



## DOCUMENTO DE TRABAJO

# METODOLOGÍA APLICADA EN LA ELABORACIÓN DEL MAPA DE COBERTURA DE BOSQUE NATIVO DE URUGUAY PARA EL AÑO 2016 CON IMÁGENES SENTINEL 2

Proyecto REDD+ Uruguay

Coordinación Técnica: Diego Martino

**Autores principales:** Lucia Bernardi, Alejandra Boccoardo, Carlos Miguel, Juan Olivera, Cecilia Penengo, Gonzalo Rama

**Autores colaboradores:** José Serafini, Adolfo Kindgard

El proyecto REDD+ es ejecutado en el marco de un acuerdo interministerial entre el Ministerio de Ganadería, Agricultura y Pesca y el Ministerio de Vivienda, Ordenamiento Territorial y Medio Ambiente. Como parte de este acuerdo participan directamente en su implementación la Dirección General Forestal, la Oficina de Programación y Políticas Agropecuarias, la Dirección Nacional de Medio Ambiente y la División de Cambio Climático. El proyecto REDD+ cuenta con apoyo financiero del Fondo Cooperativo para el Carbono de los Bosques (FCPF).

Este documento debe citarse como:

Proyecto REDD+ Uruguay (2019). Metodología aplicada en la elaboración del mapa de cobertura de bosque nativo de Uruguay para el año 2016 con imágenes Sentinel 2. Bernardi, L., Boccardo, A., Miguel, C., Olivera, J., Penengo, C., Rama, G., Serafini, J. y Kindgard, A. Ministerio de Ganadería, Agricultura y Pesca - Ministerio de Vivienda, Ordenamiento Territorial y Medio Ambiente. Montevideo.

En este documento se emplea el masculino gramatical, como término inclusivo para aludir a colectivos mixtos, o en contextos genéricos o inespecíficos.

## CONTENIDO:

A.	Introducción	4
B.	Descripción de la metodología aplicada	4
1.	Selección de imágenes satelitales	5
1.1.	Selección de escenas	5
1.2.	Correcciones geométrica, radiométrica y atmosférica	6
1.3.	Detección y enmascaramiento de nubes	6
2.	Composición del mosaico	6
3.	Clasificación supervisada	8
3.1	Selección de muestras de entrenamiento y verificación	8
3.2	Clasificación del mosaico anual de imágenes en base a las muestras de entrenamiento	9
4.	Edición visual de errores de clasificación	10
5.	Descarga de clasificación corregida y mosaico de imágenes	11
6.	Segmentación	12
7.	Asignación de clase por mayoría, ediciones finales y post-procesamiento	13
8.	Validación	16
	Bibliografía	17
	Anexo 1	18
	Anexo 2	21
	Anexo 3	30
	Anexo 4	35

## A. Introducción

En este documento se describe la metodología aplicada para la elaboración del mapa de cobertura de bosque nativo de Uruguay para el año 2016 a partir de imágenes del sensor Sentinel 2.

## B. Descripción de la metodología aplicada

Los pasos seguidos en la aplicación de la metodología se presentan en el siguiente listado y diagrama de procesos:

1. Selección de imágenes satelitales
  - 1.1. Selección de escenas
  - 1.2. Correcciones geométrica, radiométrica y atmosférica
  - 1.3. Detección y enmascaramiento de nubes
2. Composición de mosaico
3. Clasificación supervisada
  - 3.1. Selección de muestras de entrenamiento
  - 3.2. Clasificación de mosaico de imágenes en base a muestras
4. Edición visual de errores de clasificación
5. Descarga de clasificación corregida y mosaico de imágenes
6. Segmentación de imágenes
7. Asignación de clase por mayoría y ediciones finales
8. Validación

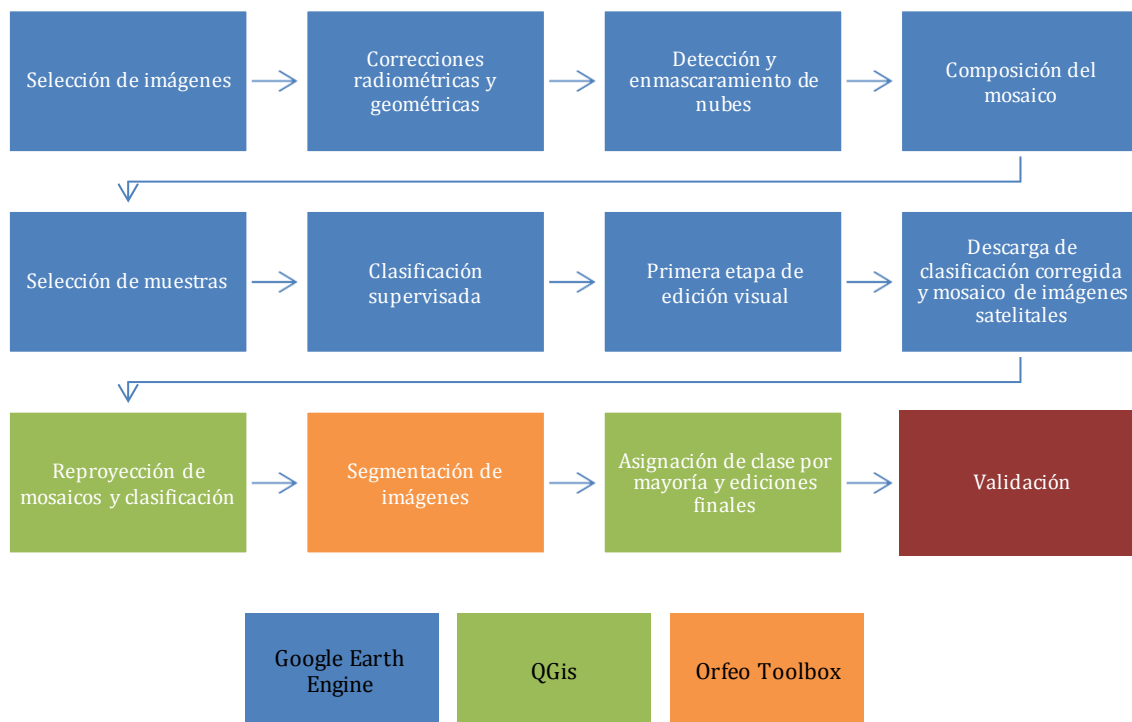


Figura 1. Diagrama de proceso

## 1. Selección de imágenes satelitales

Una imagen satelital o imagen de satélite se puede definir como la representación visual de la información capturada por un sensor montado en un satélite artificial (Chuvieco, 1995).

Los sensores utilizados para la elaboración de los mapas de cobertura de bosque nativo de Uruguay para el año 2016 fueron Landsat 8 y Sentinel 2. En este documento se incluye la descripción de la metodología aplicada para el mapa realizado utilizando el sensor Sentinel 2.

### 1.1. Selección de escenas

La selección de las escenas se realizó teniendo en cuenta factores como la presencia de nubes y la mayor cercanía posible entre las fechas de las distintas imágenes seleccionadas, tomando como referencia el período entre los meses de noviembre y enero<sup>1</sup>. Se eligen fechas próximas a la primavera para que la señal espectral que refleja la actividad fotosintética del bosque nativo sea clara. De esta forma, se evita el período de senescencia foliar de muchas especies que lo componen y se disminuye los errores de clasificación por confusiones con otras coberturas.

La colección de imágenes de Sentinel se estructura de tal forma que, definiendo los parámetros como fecha y filtro de nubes, se seleccionan todas aquellas que cumplan con estos criterios. Para la selección de escenas se desarrolló un script en la plataforma Google Earth Engine (GEE)<sup>2</sup> que se presenta a continuación:

```
37 //Función para adaptar cloudscore a Sentinel
38 var cloudScoreS2 = function(img) {
39     var rescale = img.divide(10000); //esto se hace porque las imagenes Sentinel estan reescaladas
40     var score = cloudScore(rescale).multiply(100).rename('cloudscore');
41     return img.addBands(score);
42 };
43
44 //Función para enmascarar nubes menores a x
45 var cloudthreshold = 60;
46 var maskClouds = function(image) {
47     return image.updateMask(image.select('cloudscore').lte(cloudthreshold));
48 };
49
50
51 //Colecciones Sentinel 2
52 var sentinel2 = ee.ImageCollection('COPERNICUS/S2');
53
54 //Colección Sentinel 2 2017
55 var f2017s = sentinel2.filterBounds(region)
56     .filterDate('2016-01-01', '2016-12-31')
57     .filter(ee.Filter.calendarRange(9,3,"month"))
58     .filterMetadata('CLOUDY_PIXEL_PERCENTAGE', 'less_than', 20);
59
60
61 var m2017s = f2017s.map(cloudScoreS2).map(maskClouds);
62 var m2017s2 = m2017s.median();
63 var c2017s = m2017s2.clip(region);
```

Figura 2. Script utilizado para selección de escenas de Sentinel 2

En el punto 2 de este documento se describe cómo fue realizada la composición del mosaico Sentinel y se muestran los scripts utilizados para ello.

<sup>1</sup> La definición de la ventana temporal a considerar para la selección de las escenas para cada mapa fue tomada durante un taller con expertos nacionales en mapeo, realizado en julio de 2017 en el marco del Proyecto REDD+.

<sup>2</sup> <https://code.earthengine.google.com/>

## 1.2. Correcciones geométrica, radiométrica y atmosférica

Las imágenes Sentinel ya están corregidas geométrica, radiométrica y atmosféricamente al tope de la atmósfera, por lo que no es necesario aplicarle estas correcciones.

## 1.3. Detección y enmascaramiento de nubes

Para la detección y enmascaramiento de nubes de Sentinel, se adaptó la función `landsat.simpleCloudScore`, para que utilice las bandas disponibles en Sentinel. Esta función da una puntuación similar a las nubes en cada pixel de la imagen.

