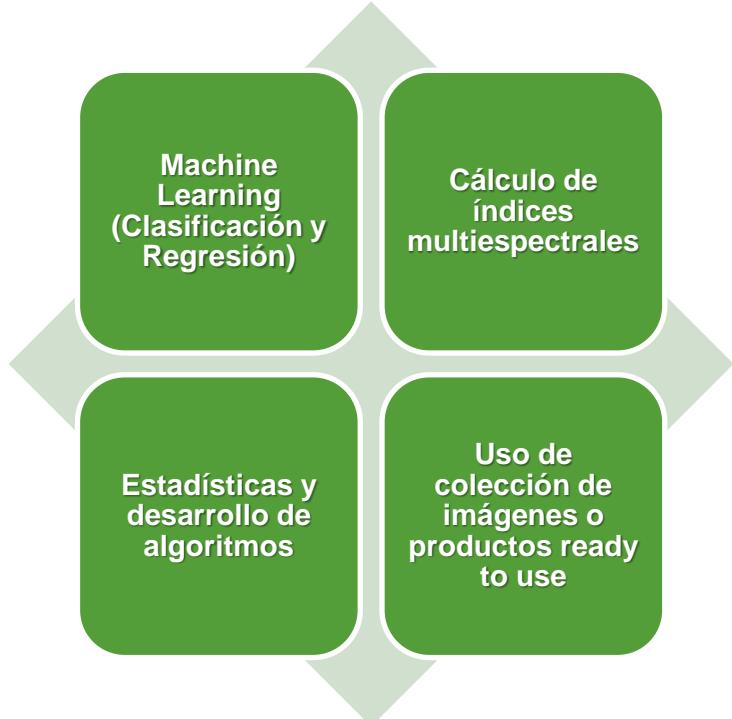
- GEE Javascript

- Earth Engine (EE) Python API

- rgee package R

- Google Earth Engine Complemento - QGIS

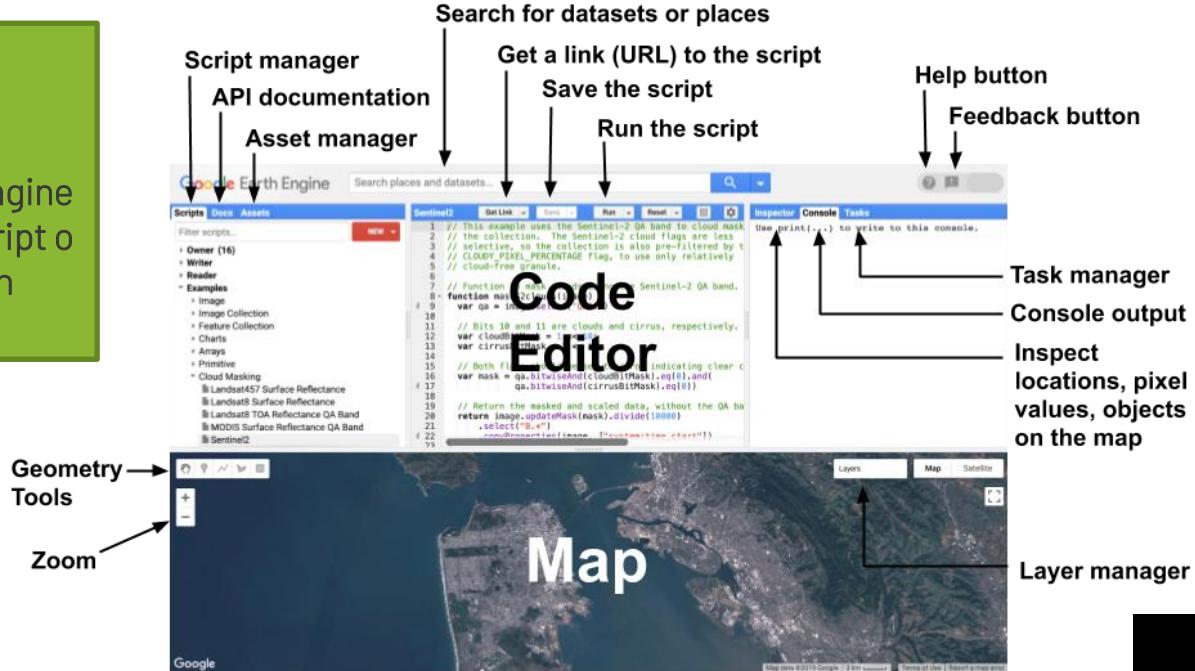
Qué podemos hacer en Google Earth Engine?



Servicio de Geoprocесamiento en la Nube – Google Earth Engine

Qué necesito?

- Cuenta de GMAIL
- Acceder a Google Earth Engine
- Conocimientos en Javascript o Python y Teledetección



Ingresar al Editor de Código GEE

Google Earth Engine

Platform Datasets Noncommercial Commercial Timelapse Case Studies FAQ [Sign Up](#)

Earth Engine for commercial use: now generally available with Google Cloud. [Get more details here](#)

A planetary-scale platform for Earth science data & analysis
Powered by Google's cloud infrastructure

▶ Watch Video

THANK YOU

Welcome to Earth Engine!

Check your inbox at cesarivanalvarezmendoza@gmail.com for information on how to get started.

