

# Tutorial para el mapeo de disturbios forestales en la Amazonia Sudoccidental usando CODED, LandTrendr y MTDD

Reygadas Langarica, Yunuen  
UNIVERSITY OF RICHMOND

## Tabla de contenidos

1. Antecedentes.....	1
2. Definiciones .....	1
3. Algoritmos.....	2
3.1. CODED .....	2
3.1.1. <i>Mapa de disturbios</i> .....	2
3.1.2. <i>Mapa de degradación y deforestación</i> .....	5
3.2. LandTrendr .....	8
3.2.1. Mapa de disturbios .....	8
3.3. MTDD.....	11
3.3.1. <i>Muestras para el entrenamiento</i> .....	11
3.3.2. <i>Mapa de degradación y deforestación</i> .....	16
Referencias .....	22

# Tutorial para el mapeo de disturbios forestales en la Amazonia Sudoccidental usando CODED, LandTrendr y MTDD

## 1. Antecedentes

Debido a la urgente necesidad de entender los efectos de los disturbios forestales en los servicios ecosistémicos, en la última década se han desarrollado múltiples algoritmos enfocados en detectar cambios en la vegetación a través de datos remotamente sensados. El reto ahora es entender qué algoritmo es el más adecuado para los objetivos y el área de estudio de cada usuario. Este tutorial comparte nuestro trabajo en la plataforma *Google Earth Engine* (GEE) derivado de la evaluación de tres algoritmos –*Continuous Degradation Detection* (CODED), *Landsat-based detection of trends in disturbance and recovery* (LandTrendr) y *Multi-variate Time-series Disturbance Detection* (MTDD)– para detectar y caracterizar disturbios forestales en la Amazonia Sudoccidental (ASO). Para conocer la totalidad del estudio, por favor consulte Reygadas et al. (2021) (en revisión).

En la sección 3.3, se comparten códigos GEE basados en el algoritmo MTDD para detectar disturbios en la ASO (i.e., Ucayali, Perú y Acre, Brasil). Tutoriales completos de [CODED](#) and [LandTrendr](#) están disponibles gratuitamente online; en este tutorial, se comparte la configuración y parametrización que empleamos para la ASO.

## 2. Definiciones

Aunque es difícil adoptar un conjunto de definiciones que armonicen con la lógica detrás de cada algoritmo, aquí nos referimos a la deforestación, degradación forestal y disturbios forestales de la manera siguiente:

- **Deforestación.** Conversión permanente, o de largo plazo, de cubierta forestal a cubierta no forestal (FAO, 2001).
- **Degradación forestal.** Si bien no existe una definición ampliamente aceptada, la degradación forestal es generalmente considerada como un proceso de largo plazo que no conduce a un cambio en la cubierta del suelo pero que afecta negativamente la estructura y la función del bosque (Sasaki and Putz, 2009; Schoene et al., 2007).
- **Disturbios forestales.** Controladores del estado y función del bosque, pueden variar desde eventos de alto impacto, como incendios o deforestación, hasta procesos sutiles y graduales, como aquellos causados por sequías prolongadas, insectos o enfermedades (Cohen et al., 2017; McDowell et al., 2015).

En concordancia, en este documento tanto la deforestación (i.e., eventos de alto impacto) como la degradación forestal (i.e., procesos sutiles y graduales) son consideradas tipos de disturbios forestales.

### 3. Algoritmos

Los tres algoritmos usan valores de reflectancia superficial de *Landsat Thematic Mapper* (TM), *Landsat Enhanced Thematic Mapper+* (ETM+) y *Landsat Operational Land Imager* (OLI).

#### 3.1. CODED

CODED v0 usa series de tiempo de fracciones espectrales y NDFI para detectar y caracterizar disturbios que posteriormente pueden ser clasificados como degradación o deforestación. Consulte Bullock et al. (2020) para detalles del algoritmo y Bullock (2020) para la implementación en GEE.

##### 3.1.1. Mapa de disturbios

Esta sección demuestra cómo ejecutar este [código](#) para producir un mapa de disturbios forestales

###### Datos de entrada

El usuario tiene que ingresar lo siguiente:

```
CODED_MapDisturbios
* Imports (2 entries)
  ↘ var AreaEstudio: Table users/retinta/Tutorials/CaseStudy1
  ↘ var ptsEntrenamiento: Table users/retinta/Tutorials/TrainingPointsCODED
1 ////////////////////////////////////////////////////////////////// ALGORITMO CODED //////////////////////////////////////////////////////////////////
2 /// Produce un mapa de disturbios forestales //////////////////////////////////////////////////////////////////
3 /// Consulte Bullock et al. (2020) para detalles del algoritmo y Bullock (2020) para la implementación en GEE //////////////////////////////////////////////////////////////////
4 /// En particular, este pequeño código para extraer productos CODED fue escrito por Y. Reygadas y V. Galati //////////////////////////////////////////////////////////////////
5
6 ////////////////////////////////////////////////////////////////// Información definida por el usuario //////////////////////////////////////////////////////////////////
7
8 // Define el área de estudio
9 var saveRegion = ee.FeatureCollection(AreaEstudio);
10
11 // Define los datos de entrenamiento
12 var trainingData = ptsEntrenamiento;
13
14 // Define los parámetros
15 var params = ee.Dictionary({
16   'cfThreshold': .01, // Umbral mínimo
17   'consec': 4, // Número de observaciones consecutivas debajo del umbral de cambio necesario para declarar una perturbación
18   'thresh': 3, // Umbral de cambio (residuo de observación normalizado por el modelo de entrenamiento RMSE)
19   'start': 2000, // Año inicial del período de estudio
20   'end': 2018, // Año final del período de estudio
21   'trainDataEnd': 2016, // Año final del período de entrenamiento
22   'trainDataStart': 2013, // Año inicial del período de entrenamiento
23   'trainLength': 3, // Número de años en el período de entrenamiento
24   'soil': [2000, 3000, 3400, 5000, 6000, 5800], // Valor de la fracción espectral de suelo para cada banda
25   'gv': [500, 900, 400, 6100, 3000, 1000], // Valor de la fracción espectral de vegetación verde para cada banda
26   'npy': [1400, 1700, 2200, 3000, 5500, 3000], // Valor de la fracción espectral vegetación no fotosintética para cada banda
27   'shade': [0, 0, 0, 0, 0, 0], // Valor de la fracción espectral de sombra para cada banda
28   'cloud': [9000, 9600, 8000, 7800, 7200, 6500], // Valor de la fracción espectral de nubes para cada banda
29   'forestLabel': 1, // Etiqueta asignada al bosque en los datos de entrenamiento
30   'window': 2, // Número máximo de años a usar en el período de monitoreo en cualquier momento dado
31   'minYears': 2, // Número mínimo de años entre disturbios
32   'numChanges': 1, // Número máximo de disturbios a detectar
33   'minObs': 5 // Número mínimo de observaciones necesarias para ajustar un modelo para entrenamiento
34 });
35
36 // Define el nombre de salida
37 var outputName = 'CODED_Disturbios';
```

1. Área de estudio: polígono del área de estudio.

2. Datos de entrenamiento: CODED requiere que el usuario colecte muestras de cubierta y uso del suelo correspondientes a un período de entrenamiento seleccionado. Estas muestras usualmente se colectan a través de un muestreo aleatorio estratificado basado en una combinación de mapas de cubierta y uso del suelo e imágenes de alta resolución especial disponibles en GEE. La columna del archivo de puntos que contiene la cubierta o uso del suelo debe ser nombrada "label".

3. Conjunto de parámetros

4. Nombre de salida: nombre deseado para el mapa de disturbios.

## Ejecución el código

Ejecute el código haciendo clic en *run*. Una vez terminado el proceso, vaya a la consola *task* y haga clic en *run* para exportar los resultados como un *asset* de GEE y como ráster.

```
CODED_MapaDisturbios
39 //////////////////////////////////////////////////////////////////// Obtiene los resultados principales de CODED //////////////////////////////////////////////////////////////////
40
41 // Agrega las funciones de CODED
42 var codedUtils = require('users/retinta/Tutorials:CODED_changeDetection');
43 var dataUtils = require('users/retinta/Tutorials:CODED_dataUtils');
44
45 // Ejecuta el algoritmo CODED
46 var results = codedUtils.submitCODED(saveRegion, params, trainingData);
47
48 // Combierte las columnas de matrices en imágenes
49 var disturbances = dataUtils.makeImage(results, 0, 'dist_', params.get('start'), params.get('end'));
50 var magnitude = dataUtils.makeImage(results, 1, 'mag_', params.get('start'), params.get('end'));
51 var postChange = dataUtils.makeImage(results, 2, 'post_', params.get('start'), params.get('end'));
52 var difference = dataUtils.makeImage(results, 3, 'dif_', params.get('start'), params.get('end'));
53 var forestFlag = dataUtils.makeImage(results, 4, 'forest_', params.get('start'), params.get('end'));
54
55 var disturbanceBands = disturbances.addBands([magnitude, postChange, difference]);
56
57 var save_output = ee.Image(dataUtils.reduceBands(ee.Image(disturbanceBands), params)
58 .addBands(forestFlag.select(0))
59 .setMulti(params));
60
61 ////////////////////////////////////////////////////////////////// Exporta los resultados //////////////////////////////////////////////////////////////////
62
63 // Exporta los resultados como un asset de GEE
64 Export.image.toAsset({
65   image: save_output,
66   description: outputName,
67   assetId: outputName,
68   maxPixels: 1e13,
69   scale: 30,
70   region: saveRegion,
71   pyramidingPolicy: {
72     '.default': 'mode'
73   }
74 });
75
76 // Exporta los resultados como un archivo ráster al Google drive
77 Export.image.toDrive({
78   image: save_output,
79   description: outputName,
80   region: saveRegion,
81   scale: 30
82 });
83
84 //////////////////////////////////////////////////////////////////
85 print("Todo listo");
86
```

**Detecta disturbios y estima sus características**

**Exporta los resultados como un asset de GEE (dato de entrada para producir el mapa de degradación y deforestación) y como un ráster (puede ser visualizado en cualquier SIG)**

The screenshots show the 'Task: Initiate image export' dialog box. Both screenshots have the 'Run' button highlighted with a red circle.

**Left Screenshot Fields:**

- Task name (no spaces) \*: CODED\_Disturbios
- Resolution \*: Scale (m/px) - 30
- Output type: Drive (selected)
- Drive folder: CODED
- Filename: CODED\_Disturbios
- Run button

**Right Screenshot Fields:**

- Task name (no spaces) \*: CODED\_Disturbios
- Resolution \*: Scale (m/px) - 30
- Output type: EE Asset (selected)
- Earth Engine Asset ID: users/retinta/ CODED\_Disturbios
- Pyramiding Policy \*: Custom
- Run button

## Productos

Este código produce un mapa de disturbios forestales con cuatro capas por disturbio (i.e., año de ocurrencia, magnitud del cambio, cubierta del suelo después del disturbio, diferencia de NDFI antes y después del disturbio) y una última capa que indica la cubierta del suelo en el año de inicio.

