

**Desarrollo de una herramienta para la detección
de nubes precipitables en imágenes
meteorológicas GOES-16 usando técnicas de
Machine Learning**

Natalia Lemos García

Santiago Rodas Arias

Trabajo de Grado

Ingeniería Topográfica



Facultad de Ingeniería

Escuela de Ingeniería Civil y Geomática

Santiago de Cali

2020

**Desarrollo de una herramienta para la detección de nubes precipitables
en imágenes meteorológicas GOES-16 usando técnicas de Machine
Learning**

Natalia Lemos García

Santiago Rodas Arias

Dirigido por

Prof. Olga Lucia Baquero Montoya, M.Sc.

Prof. Wilfredo Alfonso Morales, Eng.D.

Trabajo de Grado

Ingeniería Topográfica

Universidad del Valle

Facultad de Ingeniería

Escuela de Ingeniería Civil y Geomática

Santiago de Cali

2020

Índice general

Acrónimos	XII
1. Introducción	1
1.1. Objetivos	3
1.2. Solución propuesta	4
1.3. Organización del documento	4
2. Antecedentes y marco teórico	5
2.1. Antecedentes	6
2.2. Marco teórico	8
2.2.1. Nubes	8
2.2.1.1. Clasificación	8
2.2.2. Plataforma satelital	9
2.3. Métodos de clasificación	12

2.3.1. Support Vector Machine - SVM	12
2.3.1.1. Hiperplanos	12
2.3.1.2. Función de costo y actualizaciones de degradado	13
2.3.2. Random Forest - RF	14
2.3.2.1. Criterio de convergencia	14
2.3.2.2. Fuerza y correlación	16
2.3.3. Redes Neuronales Artificiales - RNA	17
2.4. Optimización de hiperparámetros	19
2.5. Gray-Level Co-Occurrence Matrix - GLCM	21
2.6. Rational Unified Process - RUP	22
2.6.1. Fases del RUP	23
2.7. Síntesis	23
3. Determinación de nubes precipitables en Colombia	25
3.1. Zona de estudio	27
3.2. Metodología para la identificación de nubes precipitables y no precipitables	29
3.2.1. Entrenamiento de técnicas de detección	29
3.2.2. Fuentes de información	29
3.2.3. Obtención de imágenes	30

3.2.4. Cálculo de texturas	32
3.2.5. Entrenamiento de los clasificadores	36
3.2.5.1. Random Forest (RF)	37
3.2.5.2. Máquinas de soporte Vectorial (SVM)	38
3.2.5.3. Redes Neuronales Artificiales (RNA)	39
3.3. Desarrollo de la interfaz	40
3.3.1. Requerimientos funcionales y no funcionales	40
3.3.2. Diagrama conceptual	41
3.3.3. Diagrama de clase	42
3.3.4. Interfaz gráfica	43
3.4. Síntesis	45
4. Resultados y Discusión	48
4.1. Entrenamiento de los modelos de aprendizaje	48
4.1.1. Entrenamiento Random Forest	48
4.1.2. Entrenamiento SVM	51
4.1.3. Entrenamiento MLP	52
4.2. Comparación de los clasificadores	54
4.3. Desarrollo de la interfaz	57

4.3.1. Ventana principal	57
4.3.2. Pruebas de integración	59
4.3.2.1. Búsqueda y carga de imagen	60
4.3.2.2. Búsqueda y carga de shape	62
4.3.2.3. Carga y selección de clasificadores	62
4.3.2.4. Selección de capas	65
4.3.2.5. Visualización	65
4.3.2.6. Guardar clasificación	65
4.4. Síntesis	69
5. Conclusiones y trabajos futuros	70
5.1. Conclusiones	70
5.2. Trabajos futuros	72
Referencias	75