Sentinel tiene los valores TOA re-escalados por 10000 mientras que el algoritmo está diseñado para valores TOA de 0 a 1 por lo tanto se re-escala dividiendo los valores de Sentinel como puede observarse en el siguiente script:

```
12 //Algoritmo cloud score Sentinel
13 var cloudScore = function(img) {
14   // A helper to apply an expression and linearly rescale the output.
15   var rescale = function(img, exp, thresholds) {
16     return img.expression(exp, {img: img})
17       .subtract(thresholds[0]).divide(thresholds[1] - thresholds[0]);
18   };
19   // Compute several indicators of cloudyness and take the minimum of them.
20   var score = ee.Image(1.0);
21   // Clouds are reasonably bright in the blue band.
22   score = score.min(rescale(img, 'img.B2', [0.1, 0.3]));
23
24   // Clouds are reasonably bright in all visible bands.
25   score = score.min(rescale(img, 'img.B4 + img.B3 + img.B2', [0.2, 0.8]));
26
27   // Clouds are reasonably bright in all infrared bands.
28   score = score.min(
29     rescale(img, 'img.B8 + img.B11 + img.B12', [0.3, 0.8]));
30
31   // However, clouds are not snow.
32   var ndsi = img.normalizedDifference(['B3', 'B11']);
33   return score.min(rescale(ndsi, 'img', [0.8, 0.6]));
34 };
35
36 //Funcion para adaptar cloudscore a Sentinel
37 var cloudScoresS2 = function(img) {
38   var rescale = img.divide(10000); //esto se hace porque las imagenes Sentinel estan reescaladas
39   var score = cloudScore(rescale).multiply(100).rename('cloudscore');
40   return img.addBands(score);
41 };
42
43 //Funcion para enmascarar nubes menores a x
44 var cloudthreshold = 60;
45 var maskClouds = function(image) {
46   return image.updateMask(image.select('cloudscore').lte(cloudthreshold));
47 };
```

Figura 3. Script utilizado para detección y enmascaramiento de nubes en Sentinel 2

## 2. Composición del mosaico

Considerando que el tamaño del área de estudio era todo Uruguay, se decidió dividirlo en tres zonas de modo de facilitar la distribución del trabajo entre los operadores a cargo de la elaboración del mapa, para lo que se utilizó el siguiente script:

```

1
2 var Division = ee.FeatureCollection("users/miggoncarlos/URUGUAY/DivisionesFinal");
3
4 //Seleccionar area de trabajo(1,2,3)
5
6 var region = Division.filterMetadata('id', 'equals',3);
7
8 Map.addLayer(region, {}, "Parte Seleccionada", false);
9

```

Figura 4. Script de las tres zonas de trabajo en las que se dividió el país para la elaboración del mapa

Para realizar esta división en zonas se utilizaron algunos límites naturales como ríos y otros límites artificiales como las rutas. Específicamente se utilizaron el Río Negro y las rutas 6 y 7 (ver Figura 5 a continuación).



Figura 5. Zonas de trabajo

En el caso de Sentinel 2, se seleccionó una colección de imágenes que fueron filtradas por un rango de fecha y un máximo de cobertura de nubes. El mosaico se compuso automáticamente a partir de los parámetros establecidos utilizando los siguientes comandos:

Para la zona 1 y 2:

```

51 //Colecciones Sentinel 2
52 var sentinel2 = ee.ImageCollection('COPERNICUS/S2');
53
54 //Coleccion Sentinel 2 2017
55 var f2017s = sentinel2.filterBounds(region)
56                       .filterDate('2016-01-01', '2016-12-31')
57                       .filter(ee.Filter.calendarRange(9,3,"month"))
58                       .filterMetadata('CLOUDY_PIXEL_PERCENTAGE', 'less_than', 20);

```

Figura 6. Script utilizado para la composición del mosaico Sentinel 2 para las zonas de trabajo 1 y 2

Para la zona 3, debido a que no se encontraron imágenes libres de nubes para el período considerado, se utilizó un script con diferentes parámetros, como se observa a continuación:

```

50 //Colecciones Sentinel 2
51 var sentinel2 = ee.ImageCollection('COPERNICUS/S2');
52
53 //Colección Sentinel 2 2017
54 var f2017s = sentinel2.filterBounds(region)
55                       .filterDate('2016-08-01', '2017-12-30')
56                       .filter(ee.Filter.calendarRange(8,11,"month"))
57                       .filterMetadata('CLOUDY_PIXEL_PERCENTAGE', 'less_than', 10)
58                       .filterMetadata('CLOUD_COVERAGE_ASSESSMENT', 'less_than', 10);

```

Figura 7. Script utilizado para la composición del mosaico Sentinel 2 para la zona de trabajo 3

### 3. Clasificación supervisada

La clasificación de imágenes hace referencia a la tarea de extraer clases de información de una imagen ráster multibanda. En todas las clasificaciones realizadas se utilizaron las mismas bandas: azul, verde, rojo, infrarrojo cercano, infrarrojo medio y los índices EVI, NDWI y MSAVI. En el Anexo 1 de este documento se describen las bandas de Sentinel 2 y las fórmulas de los índices utilizados para esta clasificación.

La clasificación supervisada utiliza información espectral obtenida de muestras correspondientes a distintos tipos de cobertura para clasificar toda una imagen o un mosaico de imágenes. En este caso, el clasificador utilizó la información de la combinación de bandas e índices establecidos y de la única fecha de cada imagen que compone el mosaico, asignándole una clase a cada pixel del mosaico. Esta etapa se llevó a cabo también en la plataforma de GEE, a través del siguiente script:

```

88 //Clasificación Supervisada
89
90 //Habilitar puntos de acuerdo al uso o no de muestras de bajos
91
92 //Puntos sin muestras de bajos
93 //var puntos = ee.FeatureCollection([Bosque, NoBosque, Forestacion, Agua]).flatten().filterBounds(region) ;//Training data
94
95
96 //Puntos con muestra de bajos
97 var puntos = ee.FeatureCollection([Bosque, NoBosque, Forestacion, Agua, Bajo]).flatten().filterBounds(region) ;//Training data
98
99 //Agregar Columna Random
100 var columnaR = function (feature1) { return ee.FeatureCollection(feature1).randomColumn('random', 1)};
101 var puntos = puntos.map(columnaR);
102
103 //Unir muestras con valores de pixeles
104 var samples = clasificar.sampleRegions({
105   collection: ee.FeatureCollection(puntos).flatten(),
106   properties: ['clase', 'random'],
107   scale: 30
108 });

```

Figura 8. Script utilizado para la clasificación supervisada con Sentinel 2

#### 3.1 Selección de muestras de entrenamiento y verificación

La selección de las muestras de entrenamiento para la clasificación se realizó por medio de interpretación visual, basados en un mosaico en falso color compuesto que se obtiene con la combinación de las bandas Infrarrojo Cercano, Infrarrojo medio y Rojo. Sobre este mosaico se tomaron puntos ubicados sobre los tipos de coberturas que representan las distintas clases que se pretenden diferenciar en el mapa final. Los puntos correspondieron a las coberturas Bosque Nativo, Plantaciones forestales, Agua, Bajos inundables y No Bosque. Del total de puntos definidos un 80% del total se utilizaron para entrenar al algoritmo clasificador y otro 80% se utilizaron para construir la matriz de confusión y calcular la Exactitud global, del usuario y productor.



La cantidad de puntos utilizada en cada zona se definió asumiendo un mínimo aceptable de exactitud global de 0.75. Además, se buscó que la exactitud del usuario y del productor de la clase de Bosque Nativo sea 1 en cada zona.

La cantidad de puntos totales utilizado para cada clase fueron: Bosque Nativo 260, Plantaciones forestales 250, Agua 121, Bajos Inundables 150 y No Bosque 270.

Una vez obtenidas las muestras de entrenamiento se realizó la consiguiente clasificación utilizando el algoritmo de árboles de clasificación y regresión CART (Breiman et al., 1984).

La metodología CART utiliza datos para construir árboles de clasificación o de regresión los cuales son usados para clasificar o predecir nuevos datos. Estos árboles CART pueden manipular fácilmente variables numéricas y/o categóricas. Entre otras ventajas está su robustez a valores atípicos, la invarianza en la estructura de sus árboles de clasificación o de regresión a transformaciones monótonas de las variables independientes, y sobre todo, su interpretabilidad.

El script utilizado para la selección de las muestras de entrenamiento y verificación se presenta a continuación:

```
110 //Dividir muestras para clasificación y verificación
111 var training = samples.filterMetadata('random', 'less_than', 0.8);
112 var testing = samples.filterMetadata('random', 'not_less_than', 0.8);
113
114 //Entrenar clasificador
115 var trainedClassifier = ee.Classifier.cart().train(training, 'class', clasificar.bandNames());
116
117 //Realizar verificación
118 var validation = testing.classify(trainedClassifier);
119 var errorMatrix = validation.errorMatrix('class', 'classification', [1,2,3,4,5]);
120 print('Matriz de errores:', errorMatrix);
121 print('Exactitud total:', errorMatrix.accuracy());
122 print('kappa:', errorMatrix.kappa());
123 print('Exactitud del usuario (filas):', errorMatrix.consumersAccuracy());
124 print('Exactitud del productor (columnas):', errorMatrix.producersAccuracy());
125
126 //Entrenar y clasificar con el total de muestras
127 var trainedClassifier = ee.Classifier.cart().train(samples, 'class', clasificar.bandNames());
128 var classified = clasificar.classify(trainedClassifier).clip(region);
129
130 print('Árbol de Decisiones', trainedClassifier.explain());
```

Figura 9. Script utilizado para la selección de muestras de entrenamiento y verificación en Sentinel 2

### 3.2 Clasificación del mosaico anual de imágenes en base a las muestras de entrenamiento

Como ya fuera mencionado, para la clasificación anual se tomaron en cuenta las clases Bosque Nativo, Plantaciones forestales, Agua, Bajos inundables y No Bosque (pastizales, cultivos, asentamientos humanos, suelo desnudo, rutas) y se utilizó el mosaico anual de Sentinel generado anteriormente. Dichas clases fueron consideradas a los efectos de poder aumentar la eficiencia de los árboles de decisión (CART o Random Forest). Luego de la clasificación, las clases Plantaciones forestales, Agua, Bajos inundables y No Bosque fueron agrupadas en una única clase de No Bosque, obteniéndose como resultado final un mapa con dos clases: Bosque Nativo y No Bosque.

A continuación se presentan ejemplos de muestras correspondientes a las distintas clases consideradas para la clasificación anual.