[TRY THE CODE EDITOR](#) [LOG OUT](#)

01 - Algunos conceptos de Javascript

```
print('Hola amigos');

// Variables
var city = 'Quito';
var country = 'Ecuador';
print(city, country);

var population = 3000000;
print(population);

// Lista
var majorCities = ['Quito', 'Guayaquil', 'Cuenca'];
print(majorCities);

// Diccionario
var cityData = {
  'city': city,
  'population': 8400000,
  'elevation': 930
};
print(cityData);

// Creación de Funciones
var greet = function(name) {
  return 'Hola ' + name;
};
print(greet('Señores Estudiantes'));

// Escritura de Comentarios
```

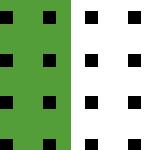
The screenshot shows a browser-based JavaScript editor interface. At the top, there's a toolbar with 'Get Link', 'Save', 'Run', 'Reset', 'Apps', and a settings gear icon. Below the toolbar is a header bar with tabs for 'Inspector', 'Console', and 'Tasks'. A sub-header below the tabs says 'Use print(...) to write to this console.' The main area contains a code editor with numbered lines of JavaScript code. Lines 1 through 10 show basic prints and variable assignments. Line 11 starts a comment block. Lines 12 through 16 define a 'majorCities' array and a 'cityData' object. Lines 17 through 20 define the 'cityData' object with properties 'city', 'population', and 'elevation'. Lines 21 and 22 print the 'cityData' object. Lines 23 and 24 start another comment block. The 'Console' tab is active, displaying the output of the printed values. The output includes 'Hola amigos' (line 1), 'Quito' and 'Ecuador' (line 2), '3000000' (line 3), the 'majorCities' array ('["Quito", "Guayaquil", "Cuenca", "Ambato"]') (line 12), the 'cityData' object ('Object (3 properties)') (line 17), and 'Hola Señores Estudiantes' (line 21). The right side of the interface shows JSON representations of the printed objects.

Output	Type
Hola amigos	JSON
Quito Ecuador	JSON
3000000	JSON
["Quito", "Guayaquil", "Cuenca", "Ambato"]	JSON
Object (3 properties)	JSON
Hola Señores Estudiantes	JSON

- ■ ■ ■ ■
- ■ ■ ■ ■
- ■ ■ ■ ■
- ■ ■ ■ ■
- ■ ■ ■ ■
- ■ ■ ■ ■
- ■ ■ ■ ■

01 - Algunos conceptos de Javascript

JS code editor	Python
<pre>var myFun = function(img) { return img; };</pre>	<pre># Define a function def my_fun(img): return img</pre>
<pre>// Define a variable var myVar = "var"; // Logical operators var match = this.and(that); var match = this.or(that); var match = this.not(that);</pre>	<pre># Define a variable my_var = "var" # Logical operators match = this.And(that) match = this.Or(that) match = this.Not(that)</pre>



02 - Acceso al catálogo de GEE

Earth Engine Data Catalog

Search Español More User profile

Home View all datasets Browse by tags Landsat MODIS Sentinel API Docs

Earth Engine Data Catalog Import

Earth Engine's public data catalog includes a variety of standard Earth science raster datasets. You can import these datasets into your script environment with a single click. You can also upload your own [raster data](#) or vector data for private use or sharing in your scripts.

Looking for another dataset not in Earth Engine yet? Let us know by [suggesting a dataset](#).

Filter list of datasets

Canada AAFC Annual Crop Inventory	Allen Coral Atlas (ACA) - Geomorphic Zonation and Benthic Habitat - v1.0	AHN Netherlands 0.5m DEM, Interpolated	AHN Netherlands 0.5m DEM, Non-Interpolated	AHN Netherlands 0.5m DEM, Raw Samples

02 - Trabajar con colección de imágenes



Landsat 9 OLI-2/TIRS-2

2021–Present

Creación de una función llamada `applyScaleFactors`, donde se requiere de entrada la variable `image`

```
// Applies scaling factors.
function applyScaleFactors(image) {
  var opticalBands = image.select('SR_B*').multiply(0.0000275).add(-0.2);
  var thermalBands = image.select('ST_B.*').multiply(0.00341802).add(149.0);
  return image.addBands(opticalBands, null, true)
    .addBands(thermalBands, null, true);
}
```

Que dará como resultado aplicar esta función

```
var dataset = ee.ImageCollection('LANDSAT/LC09/C02/T1_L2')
  .filterDate('2022-08-01', '2022-10-01');
```

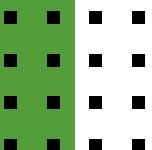
variable
Librería ee
Función para filtrar por fechas
Fecha Inicio
Fecha Fin
Función ImageCollection
Catálogo

Comentario
Función selección de todas las bandas que empiezan con SR_B
Función multiplicar
Función sumar

02 - Trabajar con colección de imágenes

```
dataset = dataset.map(applyScaleFactors);  
var visualization = {  
  bands: ['SR_B4', 'SR_B3', 'SR_B2'],  
  min: 0.0,  
  max: 0.3,  
};  
Map.setCenter(-78, 0.01, 7);  
Map.addLayer(dataset, visualization, 'True Color (432)');
```

A quién se aplicará la función
Función o algoritmo a aplicar
Función map, aplica un algoritmo sobre una colección de imágenes
Se crea una variable de visualización con las 3 bandas con colores en un mínimo y máximo
Variable Map por defecto (Mapa a visualizar)
Coordenadas de donde vamos a centrar la imagen y su zoom para visualización
Función addLayer, permite agregar al mapa a visualizar las características requeridas



03 - Filtrar imagen por coordenadas y máscaras

```
var geometry = ee.Geometry.Point([-78.5, 0.02]) ← Variable vectorial tipo geometría de punto  
/ Function to mask clouds using the Sentinel-2 QA band  
  
function maskS2clouds(image) { ← Función para enmascarar nubes y sombras Sentinel-2 usando la banda QA  
  var qa = image.select('QA60');  
  
  // Bits 10 and 11 are clouds and cirrus, respectively.  
  var cloudBitMask = 1 << 10;  
  var cirrusBitMask = 1 << 11;  
  
  // Both flags should be set to zero, indicating clear conditions.  
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0)  
    .and(qa.bitwiseAnd(cirrusBitMask).eq(0));  
  
  return image.updateMask(mask).divide(10000);  
}
```

Que retornará la función de enmascarar nubes

03 - Filtrar imagen por coordenadas y máscaras

```
var dataset = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')
    .filterDate('2020-01-01', '2020-03-30')
    // Pre-filter to get less cloudy granules.
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',20))

Filtrar con la función
maskS2clouds → .map(maskS2clouds)
    .filter(ee.Filter.bounds(geometry));           ↗
                                                ↗
Filtrar con la capa Cloudy Pixel

var visualization = {
    min: 0.0,
    max: 0.3,
    bands: ['B4', 'B3', 'B2'],
    //bands: ['B5', 'B4', 'B3'],
};

Filtrar con geometría
                                                ↗
                                                ↗
Se aplica la función media, para que nos genere
una imagen con la media de pixeles de las
fechas filtradas
                                                ↗
■ ■ ■ ■ ■ Map.centerObject(geometry, 10)
■ ■ ■ ■ ■ //Map.addLayer(dataset.mean(), visualization);
■ ■ ■ ■ ■ Map.addLayer(dataset.mean(), visualization, 'RGB');
```

03 - Filtrar imagen por coordenadas y máscaras

Google Earth Engine

Search places and datasets...