Índice de figuras

2-1. Forma de escanear la superficie del sensor	11
2-2. Proceso de entrenamiento y clasificación de RF	17
2-3. Fases de RUP	23
3-1. Metodología propuesta	26
3-2. Zona de estudio.	28
3-3. Muestras de clase nubes no-precipitables	35
3-4. Ausencia de nubes desde la imagen satelital	35
3-5. Validación Cruzada	37
3-6. Diagrama conceptual.	42
3-7. Diagrama de clase	43
3-8. Ventana principal de la interfaz.	44
3-9. Procedimiento para realizar el proceso de clasificar la imagen.	46

ÍNDICE DE FIGURAS

4-1. N°. Árboles vs Precisión.	49
4-2. Importancia de las características.	50
4-3. Parámetro de regularización vs precisión para diferentes kernels.	52
4-4. Precisión de validación determinada a partir del parametro de regularización (C) y gamma.	53
4-5. Boxplot optimizadores, función de activación, épocas vs Precisión	54
4-6. Gráfico de barras número de neuronas capa oculta, dropout, función de pérdida vs precisión de validación.	55
4-7. Comparación máscaras de detección de nubes precipitables y no precipitables.	56
4-8. Modulos de la interfaz.	58
5-1. Distribución de anexos	74

Índice de tablas

2.1. Géneros y especies de nubes	9
2.2. Nivel de nubes	9
2.3. Bandas espectrales de GOES-16	10
2.4. Modos de operación ABI	11
2.5. Niveles de datos sensor ABI	12
3.1. Bandas seleccionadas para el procesamiento	33
3.2. Medidas de texturas, abreviaturas y ecuaciones	33
3.3. Requerimientos funcionales	41
3.4. Requerimientos funcionales	41
4.1. Prueba de integración: Búsqueda y carga de la imagen	61
4.2. Prueba de integración: Búsqueda y carga del shape	63
4.3. Prueba de integración: Carga y seleccion de los clasificadores	64

ÍNDICE DE TABLAS

4.4. Prueba de integración: Selección de capas a visualizar	66
4.5. Prueba de integración: Visualización de la clasificación	67
4.6. Prueba de integración: Guardar la clasificación	68

Acrónimos

ABI Advanced Baseline Imager.

CHIRPS Climate Hazards Group InfraRed Precipitation with Station data.

CI Convective Initiation.

CNN Convolutional Neural Network.

DT Decision Tree.

EXIS Extreme Ultraviolet X - Ray Irradiance Sensors.

FY-2 Satellite Fengyun.

GFS Sistema Global de Predicción.

GLCM Gray-Level Co-Occurrence Matrix.

GLM Geostationary Lightning Mapper.

GOES-16 Geostationary Operational Environmental Satellite - 16.

IDEAM Instituto de Hidrología, Meteorología y Estudios Ambientales.

IR Infrarrojo.

ML Machine Learning.

MLC Maximum Likelihood Classification.

MLP Multilayer Perceptron.

NDVI Índice de Vegetación de Diferencia Normalizada.

NIR Infrarrojo Cercano.

OMM Organización Meteorológica Mundial.

RF Random Forest.

RMSE Raíz del Error Cuadrático Medio.

RNA Redes Neuronales Artificiales.

RUP Proceso Unificado Racional.

SEISS Space Environmental In - Situ Suite.

SUVI Solar Ultraviolet Imager.

SVM Support Vector Machine.

TRMM Misión de Medición de Lluvias Tropicales.

UNGRD Unidad Nacional para la Gestión del Riesgo y Desastre.

VC Coeficiente de Variación.

VIS Visible.

WV Vapor de Agua.

Resumen

El *Machine Learning* es un conjunto de técnicas derivadas de la inteligencia artificial, cuyo objetivo es crear algoritmos automatizados a partir de patrones relacionados con un fenómeno. Este trabajo realizó un estudio comparativo de 3 métodos de aprendizaje para desarrollar una herramienta de clasificación de nubes precipitables en Colombia. Entre los métodos seleccionados se implementaron Random Forest, Máquinas de Soporte Vectorial y Redes Neuronales Artificiales. Para este trabajo se contó con datos de precipitación de 52 estaciones meteorológicas del IDEAM e imágenes del satélite *GOES* – 16 en el mismo intervalo de tiempo, donde se reportaron presencia/ausencia de nubes (precipitables/no-precipitables). Con la información satelital se tomaron 3 patrones de textura (varianza, media y homogeneidad) en 7 de las 16 bandas: azul, rojo, infrarrojo cercano, vapor de agua nivel bajo, medio y alto, e infrarrojo térmico.

