

INTRODUCCIÓN AL DESARROLLO DE HERRAMIENTAS DE MONITOREO FORESTAL EN GOOGLE EARTH ENGINE

ERITH ALEXANDER MUÑOZ

The Food and Agriculture Organization

ONU-REDD/FAO

erith.munozrios@fao.org

28 de marzo de 2017

Índice

1. INTRODUCCIÓN	2
2. GENERACIÓN DE MOSAICOS	2
2.1. Mosaico Landsat 7	2
2.2. Generación de Mosaicos L7 multitemporales	9
2.3. Mosaico Landsat 8	10
2.4. Generación de Mosaicos L8 multitemporales	12
3. CÁLCULO DE NDVI	13
4. CLASIFICACIÓN	15

1. INTRODUCCIÓN

Partiendo desde la primicia de que el lector posee conocimientos básicos sobre Google Earth Engine (GEE), sus componentes, uso, ventajas y desventajas, en este documento se presenta a modo de guía, una breve introducción al uso de GEE para el desarrollo de herramientas computacionales para el monitoreo forestal dentro del contexto de REDD+. Desde este enfoque, el objetivo de este manual está limitado a mostrar las posibilidades de desarrollo que presta GEE, en detrimento de toda la potencialidad que la plataforma posee para usuarios enfocados a la obtención de productos especializados. Es decir, aunque actualmente existen scripts avanzados para la generación de productos cartográficos en el sector forestal, estos scripts generalmente resultan complejos para usuarios nuevos de GEE. Por este motivo, este manual pretende servir de acompañamiento durante la generación de un primer script orientado a realizar procedimientos básicos dentro de la tarea cotidiana de monitoreo forestal.

La experiencia enseña, que una de las primeras dificultades a vencer por parte de un usuario nuevo que intenta escribir un primer código de GEE, es definir como empezar. En respuesta a esta dificultad, se propone empezar de la forma más natural y rápida, que es utilizar los códigos de ejemplo disponibles en GEE, para adaptarlos a la solución de problemas particulares.

2. GENERACIÓN DE MOSAICOS

2.1. Mosaico Landsat 7

El primer ejercicio a realizar es el de conformar un mosaico de Landsat 7 (L7) para El Salvador a partir del uso del script ejemplo [Clipped Composite](#). Para esto, es importante identificar dentro de este script, las líneas de código que serían útiles para generar el mosaico sobre El Salvador. La figura 1 muestra el ambiente de trabajo de la sección de código de GEE, y además se muestra la ejecución del script [Clipped Composite](#).

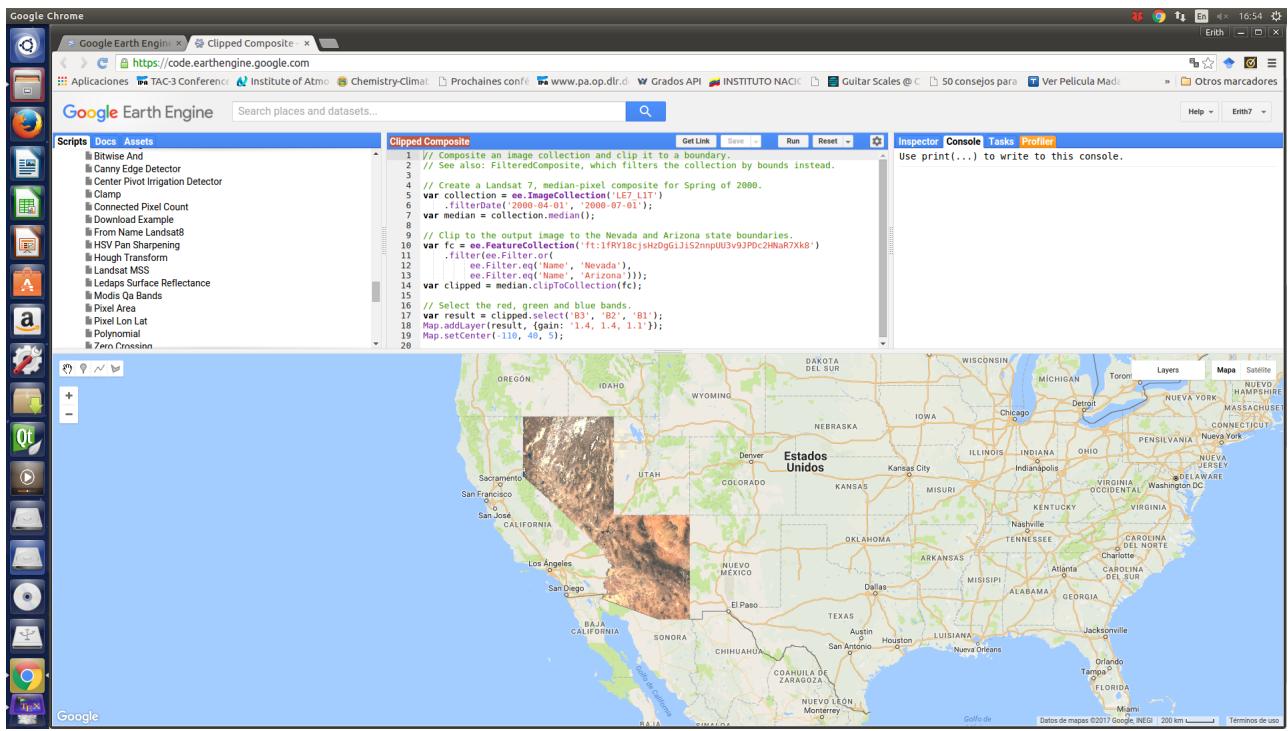


Figura 1: Ejecución del script **Clipped Composite** en GEE

A simple vista se aprecia que este código genera un composito, que es un mosaico conformado a partir de un criterio de selección de pixeles, entre varias o muchas imágenes dentro de un período específico. Este mosaico es construido tomando como límites geográficos los bordes administrativos de Nevada y Las Vegas.

Con la finalidad de crear un script propio para crear un composito de L7, es necesario tomar los comandos de las líneas 5, 7, 17, 18 y 19, del código **Clipped Composite**, que se muestra en la figura 2. En la figura 3¹ se muestra el nuevo código generado, a partir del ejemplo. Cada una de las lineas esta precedida por un comentario que describe el objetivo del comando, y un número para tener como referencia el orden de los diferentes pasos. En el código original el periodo de tiempo para búsqueda de L7 estaba limitado al año 2000, mientras que en el código nuevo, se puede observar en el comando de la línea 6, que la búsqueda ha sido extendida hasta 2002. Es importante mencionar, que en la línea 17 del nuevo código se cambió la variable **clipped** por **median** para garantizar la congruencia del código².

¹Para realizar este código, se creó una nuevo repositorio en el panel izquierdo de la interfaz de GEE. Se creó un archivo nuevo con el nombre **MosaicoL7.js**, se copió todo el contenido del código del ejemplo en el nuevo archivo, se eliminaron las líneas que no se usarían, y se realizó los comentarios.

²Esta es una consecuencia de haber eliminado el paso intermedio donde se asignan los límites administrativos de Venada y Las Vegas para definir la región de interés para la búsqueda de las L7.

The screenshot shows the Google Earth Engine Code Editor interface. On the left, there's a sidebar with icons for Scripts, Docs, and Assets. Under Scripts, there are several files listed, including 'Topographic Correction of L...', 'UntitledFile', 'UntitledFile2', 'WEEK04', 'calc_area_loss_with_tcd', 'charts_test', 'emigrant.js', 'forest_loss_AEB', 'improved Cosine vs (copy).n...', 'improved Cosine vs. no TC N...', 'lada2015', 'landsat script_IND', 'mosaicoGrid_Landsats', 'pbm', 'percentile_reducer_nyc_pan...', '+ New folder', '+ New file', and 'Repositorio_FAO_LAC' which contains 'Argentina', 'Bolivia', and 'cobertura'. The main panel displays the code for 'Clipped Composite' with a map of the United States at the bottom. The code uses Earth Engine API functions like ee.ImageCollection, ee.Filter.or, ee.Filter.eq, median, clipToCollection, select, addLayer, and setCenter.

```

1 // Composite an image collection and clip it to a boundary.
2 // See also: FilteredComposite, which filters the collection by bounds instead.
3
4 // Create a Landsat 7, median-pixel composite for Spring of 2000.
5 var collection = ee.ImageCollection('LE7_L1T')
6   .filterDate('2000-04-01', '2000-07-01');
7 var median = collection.median();
8
9 // Clip to the output image to the Nevada and Arizona state boundaries.
10 var fc = ee.FeatureCollection('ft:1fRY18cjshDgGijs2nnpUU3v9JPc2HNaR7Xk8')
11   .filter(ee.Filter.or(
12     ee.Filter.eq('Name', 'Nevada'),
13     ee.Filter.eq('Name', 'Arizona')));
14 var clipped = median.clipToCollection(fc);
15
16 // Select the red, green and blue bands.
17 var result = clipped.select('B3', 'B2', 'B1');
18 Map.addLayer(result, {gain: '1.4, 1.4, 1.1'});
19 Map.setCenter(-110, 40, 5);
20