Scripts Docs Assets

- Aplicación
- borrador
- Archive
- No accessible repositories. Click Refresh to check again.
- Examples
 - Image
 - Image Collection
 - Feature Collection
 - Charts
 - Arrays
 - Primitive
 - Cloud Masking
 - Code Editor

3 - Filtrar imagen Sentinel-2 *

```
15 // Both flags should be set to zero, indicating clear conditions.
16 var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
17 .and(qa.bitwiseAnd(cirrusBitMask).eq(0));
18
19 return image.updateMask(mask).divide(10000);
20 }
21
22 var dataset = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')
23 .filterDate('2020-01-01', '2020-03-30')
24 // Pre-filter to get less cloudy granules.
25 .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',20))
26 .map(mask2clouds)
27 .filter(ee.Filter.bounds(geometry));
28
29 var visualization = {
30   min: 0.0,
```

Get Link Save Run Reset Apps

Inspector Console Tasks

Point (-78.25381, 0.14379) at 19m/px

Pixels Layer 1: Image (23 bands)

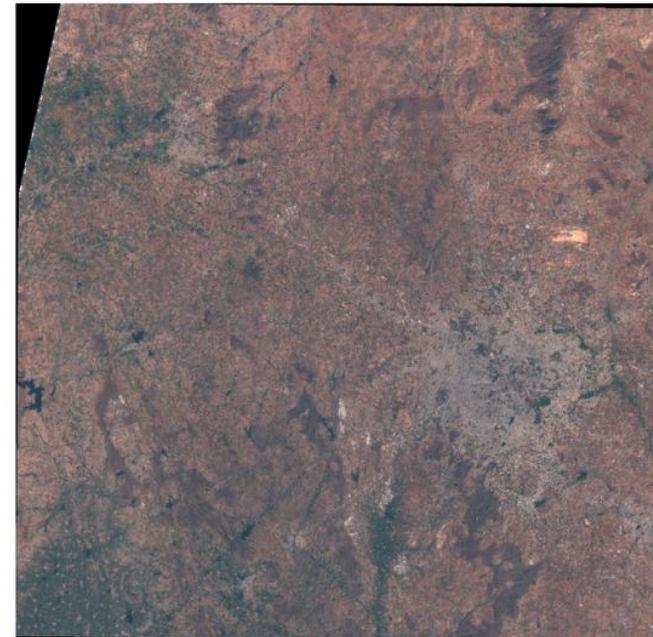
Layers Map Satélite

42

04 - Creación de mosaicos y composiciones



MOSAIC



MEDIAN COMPOSITE

Mosaic vs. Composite

04 - Creación de mosaicos y composiciones

```
var geometry = ee.Geometry.Point([-78.5, 0.02])  
  
var filtered = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')  
    .filterDate('2020-01-01', '2022-01-01')  
    // Pre-filter to get less cloudy granules.  
    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE',20))  
    .filter(ee.Filter.bounds(geometry));  
  
var mosaic = filtered.mosaic() //Crea un mosaico con los últimos pixeles de la  
colección encontrados ← Función mosaico  
  
var medianComposite = filtered.median(); //Genera una mediana de pixeles entre  
fechas ↑ Función mediana
```

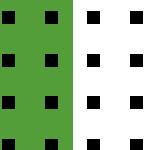
- ▪ ▪
- ▪ ▪
- ▪ ▪
- ▪ ▪
- ▪ ▪
- ▪ ▪
- ▪ ▪
- ▪ ▪

04 - Creación de mosaicos y composiciones

```
var rgbVis = {  
    min: 0.0,  
    max: 3000,  
    bands: ['B4', 'B3', 'B2'],  
};  
  
// set styling  
Map.centerObject(geometry, 15) ← Agrega cada elemento  
Map.addLayer(filtered, rgbVis, 'Colección original'); como una capa en el  
Map.addLayer(mosaic, rgbVis, 'Mosaico'); visualizador  
Map.addLayer(medianComposite, rgbVis, 'Composición mediana')  
Map.addLayer(geometry,{color: 'red', pointSize: 30, pointShape: 'circle'}, 'Punto');
```

Propiedades del Vector

Cambiar nombre de capa



04 - Creación de mosaicos y composiciones

Script Editor Interface:

- Top Bar: Scripts, Docs, Assets, NEW, Save, Run, Reset, Apps, Inspector, Console, Tasks.
- Left Sidebar: Filter scripts..., Owner (4) (users/cesarivanalvarezmendoza/CIAT, users/cesarivanalvarezmendoza/prueba).
- Code Area (Title: 4 - Mosaico y Mediana Sentinel-2):

```
i 9 var mosaic = filtered.mosaic() //Crea un mosaico con los últimos pixeles de la colección encontrados
10
11 var medianComposite = filtered.median(); //Genera una mediana de pixeles entre fechas
12
13 var rgbVis = {
14   min: 0.0,
15   max: 3000,
16   bands: ['B4', 'B3', 'B2'],
17 };
18
19 // set styling
i 20 Map.centerObject(geometry, 15)
21 Map.addLayer(filtered, rgbVis, 'Colección original');
22 Map.addLayer(mosaic, rgbVis, 'Mosaico');
i 23 Map.addLayer(medianComposite, rgbVis, 'Composición mediana')
24 Map.addLayer(geometry, {color: 'red', pointSize: 30, pointShape: 'circle'}, 'Punto');
```
- Map View: Satellite view showing a forested area with a red dot indicating the geometry center.