Se realizó un análisis de sensibilidad de hiper-parámetros con el propósito de seleccionar los mejores clasificadores entre los tres métodos. Los resultados obtenidos con el 40% del conjunto de datos, presentan desempeños de 99.87%, 99.97% y 100% para *Random Forest*, Máquinas de Soporte Vectorial y Redes Neuronales Artificiales respectivamente. Finalmente, este trabajo presenta una interfaz de usuario, para hacer la selección del clasificador y cargar la imagen multiespectral del *GOES* – 16. La interfaz permite exportar la máscara obtenida para efectos de análisis en formato *tif*.

Palabras claves: Detección de nubes; nubes precipitables; Machine Learning, RF, SVM, RNA.

Capítulo 1

Introducción

Las nubes son hidrometeoros en suspensión de partículas de agua, hielo, partículas no acuosas o sólidas, que tienen un papel fundamental en el ciclo del agua y en la estructura de la distribución global de los recursos hídricos (OMM, 2017), son uno de los principales componentes de la precipitación y del sistema climático. Según Glickman y Zenk (2000) la precipitación se define como la cantidad de partículas acuosas en fase líquida o sólida, que se originan en la atmósfera y caen a la superficie en un área durante un periodo de tiempo específico; y dentro del sistema climático cumple funciones de regulador de temperatura a través de la liberación de calor latente y de una fuente de agua potable (Michaelides y cols., 2009).

El primer paso en la medición de la precipitación desde el espacio es la distinción entre las masas nubosas precipitables y sus opuestas; por lo tanto, antes de estimar la cantidad de lluvia que puede caer de una nube en particular, se debe asegurar que realmente posea esta cualidad (Tsonis, 1988).

En Colombia, el Instituto de Hidrología, Meteorología y Estudios Ambientales (IDEAM) utiliza modelos para pronosticar el clima con información de radares en tierra y estaciones me-

teorológicas. En zonas llanas como la Amazonía y la Orinoquía las predicciones de estos modelos tienen una mayor precisión, mientras que en la región Andina suelen ser inexactas con diferencias hasta del 50%, comparadas con las estaciones; estos errores son debidos a las fuentes de información como el **Sistema Global de Predicción (GFS)**, el modelo micro, la **Misión de Medición de Lluvias Tropicales (TRMM)** y la mesoescala V5, que siendo productos a nivel global cuentan con una baja resolución espacial de 5 a 25km¹. Las nubes se encargan del transporte del agua de la atmósfera a la superficie, por lo que la medición y predicción de la precipitación son actividades de investigación con un alto impacto económico y social. La detección y la clasificación de las nubes es esencial para realizar modelos numéricos de predicción del clima, cálculo de vientos y para estimar de la precipitación. También permite proporcionar información sobre tendencias meteorológicas, cambios ambientales, estudios climáticos y pronósticos de tormentas. El Grupo de Riesgo Climático de la Universidad de California, sede Santa Barbara, se encarga de la generación y mantenimiento del producto de precipitación **Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS)**, que consiste en la fusión de datos de precipitación de estaciones meteorológicas alrededor del mundo con datos de precipitación satelital. Estos datos son obtenidos a partir del algoritmo *Cold Cloud Duration* (duración de la nube fría) que consiste en la observación de las nubes dentro de un umbral crítico donde éstas precipitan, pero para ello se debe tener en cuenta que no todas son precipitables (Funk y cols., 2015). Lo anterior es importante para grupos de investigación como la **Unidad Nacional para la Gestión del Riesgo y Desastre (UNGRD)**, el **IDEAM**, y el Ministerio de Medio Ambiente y Desarrollo Sostenible para llevar a cabo estudios de sequías, inundaciones, fenómenos de remoción en masa y prevención de desastres.