```

Figura 2: Código del ejemplo Clipped Composite en GEE

The screenshot shows the Google Earth Engine Code Editor interface with a new script named 'MosaicoL7.js'. The sidebar shows a different set of files under 'Scripts', including 'ElSalvador' (with 'Mosaico.js', 'Mosaico8.js', 'Mosaico8_ndvi.js'), 'RD' (with 'MosaicoRD.js'), 'Repository_Earth' (with 'Landsat' folder containing 'Landsat_Mosaic.js', 'Landsat_Mosaic_RD.js', 'Mosaico1.js'), 'Webinar_ES' (with 'MosaicoL7.js'), and 'Ecuador'. The main panel displays the 'MosaicoL7.js' code, which is a modified version of the 'Clipped Composite' code, specifically tailored for generating a global L7 mosaic. The code includes comments in Spanish explaining the steps: 'Generación de Mosaico L7 para El Salvador', 'Criterio para la búsqueda de imágenes', 'Criterio para la asignación de pixeles en el mosaico', 'Selección de Bandas', 'Configuración de ganancia por Bandas', and 'Definición del centro del mapa para la visualización'. A map of the central United States is visible at the bottom.

```

1 //Generación de Mosaico L7 para El Salvador
2
3 =====
4 // 1. Criterio para la búsqueda de imágenes
5 =====
6 var collection = ee.ImageCollection('LE7_L1T')
7   .filterDate('2000-04-01', '2002-07-01');
8
9 =====
10 // 2. Criterio para la asignación de pixeles en el mosaico
11 =====
12 var median = collection.median();
13
14 =====
15 // 3. Selección de Bandas
16 =====
17 var result = median.select('B3', 'B2', 'B1');
18
19 =====
20 // 4. Configuración de ganancia por Bandas
21 =====
22 Map.addLayer(result, {gain: '1.4, 1.4, 1.1'});
23
24 =====
25 // 5. Definición del centro del mapa para la visualización
26 =====
27
28 Map.setCenter(-89.033, 13.898, 12);

```

Figura 3: Código nuevo, generado a partir del código ejemplo Clipped Composite

La ejecución del código nuevo, al que se le asignó el nombre `MosaicoL7.js`, genera como resultado un mosaico L7 en RGB con dimensiones globales. Esto se debe a que en el código no se

ha ingresado un límite que defina el área de interés para la búsqueda de las imágenes L7.

Una forma de delimitar el área de interés, que en este caso es El Salvador, se puede conseguir en el comando de la línea 8 del ejemplo [Filtered Composite](#), que se muestra en la figura 5. El comando de la línea 15 muestra la forma en el que el polígono es utilizado para hacer el filtrado geográfico de las L7, y proyectar datos exclusivamente dentro del área definida por el polígono.

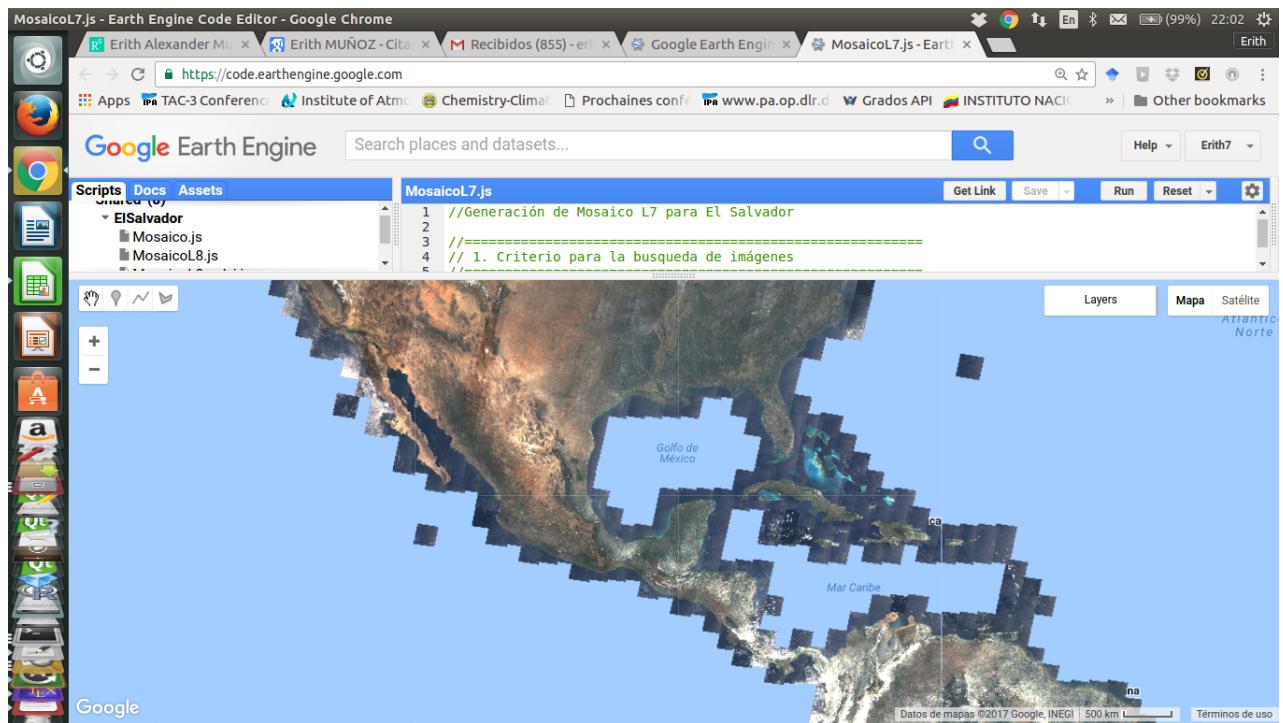


Figura 4: Muestra parcial del resultado de ejecutar el script [MosaicoL7.js](#)

Para incorporar este fragmento de código dentro de [MosaicoL7.js](#), solo debemos copiar la parte de interés, y pegarlo en la zona inicial del código, de tal forma de que sea el primer comando de [MosaicoL7.js](#). Los números que aparecen en la variable polígono representan coordenadas de 9 puntos que delimitan la zona de interés. De esta forma, es necesario realizar la selección de nueve puntos que pertenezcan al perímetro de una línea de límite imaginaria que se proyecta sobre el mapa para delimitar el área de interés.

Para seleccionar estos puntos, se debe seleccionar la herramienta INSPECTOR del panel derecho de la interfaz de GEE, dar clic sobre un punto del mapa, y tomar las coordenadas que aparecen en el INSPECTOR. Esto debe realizarse nueve veces, tomando punto que encierran el área de interés. La figura 6 muestra el INSPECTOR y la información que proporciona al hacer clic sobre un punto en el mapa.

```

Filtered Composite - Earth Engine Code Editor - Google Chrome
Erith Alexander Muñoz Cita | Erith MUÑOZ - Cita | Recibidos (855) - er | Google Earth Engine | Filtered Composite | Erith
https://code.earthengine.google.com
Apps TAC-3 Conference Institute of Atm... Chemistry-Clima Prochaines confére... www.pa.op.dlr.d Grados API INSTITUTO NACIONAL Other bookmarks
Help | Erith7
Google Earth Engine Search places and datasets...
Scripts Docs Assets Filtered Composite Get Link Save Run Reset
6 // Instead of filtering the input collection.
7 // Filter to only include images within the colorado and utah boundaries
8 var polygon = ee.Geometry.Polygon([
9 [-109.05, 37.0], [-102.05, 37.0], [-102.05, 41.0], // colorado
10 [-109.05, 41.0], [-111.05, 41.0], [-111.05, 42.0], // utah
11 [-114.05, 42.0], [-114.05, 37.0], [-109.05, 37.0]]);
12
13 // Create a Landsat 7 composite for Spring of 2000, and filter by
14 // the bounds of the FeatureCollection.
15 var collection = ee.ImageCollection('LE7_L1T')
16 .filterDate('2000-04-01', '2000-07-01')
17 .filterBounds(polygon);
18
19 // Select the median pixel.
20