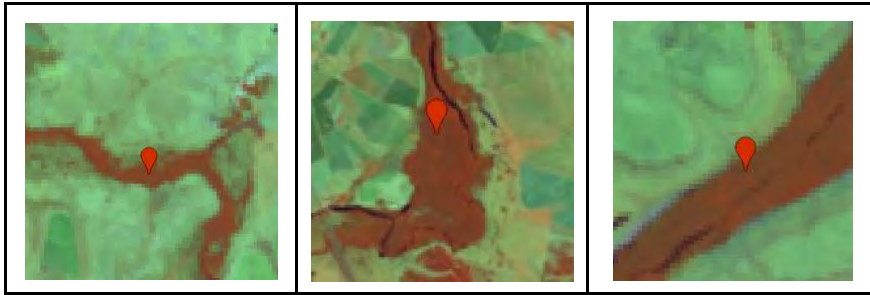


Figura 10. Ejemplo de muestras correspondientes a la clase Bosque Nativo

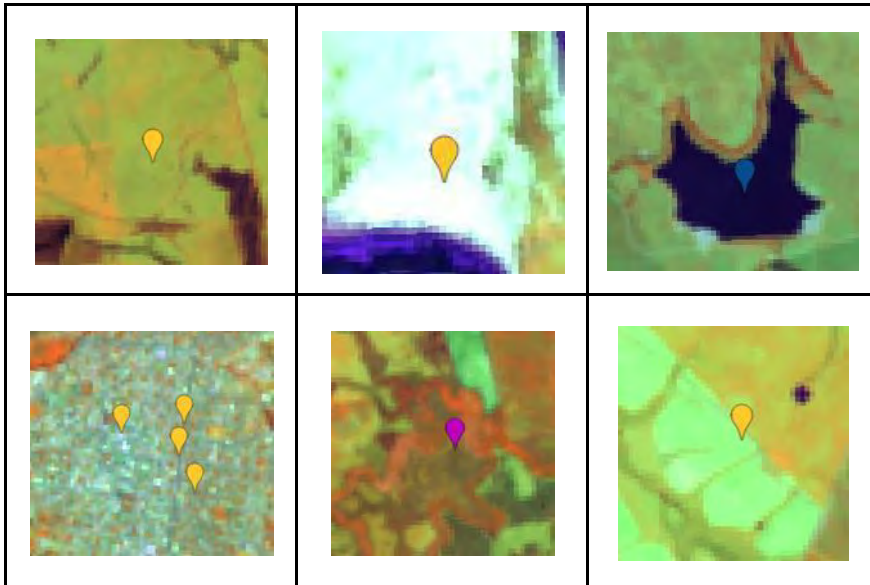


Figura 11. Ejemplo de muestras correspondientes a las clases Agua, Bajos inundables y No Bosque



Figura 12. Ejemplo de muestras correspondientes a la clase Plantaciones forestales

En el Anexo 2 de este documento se incluye el script completo de GEE utilizado para la elaboración de este mapa de cobertura de bosque nativo de Uruguay para el año 2016 con Sentinel 2.

#### 4. Edición visual de errores de clasificación

En esta etapa se realizaron correcciones en zonas en donde se detectaron errores en la clasificación de grandes áreas mal clasificadas. Esta etapa se llevó a cabo en la plataforma GEE, mediante la generación de un nuevo script para realizar las correcciones. Dicho script se presenta a continuación:

```

48 * /* Este corrector permite cambiar solo los pixeles de una clase que se
49    encuentran dentro de un poligono
50
51 Los parametros del corrector son:
52 geom = featureCollection que contiene los poligonos de correccion
53 mal = clase dentro del poligono a ser cambiada
54 bien = nuevo valor de los pixeles dentro del poligono que seran cambiados
55 clasif = archivo raster de clasificacion a ser modificado*/
56
57
58 * var corrector = function(geom, mal, bien, clasif) {
59   var binario = geom.reduceToImage(["clase"], ee.Reducer.first());
60   var condicion = binario.and(clasif.eq(mal));
61   var clasC = clasif.where(condicion, bien);
62   return clasC;
63
64
65 //descomentar la siguiente linea y completar para usar
66 var classified = clasificacionOK;
67
68 Map.addLayer(classified,{ min:1, max:5, palette: ['98ff00','d63000','fff5d2','0b4a8b', 'e9c4ec' ]},'Clasificacion original',false);
69
70 //En el siguiente ejemplo los pixeles que tengan clase 1 dentro del
71 //poligono cla2 pasan a tener clase 2 sin afectar a los pixeles de otra clase
72 var coreccion1 = corrector(c1a2,1,2,classified);
73 var coreccion2 = corrector(c1a3,1,3,coreccion1);
74 var coreccion3 = corrector(c2a1,2,1,coreccion2);
75 var coreccion4 = corrector(c2a3,2,3,coreccion3);
76 var coreccion5 = corrector(c3a1,3,1,coreccion4);
77 var coreccion6 = corrector(c3a2,3,2,coreccion5);
78 var coreccion7 = corrector(c4a1,4,1,coreccion6);
79 var coreccion8 = corrector(c4a2,4,2,coreccion7);
80 var coreccion9 = corrector(c5a1,5,1,coreccion8);
81 var coreccion10 = corrector(c5a2,5,2,coreccion9);

```

Figura 13. Script utilizado para la corrección de errores por polígonos en Sentinel 2

Las correcciones se realizaron por interpretación visual y digitalización manual de polígonos que correspondían a grandes áreas de cobertura que quedaron mal clasificadas en la etapa anterior. A dichos polígonos mal clasificados se les asignó la clase correcta de tipo de cobertura.

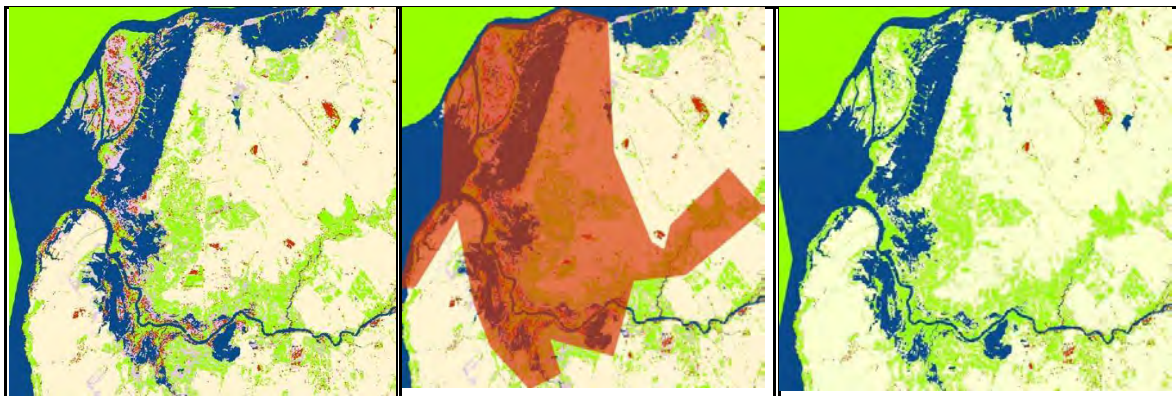


Figura 14. Ejemplo del proceso de edición visual de errores de clasificación

En el Anexo 3 de este documento se incluye el script completo de GEE utilizado para realizar esta edición visual de errores de clasificación.

## 5. Descarga de clasificación corregida y mosaico de imágenes

La descarga de la clasificación corregida por polígonos, así como del mosaico de imágenes Sentinel 2 se realizó en la proyección original WGS 84.

El mosaico de imágenes satelitales se descargó con valores al tope de la atmósfera re-escalados por un valor de 10.000. Solamente se descargaron las bandas correspondientes al verde, rojo, infrarrojo cercano e infrarrojo de onda corta 1, para su posterior utilización en el proceso de segmentación.

```

194 //Descargas
195 /*
196 //var download = classified;
197 var download = clasificar.select(['GREEN', 'RED', 'NIR']).multiply(10000).toUint16();
198 var limiteDescarga = region.geometry().bounds();
199
200 Export.image.toDrive({
201   image: download,
202   description: 'DescargaClassPararte2',
203   scale: 10,
204   region: limiteDescarga,
205   maxPixels: 1e12
206 });
207 */

```

Figura 15. Script para descargar la clasificación en Sentinel 2

Una vez descargados el mosaico y la clasificación, fueron reproyectados a la proyección UTM zona 21s utilizando el programa QGIS.

## 6. Segmentación

La segmentación se realizó por zonas, considerando las 3 mismas zonas en las que se dividió el país para la composición del mosaico.

Este proceso se realizó utilizando la herramienta Large-Scale MeanShift del software Orfeo Toolbox (versión 6.2.0).

Los parámetros utilizados para Sentinel 2 fueron: Rango Espacial: 4, Rango Espectral: 120, Tamaño mínimo de segmentos: 50, y tamaño de teselas: 2.000.

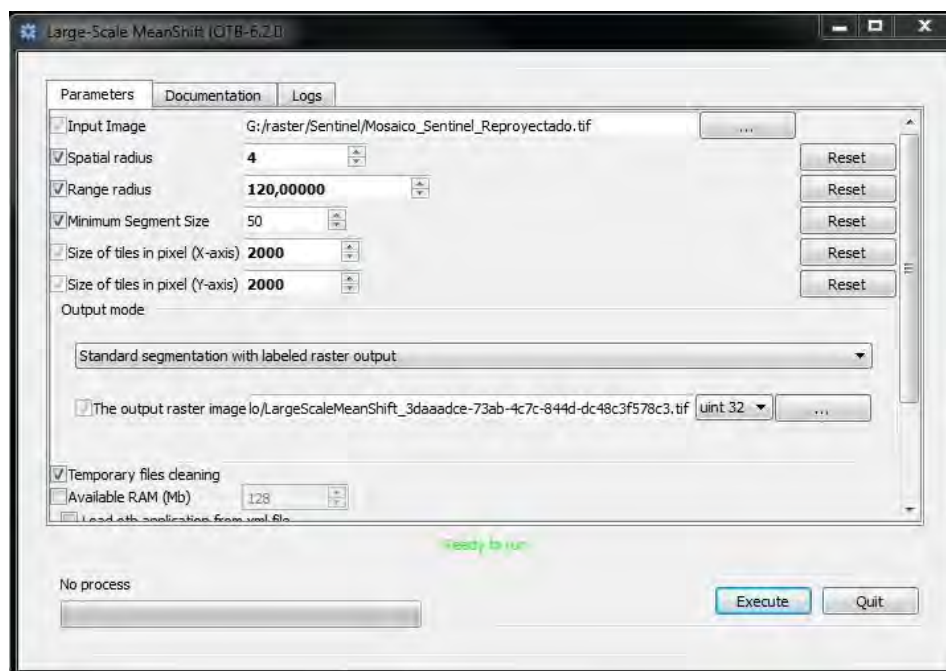


Figura 16. Herramienta Large-Scale MeanShift con los parámetros utilizados para la segmentación de Sentinel

Como resultado se obtuvo un ráster que fue luego convertido a capa vectorial con el software QGIS.