En los análisis para el estudio y monitoreo de fenómenos ambientales, específicamente en los tipos y cantidad de nubes y precipitaciones, se hace uso de técnicas de nefoanálisis que consisten en una interpretación visual de las mismas, (Pyszczyk y Snaider, 2017;

¹Información disponible en: <https://www.elespectador.com/noticias/medio-ambiente/por-que-s-e-equivocan-los-satelites-con-los-pronosticos-de-lluvias-en-colombia-articulo-750407>

Rodríguez y cols., 2014). Estas técnicas requieren de la experiencia de una persona para diferenciar las tonalidades de grises y reconocer los diferentes géneros de nubes. Es por ello que para realizar la clasificación, se han desarrollado una variedad de métodos para la caracterización de nubes a lo largo de los años como: la distinción de umbrales, la transferencia radiativa y los métodos estadísticos, siendo estos últimos los más relevantes.

Dado lo anterior, el IDEAM asume que existe una relación directa entre la concentración de nubes y la probabilidad de lluvia, por ello, los pronósticos terminan siendo imprecisos e insuficientes. Bajo este escenario se quiere responder a la pregunta:

Dada las condiciones geográficas y meteorológicas de Colombia ¿es posible detectar y clasificar nubes precipitables mediante sensores remotos?

1.1. Objetivos

Objetivo General

Desarrollar una herramienta para identificar y clasificar nubes precipitables en imágenes satelitales meteorológicas que abarquen la región de Colombia.

Objetivos Específicos

1. Caracterizar los tipos de nubes a través de sus componentes físicos y su distribución espacial.
2. Analizar los métodos de clasificación en imágenes satelitales meteorológicas.
3. Evaluar los métodos de clasificación empleados para la detección de nubes precipitables.

1.2. Solución propuesta

Con los avances en el procesamiento de imágenes a través de **Machine Learning (ML)**, este trabajo plantea una solución capaz de cargar la información multi-espectral de imágenes del **Geostationary Operational Environmental Satellite - 16 (GOES-16)** a través de una Interfaz Gráfica donde, a partir de la información espectral, se determine la probabilidad de que el tipo de nube observable sea precipitable. La interfaz cuenta con módulos para la carga de imágenes de la zona de estudio, los clasificadores diseñados con técnicas de **ML**, la visualización y por último, un módulo para almacenar el resultado obtenido.

Este trabajo toma como fuente de verificación de los resultados los datos reportados por diversas estaciones meteorológicas distribuidas en el espacio geográfico de Colombia. A partir de los registros en el tiempo entre el **GOES-16** y las estaciones meteorológicas, se construyen los patrones con características que definen nubes precipitables, nubes no-precipitables y cielo despejado. Este insumo junto con los algoritmos de **ML** como **Random Forest (RF)**, **Support Vector Machine (SVM)** y las **Redes Neuronales Artificiales (RNA)** tipo *Feedforward*, lograron obtener resultados de **99.87%**, **99.97%** y **100%** respectivamente para los datos utilizados en pruebas (es decir, aquellos no incluidos en el entrenamiento).

1.3. Organización del documento

A continuación se presenta el orden en el que esta estructurado el documento: En el capítulo **2** se presenta la revisión bibliográfica realizada sobre los métodos de clasificación empleados en imágenes satelitales, seguido del marco teórico que permite desarrollar este trabajo. En el capítulo **3** se presenta la metodología empleada para el desarrollo de este proyecto, En el capítulo **4** se muestran los resultados obtenidos aplicando las técnicas mencionadas y en el **5** se encuentran las conclusiones.