```

Figura 5: El comando de la línea 8 del ejemplo **Filtered Composite**, muestra la forma de definir un polígono para delimitar el área de interés. Por su parte, el comando de la línea 17, muestra como debe ser implementado este polígono para filtrar las L7.

```

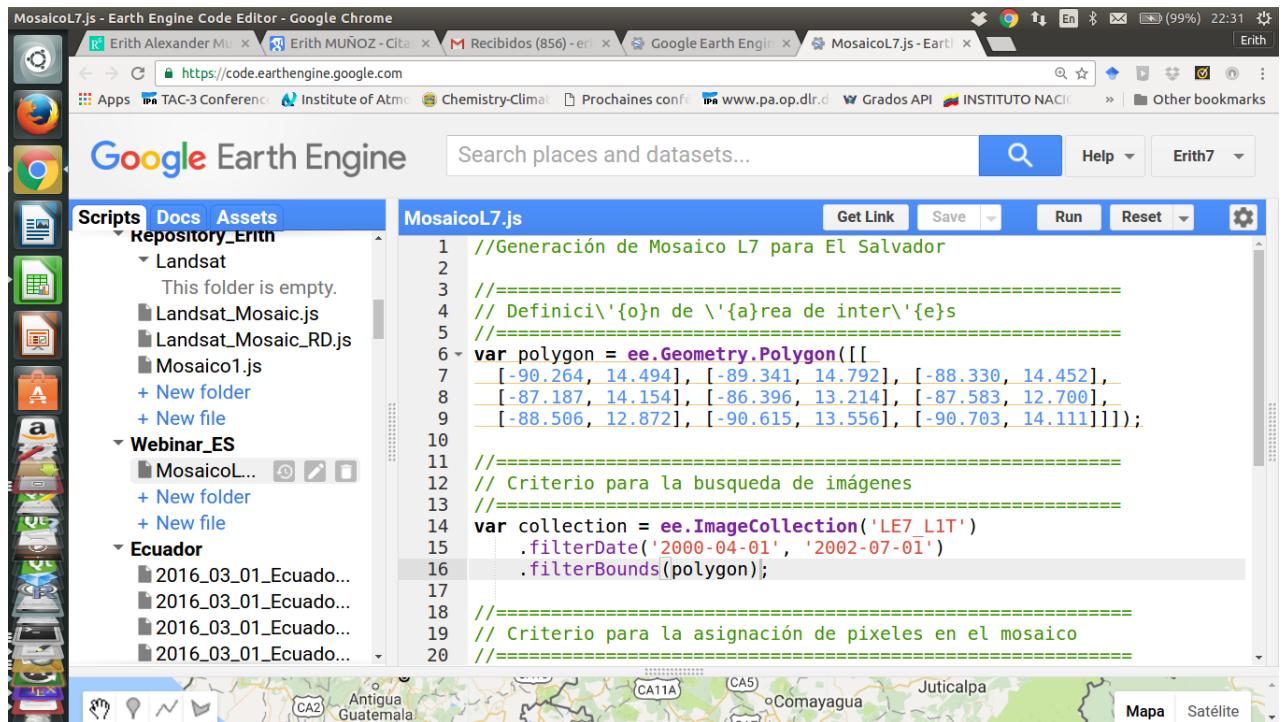
*MosaicoL7.js - Earth Engine Code Editor - Google Chrome
Erith Alexander Muñoz Cita | Erith MUÑOZ - Cita | Recibidos (856) - er | Google Earth Engine | *MosaicoL7.js - Earth Engine | Erith
https://code.earthengine.google.com
Apps TAC-3 Conference Institute of Atm... Chemistry-Clima Prochaines confére... www.pa.op.dlr.d Grados API INSTITUTO NACIONAL Other bookmarks
Help | Erith7
Google Earth Engine Search places and datasets...
Scripts Docs Assets Mosaic... Get Link Save Run Reset
1 //Generación de Mosaico L7 para El Salvador
2
3 //=====
4 // Definición de función de área de interés
5 //=====
6 var polygon = ee.Geometry.Polygon([
7 [-109.05, 37.0], [-102.05, 37.0], [-102.05, 41.0],
8 [-109.05, 41.0], [-111.05, 41.0], [-111.05, 42.0],
9 [-114.05, 42.0], [-114.05, 37.0], [-109.05, 37.0]]);
10
11

```

Figura 6: Información proporcionada por el INSPECTOR al hacer clic sobre un punto en el mapa.

En la figura 7 se muestra el código `MosaicoL7.js`, donde se han incluido los 9 puntos utilizados para delimitar a El Salvador, y en la línea 16 se observa que ha sido incluido el polígono para filtrar el área de interés para la proyección de las L7. En la figura 8 se puede apreciar el nuevo mosaico, que está integrado por las escenas L7 necesarias para cubrir el área delimitada por el polígono.

Una forma más elegante de definir el área de interés, es a partir de un límite internacional. Esto puede ser realizado guardando una capa vector del límite internacional en Google Drive usando Fusion Tables. Las Fusion Tables permiten obtener un código con una ruta absoluta que pueden luego ser proporcionadas a funciones de GEE, para poder incorporarlas en nuestro proyecto. De esta forma, si se toma el código del ejemplo `Clipped Composite` y se eliminan las líneas de comentarios, y las líneas 11, 12, 13, y se cambia la fecha de búsqueda final del 2000 al 2002, y se realizan los nuevos comentarios, el código nuevo resulta en el mostrado en la figura 9.



```

MosaicoL7.js - Earth Engine Code Editor - Google Chrome
Erith Alexander Muñoz Cita - Erith MUÑOZ - Cita | Recibidos (856) - erith | Google Earth Engine | MosaicoL7.js - Earth Engine | Erith
https://code.earthengine.google.com
Google Earth Engine | Search places and datasets... | Help | Run | Reset | Erith7 | Erith
Scripts Docs Assets
Repository_Erith
  - Landsat
    - This folder is empty.
  - Landsat_Mosaic.js
  - Landsat_Mosaic_RD.js
  - Mosaico1.js
    + New folder
    + New file
  - Webinar_ES
    - Mosaico1...
    + New folder
    + New file
  - Ecuador
    - 2016_03_01_Ecuador...
    - 2016_03_01_Ecuador...
    - 2016_03_01_Ecuador...
    - 2016_03_01_Ecuador...
MosaicoL7.js
Get Link Save Run Reset
1 //Generación de Mosaico L7 para El Salvador
2
3 //=====
4 // Definición de área de interés
5 //=====
6 var polygon = ee.Geometry.Polygon([
7   [-90.264, 14.494], [-89.341, 14.792], [-88.330, 14.452],
8   [-87.187, 14.154], [-86.396, 13.214], [-87.583, 12.700],
9   [-88.506, 12.872], [-90.615, 13.556], [-90.703, 14.111]]);
10
11 //=====
12 // Criterio para la búsqueda de imágenes
13 //=====
14 var collection = ee.ImageCollection('LE7_L1T')
15   .filterDate('2000-04-01', '2002-07-01')
16   .filterBounds(polygon);
17
18 //=====
19 // Criterio para la asignación de pixeles en el mosaico
20 //=====

```

Figura 7: Actualización del polígono con las coordenadas de los nuevos puntos, e inserción en la línea 16 del polígono para el filtrado espacial de la zona de interés.

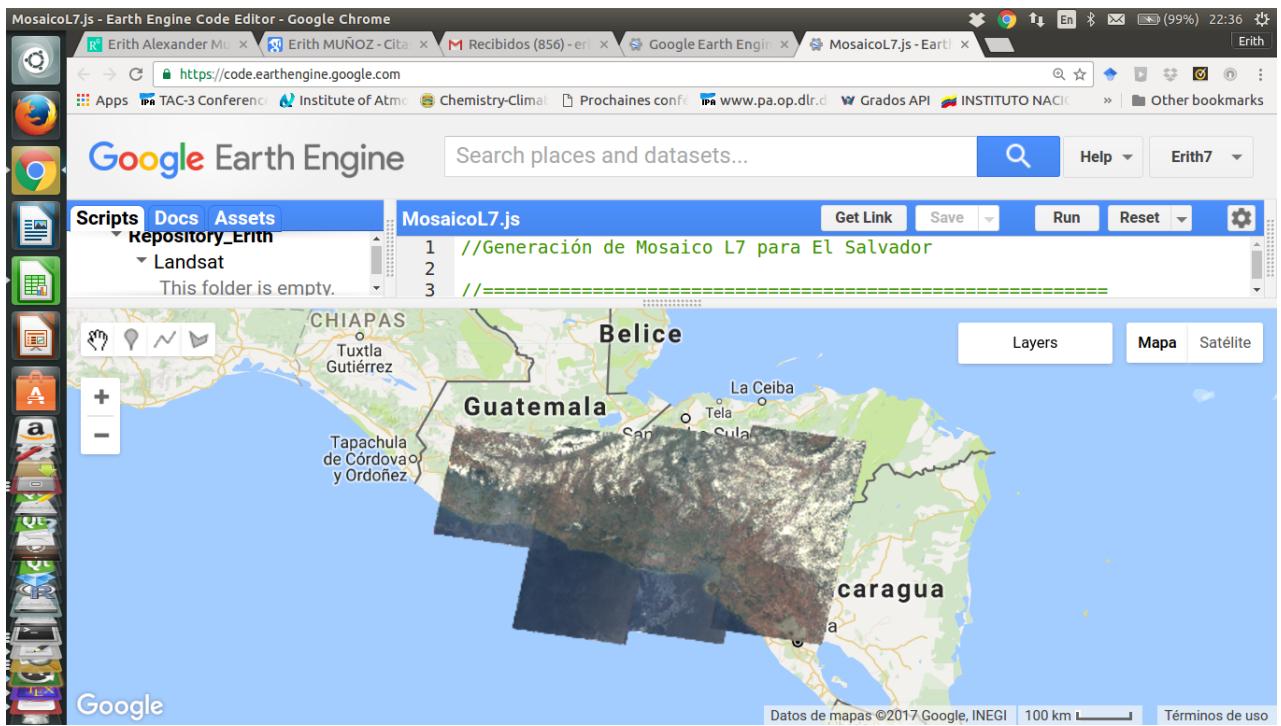


Figura 8: Resultado de la ejecución de **MosaicoL7.js** después del filtrado del área de interés a partir de la definición de un polígono.

```

MosaicoL7.js
1 //Generación de Mosaico L7 para El Salvador
2 //=====
3 // Criterio para la búsqueda de imágenes
4 //=====
5 var collection = ee.ImageCollection('LE7_L1T')
6 .filterDate('2000-04-01', '2002-07-01');
7 //=====
8 // Criterio para la asignación de pixeles en el mosaico
9 //=====
10 var median = collection.median();
11 //=====
12 // Definición de área de interés
13 //=====
14 var fc = ee.FeatureCollection('ft:1D6dB3QyC4XKEgvKTgJDZBVz8RpfdgqEcJb0jfLoY');
15 var clipped = median.clipToCollection(fc);
16 //=====
17 // Selección de Bandas
18 //=====
19 var result = clipped.select('B3', 'B2', 'B1');
20 //=====
21 // Configuración de ganancia por Bandas
22 //=====
23 Map.addLayer(result, {gain: '1.4, 1.4, 1.1'});
24 //=====
25 // Definición del centro del mapa para la visualización
26 //=====
27 Map.setCenter(-89.033, 13.898, 12);
28 //=====
29 
```

Figura 9: Versión del código **MosaicoL7.js** para la generación de mosaico en El Salvador usando el límite internacional como criterio de filtrado geográfico.