### Asset de GEE

**Image: CODED\_Disturbios**

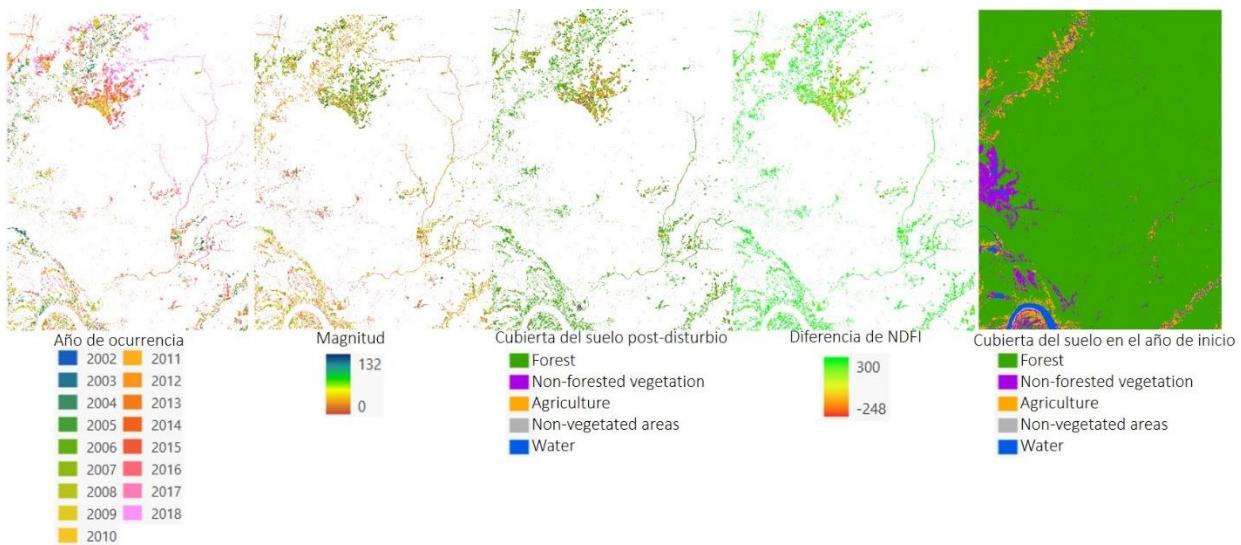
Index	Name	Type	Dimensions	CRS	Nominal Scale	Min	Max
0	dist_1	signed int16	981x1287 px	EPSG:4326	30.00000000000004	2002	2018
1	mag_1	signed int16	981x1287 px	EPSG:4326	30.00000000000004	3	132
2	post_1	signed int16	981x1287 px	EPSG:4326	30.00000000000004	0	5
3	dif_1	signed int16	981x1287 px	EPSG:4326	30.00000000000004	-248	300
4	forest_2000	signed int16	981x1287 px	EPSG:4326	30.00000000000004	1	5

**Image ID**  users/retinta/Tutorials/CODED\_Disturbios

### Ráster (archivo ".tif")

◀ CODED\_Disturbios.tif

- dif\_1
- dist\_1
- forest\_2000
- mag\_1
- post\_1



### 3.1.2. Mapa de degradación y deforestación

Esta sección demuestra cómo ejecutar este [código](#) para producir un mapa clasificado en cuatro clases: bosque, no bosque, degradación y deforestación.

#### Datos de entrada

El usuario tiene que ingresar lo siguiente:

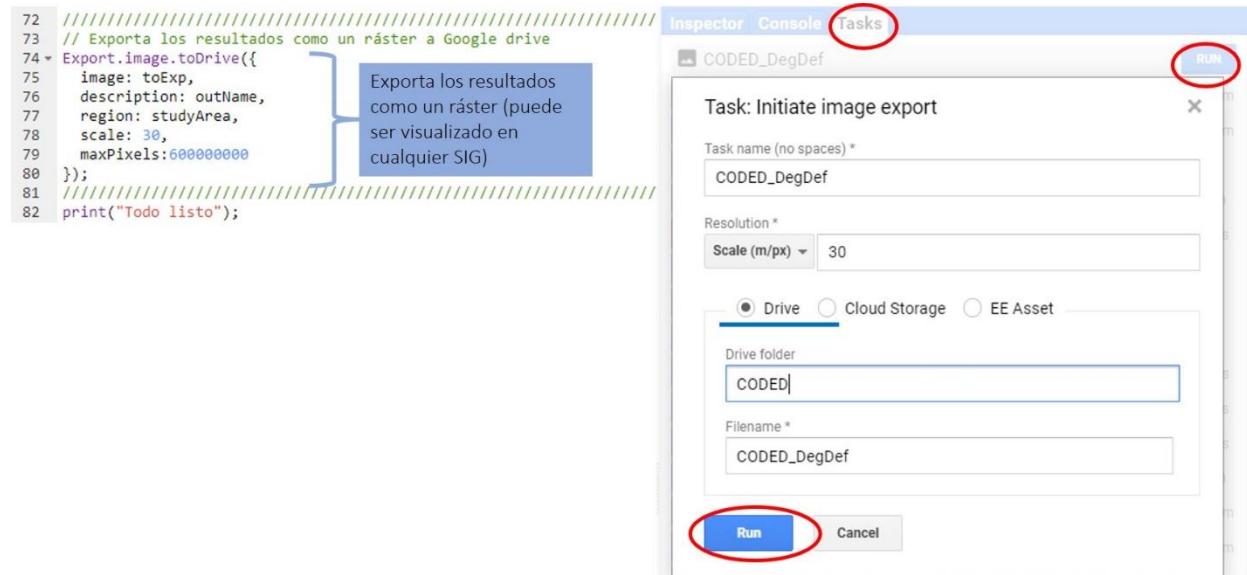
```
CODED_MapDegDef
  ↳ Imports (2 entries)
    ↳ var mapDisturbios: Image users/retinta/Tutorials/CODED_Disturbances (5 bands)
    ↳ var areaEstudio: Table users/retinta/Tutorials/CaseStudy1
1 ////////////////////////////////////////////////////////////////// ALGORITMO CODED //////////////////////////////////////////////////////////////////
2 // Produce un mapa estratificado: bosque/no bosque & degradación/deforestación //////////////////////////////////////////////////////////////////
3 /// Consulte Bullock et al. (2020) para detalles del algoritmo y Bullock (2020) para la implementación en GEE/////
4 /// En particular, este pequeño código para extraer productos CODED fue escrito por Y. Reygadas y V. Galati /////
5
6 ////////////////////////////////////////////////////////////////// Información definida por el usuario //////////////////////////////////////////////////////////////////
7 // Define el área de estudio
8 var studyArea = areaEstudio; ← 1. Área de estudio: polígono del área de estudio.
9
10 // Define el mapa de disturbios que será estratificado
11 var dist_map = mapDisturbios; ← 2. Mapa de disturbios: el producto del código GEE
   "CODED_MapDisturbios"
12
13 // Define un nombre de salida para el mapa estratificado
14 var outName = 'CODED_DegDef'; ← 3. Nombre de salida: nombre deseado para el mapa de
   degradación y deforestación.
15
```

## Ejecución del código

Ejecute el código haciendo clic en *run*. Una vez terminado el proceso, vaya a la consola *task* y haga clic en *run* para exportar los resultados como ráster.

```
CODED_MapaDeg&Def
16 //////////////////////////////////////////////////////////////////// Crea un mapa estratificado ///////////////////////////////////////////////////////////////////
17 // Función para estratificar los resultados de la función principal de CODED //
18 //((https://coded.readthedocs.io/en/latest/coded.html)//)
19
20+ var stratify = function(result) {
21   // Desenmascara las áreas sin cambio
22   var unmasked = result.unmask();
23   // Obtiene el nombre de la banda con la cubierta del suelo en el año de inicio
24   //var bands = result.bandNames();
25
26   // Clasifica BOSQUE
27   // Si en el año de inicio la cubierta del suelo ERA bosque (forest_startyear= 1)
28   // y no ocurrieron disturbios durante el período de estudio (dist_1=0), entonces bosque
29   var forested = unmasked.select('forest_2000').eq(1)
     .and(unmasked.select('dist_1').eq(0));
30
31   // Clasifica NO BOSQUE
32   // Si en el año de inicio la cubierta del suelo NO ERA bosque (forest_startyear ≠ 1)
33   // y no ocurrieron disturbios durante el período de estudio, entonces no bosque y re-etiquetar
34   var nonForested = unmasked.select('forest_2000').neq(1)
     .and(unmasked.select('dist_1').eq(0)).remap([1],[2]);
35
36   // Para los pixeles que no tienen suficientes observaciones después del disturbio, esta línea
37   // establece un umbral para distinguir entre degradación y deforestación
38   var magThreshold = 10;
39
40   // Clasifica DEGRADACIÓN
41   // Si después del primer disturbio, la cubierta del suelo aún es bosque(post_1=1),
42   // o si mag_1 <= magThreshold, entonces degradación y re-etiquetar
43   var degradation = unmasked.select('post_1').eq(1)
     .or(unmasked.select('mag_1').lte(magThreshold))
     .remap([1],[3]);
44
45   // Clasifica DEFORESTACIÓN
46   // Si después del primer disturbio, la cubierta del suelo ya no es bosque (post_1>1),
47   // o si mag_1 > magThreshold, entonces deforestación y re-etiquetar
48   var deforestation = unmasked.select('post_1').gt(1)
     .or(unmasked.select('mag_1').gt(magThreshold))
     .remap([1],[4]);
49
50   // Combina los estratos en una sola banda
51   var strata = ee.Image.cat([forested,nonForested,degradation,deforestation])
     .selfMask()
     .reduce(ee.Reducer.firstNonNull());
52
53   // Crea una paleta de colores para el mapa y despliega el mapa estratificado
54   var strataPalette = ['D9FBEE', 'CCCCCC', '38A800','E600A9'];
55   Map.addLayer(strata,[min:1,max:4,palette:strataPalette],'CODED_Strata');
56   return strata;
57 }
58
59 // Corre la función para crear el mapa estratificado
60 var toExp = stratify(dist_map);
61
62 // Exporta los resultados como un ráster a Google drive
63 Export.image.toDrive({
  image: toExp,
  description: outName,
  region: studyArea,
  scale: 30,
  maxPixels:600000000
});
64
65 print("Todo listo");
66
67 // Exporta los resultados como un ráster a Google drive
68 Export.image.toDrive({
  image: toExp,
  description: outName,
  region: studyArea,
  scale: 30,
  maxPixels:600000000
});
```

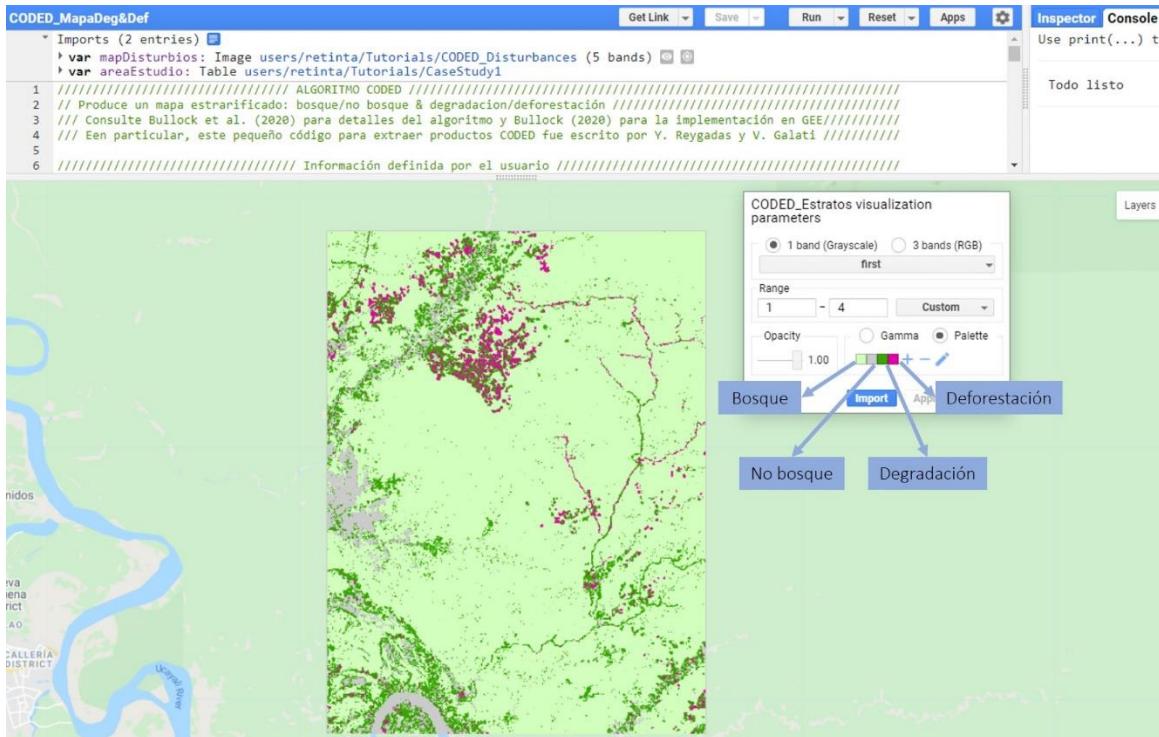
Produce y despliega un mapa clasificado como bosque, no bosque, degradación y deforestación con base en las condiciones descritas en verde