En el ráster de salida del proceso de segmentación, cada pixel correspondiente al mismo objeto tiene el mismo valor, es decir, en el mosaico segmentado se tienen tantos valores como objetos en el mosaico.



Figura 17. Segmentos no categorizados obtenidos del proceso de segmentación

En el Anexo 4 de este documento se incluye información teórica complementaria vinculada con el proceso de segmentación.

## **7. Asignación de clase por mayoría, ediciones finales y post-procesamiento**

Para la asignación de clases a cada uno de los segmentos se utilizó la herramienta de estadísticas de zona del programa QGIS. Para ello, se realizó el cálculo de mayoría por segmento, de manera de determinar la clase mayoritaria de cada uno.

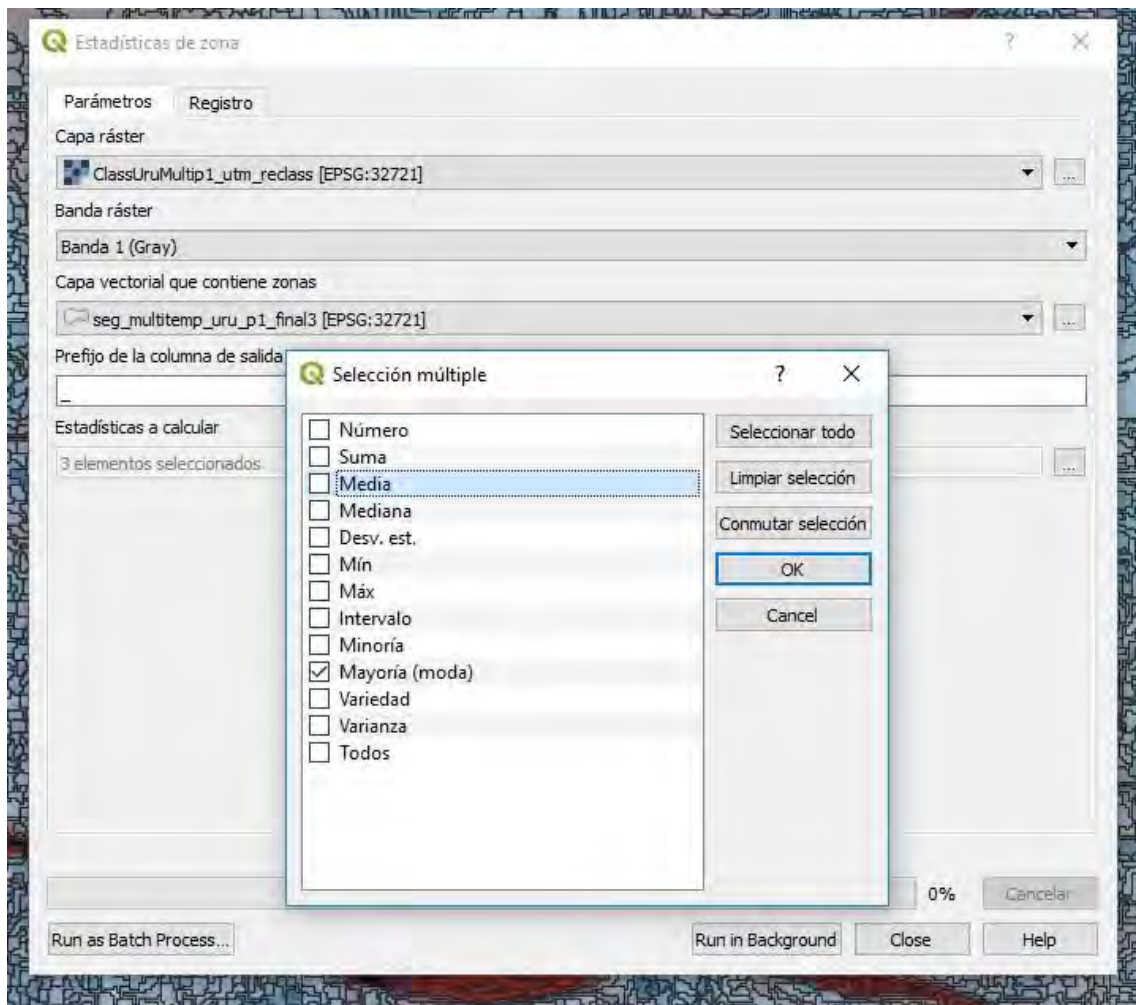


Figura 18. Herramienta de cálculo de mayoría por segmento

Este proceso genera una nueva columna en el archivo vectorial de entrada (en este caso en el archivo que contiene los segmentos) con el resultado de la clase numérica mayoritaria para cada uno de los segmentos.

Una vez obtenidos los segmentos categorizados se vuelve a realizar una revisión, por parte del operador, en la cual se verifica si su clasificación se encuentra acorde a lo observado en las imágenes satelitales. Las modificaciones necesarias se realizan directamente en la tabla de atributos de los segmentos en el programa QGIS.

Para la revisión de los segmentos categorizados se elaboró una cuadrícula que permitió mantener un orden para el trabajo y facilitó la edición por parte del observador (ver Figura 19).

Para verificar la clasificación de cada segmento se utilizó el mosaico de imágenes satelitales Sentinel (ver Figura 20) y también imágenes de alta resolución para el año en cuestión, en caso de estar disponibles (ver Figura 21).

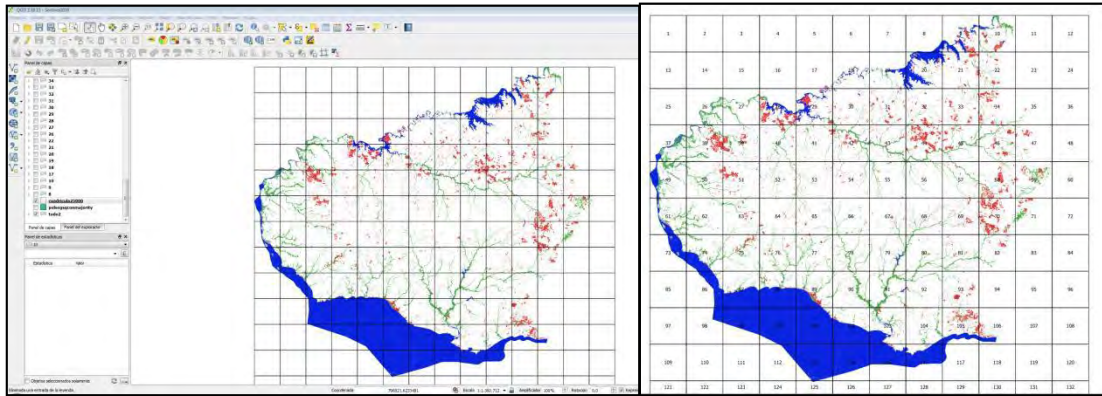


Figura 19. Cuadrícula de la zona a editar

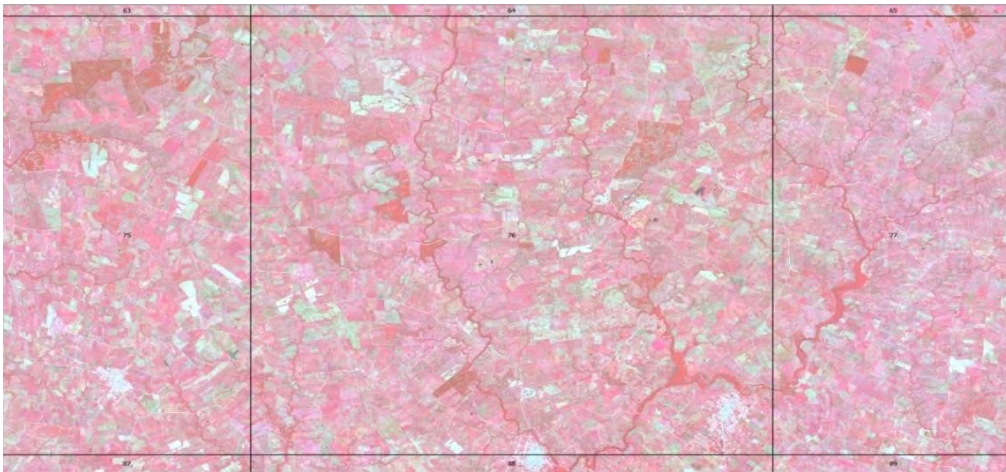


Figura 20. Mosaico de imagen satelital Sentinel en el cuadrante 76 para el año 2016



Figura 21: Imagen de alta resolución para el cuadrante 76

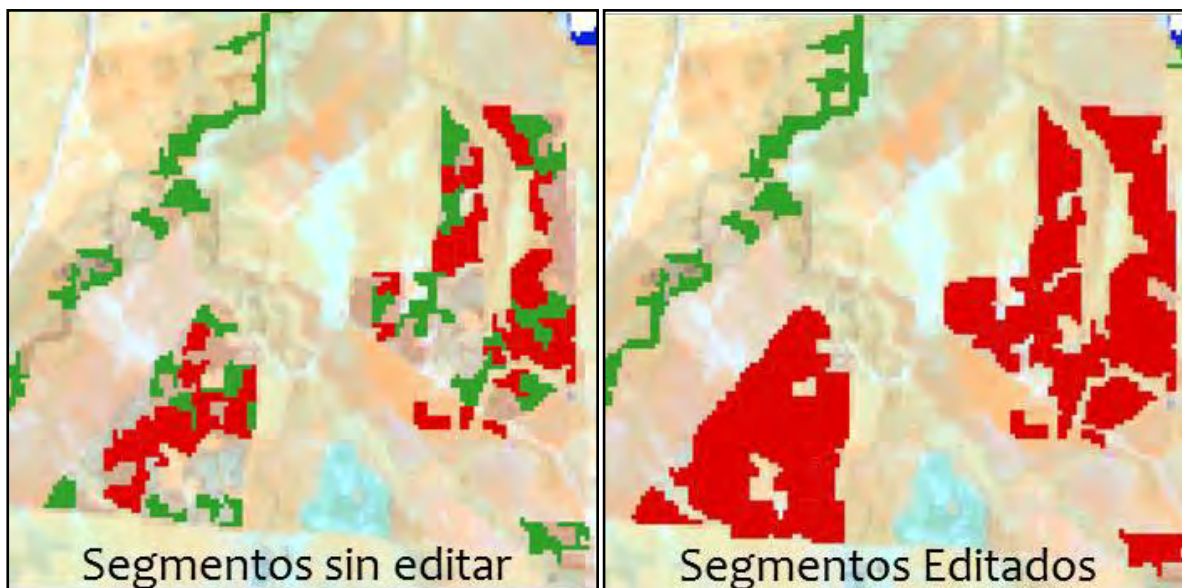


Figura 22. Imagen del proceso pre y post edición de los segmentos mal clasificados

Una vez finalizada la edición de los segmentos, se llevó a cabo una etapa de post-procesamiento del mapa. Para ello, se realizaron comprobaciones topológicas y geométricas en QGIS para la obtención de un mapa de cobertura de bosque nativo unificado. Para ello fue necesario disolver los segmentos de cada una de las 3 zonas por separado y realizar posteriormente la unión de todo el país. Como resultado de este post-proceso, se cuenta con una clasificación de dos clases pertenecientes a Bosque Nativo y No Bosque correspondiente a todo el Uruguay.