05 -Colecciones Vectoriales (Feature Collections)

```
var country = 'BL'; //Choose the country initials [Two-letter FIPS country code]  
https://en.wikipedia.org/wiki/List\_of\_FIPS\_country\_codes
```

```
// Load country features from Large Scale International Boundary (LSIB) dataset.
```

```
var countries = ee.FeatureCollection('USDOS/LSIB_SIMPLE/2017');  
var table = countries.filter(ee.Filter.eq('country_co', ee.String(country)));
```

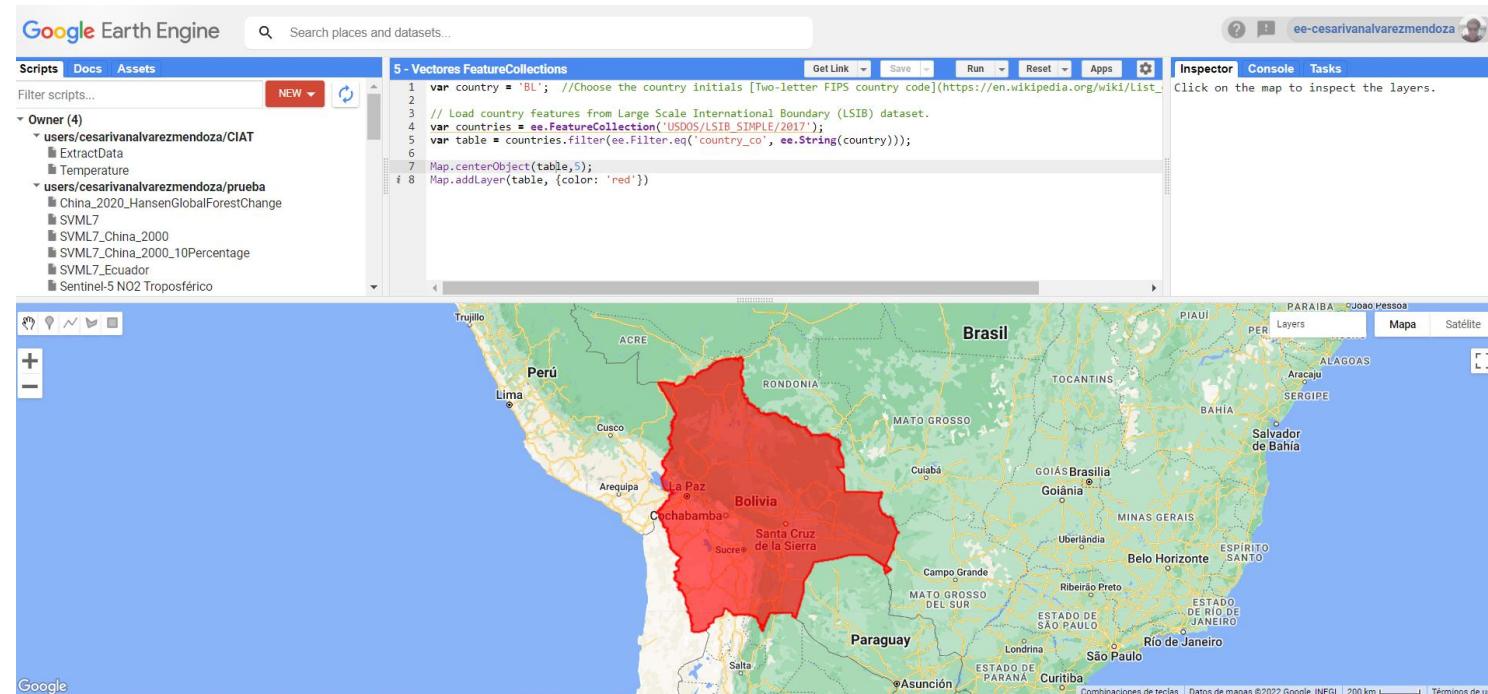
```
Map.centerObject(table,5);
```

```
Map.addLayer(table, {color: 'red'})
```