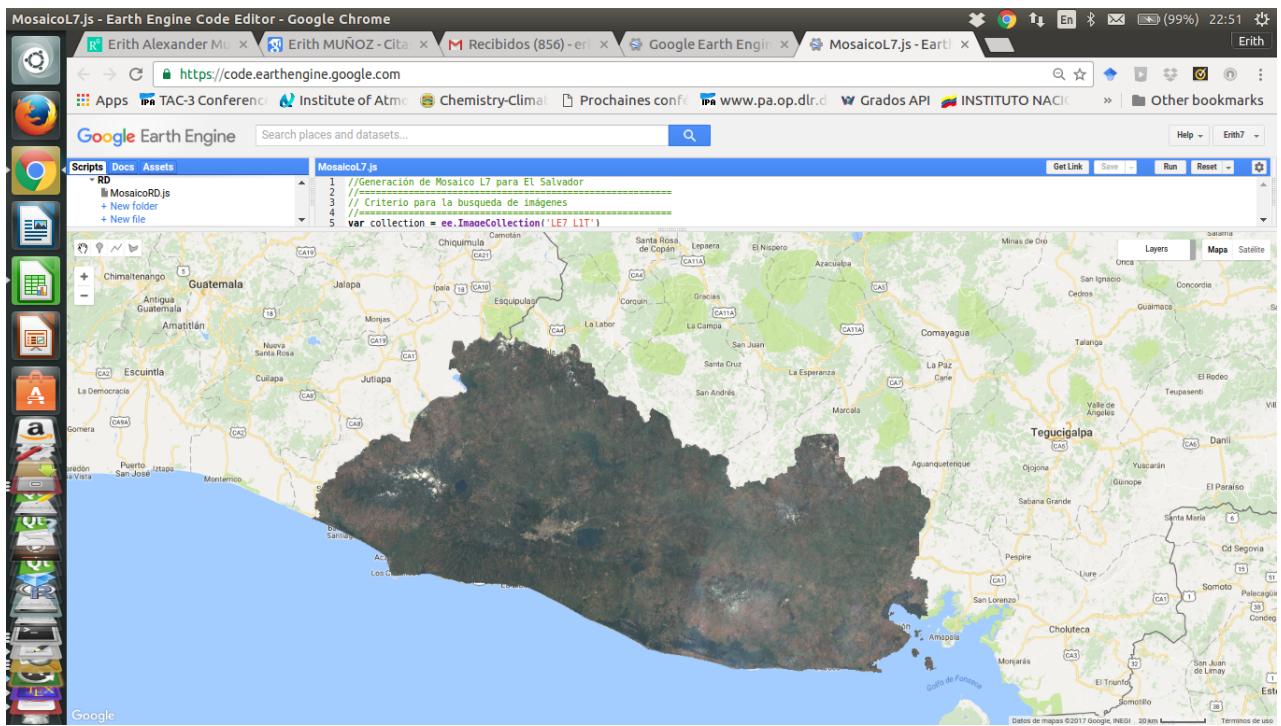


Figura 10: Resultado de la ejecución de `MosaicoL7.js` después del filtrado del área de interés a partir de la definición del límite internacional de El Salvador.

2.2. Generación de Mosaicos L7 multitemporales

A diferencia del código mostrado en la figura 9, la generación de un mosaico L7 para dos períodos de tiempo (t_1 y t_2), requiere incluir una nueva búsqueda de L7 para el segundo periodo de tiempo, que para este ejercicio se ha designado desde 2003 hasta 2005. El código para la generación de mosaicos multifechas se muestra en la figura 11.

```

3 // Criterio para la búsqueda de imágenes t1
4 =====
5 var collection = ee.ImageCollection('LE7_L1T')
6 .filterDate('2000-04-01', '2002-07-01');
7 =====
8 // Criterio para la búsqueda de imágenes t2
9 =====
10 var collection2 = ee.ImageCollection('LE7_L1T')
11 .filterDate('2003-04-01', '2005-07-01');
12 =====
13 // Criterio para la asignación de pixeles en el mosaico
14 =====
15 var median = collection.median();
16 var median2 = collection2.median();
17 =====
18 // Definición (o) de \{a\}rea de inter\{e}s
19 =====
20 var fc = ee.FeatureCollection('ft:ID6dB3QyC4XKEgvKTgJDZBVz8RpfdgqEcJb0jfLoY');
21 var clipped = median.clipToCollection(fc);
22 var clipped2 = median2.clipToCollection(fc);
23 =====
24 // Selección de Bandas
25 =====
26 var result = clipped.select('B3', 'B2', 'B1');
27 var result2 = clipped2.select('B3', 'B2', 'B1');
28 =====
29 // Configuración de ganancia por Bandas
30 =====
31 Map.addLayer(result, {gain: '1.4, 1.4, 1.1'});
32 Map.addLayer(result2, {gain: '1.4, 1.4, 1.1'});
33 =====
34 // Definición del centro del mapa para la visualización
35 =====
36 Map.setCenter(-89.033, 13.898, 12);

```

Figura 11: Resultado de la ejecución de `MosaicoL7.js` después del filtrado del área de interés a partir de la definición del límite internacional de El Salvador.

2.3. Mosaico Landsat 8

Para la generación de un mosaico de imágenes Landsat 8 (L8) para El Salvador, se utilizará como referencia el código del ejemplo [Landsat Simple Composite](#) de GEE, que se muestra en la figura 12. En su forma original, este código permite generar un composito de L8 a escala global, para un periodo de tiempo de 6 meses comprendido desde el 1 enero al 1 de julio de 2015, calibrada a reflectancia en tope de atmósfera (TOA).

De forma análoga a como se realizó para L7, se genera un archivo con el nombre `MosaicoL8.js`, se copia el código del ejemplo, y se ajusta hasta llegar al código mostrado en la figura 13

The screenshot shows the Google Earth Engine Code Editor interface. The left sidebar displays a file tree under the 'Scripts' tab, containing folders like 'Mosaico', 'RD', 'Repository_Erith', and 'Webinar_ES', along with files such as 'Mosaico.js', 'MosaicoL8.js', and 'MosaicoL8...'. The main workspace shows the code for 'Landsat Simple Composite':

```

1 // Composite 6 months of Landsat 8.
2
3 // Pick a spot with lots of clouds.
4 Map.setCenter(-47.6735, -0.6344, 12);
5
6 var L8 = ee.ImageCollection('LANDSAT/LC8_LIT');
7
8 // "asFloat: true" gives proper (floating-point) TOA output instead of
9 // the mangled-to-UINT8 outputs of the original simpleComposite().
10 var composite = ee.Algorithms.Landsat.simpleComposite({
11   collection: L8.filterDate('2015-1-1', '2015-7-1'),
12   asFloat: true});
13
14 Map.addLayer(composite, {bands: 'B7,B6,B1', max: [0.3, 0.4, 0.3]});
15

```

Figura 12: Código del ejemplo Landsat Simple Composite

The screenshot shows the Google Earth Engine Code Editor interface. The left sidebar displays a file tree under the 'Scripts' tab, containing folders like 'RD', 'Repository_Erith', 'Webinar_ES', and 'Ecuador', along with files such as 'MosaicoRD.js', 'Mosaico1.js', and 'MosaicoL8...'. The main workspace shows the code for 'MosaicoL8.js':

```

1 =====
2 // Definición del centro del mapa para la visualización
3 =====
4 Map.setCenter(-89.033, 13.898, 12);
5
6 =====
7 // Definición de la área de interés
8 =====
9 var fc = ee.FeatureCollection('ft:ID6dB30yC4XKEgvKTgJDZBVz8RpfdgqEcJb0jfLoY');
10
11 =====
12 // Definición del tipo de imagen a buscar
13 =====
14 var L8 = ee.ImageCollection('LANDSAT/LC8_LIT');
15
16 =====
17 // Definición del tipo dato y temporalidad
18 =====
19 var composite = ee.Algorithms.Landsat.simpleComposite({
20   collection: L8.filterDate('2015-1-1', '2015-7-1'),
21   asFloat: true});
22
23 =====
24 // Recorte de los datos al área de interés
25 =====
26 var clipped = composite.clipToCollection(fc);
27
28 =====
29 // Configuración de la visualización
30 =====
31 Map.addLayer(clipped, {bands: 'B7,B6,B1', max: [0.3, 0.4, 0.3]});