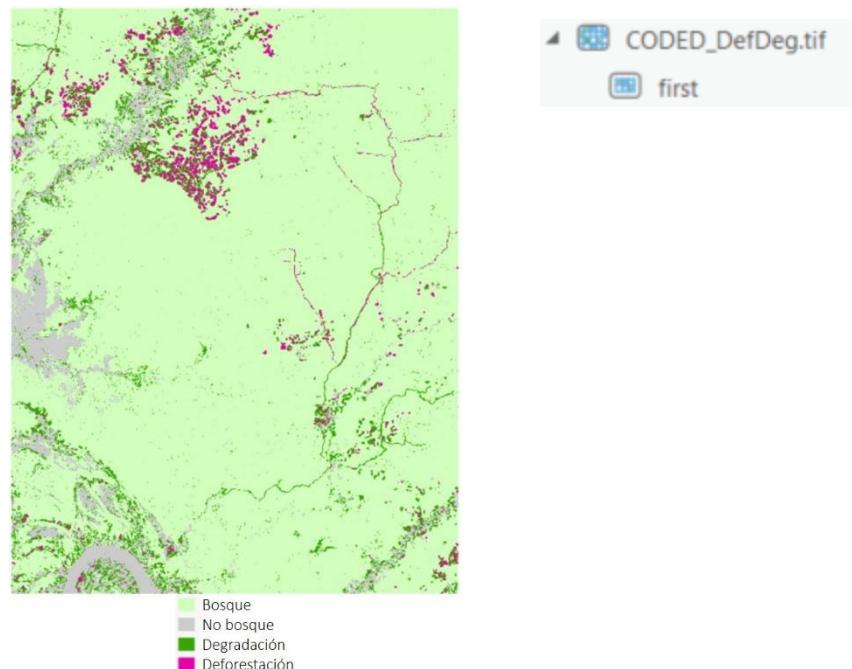
## Productos

Este código produce un mapa clasificado en cuatro clases: bosque, no bosque, degradación y deforestación.

Mapa desplegado en GEE



Ráster (archivo ".tif")



## 3.2. LandTrendr

LandTrendr detecta y caracteriza pérdidas o ganancias de vegetación a través de la segmentación y ajuste de series de tiempo de una variable definida por el usuario (variables posibles: NDFI\*agregada, NBR, NDVI, NDSI, NDMI, TCB, TCG, TCW, TCA, y regiones espectrales equivalentes a las bandas 1-5 and 7 en Landsat TM). Consulte Kennedy et al. (2010) para detalles del algoritmo y Kennedy et al. (2018) para la implementación en GEE.

### 3.2.1. Mapa de disturbios

Esta sección demuestra cómo ejecutar este [código](#) para producir un mapa de disturbios forestales.

#### Datos de entrada

El usuario tiene que ingresar lo siguiente:

```
LT_MapaDisturbios
  * Imports (1 entry)
    var areaEstudio: Table users/retinta/Tutorials/CaseStudy1
  1 //////////////////////////////////////////////////////////////////// LANDTRENDR ALGORITHM ///////////////////////////////////////////////////////////////////
  2 /// Produce un mapa, o mapas anuales, de disturbios con base en parametros específicos /////
  3 /// Consulte Kennedy et al., 2010 para detalles del algoritmo y Kennedy et al., 2018 para la implementación en GEE /////
  4 /// This particular piece of code to obtain LandTrendr outputs was written by Y. Reygadas and V. Galati /////
  5 /// En particular, este pequeño código para extraer productos LandTrendr fue escrito por Y. Reygadas y V. Galati /////
  6
  7 //////////////////////////////////////////////////////////////////// Información definida por el usuario //
  8 // Define las variable(s) de entrada
  9 var indices = ['NDFI']; // Variable(s) para ser segmentada(s)
 10
 11 // Define los parámetros para construir una colección anual Landsat
 12 var startYear = 2000; // Año de inicio
 13 var endYear = 2018; // Año de término
 14 var startDay = '01-01'; // Fecha mínima del rango estacional deseado
 15 var endDay = '12-31'; // Fecha máxima del rango estacional deseado
 16 var aoi = areaEstudio; // Área de estudio
 17 var maskThese = ['cloud', 'shadow', 'snow', 'water']; // Lista de elementos que se quieren ignorar
 18
 19 // Define los parámetros para controlar la segmentación
 20 var runParams = {
 21   maxSegments: 6, // Número máximo de segmentos para ser ajustados en la serie de tiempo
 22   spikeThreshold: 1, // Umbral para moderar anomalías (1 significa moderación nula)
 23   vertexCountOvershoot: 3, // La detección inicial de vértices potenciales puede exceder el maxSegments+1 este número de vértices
 24   preventOneYearRecovery: false, // Evita segmentos que representen recuperación
 25   recoveryThreshold: 1, // Durante el ajuste, si un segmento tiene una tasa de cambio menor que 1/año
 26   pvalThreshold: 0.05, // Si el p-value del modelo mejor ajustado excede este umbral, el segmento es descartado y otro es ajustado usando el método
 27   bestModelProportion: 0.75, // Toma el modelo con más vértices que tenga un p-value menor que el umbral
 28   minObservationsNeeded: 3 // Número mínimo de observaciones necesarias para realizar un ajuste
 29 };
 30
 31 // Define parámetros para filtrar cambios en la vegetación con base en:
 32 var changeParams = {
 33   delta: 'loss', // Pérdida o ganancia
 34   sort: 'greatest', // Tipo (greatest, least, newest, oldest, fastest, or slowest)
 35   year: {checked:true, start:startYear, end:endYear}, // Periodo de tiempo (en años)
 36   mag: {checked:true, value:150, operator:'>'}, // Magnitud (si la variable de entrada es un número)
 37   dur: {checked:true, value:19, operator:'<'}, // Duración (en años)
 38   preval: {checked:true, value:150, operator:'>'}, // Valores espectra pre-cambio (si la variable de entrada es un número)
 39   mmu: {checked:true, value:3}, // Tamaño mínimo de áreas perturbadas (número de pixels)
 40 };
 41
 42 // Define el nombre de salida que será colocado después de estos identificadores: LT_(variable de entrada)_map
 43 var outName = "Disturbios";
 44
 45 // Define el folder de salida en Google drive
 46 var outFolder = "LT";
 47
```

1. Variable(s) de entrada que será(n) segmentada(s) y a partir de la(s) cual(es) se detectarán cambios en la vegetación
2. Parámetros para construir una colección anual de datos Landsat a partir de los cuales se calculan las series de tiempo de la(s) variable(s) de entrada
3. Parámetros para controlar la segmentación de la(s) variable(s) de entrada
4. Parámetros para filtrar los cambios detectados
5. Nombre y folder de salida: el nombre deseado para el mapa de disturbios y el folder para almacenar el mapa en Google drive

## Ejecución del código

Ejecute el código haciendo clic en *run*. Una vez terminado el proceso, vaya a la consola *task* y haga clic en *run* para exportar los resultados como ráster.

```

LT_MapaDisturbios
52 // Llama la librería LandTrendR que incluye NDFI como variable de entrada posible
53 var ltgee = require('users/retinta/Tutorials:LT_Utils_NDFI');
54
55
56 // Función para ejecutar el algoritmo LandTrendr en cada variable de entrada
57 var doLT = function(indices) {
58   for(var i in indices) {
59     // Agrega la variable de entrada al objeto "changeParams"
60     changeParams.index = indices[i];
61     // Ejecuta LandTrendr
62     var LTResult = ltgee.runLT(startYear, endYear, startDay, endDay, aoi, indices[i], [], runParams, maskThese);
63     // Llama la función para obtener un mapa de disturbios
64     combinedMap(LTResult,indices[i]);
65   }
66 }
67
68
69 // Función para obtener un mapa de disturbios por cada variable de entrada
70 var combinedMap = function(result,index) {
71   changeParams.year = [checked:true, start:startYear, end:endYear];
72   // Genera las capas con las características de los disturbios (yod, mag, dur, preval, rate, dsnr)
73   var changeImg = ltgee.getChangeMap(result, changeParams);
74   // Despliega el mapa (específicamente la capa del año de ocurrencia del disturbio -yor)
75   Map.addLayer(changeImg.select(['yod']).clip(aoi), yodVizParms, String(index)+' multianual');
76   //Exporta los resultados
77   exportResults(changeImg,index,outName,outFolder);
78 }
79
80
81 // Función para exportar los resultados
82 var exportResults = function(img, index, name,folder){
83   Export.image.toDrive({
84     image: img.clip(aoi).unmask(0).short(),
85     description: 'LT_'+String(index)+'_'+name,
86     folder: folder,
87     region: aoi,
88     scale: 30,
89     maxPixels: 10000000000000
90   });
91 }
92
93
94
95 // Define los parámetros para visualizar los maps (año de ocurrencia del disturbio)
96 var palette = ['#440003', '#4B0082', '#0000FF', '#00FF00', '#FFFF00', '#FF7F00', '#FF0000'];
97 var yodVizParms = {
98   min: startYear,
99   max: endYear,
100   palette: palette
101 };
102
103 // Centro el mapa en el área de estudio
104 Map.centerObject(aoi,10);
105
106 // Ejecuta todo /**
107 doLT(indices);
108
109 print("Todo listo");