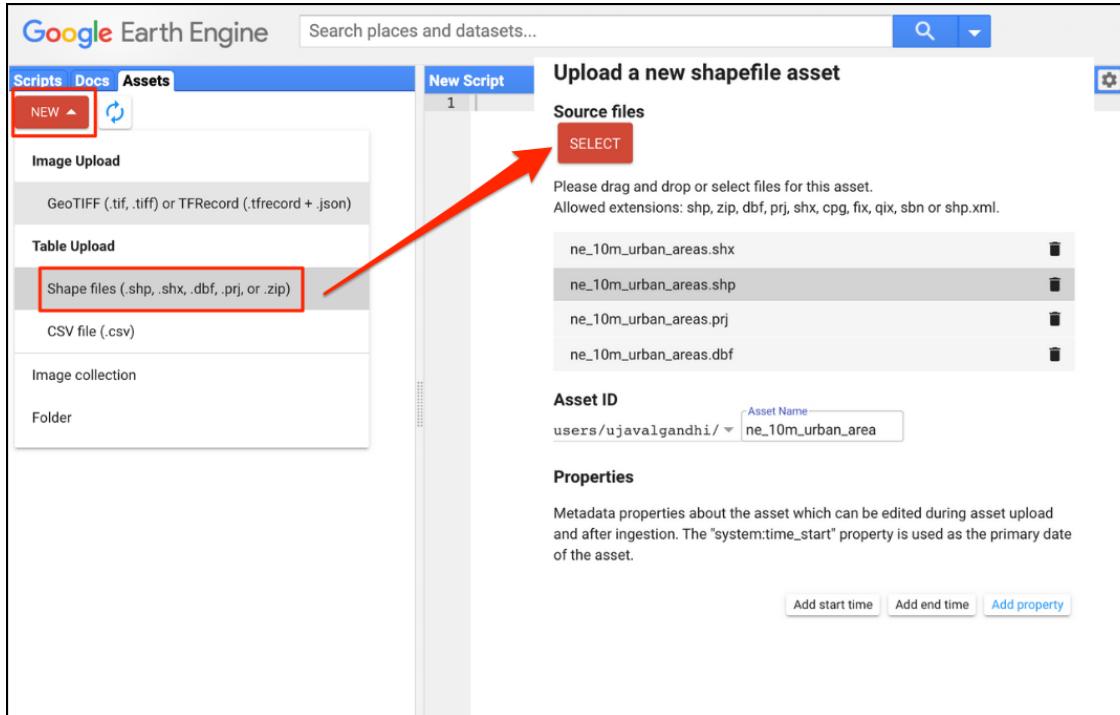
Agregar al mapa en
color rojo

Feature collection
disponible en GEE de
límite de países

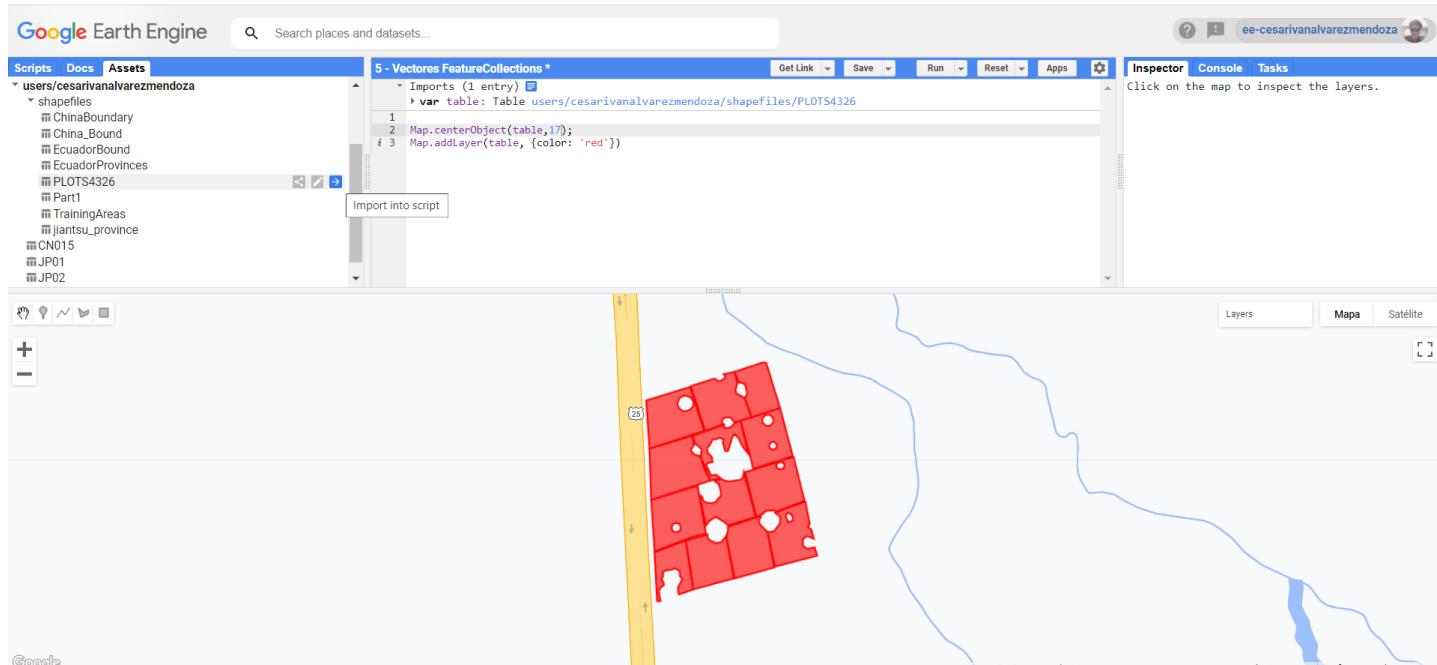
05 - Colecciones Vectoriales (Feature Collections)



06 - Importar datos (Shapefiles, Geotiff, CSV)



06 - Importar datos (Shapefiles, Geotiff, CSV)



07 - Clip Colección de Imágenes

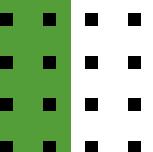
```
var s2 = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')
var urban = ee.FeatureCollection("users/cesarivanalvarezmendoza/shapefiles/PLOTS4326")

var geometry = urban.geometry()

var rgbVis = {
  min: 0.0,
  max: 3000,
  bands: ['B4', 'B3', 'B2'],
};

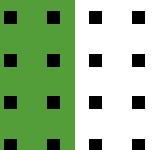
var filtered = s2.filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 30))
  .filter(ee.Filter.date('2021-01-01', '2022-01-01'))
  .filter(ee.Filter.bounds(geometry))
```

Generar la variable
del polígono desde
el featurecollection



07 - Cortar (Clip) Colección de Imágenes

```
var image = filtered.median();
var clipped = image.clip(geometry) ← Función cortar
Map.centerObject(urban,17)
Map.addLayer(clipped, rgbVis, 'Clipped')
```



07 - Cortar (Clip) Colección de Imágenes

Google Earth Engine

Search places and datasets...

Scripts Docs Assets

7 - Clip Imagen Sentinel-2

Get Link Save Run Reset Apps

Inspector Console Tasks

Use print(...) to write to this console.

```
5 var rgbVis = {  
6   min: 0.0,  
7   max: 3000,  
8   bands: ['B4', 'B3', 'B2'],  
9 };  
10  
11 var filtered = s2.filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 30))  
12   .filter(ee.Filter.date('2021-01-01', '2022-01-01'))  
13   .filter(ee.Filter.bounds(geometry))  
14  
15 var image = filtered.median();  
16  
17 var clipped = image.clip(geometry)  
18  
19 Map.centerObject(urban, 17)  
20 Map.addLayer(clipped, rgbVis, 'Clipped')
```

Writer

No accessible repositories. Click Refresh to check again.

Layers Map Satélite

The screenshot displays the Google Earth Engine (GEE) web interface. On the left, the 'Scripts' panel lists several projects and scripts, including 'Hansen treecover2000 Export by Country', 'LALModis_Extraction_by_country', 'S2', 'exportSentinel2', and a script named '7 - Clip Imagen Sentinel-2'. The main workspace shows a map of a study area with a complex blue boundary. A yellow rectangle highlights a specific region on the left side of the map. A legend in the bottom-left corner shows a white circle with a black dot inside, labeled '25'. The code editor on the right contains a JavaScript script for clipping a Sentinel-2 image collection. The script uses filters to select images with less than 30% cloudy pixels between January 1, 2021, and January 1, 2022, and then clips them against a specified geometry. It then centers the map on an urban area and adds the clipped image layer with a median filter applied. The 'Inspector' tab is active at the top, and the 'Console' tab is visible below it.