```

Figura 13: Código MosaicoL8.js

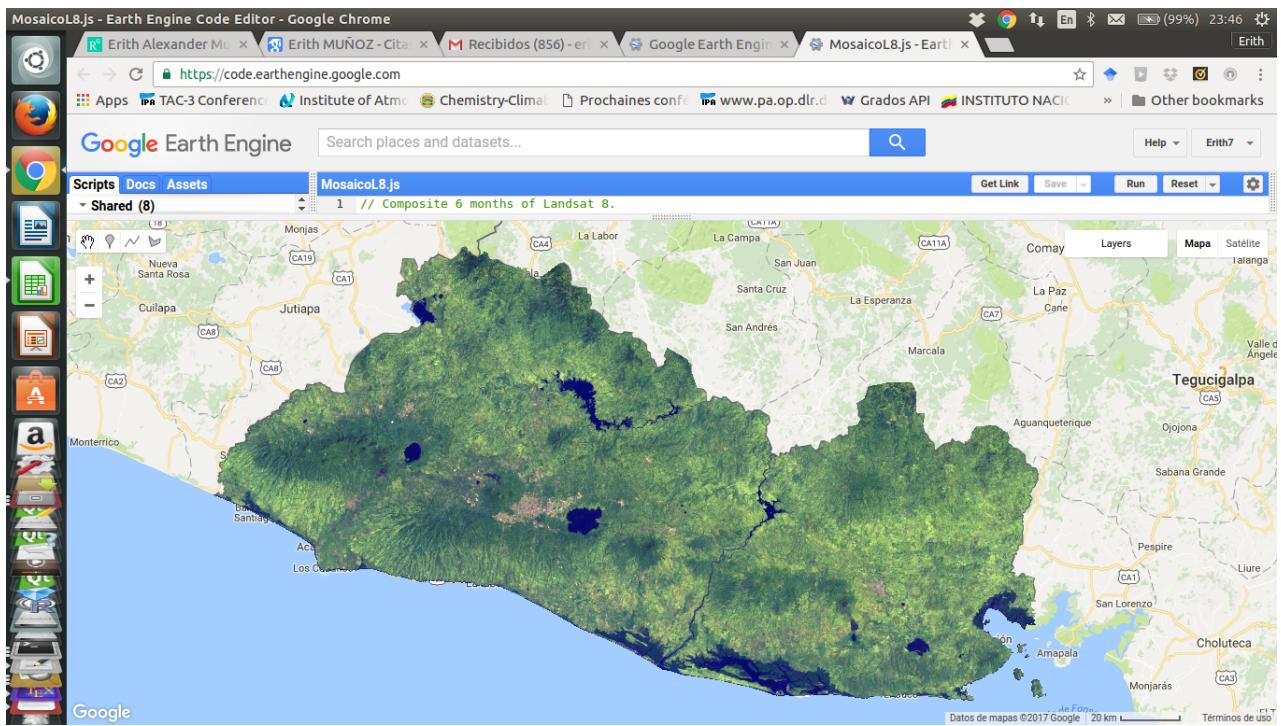


Figura 14: Resultado de la ejecución del código MosaicoL8.js

2.4. Generación de Mosaicos L8 multitemporales

```

MosaicoL8.js - Earth Engine Code Editor - Google Chrome
Erith Alexander Mu... Erith MUÑOZ - Cita... Recibidos (856) - er... Google Earth Engin... MosaicoL8.js - Earth...
https://code.earthengine.google.com
Help | Run | Reset | Scripts | Docs | Assets | MosaicoL8.js | Get Link | Save | Help | Erith7 | ...
Google Earth Engine Search places and datasets...
1 //> Composite 6 months of Landsat 8.
2 // Definición del centro del mapa para la visualización
3 // Definición de \'(a)rea de inter\'(e)s
4 var fc = ee.FeatureCollection('ft:ID6dB3QyC4XKEgvkTgJDZBVz8RpfdgqEcJb0jfLoY');
5 Map.setCenter(-89.033, 13.898, 12);
6 // Definición de \'(o)n de \'(a)rea de inter\'(e)s
7 // Definición del tipo de imagen a buscar
8 var L8 = ee.ImageCollection('LANDSAT/LC8_L1T');
9 // Definición del tipo dato y temporalidad
10 var composite = ee.Algorithms.Landsat.simpleComposite({
11   collection: L8.filterDate('2015-1-1', '2015-7-1'),
12   asFloat: true});
13 var composite2 = ee.Algorithms.Landsat.simpleComposite({
14   collection: L8.filterDate('2016-1-1', '2016-7-1'),
15   asFloat: true});
16 // Recorte de los datos al \'(á)rea de interés
17 var clipped = composite.clipToCollection(fc);
18 var clipped2 = composite2.clipToCollection(fc);
19 // configuración de la visualización
20 Map.addLayer(clipped, {bands: 'B7,B6,B1', max: [0.3, 0.4, 0.3]});
21 Map.addLayer(clipped2, {bands: 'B7,B6,B1', max: [0.3, 0.4, 0.3]});
```

Figura 15: Código para la generación de mosaicos L8 multifecha

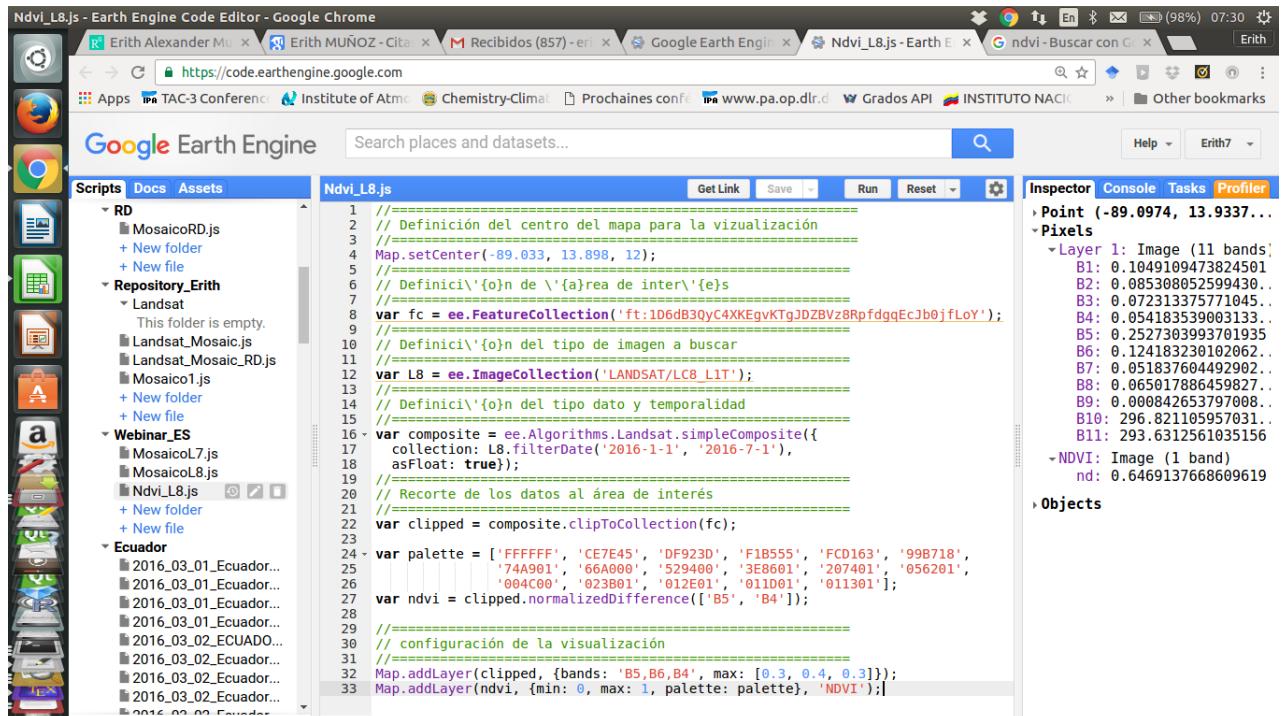
3. CÁLCULO DE NDVI

La determinación del índice de vegetación de diferencia normalizada (NDVI, por sus siglas en inglés) se realiza a partir de la ecuación 1.

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

En la ecuación 1, **NIR** representa la banda del infrarrojo cercano de la escena Landsat 8, mientras que **RED** es la banda del rojo. Los valores de NDVI están expresados entre -1 y 1, y es útil para identificar coberturas de vegetación en una escena debido a que los valores cercanos a 1 son precisamente coberturas vegetales. A medida que el NDVI se aproxima disminuye, se obtiene información sobre coberturas en las que va disminuyendo la presencia de vegetación, hasta obtener valores que indican otro tipo de coberturas.

En GEE, el cálculo de NDVI se puede realizar a partir de la ecuación 1, como de hecho se hace en el ejemplo **Expression**. También el cálculo de NDVI se puede realizar a partir de una función propia de GEE, como se hace en el ejemplo **Normalized Difference**. Para el caso práctico de este ejercicio, se utilizará como base el código mostrado en la figura 14 para la generación de un mosaico L8, conjuntamente con el código del ejemplo **Normalized Difference** para realizar el computo del NDVI.