## 8. Validación

El último paso de esta metodología consiste en validar o evaluar el mapa elaborado, de manera de estimar la exactitud temática de la clasificación. Para ello, se utilizó como información de referencia o “verdad de campo” una fuente de información de las distintas coberturas de la tierra independiente y de mayor resolución espacial que la utilizada para generar la clasificación (imágenes de Google Earth y Esri). La evaluación se realizó aplicando un muestreo aleatorio, estratificado simple por clase y considerando como unidad de análisis los segmentos del mapa de cobertura.

Para cada segmento de la muestra de validación se comparó la clase asignada en el mapa de cobertura con la clase asignada a partir de la interpretación visual de imágenes de alta resolución (información de referencia o “verdad de campo”) por parte de un equipo técnico conformado específicamente para esta evaluación. Con los resultados obtenidos de esa comparación, se procedió a construir la matriz de confusión y calcular la exactitud global, exactitud del productor y usuario del mapa de cobertura, con sus correspondientes intervalos de confianza.

El detalle de la metodología aplicada y los resultados de la validación de este mapa se incluyen en el documento “Metodología y resultados de la validación del mapa de cobertura de bosque nativo de Uruguay para el año 2016 con Sentinel 2” (Riaño et al., 2019).



## Bibliografía

Baatz, M. and Schape, A. (2000). *Multiresolution Segmentation – An Optimization Approach for High Quality Multi-Scale Image Segmentation*. Proceedings of the Angewandte Geographische Informations Verarbeitung XII, Karlsruhe, 12-23.

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984). *Classification and Regression Trees*. New York: Routledge, <https://doi.org/10.1201/9781315139470>

Chander, G. and Markham, B.L. (2003). *Revised Landsat-5 TM radiometric calibration procedures, and post-calibration dynamic ranges*. IEEE Transactions on Geoscience and Remote Sensing, 41: 2674–2677.

Chander, G., Markham, B.L. and Barsi, J.A. (2007). *Revised Landsat-5 Thematic Mapper Radiometric Calibration*. IEEE Geoscience and Remote Sensing Letters, 4: 490-494.

Chuvieco, E. 1995. *Fundamentos de teledetección ambiental* (2nd ed.). Madrid: RIALP S.A.

European Space Agency. (2019). *Copernicus Observing the Earth*. Paris, France, EU. Recuperado de [https://www.esa.int/Our\\_Activities/Observing\\_the\\_Earth/Copernicus](https://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus).

Fukunaga, K. And Hostetler, L. (1975). *The estimation of the gradient of a density function, with applications in pattern recognition*. IEEE Transactions on Information Theory, 21(1), 32-40.

MGAP (Ministerio de Ganadería, Agricultura y Pesca) – Mvotma (Ministerio de Vivienda, Ordenamiento Territorial y Medio Ambiente). (2019). *Metodología y resultados de la validación del mapa de cobertura de bosque nativo de Uruguay para el año 2016 con Sentinel 2*. Riaño, M.E., Bernardi, L., Boccardo, A., Miguel, C., Olivera, J., Penengo, C. y Rama, G. Proyecto REDD+ Uruguay. Montevideo, Uruguay, págs. 1-12.

## Anexo 1

### DESCRIPCIÓN DEL PROGRAMA COPÉRNICO, LAS BANDAS DE SUS SATÉLITES Y LOS ÍNDICES UTILIZADOS PARA LA ELABORACIÓN DEL MAPA

#### Programa Copérnico

Copérnico es el programa de observación de la Tierra diseñado para proporcionar información precisa, actualizada y de fácil acceso para mejorar la gestión del medio ambiente, comprender y mitigar los efectos del cambio climático y garantizar la seguridad ciudadana. Copérnico es el nuevo nombre del programa para la Vigilancia Mundial del Medio Ambiente y la Seguridad, antes conocido como GMES.

La ESA (European Space Agency) está desarrollando una familia de misiones Sentinel para atender a las necesidades operacionales del programa Copérnico.

Cada misión Sentinel está formada por una constelación de dos satélites, lo que permite cumplir los requisitos de cobertura y frecuencia de observación y ofrecer un conjunto robusto de datos para los servicios del programa Copérnico.

Estas misiones transportan las tecnologías más avanzadas para la observación de la Tierra, como los instrumentos radar o multiespectrales que permitirán monitorizar cualquier cambio en el terreno, en los océanos o en la atmósfera.

#### Sentinel-1

El satélite Sentinel-1 formó parte de la primera misión espacial Copérnico, lanzada por ESA en el 2014. Es un satélite Radar de Apertura Sintética (SAR) que permite obtener imágenes independientemente de las condiciones climáticas. Estas imágenes dan información vital tanto para el sector público como al privado, ayudando a monitorizar el hielo de los mares, el entorno marino, los cambios en la superficie de la tierra y también apoyo en la ayuda humanitaria en situaciones de crisis. Este satélite revisita cada punto de la Tierra en 12 días y tiene una resolución espacial de 5m.

#### Información general del satélite SENTINEL 1

Parámetros	SENTINEL 1
Altitud de Órbita	693 km
Bandas Radar	C-Band
Modo de imagen SAR	Stripmap (GSD; 5m, 8°km swath), Interferometric Wide SwathMode (GSD: 5x20m, 240km swath), Extra Wide SwathMode (400km Swath, single-look), Wave Mode (20x5m).
Polarización	VV+VH or HH+HV
Distancia de Muestreo (GSD)	5-20M
Lanzamiento/Expectativa de vida	2014-2021
Tiempo Revisita	12 días

#### Sentinel-2

El satélite de imágenes Sentinel-2 es un satélite óptico que fue lanzado el 23 de junio de 2015 cuyo objetivo es obtener imágenes que ayudan a observar los cambios terrestres de la Tierra.

Tiene un escáner multiespectral que permite obtener información en las dos longitudes de onda, visibles e infrarrojos, permitiendo monitorizar los cambios en la tierra y en la vegetación así como vigilar a nivel mundial el cambio climático, con una resolución de 10m. Las imágenes están disponibles de manera gratuita y pueden ser utilizadas en un rango amplio de aplicaciones.

### Información general del satélite SENTINEL 2

Parámetros	SENTINEL 2
Altitud de Órbita	786km
Bandas Espectrales	13 Spectral Bands (443-2190nm) with visible and Near-Infrared bands (10m), red edge and short wave infrared bands (20m) and three atmospheric correction bands (60m)
Ancho Pasada	290 km
Distancia de Muestreo (GSD)	10-60 m
Lanzamiento/Expectativa de vida	2022
Tiempo Revisita	10 días (se reducirá a 5 con el lanzamiento del Sentinel-2B)

### Información de las bandas de SENTINEL 2

Bandas Sentinel 2	Longitud de onda central (micrómetros)	Resolución (metros)
Banda 1 – Aerosol Costero	0.443	60
Banda 2 – Azul	0.49	10
Banda 3 - Verde	0.56	10
Banda 4 – Rojo	0.665	10
Banda 5 – Vegetation Red Edge	0.705	20
Banda 6 – Vegetation Red Edge	0.740	20
Banda 7 – Vegetation Red Edge	0.783	20
Banda 8 -Infrarrojo cercano (NIR)	0.842	10
Banda 8A – Vegetation Red Edge	0.865	20
Banda 9 – Vapor de Agua	0.945	60
Banda 10 – SWIR Cirrus	1.375	60
Banda 11 – SWIR	1.610	20
Banda 12 – SWIR	2.190	20

#### Sentinel-3

Transporta distintos instrumentos para medir la elevación, la temperatura o el color de la tierra y de los océanos con gran precisión y fiabilidad. Esta misión contribuye significativamente a los servicios de pronóstico marítimo y de la monitorización del clima y del medio ambiente.

#### Sentinel-4

Es un instrumento embarcado en un satélite MetOp de Segunda Generación que monitorizará la atmósfera de nuestro planeta desde una órbita polar.

#### Sentinel-5

Precursor es un satélite concebido para reducir la discontinuidad de datos entre las misiones de Envisat (y en especial, de su instrumento Sciamachy) y Sentinel-5. Esta misión estará dedicada al estudio de la atmósfera.