08 - Exportar imágenes

```
var country = 'BL'; //Choose the country initials [Two-letter FIPS country code]  
https://en.wikipedia.org/wiki/List\_of\_FIPS\_country\_codes
```

```
// Load country features from Large Scale International Boundary (LSIB) dataset.  
var countries = ee.FeatureCollection('USDOS/LSIB_SIMPLE/2017');  
var table = countries.filter(ee.Filter.eq('country_co', ee.String(country)));
```

```
// Make the clip boundary.
```

```
var clipToCol = function(image){  
    return image.clip(table);  
};
```

Función para
realizar clip

```
var collection = ee.ImageCollection('COPERNICUS/S5P/OFFL/L3_NO2')  
    .select('NO2_column_number_density')  
    .filterBounds(table)  
    .filterDate('2019-03-16', '2019-03-31')  
    .map(clipToCol);
```

08 - Exportar imágenes

```
var band_viz = {  
  min: 0,  
  max: 0.00009,  
  palette: ['green','yellow', 'red']  
};  
  
Map.addLayer(collection.median(), band_viz, 'S5P N02');  
Map.addLayer(table);  
Map.centerObject(table,5);  
print(collection.size()) ← Ver resolución de  
la imagen  
  
// Export the image, specifying scale and region. ← Exportar a Google  
Export.image.toDrive({  
  image: collection.median(), ← Drive como  
  description: 'Bolivia_2019',  
  scale: 1000,  
  region: table  
}); ← Exportar la  
mediana del image  
collection
```

08 - Exportar imágenes

Google Earth Engine

Search places and datasets...

Script Docs Assets

7 - Clip Imagen Sentinel-2 *

Get Link Save Run Reset Apps

Inspector Console Tasks

Manage tasks.

Search or cancel multiple tasks in the Task Manager.

UNSUBMITTED TASKS

Bolivia_2019 RUN

Sentinel-2-CopernicusS2_SR_432bands_sk...

Script Editor

20 max: 0.0009,
21 palette: ['green','yellow', 'red']
22 };
23
24 Map.addLayer(collection.median(), band_viz, 'SSP N02');
25 Map.addLayer(table);
26 Map.centerObject(table,5);
27
28 // Export the image, specifying scale and region.
29 Export.image.toDrive({
30 image: collection,
31 description: 'Bolivia_2019',
32 scale: 1000,
33 region: table
34});

Mapa Satélite

Brasil

Perú

ACRE RONDÔNIA MATO GROSSO MATO GROSSO DO SUL PARAGUAY

PIAUI CEARÁ PIAUÍ ALAGOAS SERGIPE

BAHIA SALVADOR DE BAÍA

GOIÁS GOIÂNIA

MINAS GERAIS

ESTADO DE RIO DE JANEIRO

Ribeirão Preto Belo Horizonte São Paulo Rio de Janeiro

Londrina Curitiba

Trujillo Acre Cusco Arequipa Lima Cusco Arequipa

Cuiabá Campo Grande

Brasília

Salvador de Bahia

Aracaju

Mapa Satélite

Layers

ee-cesarivanalvarezmendoza

7m

```
20 max: 0.0009,  
21 palette: ['green','yellow', 'red']  
22 };  
23  
24 Map.addLayer(collection.median(), band_viz, 'SSP N02');  
25 Map.addLayer(table);  
26 Map.centerObject(table,5);  
27  
28 // Export the image, specifying scale and region.  
29 Export.image.toDrive({  
30 image: collection,  
31 description: 'Bolivia_2019',  
32 scale: 1000,  
33 region: table  
34});
```

08 - Exportar imágenes

Task: Initiate image export

Task name (no spaces) *
Bolivia_2019

Coordinate Reference System (CRS)
EPSG:3857

Scale (m/px)
1000

DRIVE **CLOUD STORAGE** **EE ASSET**

Drive folder
Drive folder name or blank for root

Filename *
Bolivia_2019

File format *
GEO_TIFF

CANCEL **RUN**

Google Earth Engine

Scripts Docs Assets

7 - Clip Imagen Sentinel-2 *

```
22 );
23 Map.addLayer(collection.median(), band_viz, 'SSP_N02');
24 Map.addLayer(table);
25 Map.centerObject(table,5);
26 print(collection.size());
27
28
29
30 // Export the image, specifying scale and region.
31 Export.image.toDrive({
32   image: collection.median(),
33   description: 'Bolivia_2019',
34   scale: 10,
35   region: table
36 });
37
```

Inspector Console Tasks

Manage tasks. Search or cancel multiple tasks in the Task Manager.

Bolivia_2019

ID: BNZOJXKMLVYA6V3ULJRXP4

Phase: Running on server

Runtime: 2s (started 2022-09-23 17:11:23-0500)

Attempted 1 time



09 – Cálculo de índices - NDVI

- ```
■ ■ ■ ■ Map.addLayer(composite, rgbVis, 'RGB Ecuador')
```

## Generar una imagen compuesta por la mediana

# 09 – Cálculo de índices - NDVI

```
// Write a function that computes NDVI for an image and adds it as a band
function addNDVI(image) {
 var ndvi = image.normalizedDifference(['B8', 'B4']).rename('ndvi');
 return image.addBands(ndvi);
}
```



Función NDVI

```
// Map the function over the collection
var withNdvi = filtered.map(addNDVI);
var composite = withNdvi.median()
var ndviComposite = composite.select('ndvi').clip(geometry)
```

```
var palette = [
 'FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718',
 '74A901', '66A000', '529400', '3E8601', '207401', '056201',
 '004C00', '023B01', '012E01', '011D01', '011301'];
```

```
var ndviVis = {min:0, max:1, palette: palette }
Map.addLayer(ndviComposite, ndviVis, 'NDVI')
```

```
Map.centerObject(table,7);
Map.addLayer(table, {color: 'red'}, 'Limite Ecuador')
```