The screenshot shows the Google Earth Engine Code Editor interface. The left sidebar displays a file tree with categories like RD, Repository_Erith, Webinar_ES, and Ecuador, each containing various JS files. The main editor area contains the code for NDVI_L8.js:

```
1 //=====
2 // Definición del centro del mapa para la visualización
3 //=====
4 Map.setCenter(-89.033, 13.898, 12);
5 //=====
6 // Definición de la área de interés
7 //=====
8 var fc = ee.FeatureCollection('ft:ID6d830yC4XKEgvkTgJDZBVz8RpfdgqEcJb0jfLoY');
9 //=====
10 // Definición del tipo de imagen a buscar
11 //=====
12 var L8 = ee.ImageCollection('LANDSAT/LC8_L1T');
13 //=====
14 // Definición del tipo dato y temporalidad
15 //=====
16 var composite = ee.Algorithms.Landsat.simpleComposite({
17   collection: L8.filterDate('2016-1-1', '2016-7-1'),
18   asFloat: true});
19 //=====
20 // Recorte de los datos al área de interés
21 //=====
22 var clipped = composite.clipToCollection(fc);
23 //=====
24 var palette = ['FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718',
25   '74A901', '66A000', '529400', '3E8601', '207401', '056201',
26   '004C00', '023B01', '012E01', '01D001', '011301'];
27 var ndvi = clipped.normalizedDifference(['B5', 'B4']);
28 //=====
29 // Configuración de la visualización
30 //=====
31 //=====
32 Map.addLayer(clipped, {bands: 'B5,B6,B4', max: [0.3, 0.4, 0.3]});
33 Map.addLayer(ndvi, {min: 0, max: 1, palette: palette}, 'NDVI');
```

The right panel shows the Inspector, Console, Tasks, and Profiler tabs. The Inspector tab displays a point at coordinates (-89.0974, 13.9337) with several bands (B1-B11) listed. The Profiler tab shows the execution time for the NDVI calculation.

Figura 16: Código **NDVI_L8.js**, para el cálculo de NDVI para un mosaico L8

En la figura 16 se muestra el código para el cálculo de NDVI en el mosaico L8. Los comandos de las líneas 4, 8, 12, 16, 22, y 32 corresponden al código de la figura 14 para generar un mosaico L8. Sin embargo, en el comando de la línea 16 se ha modificado el periodo de tiempo para la conformación del composito, y en la línea 32 se ha modificado las bandas a visualizar, con la finalidad de tener a la mano las bandas 4 y 5 de L8, que son las que se utilizan para la estimación de NDVI, de acuerdo a la ecuación 1.

Además de estas modificaciones, se utilizó el ejemplo **Normalized Difference** para realizar el cálculo del NDVI, a partir del comando de la línea 27. Previamente en la línea 24 se incluyó una paleta de colores para poder visualizar el NDVI a color, en lugar de tono de grises, y finalmente se agrego la capa generada de NDVI a la visualización del mapa, mediante el comando de la línea 33.

En la figura 17 se muestra un mosaico L8 utilizando para la visualización el arreglo R=B5, G=B6, B=B4. Entre las principales características que tiene esta combinación de bandas, resalta que los cuerpos de agua son rápidamente identificables a partir del color negro. Los bosques primarios y densos están caracterizados por tonos de color naranja oscuro, otros tipos de coberturas boscosas presentan tonos naranjas más débiles. Coberturas asociadas a pastizales presentan colores amarillos, las zonas de cultivos presentan tonos verdes.

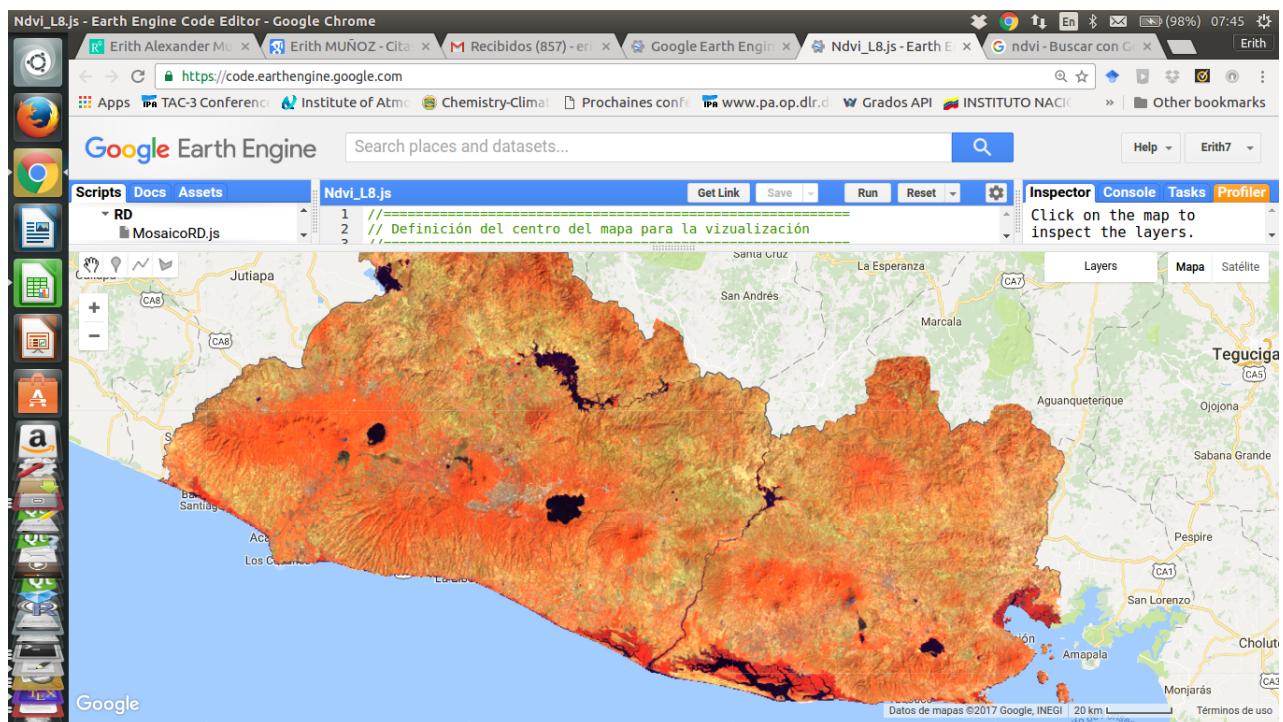


Figura 17: Mosaico L8 utilizando para la visualización el arreglo R=B5, G=B6, B=B4

Por otra parte, en la figura 18 se muestra el mapa de NDVI. En este mapa los tonos verdes oscuros corresponden a valores altos de ndvi, lo que generalmente está asociado a coberturas vege-

tales densas y vigorosas como los bosques. Los tonos de verdes claros representan niveles de NDVI medios, los amarillos niveles bajos, generalmente vinculados a zonas de cultivos, y los que toman colores claros, cercanos a blanco, son valores cercanos a cero, y/o negativos.

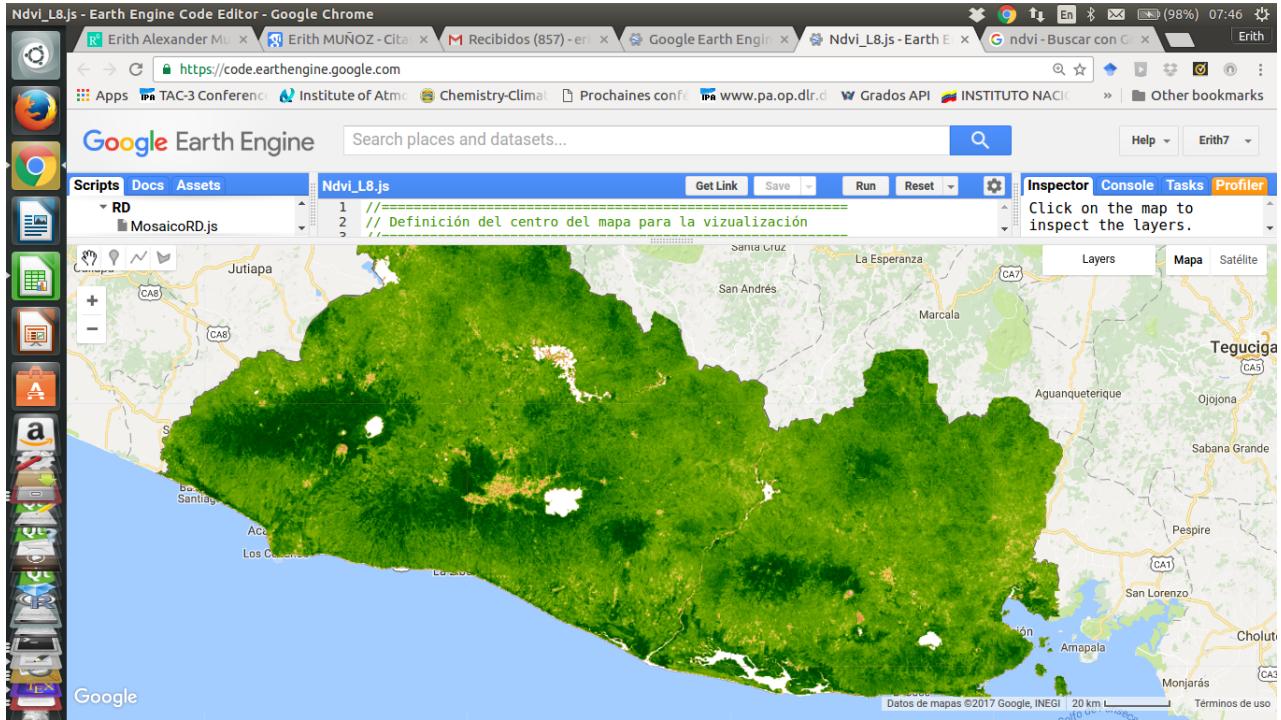


Figura 18: Mapa del indice NDVI para el mosaico de la figura 17

4. CLASIFICACIÓN

Para realizar un clasificación, se pueden utilizar algoritmos de inteligencia artificial implementados en GEE, como lo son random forest, cart, o maquina de vectores de soporte (svm). Estos algoritmos, permiten realizar clasificación supervisada a partir de puntos de entrenamientos que suministra el usuario. De esta forma, a modo de ejemplo se construye una clasificación de bosque no bosque (BNB) para el mosaico generado a partir del código de la figura 14.