## Información de los índices utilizados para la elaboración del mapa

### *Enhanced Vegetation Index (EVI)*

EVI incorpora un valor "L" para ajustar el fondo del dosel, los valores "C" como coeficientes de resistencia atmosférica y valores de la banda azul. Estas mejoras permiten el cálculo del índice como una relación entre los valores Rojo y NIR, permite reducir el ruido de fondo, el ruido atmosférico y la saturación en la mayoría de los casos.

$$EVI = G * ((NIR - Rojo) / (NIR + C1 * Rojo - C2 * Azul + L))$$

$$EVI = 2.5 * ((NIR - Rojo) / (NIR + 6 * Rojo - 7.5 * Azul + 1))$$

### *Normalized Difference Water Index (NDWI)*

Este índice es comúnmente utilizado para detectar cuerpos de agua.

$$NDWI = (Verde - NIR) / (Verde + NIR)$$

### *Modified Soil Adjust Vegetation Index (MSAVI)*

MSAVI se calcula como una relación entre los valores Rojo y NIR con una L inductiva, función aplicada para maximizar la reducción de los efectos del suelo sobre la señal de vegetación.

$$MSAVI = (2 * NIR + 1 - \sqrt{(2 * NIR + 1)^2 - 8 * (NIR - Rojo)}) / 2$$

## Anexo 2

### SCRIPT COMPLETO DE GEE UTILIZADO PARA LA ELABORACIÓN DEL MAPA DE COBERTURA DE BOSQUE NATIVO DE URUGUAY PARA EL AÑO 2016 CON SENTINEL 2

#### Clasificación Sentinel 2 – Zonas 1 y 2

```
var bosque_nativo = ee.FeatureCollection("users/joseserafinig/Uruguay/IFN_NATIVO"),
    plantaciones = ee.FeatureCollection("users/joseserafinig/Uruguay/IFN_PLANTACIONES");
var Uruguay = ee.FeatureCollection("users/miggoncarlos/URUGUAY/ContornoPaís");
var Division = ee.FeatureCollection("users/miggoncarlos/URUGUAY/DivisionesFinal");

//Seleccionar area de trabajo(1,2,3)

var region = Division.filterMetadata('id', 'equals',2);

Map.addLayer(region, {}, "Parte Seleccionada", false);

//Algoritmo cloud score Sentinel
var cloudScore = function(img) {
  // A helper to apply an expression and linearly rescale the output.
  var rescale = function(img, exp, thresholds){
    return img.expression(exp, {img: img})
      .subtract(thresholds[0]).divide(thresholds[1] - thresholds[0]);
  };
  // Compute several indicators of cloudyness and take the minimum of them.
  var score = ee.Image(1,0);
  // Clouds are reasonably bright in the blue band.
  score = score.min(rescale(img, 'img.B2', [0.1, 0.3]));

  // Clouds are reasonably bright in all visible bands.
  score = score.min(rescale(img, 'img.B4 + img.B3 + img.B2', [0.2, 0.8]));

  // Clouds are reasonably bright in all infrared bands.
  score = score.min(
    rescale(img, 'img.B8 + img.B11 + img.B12', [0.3, 0.8]));

  // However, clouds are not snow.
  var ndsi = img.normalizedDifference(['B3', 'B11']);
  return score.min(rescale(ndsi, 'img', [0.8, 0.6]));
};

//Funcion para adaptar cloudscore a Sentinel
var cloudScoreS2 = function(img){
  var rescale = img.divide(10000); //esto se hace porque las imagenes Sentinel estan reescaladas
  var score = cloudScore(rescale).multiply(100).rename('cloudscore');
  return img.addBands(score);
};

//Funcion para enmascarar nubes menores a x
var cloudthreshold = 60;
var maskClouds = function(image) {
  return image.updateMask(image.select('cloudscore').lte(cloudthreshold));
};
```

```

//Colecciones Sentinel 2
var sentinel2 = ee.ImageCollection('COPERNICUS/S2');

//Coleccion Sentinel 2 2017
var f2017s = sentinel2.filterBounds(region)
    .filterDate('2016-01-01', '2016-12-31')
    .filter(ee.Filter.calendarRange(9,3,"month"))
    .filterMetadata('CLOUDY_PIXEL_PERCENTAGE', 'less_than', 20);

var m2017s = f2017s.map(cloudScoreS2).map(maskClouds);
var m2017s2 = m2017s.median();
var c2017s = m2017s2.clip(region);

Map.addLayer(c2017s, {'bands': 'B8,B4,B3', 'min': [0,0,0], 'max': [5000,3000,3000]}, 'Mosaico Sentinel 2
2016 v843', false);
Map.addLayer(c2017s, {'bands': 'B8,B11,B3', 'min': [0,0,0], 'max': [5000,5000,3000]}, 'Mosaico Sentinel 2
2016 v8113', true);

//Funciones para renombrar

var changeBandNameS2 = function(image){
    return image.select(
        ['B2', 'B3', 'B4', 'B8', 'B11', 'B12'],
        ['BLUE', 'GREEN', 'RED', 'NIR', 'SWIR1', 'SWIR2'])
    };

//Funciones para agregar indices

var calcularinidices = function (image) {
    var evi = image.expression('2.5 * ((NIR - RED)/(NIR + 6 * RED - 7.5 * BLUE + 1))',
    {
        'NIR': image.select('NIR'),
        'RED': image.select('RED'),
        'BLUE': image.select('BLUE')
    }).rename('evi');
    var msavi = image.expression('(2 * NIR + 1 - ((2 * NIR + 1)**2 - 8 * (NIR - RED))**0.5)/2',
    {
        'NIR': image.select('NIR'),
        'RED': image.select('RED'),
        'BLUE': image.select('BLUE')
    }).rename('msavi');
    var ndwi = image.normalizedDifference(['RED', 'SWIR1']).rename('ndwi');
    return image.addBands(evi).addBands(ndwi).addBands(msavi);
};

var clasificar = calcularinidices(changeBandNameS2(c2017s));

//Elegir bandas que seran utilizadas para clasificar
var clasificar = clasificar.select(['BLUE', 'GREEN', 'RED', 'NIR', 'SWIR1', 'SWIR2', 'evi', 'ndwi', 'msavi']);

//Habilitar Clasificacion despues de tomar puntos de muestreo

//Clasificacion Supervisada

```

```

//Habilitar puntos de acuerdo al uso o no de muestras de bajos

//Puntos sin muestras de bajos
var puntos = ee.FeatureCollection([Bosque, NoBosque, Forestacion, Agua]).flatten().filterBounds(region)
;//Training data

//Puntos con muestra de bajos
//var puntos = ee.FeatureCollection([Bosque, NoBosque, Forestacion, Agua, Bajo]).flatten().filterBounds(region);//Training data

//Agregar Columna Random
//for (var i = 0; i < puntos.length; i++){puntos[i] = puntos[i].randomColumn('random', 1001)}
var columnaR = function(feature1){ return ee.FeatureCollection(feature1).randomColumn('random', 1)};
var puntos = puntos.map(columnaR);

//Unir muestras con valores de pixeles
var samples = clasificar.sampleRegions({
  collection: ee.FeatureCollection(puntos).flatten(),
  properties: ['clase', 'random'],
  scale: 30
});

//Dividir muestras para clasificacion y verificacion
var training = samples.filterMetadata('random', 'less_than', 0.8);
var testing = samples.filterMetadata('random', 'not_less_than', 0.8);

//Entrenar clasificador
var trainedClassifier = ee.Classifier.cart().train(training, 'clase', clasificar.bandNames());

//Realizar verificacion
var validation = testing.classify(trainedClassifier);
var errorMatrix = validation.errorMatrix('clase', 'classification', [1,2,3,4]);
print('Matriz de errores:', errorMatrix);
print('Exactitud total:', errorMatrix.accuracy());
print('Exactitud del usuario (filas):', errorMatrix.consumersAccuracy());
print('Exactitud del productor (columnas):', errorMatrix.producersAccuracy());

//Entrenar y clasificar con el total de muestras
var trainedClassifier = ee.Classifier.cart().train(samples, 'clase', clasificar.bandNames());
var classified = clasificar.classify(trainedClassifier).clip(region);

print('Árbol de Decisiones', trainedClassifier.explain());

//Visualizacion de capas

Map.addLayer(classified, { min:1, max:4, palette: ['3e7531','ff3126','eaf6b4','1177e0'] }, 'Clasificacion');

//Map.addLayer(classified, { min:1, max:5, palette: ['3e7531','ff3126','eaf6b4','1177e0', 'bf04c2']
}, 'Clasificacion con bajos');

//Datos de apoyo
//Bosque Hansen

```

```

var gfclImage = ee.Image('UMD/hansen/global_forest_change_2015_v1_3');
var forest = gfclImage.select(['treecover2000']).clip(Uruguay).clip(region);
var forest10 = forest.where(forest.lt(10), 1);
var forest2000 = forest10.where(forest10.gte(10), 2);
Map.addLayer(forest2000, {"min":1,"max":2,"palette":["f7ffcd","31a523"]}, "Bosque Hansen 2000", false);

var loss = gfclImage.select(['loss']);
var forest2015 = forest2000.where(loss.eq(1), 0);
Map.addLayer(forest2015, {"min":1,"max":2,"palette":["f7ffcd","31a523"]}, 'Bosque Hansen 2015', false);

//Datos IFN
Map.addLayer(bosque_nativo.filterBounds(region), {color:'98ff00'}, 'Bosque Nativo', false);
Map.addLayer(plantaciones.filterBounds(region), {color:'4a7d00'}, 'Plantaciones', false);

//Calculos de area
//Poner la clasificación en imagenA para calcular el area
var imagenA = classified;

var bosque1 = imagenA.eq(1);

var arealImage = bosque1.multiply(ee.Image.pixelArea().divide(10000));
var stats_area = arealImage.reduceRegion({
  reducer: ee.Reducer.sum(),
  scale:10,
  geometry: region.geometry(),
  maxPixels: 1e12
});
var area = ee.Number(stats_area.get('classification'));

// Print stats for an area property.
print('Area total Bosque p2', area.format('%,.2f'), 'ha');

//Descargas

//var download = classified;
var download = clasificar.select(['GREEN', 'RED', 'NIR']).multiply(10000).toUint16();
var limiteDescarga = region.geometry().bounds();

Export.image.toDrive({
  image: download,
  description: 'DescargaClassPararte1',
  scale: 10,
  region: limiteDescarga,
  maxPixels: 1e12
});