## — Función NDVI

# Paleta de colores NDVI

# 09 - Cálculo de índices - NDVI

Google Earth Engine Search places and datasets... ee-cesarivanalvarezmendoza

Scripts Docs Assets

- ForestGainLossbyCountryUsingShapefiles 200...
- ForestLossbyYear
- HansenTreeCover2000 Export by Country
- LAI\_Modis\_Extraction\_by\_country
- S2
- exportSentinel2
- users/cesarivanalvarezmendoza/UASB**
  - 1 - Introducción JavaScript
  - 2 - Visualización Imagen Landsat 9
  - 3 - Filtrar Imagen Sentinel-2
  - 4 - Mosaico y Mediana Sentinel-2

9 - NDVI \*

```
23 var withNdvi = filtered.map(addNDVI);
i 24 var composite = withNdvi.median()
i 25 var ndviComposite = composite.select('ndvi').clip(geometry)
26
27 var palette =
28 'FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718',
29 '74A901', '66A000', '529400', '3BEB01', '207401', '056201',
30 '004C00', '023B01', '012E01', '011D01', '011301'];
31
32 var ndviVis = {min:0, max:1, palette: palette }
33 Map.addLayer(ndviComposite, ndviVis, 'NDVI')
34
35 Map.centerObject(table,:);
36 Map.addLayer(table, {color: 'red'}, 'Limite Ecuador')
```

Inspector Console Tasks

Use print(...) to write to this console.

The screenshot shows the Google Earth Engine interface. On the left, there's a sidebar with 'Scripts', 'Docs', and 'Assets'. Under 'Assets', there's a folder for 'users/cesarivanalvarezmendoza/UASB' containing four scripts: '1 - Introducción JavaScript', '2 - Visualización Imagen Landsat 9', '3 - Filtrar Imagen Sentinel-2', and '4 - Mosaico y Mediana Sentinel-2'. The main area has tabs for '9 - NDVI \*', 'Get Link', 'Save', 'Run', 'Reset', 'Apps', and a 'Layers' button. Below these are code snippets for calculating NDVI and visualizing it. To the right is an 'Inspector' panel with 'Console' and 'Tasks' tabs, and a note about using print(). At the bottom is a map of Ecuador with a green NDVI visualization and a red polygon representing the 'Limite Ecuador' boundary. The map also shows various place names like Pedernales, Cuenca, Manabí, Salinas, Machala, La Chorrera, El Encanto, El Estrecho, Indiana, and Pacoa. There are also buttons for zooming in and out, and a 'Mapa' button.

# 10 – Reductores (Reducers)

```
//Forest loss between 2000 to 2020
var year = 20; //Choose the year between 2000 (00) to 2020 (20)
var country = 'EC'; //Choose the country initials [Two-letter FIPS country
code](https://en.wikipedia.org/wiki/List_of_FIPS_country_codes

// Load country features from Large Scale International Boundary (LSIB) dataset.
var countries = ee.FeatureCollection('USDOS/LSIB_SIMPLE/2017');
var selected = countries.filter(ee.Filter.eq('country_co', ee.String(country)));

// Make the clip boundary.
var clipToCol = function(image){
 return image.clip(selected);
};

var gfc2020 = ee.Image('UMD/hansen/global_forest_change_2020_v1_8');

// Canopy loss during the selected year
var loss = gfc2020.select(['lossyear']).eq(year).clip(selected) //ClipGeometry can any geometry or feature or
shapfile
```

Catalog Global Forest

Filtrar por año

# 10 – Reductores (Reducers)

```
// Get the forest loss in square kilometers.
var arealmage = loss.multiply(ee.Image.pixelArea()).divide(1000000);

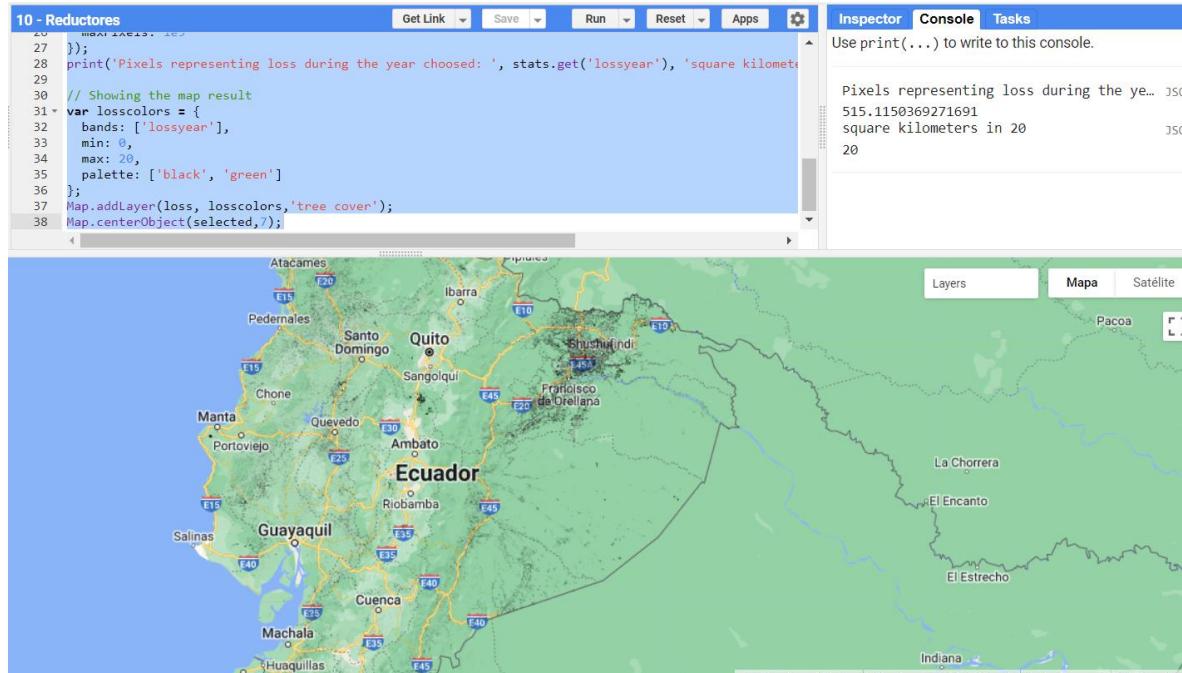
var stats = arealmage.reduceRegion({
 reducer: ee.Reducer.sum(),
 geometry: selected,
 scale: 30,
 maxPixels: 1e9
});
print('Pixels representing loss during the year choosed: ', stats.get('lossyear'), 'square kilometers in 20', year);

// Showing the map result
var losscolors = {
 bands: ['lossyear'],
 min: 0,
 max: 20,
 palette: ['black', 'green']
};
Map.addLayer(loss, losscolors,'tree cover');
Map.centerObject(selected,7);
```