Capítulo 2

Antecedentes y marco teórico

En este capítulo se presenta una revisión bibliográfica (Sección 2.1) de los métodos más relevantes para la detección de nubes precipitables y las diferentes técnicas utilizadas en percepción remota en los procesos de clasificación. Posteriormente, la Sección 2.2 presenta, las definiciones de nubes, su clasificación y las especificaciones de la plataforma satelital GOES-16 como las diferentes resoluciones que ésta posee, su proceso de adquisición de datos y la forma de acceder a dichos datos. En la Sección 2.3 se explica los conceptos básicos necesarios para el desarrollo de las técnicas de clasificación empleadas en esta investigación. Posteriormente, se hace una presentación general de los mecanismos para la optimización de hiper-parámetros para los diferentes algoritmos de clasificación (Sección 2.4). Para la definición del vector de características que finalmente se convierte en el patrón de entrada se hace uso de la Gray-Level Co-Occurrence Matrix (GLCM), la cual es presentada en la Sección 2.5. Finalmente, en la Sección 2.6 se muestra el uso del Proceso Unificado Racional (RUP) como elemento para la construcción de la interfaz de usuario.

2.1. Antecedentes

Una nube es un hidrometeoro en suspensión en la atmósfera de partículas de agua líquida y/o de hielo de diferentes dimensiones, que puede contener partículas no acuosas o partículas sólidas como gases industriales, humo, polvo o cenizas volcánicas (Rochas, 1996). La clasificación de éstas es esencial para realizar modelos numéricos de predicción del tiempo, el cálculo de vientos, la estimación de precipitación, entre otros fenómenos meteorológicos (Orun, 2000; Menzel, 2001; Szantai y cols., 2002; Manoj y cols., 2010) que se han venido desarrollando desde el *boom* de la era de los satélites con el fin de comprender la dinámica del clima de nuestro planeta.

Conforme avanzan las nuevas tecnologías y a medida que aumenta la capacidad de los sistemas de observación planetaria, también aumentan las capacidades para monitorear y pronosticar fenómenos meteorológicos a diferentes escalas. El procesamiento de datos satelitales geoestacionarios, en tiempo casi real, es una forma óptima para evaluar el proceso de **Convective Initiation (CI)** (Mecikalski y Bedka, 2006) generando una máscara de nubes tipo *Cumulonimbus* que, aparte de causar fuertes precipitaciones, son las causantes de las tormentas eléctricas; para ello, se han utilizado técnicas de nefoanálisis (Rodríguez y cols., 2014; Pyszczyk y Snajder, 2017) que consisten en la interpretación visual de imágenes por parte de un experto distinguiendo entre diferentes tonalidades de grises, hasta el uso de técnicas más avanzadas como aquellas basadas en aprendizaje automático (Cai y Wang, 2017; Wohlfarth y cols., 2018; Jin y cols., 2019). Para la separación de coberturas se han implementado clasificadores paramétricos, como la **Maximum Likelihood Classification (MLC)**, los cuales brindan excelentes resultados cuando se trata de datos unidimensionales; sin embargo, tienen limitaciones al tratar con conjuntos de datos de entrada multidimensionales, como se presenta en los estudios de percepción remota, porque asumen una distribución de datos normal (Liu y cols., 2011).

Las propiedades ópticas de las gotas de agua y las partículas de hielo que conforman mayormente las nubes, generan una alta reflectividad en las imágenes en la región del espectro visible, mientras que la reflectancia espectral empieza a disminuir a medida que aumenta la longitud de onda (Bai, Li, Sun, Chen, y Li, 2016). Para una identificación correcta de nubes precipitables es necesario utilizar otros tipos de información además de la espectral y por ello se ha empleado información de texturas a partir de la **GLCM** (Arab y cols., 2014; Taravat, Frate, y cols., 2015; Cheng y Yu, 2015; Bai y cols., 2016; Wang y cols., 2018; Wu y Li, 2018) usando métodos de clasificación como: **Decision Tree (DT)**, **SVM**, **RF**, y arquitecturas de **RNA** como **Convolutional Neural Network (CNN)** y **Multilayer Perceptron (MLP)** principalmente (Ramírez y Lizarazo, 2014; Taravat, Proud, y cols., 2015; Cai y Wang, 2017; Wohlfarth y cols., 2018).