```



## Clasificación Sentinel 2 – Zona 3

```
var Uruguay = ee.FeatureCollection("users/miggoncarlos/URUGUAY/ContornoPais");
var Division = ee.FeatureCollection("users/miggoncarlos/URUGUAY/DivisionesFinal");

//Seleccionar area de trabajo(1,2,3)

var region = Division.filterMetadata('id', 'equals',3);

Map.addLayer(region, {}, "Parte Seleccionada", false);

//Algoritmo cloud score Sentinel
var cloudScore = function(img) {
  // A helper to apply an expression and linearly rescale the output.
  var rescale = function(img, exp, thresholds){
    return img.expression(exp, {img: img})
      .subtract(thresholds[0]).divide(thresholds[1] - thresholds[0]);
  };
  // Compute several indicators of cloudyness and take the minimum of them.
  var score = ee.Image(1.0);
  // Clouds are reasonably bright in the blue band.
  score = score.min(rescale(img, 'img.B2', [0.1, 0.3]));

  // Clouds are reasonably bright in all visible bands.
  score = score.min(rescale(img, 'img.B4 + img.B3 + img.B2', [0.2, 0.8]));

  // Clouds are reasonably bright in all infrared bands.
  score = score.min(
    rescale(img, 'img.B8 + img.B11 + img.B12', [0.3, 0.8]));

  // However, clouds are not snow.
  var ndsi = img.normalizedDifference(['B3', 'B11']);
  return score.min(rescale(ndsi, 'img', [0.8, 0.6]));
};

//Funcion para adaptar cloudscore a Sentinel
var cloudScoreS2 = function(img){
  var rescale = img.divide(10000); //esto se hace porque las imagenes Sentinel estan reescaladas
  var score = cloudScore(rescale).multiply(100).rename('cloudscore');
  return img.addBands(score);
};

//Funcion para enmascarar nubes menores a x
var cloudthreshold = 60;
var maskClouds = function(image) {
  return image.updateMask(image.select('cloudscore').lte(cloudthreshold));
};

//Colecciones Sentinel 2
var sentinel2 = ee.ImageCollection('COPERNICUS/S2');

//Coleccion Sentinel 2 2017
var f2017s = sentinel2.filterBounds(region)
  .filterDate('2016-08-01', '2017-12-30');
```

```

.filter(ee.Filter.calendarRange(8,11,"month"))
.filterMetadata('CLOUDY_PIXEL_PERCENTAGE','less_than',10)
.filterMetadata('CLOUD_COVERAGE_ASSESSMENT','less_than',10);

print(f2017s)

var m2017s = f2017s.map(cloudScoreS2).map(maskClouds);
var m2017s2 = m2017s.median();
var c2017s = m2017s2.clip(region);

Map.addLayer(c2017s,{'bands': 'B8,B4,B3','min':[0,0,0],'max':[5000,3000,3000]}, 'Mosaico Sentinel 2
2017 v843', false);
Map.addLayer(c2017s,{'bands': 'B8,B11,B3','min':[0,0,0],'max':[5000,5000,3000]}, 'Mosaico Sentinel 2
2017 v8113', true);

//Funciones para renombrar

var changeBandNameL5L7 = function(image){
return image.select(
['B1','B2','B3','B4','B5','B7'],
['BLUE','GREEN','RED','NIR','SWIR1','SWIR2']);

var changeBandNameL8 = function(image){
return image.select(
['B2','B3','B4','B5','B6','B7'],
['BLUE','GREEN','RED','NIR','SWIR1','SWIR2'])
};

//Funciones para agregar indices

var calcularinidices = function (image) {
var evi = image.expression('2.5 * ((NIR - RED)/(NIR + 6 * RED - 7.5 * BLUE + 1))',
{
'NIR': image.select('NIR'),
'RED': image.select('RED'),
'BLUE': image.select('BLUE')
}).rename('evi');
var msavi = image.expression('(2 * NIR + 1 - ((2 * NIR + 1)**2 - 8 * (NIR - RED))**0.5)/2',
{
'NIR': image.select('NIR'),
'RED': image.select('RED'),
'BLUE': image.select('BLUE')
}).rename('msavi');
var ndwi = image.normalizedDifference(['RED', 'SWIR1']).rename('ndwi');
return image.addBands(evi).addBands(ndwi).addBands(msavi);
};

var clasificar = calcularinidices(changeBandNameL5L7(c2017s));

//Elegir bandas que seran utilizadas para clasificar
var clasificar = clasificar.select(['BLUE','GREEN','RED','NIR','SWIR1','SWIR2','evi','ndwi','msavi']);

//Habilitar Clasificacion despues de tomar puntos de muestreo

//Clasificacion Supervisada

```

```

//Habilitar puntos de acuerdo al uso o no de muestras de bajos

//Puntos sin muestras de bajos
var puntos = ee.FeatureCollection([Bosque, NoBosque, Forestacion, Agua]).flatten().filterBounds(region)
;//Training data

//Puntos con muestra de bajos
var puntos = ee.FeatureCollection([Bosque, NoBosque, Forestacion, Agua,
Bajo]).flatten().filterBounds(region);//Training data

//Agregar Columna Random
for (var i = 0; i < puntos.length; i++) {puntos[i] = puntos[i].randomColumn('random', 1001)}
var columnaR = function (feature1) { return ee.FeatureCollection(feature1).randomColumn('random', 1)};
var puntos = puntos.map(columnaR);

//Unir muestras con valores de pixeles
var samples = clasificar.sampleRegions({
collection: ee.FeatureCollection(puntos).flatten(),
properties: ['clase', 'random'],
scale: 30
});

//Dividir muestras para clasificacion y verificacion
var training = samples.filterMetadata('random', 'less_than', 0.8);
var testing = samples.filterMetadata('random', 'not_less_than', 0.8);

//Entrenar clasificador
var trainedClassifier = ee.Classifier.cart().train(training, 'clase', clasificar.bandNames());

//Realizar verificacion
var validation = testing.classify(trainedClassifier);
var errorMatrix = validation.errorMatrix('clase', 'classification', [1,2,3,4,5]);
print('Matriz de errores:', errorMatrix);
print('Exactitud total:', errorMatrix.accuracy());
print('Exactitud del usuario (filas):', errorMatrix.consumersAccuracy());
print('Exactitud del productor (columnas):', errorMatrix.producersAccuracy());

//Entrenar y clasificar con el total de muestras
var trainedClassifier = ee.Classifier.cart().train(samples, 'clase', clasificar.bandNames());
var classified = clasificar.classify(trainedClassifier).clip(region);

print('Árbol de Decisiones', trainedClassifier.explain());

//Visualizacion de capas

//Map.addLayer(classified,{ min:1, max:4, palette:['54c924','238b35','ffd79b','1177e0']},'Clasificacion');

Map.addLayer(classified,{ min:1, max:5, palette:['54c924','238b35','ffd79b','1177e0','ffd79b']
},'Clasificacion con bajos');

//Datos de apoyo

```

```

//Bosque Hansen
var gfcImage = ee.Image('UMD/hansen/global_forest_change_2015_v1_3');
var forest = gfcImage.select(['treecover2000']).clip(Uruguay).clip(region);
var forest10 = forest.where(forest.lt(10), 1);
var ForestAt2000 = forest10.where(forest10.gte(10), 2);
Map.addLayer(ForestAt2000, {"min":1,"max":2,"palette":["f7ffcd","31a523"]}, "Bosque Hansen 2000",
false);

/*
//Cartografia Forestal
var cartForestal2000 = cartografiaForestal.clip(Uruguay).clip(region);
Map.addLayer(cartForestal2000, {"min":0,"max":1,"palette":["e6ea16","31a523"]}, "Cartografia Forestal
2000", false);

//Datos IFN
Map.addLayer(bosque_nativo.filterBounds(region), {color:'98ff00'}, 'Bosque Nativo', false);
Map.addLayer(plantaciones.filterBounds(region), {color:'4a7d00'}, 'Plantaciones', false);

*/

//calcular area
var imagenA = classified;

var bosque1 = imagenA.eq(1);

var arealImage = bosque1.multiply(ee.Image.pixelArea().divide(10000));
var stats_area = arealImage.reduceRegion({
  reducer: ee.Reducer.sum(),
  scale:30,
  geometry: region.geometry(),
  maxPixels: 1e12
});
var area = ee.Number(stats_area.get('classification'));

// Print stats for an area property.
print('Area total Bosque p3', area.format('%,.2f'), 'ha');

//Descargas
/*
var download = classified;
//var download = clasificar.select().multiply(10000).toUint16()
var limiteDescarga = region.geometry().bounds();

Export.image.toDrive({
  image: download,
  description: 'Class2016_Sentinel_Zona3_v1',
  scale: 30,
  region: limiteDescarga,
  maxPixels: 1e12
});
*/

//MOSAICO

```

```
//var download = classified;
var download = clasificar.select(['GREEN','RED','NIR']).multiply(10000).toUint16();
var limiteDescarga = region.geometry().bounds();

Export.image.toDrive({
  image: download,
  description: 'Mosaico_Sentinel_2016_Zona3',
  scale: 10,
  region: limiteDescarga,
  maxPixels: 1e12
});
```