```

Llama la librería LandTrendr que también soporta NDFI como variable de entrada

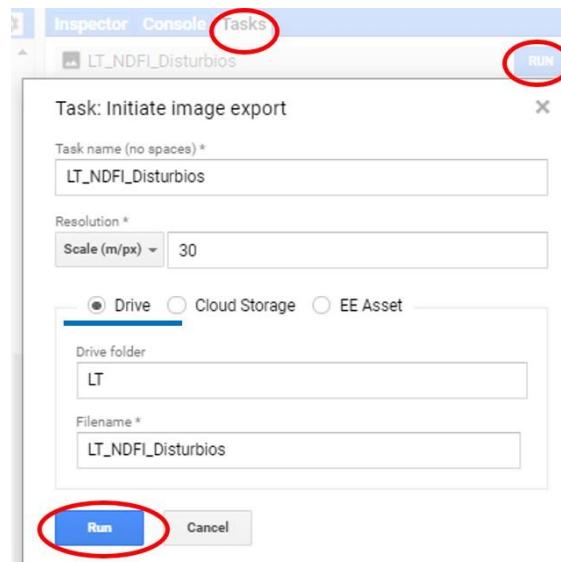
Segmenta la(s) variable(s) de entrada

Produce un mapa de disturbios con los filtros definidos por el usuario

Exporta los resultados como rásters

Configura los parámetros para visualizar el mapa

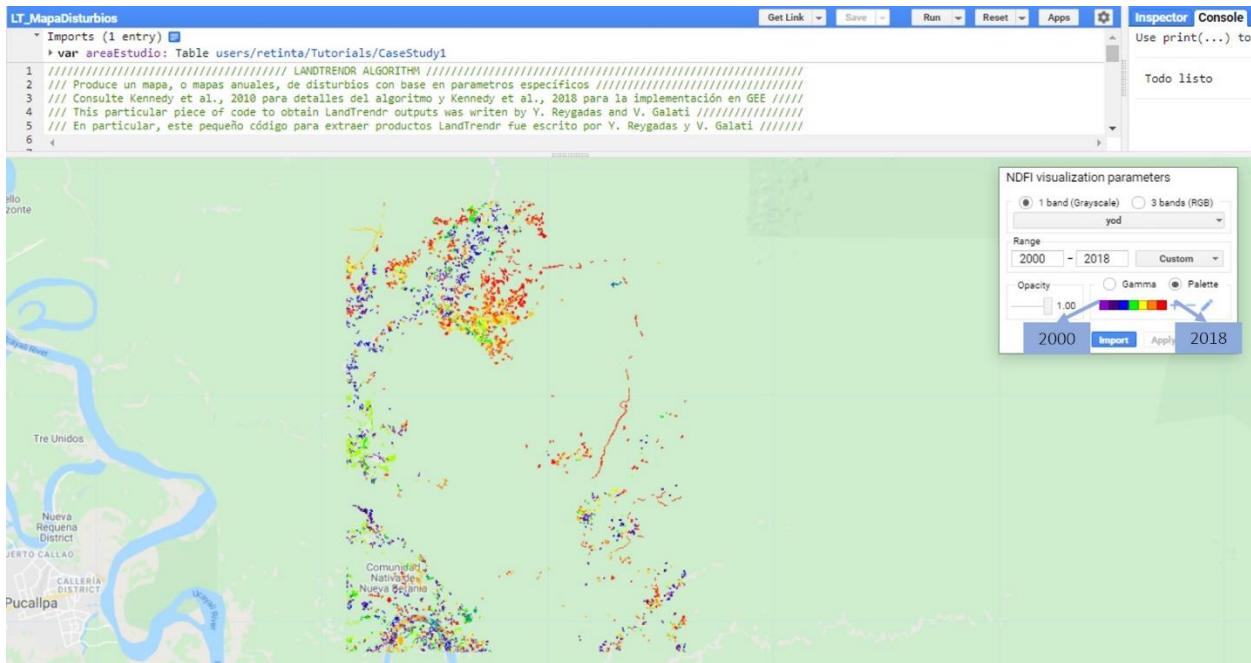
Ejecuta todas las funciones



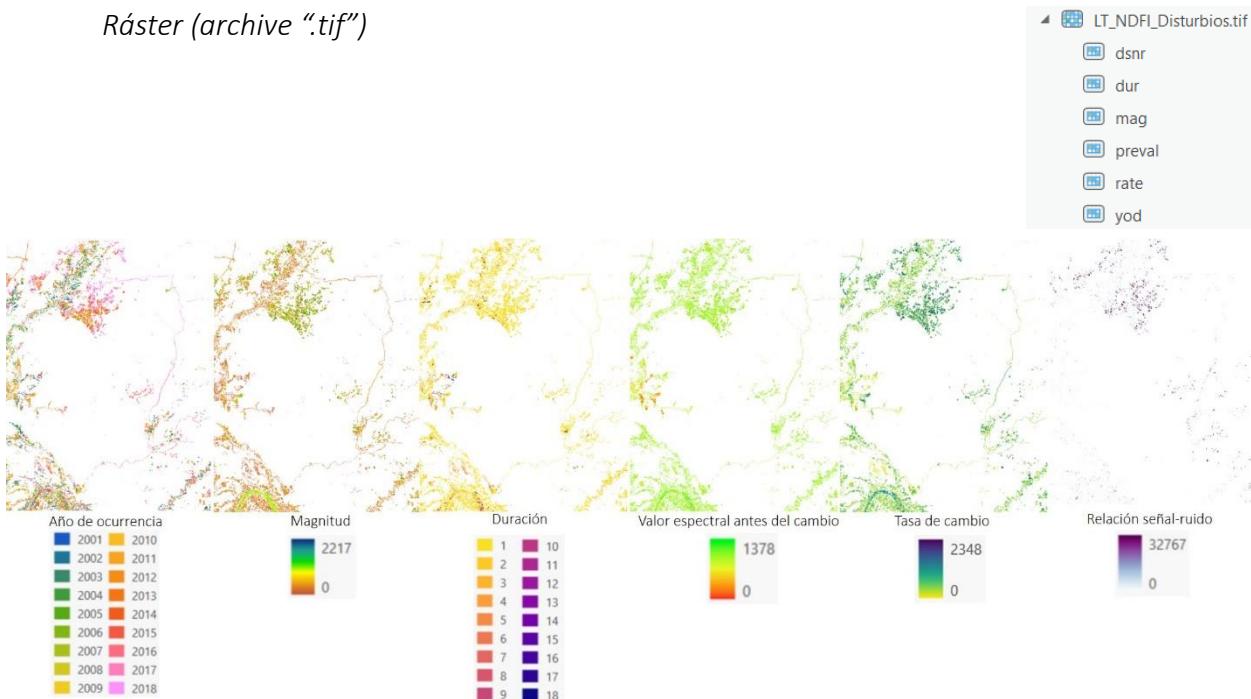
## Productos

Este código produce un mapa de disturbios forestales con seis capas: año de ocurrencia, magnitud del cambio, duración, valor espectral antes del cambio, tasa de cambio y relación señal-ruido.

*Mapa desplegado en GEE (solo la capa “año de ocurrencia”)*



*Ráster (archive “.tif”)*



### 3.3. MTDD

Este algoritmo basado en MTDD clasifica áreas inicialmente forestales como intactas, degradadas o deforestadas entrenando un modelo *random forest* con sesenta y seis métricas derivadas de seis series de tiempo anuales series (i.e., NDVI, dos SWIR regiones espectrales, dos índices NDWI y SAVI) de las cuales se calculan once estadísticas descriptivas (i.e., mínimo, máximo, rango, promedio, desviación estándar, coeficiente de variación, curtosis, sesgo, pendiente, pendiente máxima en 5 años y valor más reciente). Este código GEE fue construido con base en Wang et al. (2019) y adaptado a la ASO.

#### 3.3.1. *Muestras para el entrenamiento*

Esta sección demuestra cómo ejecutar este [código](#) para producir las muestras que posteriormente entrenarán el clasificador *random forest*.

Las muestras son seleccionadas aleatoriamente con base en las condiciones siguientes:

Clase	Condición	Datos de referencia
Intacto	Cubierta forestal en todo el periodo de estudio	MapBiomas (MapBiomas, 2020)
	Sin deforestación en todo el periodo de estudio	RAISG (RAISG, 2020)
	Sin pérdida de bosque en todo el periodo de estudio	GFC (Hansen et al., 2013)
	Cubierta arbórea mayor a 75% <sup>a</sup> en 2000	GFC (Hansen et al., 2013)
	Cubierta arbórea mayor a 75% <sup>a</sup> en 2015	GFCC (Sexton et al., 2013)
Degradado	Cubierta forestal en el último año	MapBiomas (MapBiomas, 2020)
	Deforestado en algún momento del periodo de estudio	RAISG (RAISG, 2020)
	Cubierta arbórea mayor a 30% <sup>b</sup> y menor a 75% <sup>a</sup> en 2015	GFCC (Sexton et al., 2013)
Deforestado	Cubierta no forestal en el último año	MapBiomas (MapBiomas, 2020)
	Deforestado en algún momento del periodo de estudio	RAISG (RAISG, 2020)
	Pérdida de bosque en algún momento del periodo de estudio	GFC (Hansen et al., 2013)

Los umbrales de cubierta arbórea <sup>a</sup> y <sup>b</sup> pueden ser modificados en el código GEE

## Datos de entrada

El usuario tiene que ingresar lo siguiente:

```
1 ////////////////////////////////////////////////////////////////// ALGORITMO MDTT //////////////////////////////////////////////////////////////////
2 ////////////////////////////////////////////////////////////////// Muestreo //////////////////////////////////////////////////////////////////
3 // Produce un número definido por el usuario de muestras (90% para entrenamiento y 10% para validación) //////////////////////////////////////////////////////////////////
4 // de bosque intacto, bosque degradado, y áreas deforestadas. Si el archivo de salida contiene menos //////////////////////////////////////////////////////////////////
5 // muestras de las esperadas, es porque el código se asegura de que las muestras para entrenamiento //////////////////////////////////////////////////////////////////
6 // no estén correlacionadas con las muestras para validación a través de la eliminación de muestras //////////////////////////////////////////////////////////////////
7 // que estén a menos de cierta distancia una de la otra //////////////////////////////////////////////////////////////////
8 //////////////////////////////////////////////////////////////////
9 // Con base en Wang et al. (2019) y adaptado por Reygadas et al. 2021 a la Amazonia Sudoccidental //////////////////////////////////////////////////////////////////
10 ////////////////////////////////////////////////////////////////// para detectar bosque intacto, bosque perturbado y áreas deforestadas //////////////////////////////////////////////////////////////////
11 //////////////////////////////////////////////////////////////////
12 ////////////////////////////////////////////////////////////////// Información definida por el usuario //////////////////////////////////////////////////////////////////
13 //////////////////////////////////////////////////////////////////
14 // Define el periodo de estudio
15 var startYear = 2000; } 1. Período de estudio: año inicial y final
16 var endYear = 2018;
17
18 // Define el número de puntos de muestreo por clase:
19 // bosque intacto, bosque degradado y área deforestada
20 // (90% se usarán para entrenamiento y 10% para validación)
21 var samPoints = 500; } 2. Puntos de muestreo: número de puntos de muestreo por clase
22
23 // Define el área de estudio
24 var aoi = areaEstudio; } 3. Área de estudio: polígono del área de estudio
25
26 // Define las bases de datos usadas para construir los condicionales para detectar bosque intacto, bosque degradado y áreas deforestadas
27 var gfw = GFW; // Global Forest Watch (GFC)
28 var gfcc = GFCC_TC; // Global Forest Cover Change (GFCC) Tree Cover Multi-Year Global 30m
29 var mapBiomas = MAPBIOMAS; // Cubierta y uso del suelo, MapBiomas
30 var raosg = RAISG; // Deforestación, RAISG } 4. Bases de datos: referencias a partir de las cuales
31 // se construyen las condiciones (ver tabla de arriba)
32
33 // Define los umbrales de cobertura arbórea "a" y "b" (ver la tabla de la sección 3.3.1 en el tutorial)
34 var tcMaxThres = 75; } 5. Umbral de cobertura arbórea: umbral ("a" y "b") usado en algunas de las condiciones (ver la tabla de arriba)
35 var tcMinThres = 30;
36
37 // Define un nombre de salida que será agregado a algunos identificadores (eg., validationPts_"outputName")
38 var outputName = "MDTT"; } 6. Nombre de salida: nombre deseado para los puntos de muestreo
```

Nota: Algunas de las bases de datos usadas como referencia para construir las condiciones tienen datos hasta 2018. Por lo tanto, el periodo de estudio tiene que estar comprendido entre 2000 y 2018.