Bai y cols. (2016) emplean la información espectral de las bandas azul, verde, roja e **Infrarrojo Cercano (NIR)** acompañado de características de textura **GLCM**, seleccionando el tamaño de la ventana a partir del **Coefficiente de Variación (VC)** y calculando el promedio de la media, la varianza, la uniformidad, el contraste, la correlación y la entropía para las direcciones $0^\circ, 45^\circ, 90^\circ, 135^\circ$ con el fin de evitar la varianza rotacional. Además, emplean el **Índice de Vegetación de Diferencia Normalizada (NDVI)** y un clasificador **SVM** en imágenes de los satélites *GF-1* y *GF-2* con una detección de nubes superior al 91.45%. Wu y Li (2018) clasifican 3 clases de nubes en imágenes del **Satelite Fengyun (FY-2)** mediante **SVM**. Las entradas a este modelo corresponden al **GLCM** (contraste, correlación, energía, homogeneidad y desviación estándar) y la transformada de Gabor. El modelo es entrenado con 60 imágenes, además de 20 imágenes para la validación, la evaluación se realizó mediante la matriz de confusión obteniendo una precisión del 93.33%. Por otro lado, Taravat, Proud, y cols. (2015) evalúan la detección de nubes en imágenes en tierra de vista de cielo a partir de los métodos de clasificación **SVM** y **MLP**. Los hiper-parámetros fueron optimizados por el método de búsqueda de cuadrícula (ver Sección 2.4). Este trabajo obtuvo una precisión de 95.07% y 93.66% para **MLP** y **SVM**, respectivamente.

2.2. Marco teórico

2.2.1. Nubes

Según la [Organización Meteorológica Mundial \(OMM\)](#) las nubes son hidrometeoros en suspensión en la atmósfera de partículas de agua líquida y/o de hielo de diferentes dimensiones, ([OMM, 2011, s.f.](#)). Éstas se crean por la interacción de masas de aire cálido y frío desplazadas por las corrientes de vientos, que contienen vapor de agua y cuando llegan a la atmósfera alcanzan un punto de saturación, a partir de allí, el nuevo vapor entrante a la atmósfera forma gotas y cristales de hielo que posteriormente se unen formando nubes. Las nubes tienen un papel decisivo en las predicciones y los avisos meteorológicos, ya que contribuyen al ciclo del agua y a todo el sistema climático.

2.2.1.1. Clasificación

Las nubes se clasifican en 10 clases denominadas géneros, estas describen las características principales de sus formas, además estas se subdividen en especies, que dependen de la singularidad de su forma y de su estructura interna ([Tapakis y Charalambides, 2013](#)). Hay variedades que definen las características especiales de una nube como se observa en la figura [2.1](#)

Por otro lado, se distinguen regiones, divididas en tres niveles: nubes bajas, medias o altas. Cada nivel está definido por un rango de altura que varía según la región del planeta en la que se encuentren. Las regiones consideradas para la clasificación se observan en la tabla [2.5](#)

Los géneros de las nubes se suelen acomodar de acuerdo a la altura, en la que con mayor frecuencia se encuentran sus bases en:

Tabla 2.1: Géneros y especies de nubes

Género	Especie	Variedades
Cirrus (Ci)	uncinus, fibratus, spissatus, castellanus	intortus, radiatus, vertebratus
Cirrocumulus (Cc)	castellanus, floccus, lenticularis	undulatus
Cirrostratus (Cs)	nebulosus, fibratus	-
Altostratus (As)	-	translucidus, opacus
Alto cumulus (Ac)	castellanus, floccus, lenticularis	translucidus, opacus, undulatus, perlucidus
Stratus (St)	fractus, nebulosus	-
Stratocumulus (Sc)	castellanus, lenticularis	translucidus opacus
Nimbostratus (Ns)	-	-
Cumulus (Cu)	fractus, humilis, mediocris, congestus	-
Cumulonimbus (Cb)	calvus, capillatus	-

Fuente: adaptado de ([Tapakis y Charalambides, 2013](#)).

Tabla 2.2: Nivel de nubes

Nivel	Región Polar	Región Templada	Región Tropical
Nubes altas	3 a 8 km	5 a 13 km	6 a 18 km
Nubes medias	2 a 4 km	2 a 7 km	2 a 8 km
Nubes bajas	superficie a 2 km	superficie a 2 km	superficie a 2 km

Fuente: adaptado de ([Tapakis y Charalambides, 2013](#)).