Suma los píxeles dentro del polígono de featurecollection

Función get

# 10 – Reductores (Reducers)



# 11 - Serie de Tiempo

```
var s2 = ee.ImageCollection("COPERNICUS/S2_SR_HARMONIZED");
var geometry = ee.Geometry.Polygon([
 [82.60642647743225, 27.16350437805251],
 [82.60984897613525, 27.1618529901377],
 [82.61088967323303, 27.163695288375266], ← Creando un polígono de estudio o ROI
 [82.60757446289062, 27.16517483230927]
]);
Map.addLayer(geometry, {color: 'red'}, 'Farm')
Map.centerObject(geometry)
var rgbVis = {min: 0.0, max: 3000, bands: ['B4', 'B3', 'B2']};

var filtered = s2.filter(ee.Filter.date('2017-01-01', '2022-01-01'))
 .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 30))
 .filter(ee.Filter.bounds(geometry))

// Write a function for Cloud masking
function maskS2clouds(image) {
 var qa = image.select('QA60')
 var cloudBitMask = 1 << 10;
 var cirrusBitMask = 1 << 11;
 var mask = qa.bitwiseAnd(cloudBitMask).eq(0).and(
 qa.bitwiseAnd(cirrusBitMask).eq(0))
 return image.updateMask(mask)//.divide(10000)
 .select("B.*")
 .copyProperties(image, ["system:time_start"])
}
```

# 11 - Serie de Tiempo

```
var filtered = filtered.map(maskS2clouds)
// Write a function that computes NDVI for an image and adds it as a band
function addNDVI(image) {
 var ndvi = image.normalizedDifference(['B8', 'B4']).rename('ndvi');
 return image.addBands(ndvi);
}

// Map the function over the collection
var withNdvi = filtered.map(addNDVI);

// Display a time-series chart
var chart = ui.Chart.image.series({
 imageCollection: withNdvi.select('ndvi'),
 region: geometry,
 reducer: ee.Reducer.median(),
 scale: 10
}).setOptions({
 lineWidth: 1,
 title: 'Serie de Tiempo NDVI',
 interpolateNulls: true,
 vAxis: {title: 'NDVI'},
 hAxis: {title: '', format: 'YYYY-MMM'}
})
print(chart);
```

← Función NDVI

← Gráfico Serie de Tiempo

# 11 – Serie de Tiempo

```
var withNDVI = withNdvi.select('ndvi')
var withNDVI = ee.ImageCollection(withNDVI).toBands();

// Get Zonal Statistics
var reduced = withNDVI.reduceRegions({
 collection: geometry,
 reducer: ee.Reducer.mean(),
 scale: 100,
});
print(reduced);

// the resulting mean is a FeatureCollection
// so you can export it as a table
Export.table.toDrive({
 collection: reduced,
 description: 'NDVIpolygons',
 fileNamePrefix: 'NDVIpolygons',
 fileFormat: 'CSV'
})
```

← Cálculo de estadísticas NDVI

← Exportar a CSV

- ■ ■
- ■ ■
- ■ ■
- ■ ■
- ■ ■
- ■ ■
- ■ ■

# 11 - Serie de Tiempo

Google Earth Engine Search places and datasets...

Scripts Docs Assets Get Link Save Run Reset Apps

11 - Series de Tiempo ee-cesarivanalvarezmendoza

Inspector Console Tasks

Use print(...) to write to this console.

Serie de Tiempo NDVI

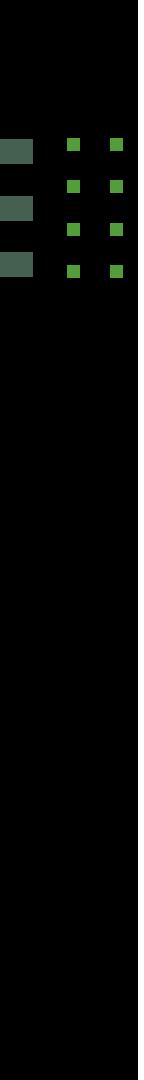
NDVI

2019-Jan 2020-Jan 2021-Jan 2021-Sep

10  
20  
30  
40  
50  
60  
70  
80  
90  
100  
110  
120  
130  
140  
150  
160  
170  
180  
190  
200  
210  
220  
230  
240  
250  
260  
270  
280  
290  
300  
310  
320  
330  
340  
350  
360  
370  
380  
390  
400

var cloudBitMask = 1 << 10;  
var cirrusBitMask = 1 << 11;  
var mask = qa.bitwiseAnd(cloudBitMask).eq(0).and(  
 qa.bitwiseAnd(cirrusBitMask).eq(0))  
return image.updateMask(mask).divide(10000)  
 .select("B.\*")  
 .copyProperties(image, ["system:time\_start"]);  
}  
  
var filtered = filtered.map(maskS2clouds);  
// Write a function that computes NDVI for an image and adds it as a band  
function addNDVI(image){  
 var ndvi = image.normalizedDifference(['B8', 'B4']).rename('ndvi');  
 return image.addBands(ndvi);  
}  
  
// Map the function over the collection  
var withNdvi = filtered.map(addNDVI);  
  
// Display a time-series chart  
var chart = ui.Chart.image.series({

Layers Mapa Satélite



César Iván Alvarez Mendoza  
[cesarivanalvarezmendoza@gmail.com](mailto:cesarivanalvarezmendoza@gmail.com)

**Gracias por  
su atención!**