### Anexo 3

## SCRIPT COMPLETO DE GEE UTILIZADO PARA LA EDICIÓN VISUAL DE ERRORES DE CLASIFICACIÓN

### Script corrección por polígonos Sentinel 2 para zonas 1 y 2

```
var Uruguay = ee.FeatureCollection("users/miggoncarlos/URUGUAY/ContornoPais");
var Division = ee.FeatureCollection("users/miggoncarlos/URUGUAY/DivisionesFinal");

//Seleccionar area de trabajo(1,2,3)

var region = Division.filterMetadata('id', 'equals',2);

Map.addLayer(region, {}, "Parte Seleccionada", false);

//Algoritmo cloud score Sentinel
var cloudScore = function(img) {
  // A helper to apply an expression and linearly rescale the output.
  var rescale = function(img, exp, thresholds){
    return img.expression(exp, {img: img})
      .subtract(thresholds[0]).divide(thresholds[1] - thresholds[0]);
  };
  // Compute several indicators of cloudyness and take the minimum of them.
  var score = ee.Image(1.0);
  // Clouds are reasonably bright in the blue band.
  score = score.min(rescale(img, 'img.B2', [0.1, 0.3]));

  // Clouds are reasonably bright in all visible bands.
  score = score.min(rescale(img, 'img.B4 + img.B3 + img.B2', [0.2, 0.8]));

  // Clouds are reasonably bright in all infrared bands.
  score = score.min(
    rescale(img, 'img.B8 + img.B11 + img.B12', [0.3, 0.8]));

  // However, clouds are not snow.
  var ndsi = img.normalizedDifference(['B3', 'B11']);
  return score.min(rescale(ndsi, 'img', [0.8, 0.6]));
};

//Funcion para adaptar cloudscore a Sentinel
var cloudScoreS2 = function(img){
  var rescale = img.divide(10000); //esto se hace porque las imagenes Sentinel estan reescaladas
  var score = cloudScore(rescale).multiply(100).rename('cloudscore');
  return img.addBands(score);
};

//Funcion para enmascarar nubes menores a x
var cloudthreshold = 60;
var maskClouds = function(image) {
  return image.updateMask(image.select('cloudscore').lte(cloudthreshold));
};

//Colecciones Sentinel 2
var sentinel2 = ee.ImageCollection('COPERNICUS/S2');
```

```

//Coleccion Sentinel 2 2017
var f2017s = sentinel2.filterBounds(region)
    .filterDate('2016-01-01', '2016-12-31')
    .filter(ee.Filter.calendarRange(9,3,"month"))
    .filterMetadata('CLOUDY_PIXEL_PERCENTAGE', 'less_than', 20);

var m2017s = f2017s.map(cloudScoreS2).map(maskClouds);
var m2017s2 = m2017s.median();
var c2017s = m2017s2.clip(region);

Map.addLayer(c2017s, {'bands': 'B8,B4,B3', 'min': [0,0,0], 'max': [5000,3000,3000]}, 'Mosaico Sentinel 2
2016 v843', false);
Map.addLayer(c2017s, {'bands': 'B8,B11,B3', 'min': [0,0,0], 'max': [5000,5000,3000]}, 'Mosaico Sentinel 2
2016 v8113', true);

/* Este corrector permite cambiar solo los pixeles de una clase que se
encuentran dentro de un poligono

Los parametros del corrector son:
geom = featureCollection que contiene los poligonos de correccion
mal = clase dentro del poligono a ser cambiada
bien = nuevo valor de los pixeles dentro del poligono que seran cambiados
clasif = archivo raster de clasificacion a ser modificado*/

var corrector = function(geom, mal, bien, clasif) {
  var binario = geom.reduceToImage(["dase"], ee.Reducer.first());
  var condicion = binario.and(clasif.eq(mal));
  var clasC = clasif.where(condicion, bien);
  return clasC};

//descomentar la siguiente linea y completar para usar
var classified = clasificacion2;

Map.addLayer(classified, { min:1, max:5, palette: ['98ff00','d63000','fff5d2','0b4a8b', 'e9c4ec']
}, 'Clasificacion original');

//En el siguiente ejemplo los pixeles que tengan clase 1 dentro del
//poligono c1a2 pasan a tener clase 2 sin afectar a los pixeles de otra clase
var coreccion1 = corrector(c1a2,1,2,classified);
var coreccion2 = corrector(c1a3,1,3,coreccion1);
var coreccion3 = corrector(c2a1,2,1,coreccion2);
var coreccion4 = corrector(c2a3,2,3,coreccion3);
var coreccion5 = corrector(c3a1,3,1,coreccion4);
var coreccion6 = corrector(c3a2,3,2,coreccion5);
var coreccion7 = corrector(c4a1,4,1,coreccion6);
var coreccion8 = corrector(c4a2,4,2,coreccion7);
var coreccion9 = corrector(c5a1,5,1,coreccion8);
var coreccion10 = corrector(c5a2,5,2,coreccion9);

Map.addLayer(coreccion10, { min:1, max:5, palette: ['98ff00','d63000','faffd2','0b4a8b', 'faffd2']
}, 'Clasificacion Corregida', false);

//Calculos de area
//Poner la clasificación en imagenA para calcular el area
var imagenA = coreccion10;

```

```

var bosque1 = imagenA.eq(1);

var arealImage = bosque1.multiply(ee.Image.pixelArea().divide(10000));
var stats_area = arealImage.reduceRegion({
  reducer: ee.Reducer.sum(),
  scale:10,
  geometry: region.geometry(),
  maxPixels: 1e12
});
var area = ee.Number(stats_area.get('classification'));

// Print stats for an area property.
print('Area total Bosque p2', area.format('%,.2f'), 'ha');

//Descargas
/*
var download = coreccion10;
//var download = clasificar.select().multiply(10000).toUint16()
var limiteDescarga = region.geometry().bounds();

Export.image.toDrive({
  image: download,
  description: 'DescargaClassParrote1',
  scale: 10,
  region: limiteDescarga,
  maxPixels: 1e12
});
*/

```

### Script corrección por polígonos Sentinel 2 para zona 3

```

var Uruguay = ee.FeatureCollection("users/miggoncarlos/URUGUAY/ContornoPais");
var Division = ee.FeatureCollection("users/miggoncarlos/URUGUAY/DivisionesFinal");

//Seleccionar area de trabajo(1,2,3)

var region = Division.filterMetadata('id', 'equals',3);

Map.addLayer(region, {}, "Parte Seleccionada", false);

//Composicion del mosaico

//Algoritmo cloud score Sentinel
var cloudScore = function(img) {
  // A helper to apply an expression and linearly rescale the output.
  var rescale = function(img, exp, thresholds){
    return img.expression(exp, {img: img})
      .subtract(thresholds[0]).divide(thresholds[1] - thresholds[0]);
  };
  // Compute several indicators of cloudyness and take the minimum of them.
  var score = ee.Image(1.0);
  // Clouds are reasonably bright in the blue band.
  score = score.min(rescale(img, 'img.B2', [0.1, 0.3]));

  // Clouds are reasonably bright in all visible bands.
  score = score.min(rescale(img, 'img.B4 + img.B3 + img.B2', [0.2, 0.8]));
}

```



```

// Clouds are reasonably bright in all infrared bands.
score = score.min(
  rescale(img, 'img.B8 + img.B11 + img.B12', [0.3, 0.8]));

// However, clouds are not snow.
var ndsi = img.normalizedDifference(['B3', 'B11']);
return score.min(rescale(ndsi, 'img', [0.8, 0.6]));
};

//Funcion para adaptar cloudscore a Sentinel
var cloudScoreS2 = function(img){
  var rescale = img.divide(10000); //esto se hace porque las imagenes Sentinel estan reescaladas
  var score = cloudScore(rescale).multiply(100).rename('cloudscore');
  return img.addBands(score);
};

//Funcion para enmascarar nubes menores a x
var cloudthreshold = 60;
var maskClouds = function(image) {
  return image.updateMask(image.select('cloudscore').lte(cloudthreshold));
};

//Colecciones Sentinel 2
var sentinel2 = ee.ImageCollection('COPERNICUS/S2');

//Coleccion Sentinel 2 2017
var f2017s = sentinel2.filterBounds(region)
  .filterDate('2016-08-01', '2017-12-30')
  .filter(ee.Filter.calendarRange(8,11,"month"))
  .filterMetadata('CLOUDY_PIXEL_PERCENTAGE', 'less_than', 10)
  //.filterMetadata('CLOUD_COVERAGE_ASSESSMENT', 'less_than', 10);

print(f2017s)
var m2017s = f2017s.map(cloudScoreS2).map(maskClouds);
var m2017s2 = m2017s.median();
var c2017s = m2017s2.clip(region);
Map.addLayer(c2017s, {'bands': 'B8,B4,B3', 'min': [0,0,0], 'max': [5000,3000,3000]}, 'Mosaico Sentinel 2 2017 v843', false);
Map.addLayer(c2017s, {'bands': 'B8,B11,B3', 'min': [0,0,0], 'max': [5000,5000,3000]}, 'Mosaico Sentinel 2 2017 v8113', true);

/* Este corrector permite cambiar solo los pixeles de una clase que se encuentran dentro de un poligono

Los parametros del corrector son:
geom = featureCollection que contiene los poligonos de correccion
mal = clase dentro del poligono a ser cambiada
bien = nuevo valor de los pixeles dentro del poligono que seran cambiados
clasif = archivo raster de clasificacion a ser modificado*/

var corrector = function(geom, mal, bien, clasif) {
  var binario = geom.reduceToImage(["clase"], ee.Reducer.first());
  var condicion = binario.and(clasif.eq(mal));
  var clasC = clasif.where(condicion, bien);
  return clasC};

```

```

//descomentar la siguiente linea y completar para usar (bajo ob4a8b)
var classified = clasificacion;

Map.addLayer(classified,{ min:1, max:5, palette: ['98ff00','d63000','fff5d2','#1a94dc', 'fff5d2']
},'Clasificacion original');

//En el siguiente ejemplo los pixeles que tengan clase 1 dentro del
//poligono c1a2 pasan a tener clase 2 sin afectar a los pixeles de otra clase
var coreccion1 = corrector(c1a2,1,2,classified);
var coreccion2 = corrector(c1a3,1,3,coreccion1);
var coreccion3 = corrector(c2a1,2,1,coreccion2);
var coreccion4 = corrector(c2a3,2,3,coreccion3);
var coreccion5 = corrector(c3a1,3,1,coreccion4);
var coreccion6 = corrector(c3a2,3,2,coreccion5);
var coreccion7 = corrector(c4a1,4,1,coreccion6);
var coreccion8 = corrector(c4a2,4,2,coreccion7);
var coreccion9 = corrector(c5a1,5,1,coreccion8);
var coreccion10 = corrector(c5a2,5,2,coreccion9);

Map.addLayer(coreccion10,{ min:1, max:5, palette: ['98ff00','d63000','fff5d2','ob4a8b', 'fff5d2']
},'Clasificacion Corregida',false);

var imagenA = coreccion10;
var bosque1 = imagenA.eq(1);

var arealImage = bosque1.multiply(ee.Image.pixelArea().divide(10000));
var stats_area = arealImage.reduceRegion({
  reducer: ee.Reducer.sum(),
  scale:30,
  geometry: region.geometry(),
  maxPixels: 1e12
});
var area = ee.Number(stats_area.get('b1'));

// Print stats for an area property.
print('Area total Bosque p3', area.format('%,.2f'), 'ha');
//Descargas
/*
var download = coreccion10;
//var download = clasificar.select().multiply(10000).toUint16()
var limiteDescarga = region.geometry().bounds();

Export.image.toDrive({
  image: download,
  description: 'Class2016_Sentinel_Zona3_v2',
  scale: 30,
  region: limiteDescarga,
  maxPixels: 1e12
});
*/

```

## Anexo 4

### PROCESO DE SEGMENTACIÓN

La segmentación de imágenes es el proceso que consiste en dividir una imagen agrupando los píxeles similares en aglomeraciones denominadas objetos, también llamados regularmente segmentos o polígonos, basados en la similitud del espectro y la particularidad espacial (Baatz y Schappe, 2000).

La operación de segmentación trata de distinguir si un píxel pertenece o no a un objeto de interés. Los algoritmos de segmentación se basan en una de estas dos propiedades básicas de los valores del nivel de gris: discontinuidad o similitud entre los niveles de gris de píxeles vecinos.

Mean-Shift es un algoritmo de agrupamiento no paramétrico introducido por Fukunaga y Hosteltler (1975) que clasifica un conjunto de elementos en base a propiedades de estos de forma iterativa. Este procedimiento se usa para buscar las modas de una función de densidad de probabilidad, representada por un grupo finito de muestras. El algoritmo Mean-Shift sustituye los píxeles de la imagen original por el promedio de los píxeles dentro de su vecindad, incluido el mismo, pero promediando sólo aquellos píxeles cuyo valor de intensidad esté dentro de un intervalo de ponderación del valor de intensidad del píxel central.