En este contexto, se generarán punto de entrenamiento para tres clases diferentes, bosque, no-bosque, y cuerpos de agua. Para tomar los puntos de control se debe seguir los siguientes pasos:

1. Se ejecuta el código de la figura 14, para tener como referencia el mosaico L8 para la captura de los puntos correspondientes a cada clase. Las bandas utilizadas para el mosaico son B5, B6, y B4.
2. En la parte superior izquierda del mapa generado, se selecciona la herramienta **añadir marca-**

dor.

3. De acuerdo, a la figura 19, se da clic en la sección **geometry imports**.
4. Se obtiene la ventana mostrada en la figura 20, donde se asigna el tipo de colección de los puntos de entrenamiento, se asigna un nombre, una etiqueta para la clase, y el valor.
5. Luego se añade la segunda clase, dando clic en el link **new layer**, mostrado en la figura 21, y se realiza el mismo procedimiento que con la clase 1.
6. Se repite este procedimiento para la clase agua, y se obtiene la figura 22, en donde se pueden apreciar las tres clases mostradas en el mapa, y también agregadas como variables en la parte superior del código para la generación del mosaico L8.
7. Finalmente se inserta un polígono con característica de geometría, necesario para el proceso de clasificación, tal como se muestra en la figura 23.

En la figura 24, se muestra la fracción del código que es necesario agregar para la clasificación supervisada.

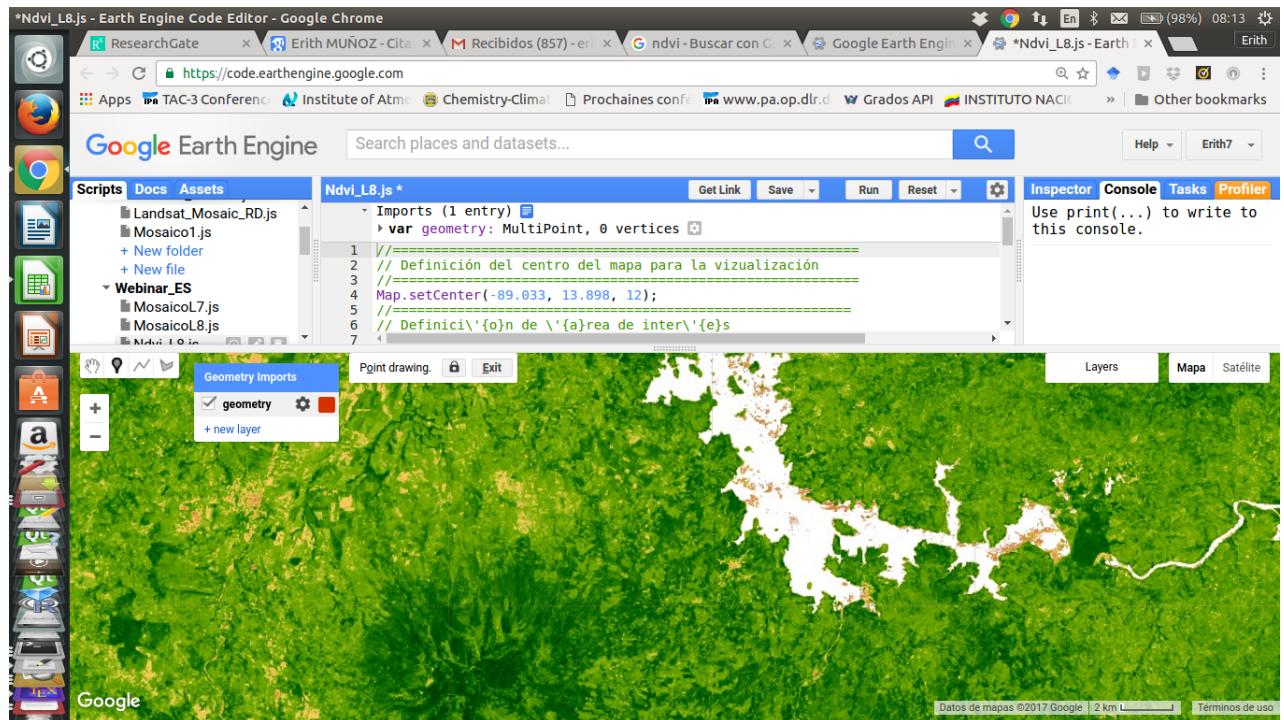


Figura 19: Configuración de la clase bosque

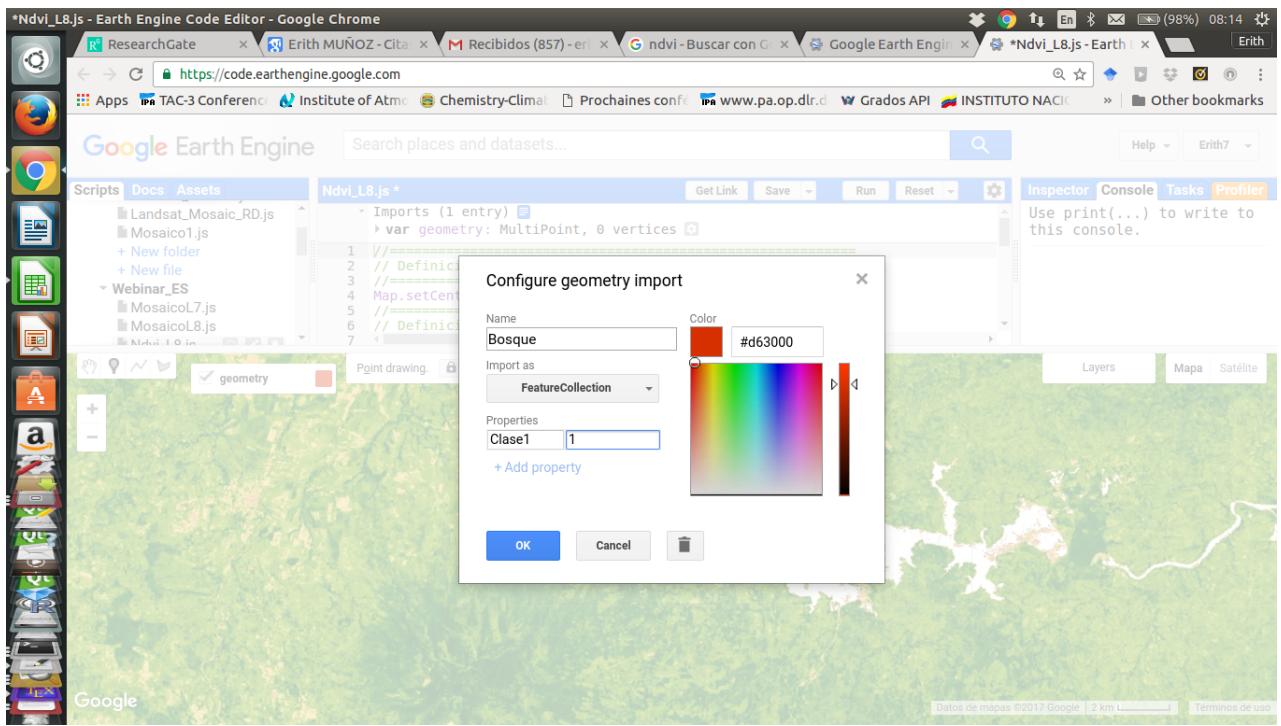


Figura 20: Configuración de la clase bosque, definición de parámetros de la clase