## Ejecución del código

Ejecute el código haciendo clic en *run*. Una vez terminado el proceso, vaya a la consola *task* y haga clic en *run* para exportar los resultados como un *asset* de GEE.

```
MTDD_Muestreo
 40 //////////////////////////////////////////////////////////////////// Todas las condiciones ///////////////////////////////////////////////////////////////////
 41
 42 //////////////////////////////////////////////////////////////////// Deforestado/no deforestado en algún momento del periodo de estudio (RAISG) /////
 43 var rStartYear; // RAISG starts in 2001, this conditional selects 2001 as startYear if the actual startYear is lower (2000)
 44 if (startYear < 2001) {rStartYear = 2001;} else {rStartYear = startYear}
 45 var def = raig.lte(endYear).and(raig.gte(rStartYear));//Deforestación aparece como 1
 46 var noDef = def.not();// No deforestación aparece como 1
 47
 48 //////////////////////////////////////////////////////////////////// Pérdida/no pérdida de bosque en algún momento del periodo de estudio (GFW) /////
 49 var endYearN = endYear-2000; // convierte el año final en un número de dos dígitos
 50 var startYearN = startYear-2000; // convierte el año inicial en un número de dos dígitos
 51 var loss = gfw.select("lossyear").lte(endYearN).and(gfw.select("lossyear").gte(startYearN)).unmask();
 52 var noLoss = loss.not();
 53
 54 //////////////////////////////////////////////////////////////////// Cubierta arbórea mayor a 75% ("a") en 2000 (GFW) /////
 55 var tc75_2000 = gfw.select("treecover2000").gt(tcMaxThres);
 56
 57 //////////////////////////////////////////////////////////////////// Criterios de cubierta arbórea (GFCC): /////
 58 // GFCC tiene datos para 2005, 2010 and 2015, por lo tanto
 59 // construye un condicional para seleccionar los datos más cercanos al año final del periodo de estudio
 60 var tcYear;
 61 if (endYear < 2007) {tcYear = 2005;}
 62 else if (endYear >= 2008 && endYear <= 2012) {tcYear = 2010;}
 63 else if (endYear > 2013) {tcYear = 2015;}
 64 // Función para aplicar un filtro que reduce el ruido en GFCC (especialmente los años 2005 y 2010 tienen mucho ruido)
 65 var filter = function (image) {
 66   // Aplica un filtro de 90m X 90m
 67   var imageFiltered = image.focal_mode();
 68   // Esto es solo con propósitos de visualización. La reproyección es suficiente para forzar que el cálculo ocurra a la escala nativa
 69   return imageFiltered.reproject('EPSG:4326', null, 30);
 70 };
 71
 72 //////////////////////////////////////////////////////////////////// Cubierta arbórea mayor a 75% ("a") en 2015 (o cerca del año final del periodo de estudio) (GFCC)
 73 var tc75_endYear = gfc.filterBounds(aoi).select("tree_canopy_cover")
 74   .filterMetadata("year","equals",tcYear).mean().gt(tcMaxThres); // mean es solo para convertirlo en una sola imagen
 75 var tc75_endYear_fil = filter (tc75_endYear);
 76
 77 //////////////////////////////////////////////////////////////////// Cubierta arbórea mayor o igual a 30% ("b") y menor o igual a 75% ("a") en 2015 (o cerca del año final del periodo de estudio) (GFCC)
 78 var tc30_endYear = gfc.filterBounds(aoi).select("tree_canopy_cover")
 79   .filterMetadata("year","equals",tcYear).mean().gt(tcMinThres); // mean es solo para convertirlo en una sola imagen
 80 var tcless75_endYear = gfc.filterBounds(aoi).select("tree_canopy_cover")
 81   .filterMetadata("year","equals",tcYear).mean().lte(tcMaxThres);
 82 var tc30_75_endYear = tc30_endYear.and(tcless75_endYear);
 83 var tc30_75_endYear_fil = filter (tc30_75_endYear);
 84
 85 //////////////////////////////////////////////////////////////////// Cubierta forestal durante todo el periodo de estudio (MapBiomas) /////
 86 var rename = function (image){// Función para asignarle el mismo nombre a todas las bandas
 87   return image.select(0).rename("bl");
 88 var mapBiomas_r= mapBiomas.map(rename);
 89 var mapBiomas_fil = mapBiomas_r.filter(ee.Filter.and// Crea una colección con el periodo de estudio (solo es posible entre 2000 y 2018)
 90   ee.Filter.gte('year', startYear),
 91   ee.Filter.lte('year', endYear));
 92 var mapBiomas_minMax = mapBiomas_fil.reduce(ee.Reducer.minMax()).rename("min","max");
 93 var mapBiomas_range = mapBiomas_minMax.expression('b\'(\min\')-b\'(\max\')').rename('range');
 94 var forestAll = mapBiomas_minMax.select("range").lte(6)//https://storage.googleapis.com/mapbiomas-public/RAISG/COLECAO2/LEGENDA/codigo_de_la_leyenda_colección-2.pdf
 95   .and(mapBiomas_range.eq(0)).rename("allForest");// bosque es 3 o 6 en toda la región Acre-Ucayali durante todos los años
 96
 97 //////////////////////////////////////////////////////////////////// Bosque/no bosque en el año final del periodo de estudio (MapBiomas) /////
 98 var forestEndYear = mapBiomas_r.filter(ee.Filter.eq("year",endYear)).mean().lte(6); // mean es solo para convertirlo en una sola imagen
 99 var noForestEndYear = forestEndYear.not();
100
101 //////////////////////////////////////////////////////////////////// Selección de áreas de bosque intacto, bosque perturbado, y áreas deforestadas con base en los criterios anteriores /////
102
103 //////////////////////////////////////////////////////////////////// Bosque intacto (1)
104 // (condiciones: nunca clasificado como deforestado, CA>75% en 2000,
105 // sin pérdida de bosque, CA>75% en 2015 -o cerca del año final del periodo de estudio-
106 // y clasificado como bosque durante todo el periodo de estudio)
107 var intactForest = noDef.and(tc75_2000).and(noLoss)
 108   .and(tc75_endYear).and(forestAll).clip(aoi);
109
110 //////////////////////////////////////////////////////////////////// Bosque degradado (2)
111 // (condiciones: deforestado en algún momento del periodo de estudio, CA>=30<=75% en 2015 -o cerca del año final del periodo
112 // y clasificado como bosque en el año final del periodo de estudio)
113 var degradedForest = def.and(tc30_75_endYear_fil)
 114   .and(forestEndYear).remap([0,1],[0,2]).clip(aoi);
115
116 //////////////////////////////////////////////////////////////////// Área deforestada (3)
117 // (condiciones: Clasificado como deforestado en algún momento del periodo de estudio, pérdida de bosque en algún momento de
118 // y clasificado como no bosque en el año final del periodo de estudio
119 var deforested = def.and(loss)
 120   .and(noForestEndYear).remap([0,1],[0,3]).clip(aoi);
121
122 //////////////////////////////////////////////////////////////////// Une todas las clases y despliega los resultados en el mapa (Image.cat para combinar provoca un error en la obtención de los puntos)
123 var IntactDegDef = intactForest.add(degradedForest).remap([1,2,3],[1,2,3]).rename("class");
124 Map.addLayer(IntactDegDef, {"opacity":1,"bands":["class"],"palette":["#a0ff95","fff6131","b05aff"]}, "Bosque intacto, Bosque degradado");
125 Map.centerObject(aoi,10);
126
```

Construye cada una de las condiciones descritas en la tabla de arriba

Selecciona bosque intacto, bosque degradado y áreas deforestadas con base en el conjunto de condiciones descritas en verde y en la tabla de arriba

Despliega bosque intacto, bosque degradado y áreas deforestadas en el mapa

```
128 ////////////////////////////////////////////////////////////////// Muestreo aleatorio //////////////////////////////////////////////////////////////////
129
130 // Crea un muestreo aleatorio con base en el número de muestras por clase definido por el usuario
131 var samplingPts = IntactDefDef.stratifiedSample({
132   numPoints: 0, // puntos para valores de pixeles diferentes de 1,2 o 3
133   seed:0,
134   classValues: [1,2,3], // valores de pixeles a muestrear
135   classPoints: [samPoints,samPoints,samPoints], // número de puntos de muestreo por clase
136   scale: 30, // Resolución espacial Landsat
137   geometries: true, // Retiene la información espacial de los puntos de muestreo, esto se necesita para el "spatial join"
138 });
139
140 // Divide los puntos de muestreo en puntos para entrenamiento y puntos para validación
141 samplingPts = samplingPts.randomColumn(); // agrega una columna (llamada random) de números aleatorios uniformes
142 var split = 0.9; // aproximadamente 90% para entrenamiento, 10% para validación
143 var training = samplingPts.filter(ee.Filter.lt('random', split));
144 var validation = samplingPts.filter(ee.Filter.gte('random', split));
145
146 // Asegura que los puntos de entrenamiento no estén correlacionados con los puntos de validación
147 // a través de la eliminación de muestras que estén a menos de 500 metros una de la otra
148 var distFilter = ee.Filter.withinDistance({//spatial join
149   distance: 500,
150   leftField: '.geo',
151   rightField: '.geo',
152   maxError: 10
153 });
154 var join = ee.Join.inverted();
155 training = join.apply(training, validation, distFilter); // aplica el "spatial join"
156
157 ////////////////////////////////////////////////////////////////// Imprime y exporta los resultados //////////////////////////////////////////////////////////////////
158
159 // Imprime el número de puntos de entrenamiento y los despliega en el mapa
160 print("Puntos de entrenamiento", training.size());
161 //print(training);
162 Map.addLayer(training, {}, "Puntos de entrenamiento");
163
164 print("Puntos de validación", validation.size());
165 //print(validation);
166 Map.addLayer(validation, {}, "Puntos de validación");
167
168 // Export.table.toAsset({collection:training,
169 //   description:"PuntosEntrenamiento_" + outputName,
170 //   assetId: "ptsEntrenamiento_" + outputName
171 // });
172
173 // Export.table.toAsset({collection: validation,
174 //   description:"PuntosValidacion_" + outputName,
175 //   assetId: "ptsValidacion_" + outputName
176 // });
177
178 print("Todo listo!");
179
```

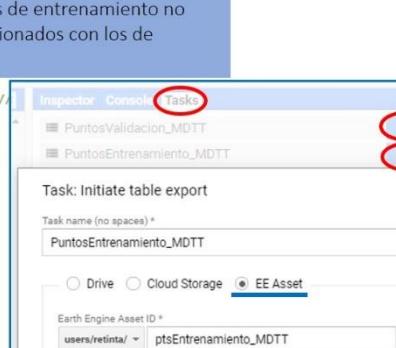
Selecciona aleatoriamente el número de muestras por clase definido por el usuario

Divide los puntos de muestreo en puntos de entrenamiento (90%) y validación (10%)

Elimina muestras que estén a menos de 500 metros una de la otra para asegurar que los puntos de entrenamiento no estén correlacionados con los de validación.