- Nubes bajas: cumulus, stratus, stratocumulus, cumulonimbus.
- Nubes medias: alto cumulus, altostratus y nimbostratus.
- Nubes altas: cirrus, cirrostratus y cirrocumulus.
- Nubes de desarrollo vertical : cumulonimbus.

2.2.2. Plataforma satelital

El **GOES-16** tiene como principal objetivo estudiar las condiciones meteorológicas desde la órbita geoestacionaria. El satélite posee seis instrumentos de medición entre ellos se encuentra el **Advanced Baseline Imager (ABI)**, que observa una porción del planeta en 16 bandas espectrales comprendidas entre el **Visible (VIS)** e **Infrarrojo (IR)**. Además, para el estudio

2.2 Marco teórico

de las tormentas solares y el sol cuenta con el **Space Environmental In - Situ Suite (SEISS)**, el **Extreme Ultraviolet X - Ray Irradiance Sensors (EXIS)**, el **Solar Ultraviolet Imager (SUVI)** y un magnetómetro. Por último, para la observación de relámpagos en el planeta posee un **Geostationary Lightning Mapper (GLM)** (ver Tabla 2.3).

Tabla 2.3: Bandas espectrales de **GOES-16**.

N. de banda	Res. Espacial (km)	Longitud de onda (μm)	Nombre
01	1	0.47	Banda azul
02	0.5	0.64	Banda roja
03	1	0.86	Banda <i>Veggie</i>
04	2	1.37	Banda cirrus
05	1	1.60	Banda nieve/hielo
06	2	2.24	Banda tamaño de partículas de nube
07	2	3.90	Banda de ventana de onda corta
08	2	6.20	Banda Vapor de Agua (WV) de la troposfera superior
09	2	6.90	Banda WV de la troposfera nivel medio
10	2	7.30	Banda WV de la troposfera de bajo nivel
11	2	8.40	Banda de fase de la cima de la nube
12	2	9.60	Banda de ozono
13	2	10.3	Banda de onda larga de IR limpio
14	2	11.2	Banda de onda larga de IR
15	2	12.3	Banda de onda larga de IR sucio
16	2	13.3	Banda de onda larga de CO₂

Fuente: adaptado de (HARRIS, 2018).

GOES-16 observa continuamente los Estados Unidos, los océanos Pacífico y Atlántico, Centro y Sudamérica y el sur de Canadá. Está ubicado en la meridiano 75.2°W, orbitando a 35.800 *km* sobre el ecuador terrestre a una velocidad igual a la rotación del planeta, es por ello que conserva una posición fija de observación (geoestacionario). La disponibilidad de datos depende de la escala que se requiere. En la Tabla 2.4 se presentan las escalas disponibles de los productos y su resolución temporal.

El instrumento a utilizar de **GOES-16** es el sensor **ABI** que incluye dos canales **VIS**, cuatro canales **NIR** y diez canales de **IR** comprendidas entre los 0.45 – 13.3 μm .

Tabla 2.4: Modos de operación **ABI**

Escala	Extensión espacial	Res. temporal
Disco completo	Cobertura hemisférica de 83° del ángulo cenital local	10min
Mesoescala	Cobertura en un cuadro de 1000x1000 km	30s
CONUS	Cobertura del rectángulo de 5000km (este / oeste) y 3000km (norte / sur) sobre los Estados Unidos continentales o el océano Pacífico	5min

Fuente: adaptado de (HARRIS, 2018).

ABI es un radiómetro de imágenes pasivas multicanal diseñado para observar el hemisferio occidental y proporcionar imágenes de área variable e información radiométrica de la superficie de la Tierra, la atmósfera y la cubierta de nubes. En la figura 2-1 se presenta la forma en que el sensor escanea la superficie del planeta y obtiene las respuestas espectrales.