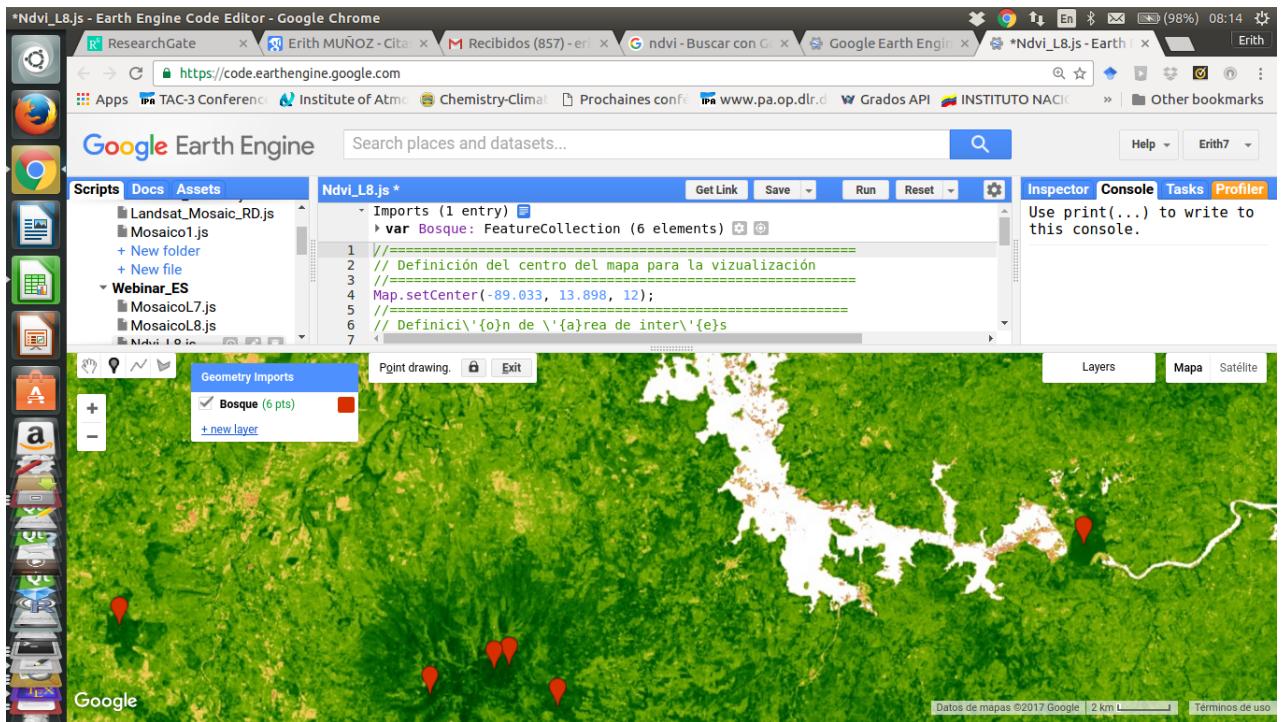


Figura 21: Link para insertar una nueva clase de puntos de entrenamiento

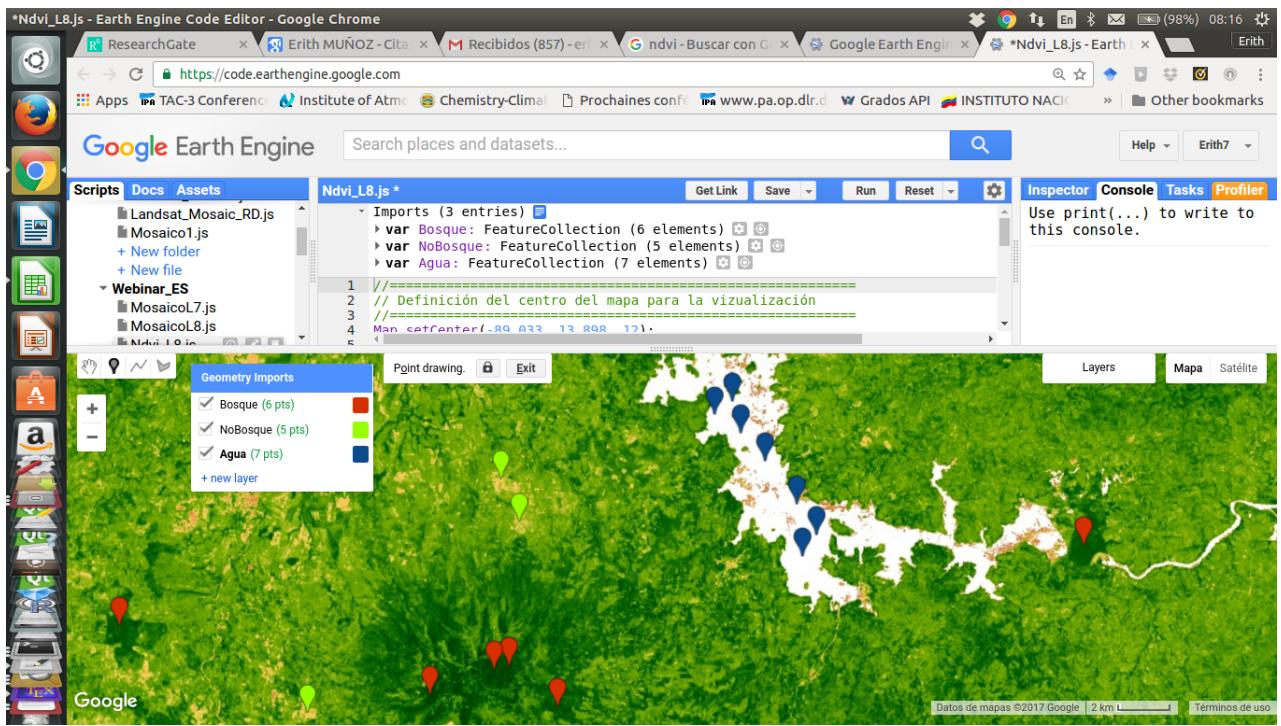


Figura 22: Resultado obtenido a partir de la selección de clases y puntos de entrenamiento

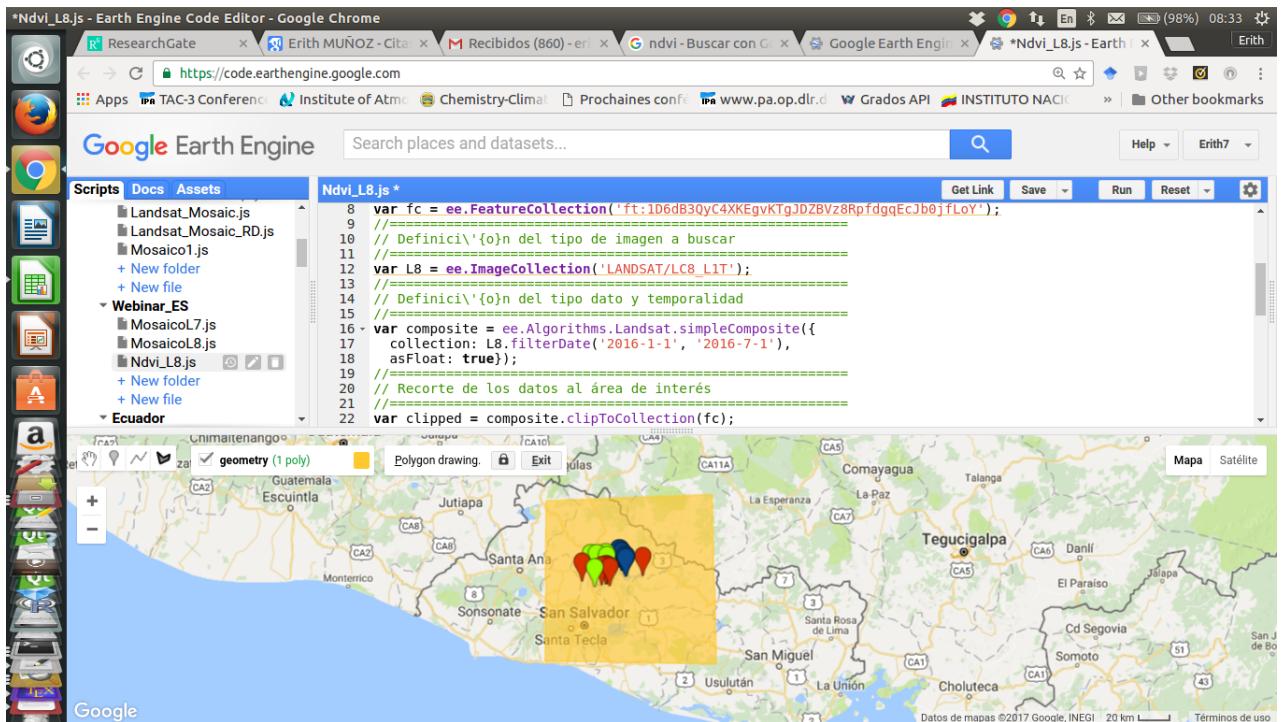


Figura 23: Dibujo de polígono para definir el límite de la clasificación BNB

The screenshot shows the Google Earth Engine Code Editor interface. The left sidebar displays a file tree under the 'Scripts' tab, containing various projects and scripts such as 'Repository_Erit', 'Webinar_ES', and 'Ecuador'. The main editor area shows the code for 'Ndvi_L8.js'. The code performs several steps: it clips a composite to a specific area of interest; defines a color palette; calculates NDVI (Normalized Difference Vegetation Index); samples regions from the clipped data; trains a CART classifier on the training data; classifies the entire dataset; and adds layers to the map for visualization.

```
18     asFloat: true});
19 //=====
20 // Recorte de los datos al área de interés
21 //=====
22 var clipped = composite.clipToCollection(fc);
23
24 var palette = ['FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718',
25   '7A901', '66A000', '529400', '3E8601', '207401', '056201',
26   '004C00', '023B01', '012E01', '011D01', '011301'];
27
28 var ndvi = clipped.normalizedDifference(['B5', 'B4']);
29
30
31 //=====
32 // Clasificación Supervisada
33 //=====
34 //Clasificacion Supervisada
35 var puntos = ee.FeatureCollection([Bosque, NoBosque, Agua]).flatten();
36 var training = clipped.sampleRegions({
37   collection: puntos,
38   properties: ['clase'],
39   scale: 30
40 });
41
42 var bandas = ['B4', 'B5', 'B6'];
43 var trained = ee.Classifier.cart().train(training, 'clase', bandas);
44 var classified = clipped.select(bandas).classify(trained);
45 Map.addLayer(classified, {min:1, max:3, palette: ['339820', '0446ff', 'e6f0c2'] }, 'Clasificacion');
46
47 //=====
48 // configuración de la visualización
49 //=====
50 Map.addLayer(clipped, {bands: 'B5,B6,B4', max: [0.3, 0.4, 0.3]});
51 Map.addLayer(ndvi, {min: 0, max: 1, palette: palette}, 'NDVI');
```

Figura 24: Código para el porceso de clasificación supervisada, mediante el algoritmo cart.