Imprime en la consola el número de puntos de entrenamiento y validación resultantes y los despliega en el mapa

Exporta los resultados como assets de GEE (datos de entrada para clasificar bosque intacto, bosque degradado, y áreas deforestadas)



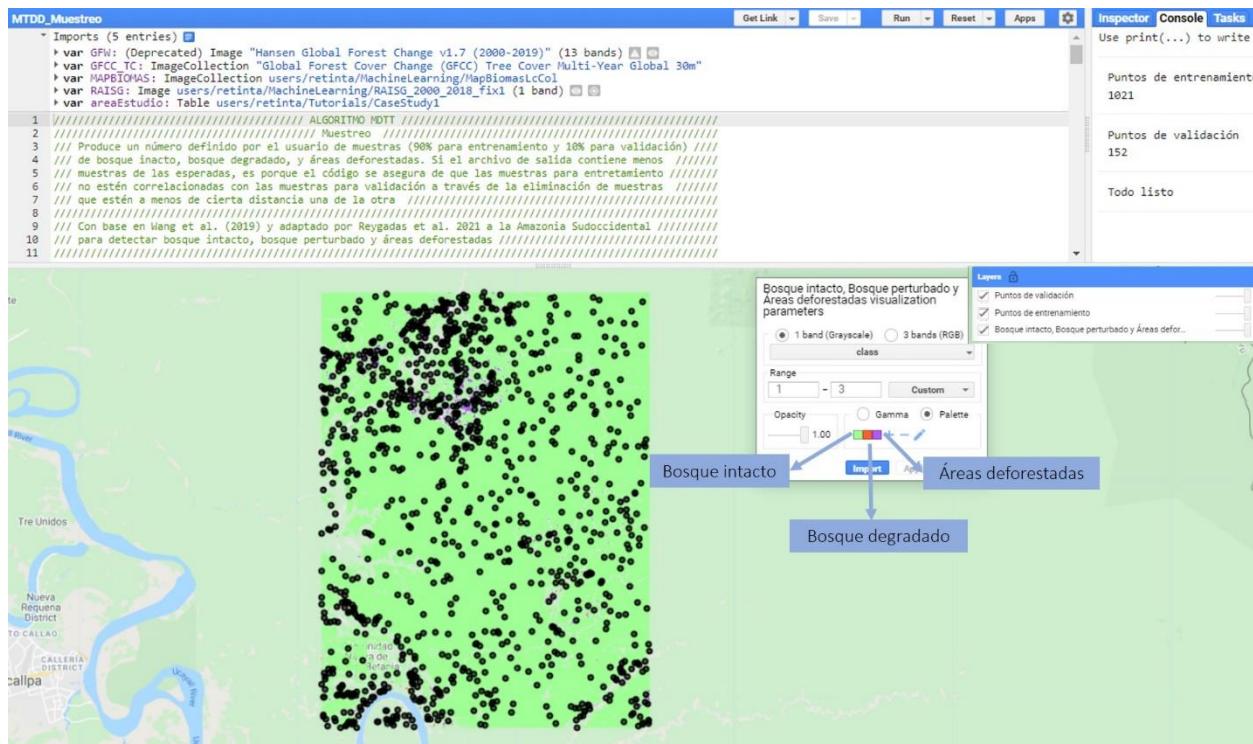
## *Productos*

Este código produce dos archivos de puntos: uno para entrenar y otro para validar.

Assets de GEE

Table: ptsEntrenamiento_MDTT				Table: ptsValidacion_MDTT			
DESCRIPTION		FEATURES		DESCRIPTION		FEATURES	
Feature Index	class (Long)	random (Float)	system:index (String)	Feature Index	class (Long)	random (Float)	system:index (String)
0	1	0.5397374669840993		0	1	0.9287363229116009	
1	1	0.17310481412728573		1	1	0.9962982589520231	
2	1	0.3011276520055456		2	1	0.9668643293210513	
3	1	0.5262166033693878		3	1	0.9393284494480373	
4	1	0.08721550293471602		4	1	0.952910424892686	

## Mapa desplegado en GEE



### 3.3.2. Mapa de degradación y deforestación

Este mapa demuestra cómo ejecutar este [código](#) para producir un mapa clasificado como bosque intacto, degradación y deforestación

#### Datos de entrada

El usuario tiene que ingresar lo siguiente:

```
MTDD_MapaDisturbios
 1 ////////////////////////////////////////////////////////////////// ALGORITMO MTDD //////////////////////////////////////////////////////////////////
 2 ////////////////////////////////////////////////////////////////// Clasificación //////////////////////////////////////////////////////////////////
 3 /// Produce una imagen clasificada en No bosque (0), Bosque intacto (1), Bosque degradado (2),
 4 /// y Deforestación (3), también valida el modelo y exporta los resultados en formato csv //////////////////////////////////////////////////////////////////
 5 /// Con base en Wang et al. (2019) y adaptado por Reygadas et al. 2021 a la Amazonía Sudoccidental //
 6 /// para detectar bosque intacto, bosque perturbado y áreas deforestadas //////////////////////////////////////////////////////////////////
 7 //////////////////////////////////////////////////////////////////
 8 //////////////////////////////////////////////////////////////////
 9 // Vaya a este link antes de comenzar: https://code.earthengine.google.com/?accept\_repo=users/emaprlab/public;
10 // Si se desea desplegar la clasificación en el mapa e imprimir los resultados de la validación en la consola,
11 // descomente las líneas 267-270 & 284-285. Si el área de estudio es grande, esta acción usualmente provoca uno de estos errores "time computation" or "memory limit"
12 ////////////////////////////////////////////////////////////////// Información definida por el usuario //////////////////////////////////////////////////////////////////
13 //////////////////////////////////////////////////////////////////
14 // Define los años del muestreo (años posibles: 2000-2018)
15 var startYearT = 2000;
16 var endYearT = 2018;
17 // Define los años para la clasificación (possible years 2000-present)
18 var startYearC = 2000;
19 var endYearC = 2020;
20 // Define los parámetros para construir una colección de imágenes Landsat
21 var startDayB = '01-01';
22 var endDayB = '12-31';
23 var aoiB = areaEstudio;
24 var maskTheseB = ['cloud', 'shadow', 'snow', 'water'];
25 // Define las muestras para entrenamiento y validación
26 var trainingPts = pe;
27 var validationPts = pv;
28 // Define los datos de cubierta y uso de suelo para construir un mosaico de imágenes de fondo
29 var mapBiomass = MAPBIOMAS; // Annual Land Cover
30 // Define un nombre de salida que será colocado después de estos identificadores: MTDD_Clasi
31 var outName = "2020";
32 //////////////////////////////////////////////////////////////////
33 //////////////////////////////////////////////////////////////////
34 //////////////////////////////////////////////////////////////////
35 //////////////////////////////////////////////////////////////////
36 //////////////////////////////////////////////////////////////////
37 //////////////////////////////////////////////////////////////////
38 //////////////////////////////////////////////////////////////////
```

1. Período de entrenamiento: año inicial y final usados para generar los puntos de entrenamiento (año inicial y final en el código GEE "MTDD\_Muestreo")

2. Período de clasificación: año inicial y final para la clasificación (puede o no coincidir con los años del periodo de entrenamiento)

3. Parámetros para construir una colección anual de datos Landsat a partir de los cuales se calculan las series de tiempo

4. Puntos de entrenamiento y validación: assets generados con el código GEE "MTDD\_Muestreo"

5. Mapas de cubierta y uso del suelo: usados para delimitar áreas boscosas

6. Nombre de salida: un nombre deseado para el mapa de bosque intacto, bosque degradado y áreas deforestadas

## Ejecución del código

Ejecute el código haciendo clic en *run*. Una vez terminado el proceso, vaya a la consola *task* y haga clic en *run* para exportar los resultados como un *ráster* (mapa clasificado) y como archivos csv (precisión general y matriz de error).

```
MTDD_MapaDisturbios
Get Link | Save | Run | Reset | Apps | 
40 /////////////////////////////////////////////////////////////////// Función para calcular TODAS las métricas
41 /////////////////////////////////////////////////////////////////// Series de tiempo
42 var metricsGrbl = function (startYear, endYear, startDay, endDay, aoi, maskThese){ ← Inicio de la función que calcula las 66 métricas
43
44 /////////////////////////////////////////////////////////////////// Construye un compuesto anual (con base en medoids) de reflectancia de las regiones espectrales
45 // equivalentes a las bandas 1(blue),2(Green),3(Red),4(NIR),5(SWIR1640) y 7(SWIR2130) en Landsat TM
46 var ltgee = require("users/emaplab/public:Modules/LandTrendr.js"); // Llama la librería LandTrendr
47 var annualSRC = ltgee.buildSRCollection(startYear, endYear, startDay, endDay, aoi, maskThese);
48
49 // Función para convertir los valores de reflectancia originales a un rango de 0 a 1
50 // Este no es un paso crucial, pero ayudará a tener un rango de valores más homogéneo entre los índices (NDVI, NDWI, SAVI) y las bandas (B5 and B7)
51 var multiply = function (image) {
52   var multiplication = image.multiply(ee.Image(0.0001));
53   return multiplication.copyProperties(image, ['system:time_start']);
54 };
55 var annualSRC_md = annualSRC.map(multiply);
56
57 // Función para agregar una banda indicando el tiempo (necesaria para calcular las métricas temporales)
58 var createTimeBand = function(image) {
59   // Escala (divide) los milisegundos para evitar pendientes muy pequeñas
60   // al calcular la regresión lineal (metricas temporales)
61   var time = image.metadata('system:time_start').divide(1e18).rename('time');
62   return image.addBands(time);
63 };
64
65 var annualSRC_md_tm = annualSRC_md.map(createTimeBand);
66
67 // Función para calcular los índices y agregarlos como nuevas bandas a cada imagen de la colección
68 var addIndices = function(image) {
69   var ndvi = image.expression('(b('B4')-b('B3'))/(b('B4')+b('B3'))').rename('NDVI');
70   var ndwi2130= image.expression('(b('B4')-b('B7'))/(b('B4')+b('B7'))').rename('NDWI2130');
71   var ndwi1640= image.expression('(b('B4')-b('B5'))/(b('B4')+b('B5'))').rename('NDWI1640');
72   var savi= image.expression('1.5*(b('B4')-b('B3'))/(b('B4')+b('B3')+0.5)').rename('SAVI');
73   return image.addBands([ndvi,ndwi2130,ndwi1640,savi]);
74 };
75 var annualSRC_md_tm_indices = annualSRC_md_tm.map(addIndices);
76
```

Construye una colección con series de tiempo anuales de las regiones espectrales equivalentes a las bandas 1-5 & 7 en Landsat TM, NDVI, NDWI1, NDWI2 y SAVI

```

77 //////////////////////////////////////////////////////////////////// Estadísticas descriptivas ///////////////////////////////////////////////////////////////////
78
79 ///// Métricas de localización: min, max, rango, promedio /////
80 var min = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','SAVI'])
81 .reduce(ee.Reducer.min());
82 var max = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','SAVI'])
83 .reduce(ee.Reducer.max());
84 var range = max.subtract(min).rename(['B5_range','B7_range','NDVI_range','NDWI2130_range','NDWI1640_range','SAVI_range']);
85 var mean = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
86 .reduce(ee.Reducer.mean());
87
88 ///// Métricas de escala: desviación estándar, C.V., curtosis, sesgo /////
89 var stdDev = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
90 .reduce(ee.Reducer.stdDev());
91 var cv = mean.divide(annualDev).rename(['B5_cv','B7_cv','NDVI_cv','NDWI2130_cv','NDWI1640_cv','SAVI_cv']);
92 var kurt = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
93 .reduce(ee.Reducer.kurtosis());
94 var skew = annualSRC_md_tm_indices.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI'])
95 .reduce(ee.Reducer.skew());
96
97 ///// Métricas temporales: pendiente, pendiente máxima en 5 años, valor más reciente /////
98 // Pendiente (produce dos bandas: 'offset' (y-intersección) y 'scale' (pendiente))
99 var slopeB5 = annualSRC_md_tm_indices.select(['time','B5'])
100 .reduce(ee.Reducer.linearFit()).select('scale').rename('B5_slp');
101 var slopeB7 = annualSRC_md_tm_indices.select(['time','B7'])
102 .reduce(ee.Reducer.linearFit()).select('scale').rename('B7_slp');
103 var slopeNDVI = annualSRC_md_tm_indices.select(['time','NDVI'])
104 .reduce(ee.Reducer.linearFit()).select('scale').rename('NDVI_slp');
105 var slopeNDWI2130 = annualSRC_md_tm_indices.select(['time','NDWI2130'])
106 .reduce(ee.Reducer.linearFit()).select('scale').rename('NDWI2130_slp');
107 var slopeNDWI1640 = annualSRC_md_tm_indices.select(['time','NDWI1640'])
108 .reduce(ee.Reducer.linearFit()).select('scale').rename('NDWI1640_slp');
109 var slopeSAVI = annualSRC_md_tm_indices.select(['time','SAVI'])
110 .reduce(ee.Reducer.linearFit()).select('scale').rename('SAVI_slp');
111
112 // Pendiente máxima (pendiente máxima absoluta derivada de regresiones lineales aplicadas en ventanas móviles de 5-años)
113 // Crea ventanas móviles de 5-años
114 var join = ee.Join.saveAll({// Regresa un "join" que junta cada elemento de la colección anual
115   matchesKey: 'images' // con un grupo de elementos coincidentes en la colección de 5-años
116 }); // La lista de elementos coincidentes se agrega a cada resultado como una propiedad adicional
117 var diffFilter = ee.Filter.maxDifference({// Establece un filtro de 5-años
118   difference: 78892300000, // 2.5 años en milisegundos, el filtro selecciona 2 años antes y después del año objetivo
119   leftField: 'system:time_start', // por lo tanto, las ventanas de los extremos contienen solo 3-4 años
120   rightField: 'system:time_start'
121 });
122 var fiveWindowJoin = join.apply({// Colección en la cual cada imagen es asociada con las imágenes de la ventana de 5-años (ver p
123   primary: annualSRC_md_tm_indices,
124   secondary: annualSRC_md_tm_indices,
125   condition: diffFilter
126 });
127 // Función para calcular la pendiente máxima absoluta de ventanas móviles de 5-años
128 var maxAbsMovSlope = function (invar, outvar){
129   // Calcula pendientes en ventanas de 5-años
130   var mslp = ee.ImageCollection(fiveWindowJoin.map(function(image) {
131     var annualSRC_md_tm_indices = ee.ImageCollection.fromImages(image.get('images'));
132     return ee.Image(image).addBands(annualSRC_md_tm_indices.select(['time',invar])
133       .reduce(ee.Reducer.linearFit()));
134   }));
135   // Elimina las ventanas extremas que solo contienen 3 o 4 años
136   var range1 = mslp.reduceColumns(ee.Reducer.minMax(), ['system:time_start']); // Obtiene el rango de fechas de las imágenes en la
137   var filter1 = mslp.filter(ee.Filter.date(range1.get('min')).not());
138   var filter2 = filter1.filter(ee.Filter.date(range1.get('max')).not());
139   var range2 = filter2.reduceColumns(ee.Reducer.minMax(), ['system:time_start']);
140   var filter3 = filter2.filter(ee.Filter.date(range2.get('min')).not());
141   var filter4 = filter3.filter(ee.Filter.date(range2.get('max')).not()); // Imprimir esto para ver las pendientes de todas las ventanas de 5 años
142   // Convierte las pendientes a números absolutos
143   var abs = function (image){return image.select('scale').abs()};
144   // Extrae la pendiente máxima absoluta
145   return filter4.map(abs).reduce(ee.Reducer.max()).rename(outvar);
146 };
147 // Aplica la función "maxAbsMovSlope" a todas las variables
148 var mxSlpB5 = maxAbsMovSlope('B5','B5_mxSlp');
149 var mxSlpB7 = maxAbsMovSlope('B7','B7_mxSlp');
150 var mxSlpNDVI = maxAbsMovSlope('NDVI','NDVI_mxSlp');
151 var mxSlpNDWI2130 = maxAbsMovSlope('NDWI2130','NDWI2130_mxSlp');
152 var mxSlpNDWI1640 = maxAbsMovSlope('NDWI1640','NDWI1640_mxSlp');
153 var mxSlpSAVI = maxAbsMovSlope('SAVI','SAVI_mxSlp');
154
155 // Valor más reciente
156 var rangeDates = annualSRC_md_tm_indices.reduceColumns(ee.Reducer.minMax(), ['system:time_start']);
157 var lastYearFil = annualSRC_md_tm_indices.filter(ee.Filter.date(rangeDates.get('max')));
158 var lastYear = lastYearFil.select(['B5','B7','NDVI','NDWI2130','NDWI1640','SAVI']).toBands();
159 var lastYearVal= lastYear.rename(['B5_last','B7_last','NDVI_last','NDWI2130_last','NDWI1640_last','SAVI_last']);
160
161 ///// Todas las métricas /////
162 // Crea una colección con todas las métricas
163 var metricsCollection = ee.ImageCollection([min,max,rangemean,stdDev, cv,kurt,skew,
164   slopeB5,slopeB7,slopeNDVI,slopeNDWI2130,slopeNDWI1640,slopeSAVI,
165   mxSlpB5,mxSlpB7,mxSlpNDVI,mxSlpNDWI2130,mxSlpNDWI1640,mxSlpSAVI,lastYearVal]);
166
167 // Convierte la colección de métricas en una sola imagen con múltiples bandas
168 var metrics= metricsCollection.toBands();
169 return metrics;
170 };

```

Calcula 11 estadísticas descriptivas (i.e., mínimo, máximo, rango, promedio, desviación estándar, coeficiente de variación, curtosis, sesgo, pendiente, pendiente máxima en 5 años y valor más reciente) para cada una de las series de tiempo (i.e., NDVI, dos SWIR regiones espetrales, dos índices NDWI y SAVI)

Fin de la función que calcula las 66 métricas

```

173 //////////////////////////////////////////////////////////////////// Máscara: delimitación de áreas forestales ///////////////////////////////////////////////////////////////////
174
175 //////////////////////////////////////////////////////////////////// Selecciona áreas que fueron bosque por lo menos 3 años consecutivos durante el periodo de estudio ///////////////////////////////////////////////////////////////////
176 var rename = function (image){// Funcion for assigning the same name to all bands
177   return image.select(0).rename("b1");}
178 var mapBiomas_r= mapBiomas.map(rename);
179 var mapBiomas_fil = mapBiomas_r.filter(ee.Filter.and());//Crea una colección que contiene los años dentro del periodo de estudio (hasta 2018)
180 ee.Filter.gte('year', startYearT),
181 ee.Filter.lte('year', endYearT));
182 // Función para detectar pixeles cubiertos por bosque
183 // https://storage.googleapis.com/mapbiomas-public/RAISG/COLECAO2/LEGENDA/codigo_de_la_leyenda_coleccion-2.pdf
184 // bosque es 3 o 6 en toda la región Acre-Ucayali durante todos los años
185 var forest = function (image) {
186   return image.eq(3).or(image.eq(6)).rename("forest").copyProperties(image, ['year']);
187 var forestCol = mapBiomas.map(forest);
188
189 // Función para añadir una banda "tiempo" (necesaria para extraer la cubierta de suelo por año)
190 var createTimeBandMask = function(image) {
191   var time = image.metadata('year').rename('time');
192   return image.addBands(time);
193 };
194 var forestCol_tm = forestCol.map(createTimeBandMask);
195
196 // Bosque durante 3 años consecutivos
197 // Crea ventanas móviles de 3-años
198 var join1 = ee.Join.saveAll({ matchesKey: 'images'});
199 var diffFilter1 = ee.Filter.maxDifference({ difference: 1.5, leftField: 'year',rightField: 'year'});
200 var threeWindowJoin = join1.apply({primary: forestCol_tm , secondary: forestCol_tm , condition: diffFilter1});
201
202 // Función para detectar cubierta forestal durante por lo menos 3 años consecutivos
203 var forest3years = function (invar, outvar){
204   // Calcula una suma en ventanas de 3-años, si siempre hay bosque la suma será 3
205   var f3y = ee.ImageCollection(threeWindowJoin.map(function(image) {
206     var forestCol_tm = ee.ImageCollection.fromImages(image.get('images'));
207     return ee.Image(image).addBands(forestCol_tm .select(["time",invar]).reduce(ee.Reducer.sum()));
208   }));
209   // Extrae la suma más alta
210   return f3y.reduce(ee.Reducer.max());
211 };
212
213 // Aplica la función para "detectar cubierta forestal durante 3-años consecutivos"
214 var MaxForest3years = forest3years('forest', 'forest_3years').select("forest_sum_max");
215 // Obtiene la máscara de bosque (áreas que fueron bosque por lo menos 3 años consecutivos durante el periodo de estudio)
216 var forestMask= MaxForest3years.gte(3);
217

```

Crea una mascara compuesta de todas las áreas cubiertas por bosque por lo menos tres años consecutivos durante el periodo de estudio

```

218 //////////////////////////////////////////////////////////////////// Clasificador random forest ///////////////////////////////////////////////////////////////////
219
220 // *Crea una colección con todas las métricas con base en los años del muestreo
221 var metricsTraining = metricsGral(startYearT, endYearT,startDayB, endDayB, aoiB, maskTheseB);
222 print("Métricas muestreo", metricsTraining);
223
224 // **Crea una colección con todas las métricas con base en los años para la clasificación
225 var metricsClassification = metricsGral(startYearC, endYearC,startDayB, endDayB, aoiB, maskTheseB);
226 print("Métricas clasificación", metricsClassification);
227
228 // Sobrepone los puntos de entrenamiento con las métricas*
229 var collectMetrics = function (feature){// usar una funcion permite a GEE trabajar con más puntos antes de exceder su capacidad
230   return metricsTraining.sampleRegions({collection:feature,
231   properties: ["class"],
232   scale: 30,
233   tileScale: 2});
234 };
235 var trainingCol = trainingPts.map(collectMetrics);
236 var trainingPoints = trainingCol.flatten();
237
238 // Crea un clasificador random forest con 500 árboles y lo entrena
239 var classifier = ee.Classifier.smileRandomForest({numberOfTrees: 50})
240 .train({features: trainingPoints, classProperty:'class',
241 inputProperties:['0_B5_min','0_B7_min','0_NDVI_min','0_NDWI2130_min','0_NDWI1640_min','0_SAVI_min',
242 '1_B5_max','1_B7_max','1_NDVI_max','1_NDWI2130_max','1_NDWI1640_max','1_SAVI_max',
243 '2_B5_range','2_B7_range','2_NDVI_range','2_NDWI2130_range','2_NDWI1640_range','2_SAVI_range',
244 '3_B5_mean','3_B7_mean','3_NDVI_mean','3_NDWI2130_mean','3_NDWI1640_mean','3_SAVI_mean',
245 '4_B5_stdDev','4_B7_stdDev','4_NDVI_stdDev','4_NDWI2130_stdDev','4_NDWI1640_stdDev','4_SAVI_stdDev',
246 '5_B5_cv','5_B7_cv','5_NDVI_cv','5_NDWI2130_cv','5_SAVI_cv',
247 '6_B5_kurtosis','6_B7_kurtosis','6_NDVI_kurtosis','6_NDWI2130_kurtosis','6_NDWI1640_kurtosis','6_SAVI_kurtosis',
248 '7_B5_skew','7_B7_skew','7_NDVI_skew','7_NDWI2130_skew','7_NDWI1640_skew','7_SAVI_skew',
249 '8_B5_slp','9_B7_slp','10_NDVI_slp','11_NDWI2130_slp','12_NDWI1640_slp','13_SAVI_slp',
250 '14_B5_mxSlp','15_B7_mxSlp','16_NDVI_mxSlp','17_NDWI2130_mxSlp','18_NDWI1640_mxSlp','19_SAVI_mxSlp',
251 '20_B5_last','20_B7_last','20_NDVI_last','20_NDWI2130_last','20_NDWI1640_last','20_SAVI_last'
252 });
253
254
255 // Clasifica las métricas deseadas**
256 var metricsMask = metricsClassification.mask(forestMask); // Enmascara las areas no forestales
257 var classified = metricsMask.classify(classifier); //No bosque (0), bosque intacto (1), bosque degradado (2), áreas deforestadas (3)
258 print("Mapa clasificado");
259

```

Entrena un clasificador *random forest* con las 66 métricas de los datos de entrenamiento y la clasifica las imágenes deseadas

```

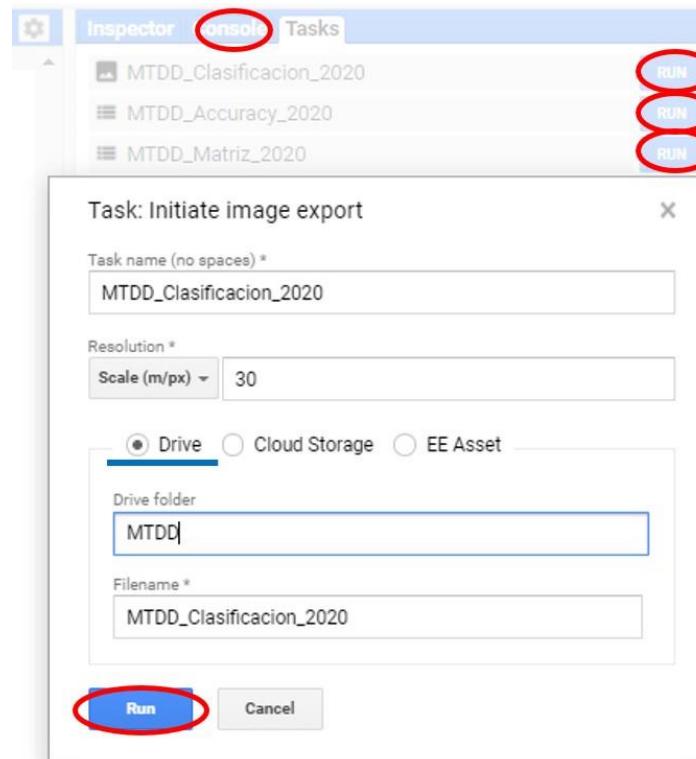
260 ///////////////// Filtro para reducir ruido ///////////////////
261 var classified_fil = classified.focal_mode().mask(forestMask);
262 // Esto es solo con propósitos de visualización. La reproyección es suficiente para forzar que el cálculo ocurra a la escala nativa.
263 var classified_fil_vis = classified_fil.reproject('EPSG:4326', null, 30);
264 // Despliega los resultados
265 //Map.centerObject(aoi,10);
266 //Map.addLayer(classified_fil_vis.clip(aoi),
267 //  {"opacity":1,"bands":[["classification"]],"min":0,"max":3,"palette":["ffffff","7dff5c","0a9b15","ff5514"]},
268 //  "Clasificación");
269
270 ///////////////// Validación ///////////////////
271
272 // Sobrepone los puntos de validación con las métricas*
273 // Overlays the validation points over the metrics
274 var validationCol = validationPts.map(collectMetrics);
275 var validationPoints = validationCol.flatten();
276
277 // Clasifica los puntos de validación
278 var validated = validationPoints.classify(classifier);
279
280 // Genera una matriz de confusión
281 var valAccuracy = validated.errorMatrix("class","classification");
282 //print('Matriz de errores: ', valAccuracy);
283 //print('Precisión general: ', valAccuracy.accuracy());
284
285 ///////////////// Exporta los resultados ///////////////////
286
287 // Exporta la imagen clasificada
288 Export.Image.toDrive({image:classified_fil,
289   description: "MTDD_Clasificacion_" + outName ,
290   fileNamePrefix: "MTDD_Clasificacion_" + outName,
291   region:aoiB,
292   scale:30,
293   maxPixels:600000000
294 });
295
296 // Exporta la matriz de confusión
297 var toExportMatrix = ee.Feature(null, {matrix: valAccuracy.array()});
298 Export.table.toDrive({
299   collection: ee.FeatureCollection(toExportMatrix),
300   description: "MTDD_Matriz_" + outName,
301   fileFormat: 'CSV'
302 });
303
304 // Exporta la precisión general
305 var toExportAccuracy = ee.Feature(null, {matrix: valAccuracy.accuracy()});
306 Export.table.toDrive({
307   collection: ee.FeatureCollection(toExportAccuracy),
308   description: "MTDD_Accuracy_" + outName,
309   fileFormat: 'CSV'
310 });
311
312 });
313
314
315
316 print("Todo listo");

```

Aplica un filtro para reducir ruido

Valida los resultados usando los puntos de validación y genera una matriz de confusión

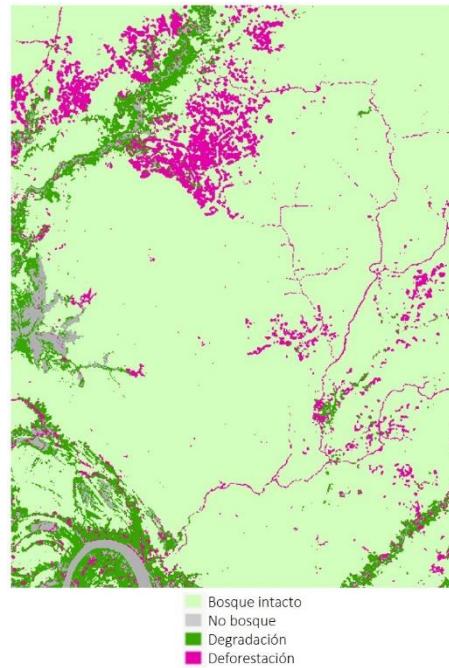
Exporta la clasificación como un raster, y la matriz de confusión y la precision general como un archivo csv



## Productos

Este código produce un mapa clasificado en cuatro clases: no bosque, bosque intacto, degradación y deforestación

Ráster (archive ".tif")



Precisión general y matriz de error (archivos ".csv")

	A	B	C
1	system:in	matrix	.geo
2	0	0.927632	
3			

	A	B	C	D	E	F
1	system:in	matrix	.geo			
2	0	[[0, 0, 0, 0], [0, 60, 0, 0], [0, 0, 47, 4], [0, 1, 6, 34]]				
3						
4						

## Referencias

- Bullock, E., 2020. Continuous Degradation Detection (CODED) — coded 0.2 documentation [WWW Document]. URL <https://coded.readthedocs.io/en/latest/old.html> (accessed 2.9.21).
- Bullock, E.L., Woodcock, C.E., Olofsson, P., 2020. Monitoring tropical forest degradation using spectral unmixing and Landsat time series analysis. *Remote Sens. Environ.* 238, 110968. <https://doi.org/10.1016/j.rse.2018.11.011>
- Cohen, W.B., Healey, S.P., Yang, Z., Stehman, S. V., Brewer, C.K., Brooks, E.B., Gorelick, N., Huang, C., Hughes, M.J., Kennedy, R.E., Loveland, T.R., Moisen, G.G., Schroeder, T.A., Vogelmann, J.E., Woodcock, C.E., Yang, L., Zhu, Z., 2017. How Similar Are Forest Disturbance Maps Derived from Different Landsat Time Series Algorithms? *Forests* 8, 98. <https://doi.org/10.3390/f8040098>
- FAO, 2001. Global Forest Resources Assessment 2000. Rome, Italy.
- Hansen, M.C., Potapov, P. V., Moore, R., Hancher, M., Turubanova, S.A., Tyukavina, A., Thau, D., Stehman, S. V., Goetz, S.J., Loveland, T.R., Kommareddy, A., Egorov, A., Chini, L., Justice, C.O., Townshend, J.R.G., 2013. High-resolution global maps of 21st-century forest cover change. *Science* (80-. ). 342, 850–853. <https://doi.org/10.1126/science.1244693>
- Kennedy, R.E., Yang, Z., Cohen, W.B., 2010. Detecting trends in forest disturbance and recovery using yearly Landsat time series: 1. LandTrendr-Temporal segmentation algorithms. <https://doi.org/10.1016/j.rse.2010.07.008>
- Kennedy, R.E., Yang, Z., Gorelick, N., Braaten, J., Cavalcante, L., Cohen, W.B., Healey, S., 2018. Implementation of the LandTrendr algorithm on Google Earth Engine. *Remote Sens.* 10, 691. <https://doi.org/10.3390/rs10050691>
- MapBiomas, 2020. MapBiomas Amazonia Project-Collection 2.0 of annual land-cover and land-use maps [WWW Document]. URL <https://code.earthengine.google.com/toolkit-download>
- McDowell, N.G., Coops, N.C., Beck, P.S.A., Chambers, J.Q., Gangodagamage, C., Hicke, J.A., Huang, C. ying, Kennedy, R., Kroccheck, D.J., Litvak, M., Meddens, A.J.H., Muss, J., Negrón-Juarez, R., Peng, C., Schwantes, A.M., Swenson, J.J., Vernon, L.J., Williams, A.P., Xu, C., Zhao, M., Running, S.W., Allen, C.D., 2015. Global satellite monitoring of climate-induced vegetation disturbances. *Trends Plant Sci.* <https://doi.org/10.1016/j.tplants.2014.10.008>
- RAISG, 2020. Amazon Geo-Referenced Socio-Environmental Information Network [WWW Document]. URL <https://www.amazoniasocioambiental.org/> (accessed 3.30.21).
- Reygadas, Y., Spera, S., Galati, V., Salisbury, D.S., Silva, S., Novoa, S., 2021. Mapping Forest Disturbances Across the Southwestern Amazon: Evaluation and Comparison of Optical Remote Sensing Algorithms. *Remote Sens. Environ.* En revisión.
- Sasaki, N., Putz, F.E., 2009. Critical need for new definitions of “forest” and “forest degradation” in global climate change agreements. *Conserv. Lett.* 2, 226–232. <https://doi.org/10.1111/j.1755-263X.2009.00067.x>

Schoene, D., Killmann, W., von Lüpke, H., LoycheWilkie, M., 2007. Definitional Issues Related to Reducing Emissions from Deforestation in Developing Countries, Vol. 5. ed. Food and Agriculture Organization of the United Nations, Rome.

Sexton, J.O., Song, X.-P., Feng, M., Noojipady, P., Anand, A., Huang, C., Kim, D.-H., Collins, K.M., Channan, S., Dimiceli, C., Townshend, J.R., 2013. International Journal of Digital Earth Global, 30-m resolution continuous fields of tree cover: Landsat-based rescaling of MODIS vegetation continuous fields with lidar-based estimates of error. *Int. J. Digit. Earth* 6, 427–448. <https://doi.org/10.1080/17538947.2013.786146>

Wang, Y., Ziv, G., Adami, M., Mitchard, E., Batterman, S.A., Buermann, W., Schwantes Marimon, B., Marimon Junior, B.H., Matias Reis, S., Rodrigues, D., Galbraith, D., 2019. Mapping tropical disturbed forests using multi-decadal 30 m optical satellite imagery. *Remote Sens. Environ.* 221, 474–488. <https://doi.org/10.1016/j.rse.2018.11.028>