Figura 2-1: Forma de escanear la superficie del sensor

Fuente: adaptado de (NOAA y GOES-R, 2019).

Los datos del sensor **ABI** están disponibles en tres tipos de procesamiento diferente que se presentan en la tabla 2.5, además es posible descargar el compilado completo a una resolución espacial de 2 km o descargar cada banda en su resolución de origen¹.

¹Disponible en: <https://noaa-goes16.s3.amazonaws.com> o <https://osdc.rcc.uchicago.edu>

Tabla 2.5: Niveles de datos sensor **ABI**

Nivel	Descripción
0	Datos del instrumento sin procesar a resolución completa.
1b	Datos de nivel 0 con corrección radiométrica y geométrica aplicada para producir parámetros en unidades físicas.
2+	Variabes ambientales derivadas con resolución espacial y temporal comparable a nivel 1.

Fuente: adaptado de ([NOAA y NASA, 2018](#)).

2.3. Métodos de clasificación

2.3.1. Support Vector Machine - SVM

Es una técnica aplicada fundamentalmente al procesamiento de grandes cantidades de información y consiste en encontrar un hiperplano en el espacio *N-dimensional* que clasifique correctamente los conjuntos de datos. Para separar estos datos existen diferentes hiperplanos, por lo que el objetivo es encontrar un plano que tenga un margen máximo, es decir, la distancia máxima entre los puntos de datos de cada clase, al maximizar la distancia del margen se proporciona un nivel de confianza mayor para los datos futuros ([Hastie, Tibshirani, y Friedman, 2009](#)).

2.3.1.1. Hiperplanos

Los hiperplanos son límites de decisión que ayudan a clasificar los datos que corresponden a las diferentes clases. La dimensión del hiperplano depende del número de características, por lo que, si la dimensión de los datos de entrada es 2, entonces el hiperplano es solo una línea; si es 3, entonces el hiperplano se convierte en un plano. En general si la dimensión es n el hiperplano es $n - 1$. Así, los datos de entrenamiento consisten en N pares $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$, con $x_i \in \mathbb{R}^n$ y $y_i \in \{-1, 1\}$.

2.3.1.2. Función de costo y actualizaciones de degradado

Con esta función se busca maximizar el margen de separación entre los puntos que representan los datos y el hiperplano. Esta función de costo o pérdida esta dada por la ecuación 2.1.

$$c(x, y, f(x)) = \begin{cases} 0 & \text{si } y \cdot f(x) \geq 1 \\ 1 - y \cdot f(x) & \text{en otro caso} \end{cases} \quad (2.1)$$

El costo es 0 si el valor predicho y el valor real son del mismo signo. Si no lo son, entonces se calcula el valor de la pérdida. También se agrega un parámetro de orden a la función de costo. El objetivo del parámetro de orden es equilibrar la maximización del margen y la pérdida. Después de agregar el parámetro de orden, las funciones de costo serían:

$$\min_w \lambda |w|^2 + \sum_{i=1} 1 - y_i(x_i, w) \quad (2.2)$$

Ahora que se tiene la función de pérdida, se toman derivadas parciales con respecto a los pesos para encontrar los gradientes. Usando los gradientes, se puede actualizar los pesos.

$$\frac{\delta}{\delta w_k} \lambda |w|^2 = 2\lambda w_k \quad (2.3)$$

Cuando no hay errores de clasificación, es decir, el modelo predice correctamente la clase de punto de datos, basta con actualizar el gradiente desde el parámetro de orden:

$$w = w - \alpha(2\lambda w) \quad (2.4)$$

Cuando hay una clasificación errónea, es decir, el modelo comete un error en la predicción

de la clase de punto de datos, se incluye la pérdida junto con el parámetro de orden para realizar la actualización del gradiente.

$$w = w + \alpha(y_i * x_i - 2\lambda w) \quad (2.5)$$

2.3.2. Random Forest - RF

Un **RF** se expresa como un conjunto de clasificadores en forma de árbol o **DT**, donde θ_k es el vector aleatorio del k -ésimo árbol, independiente de los últimos vectores aleatorios $\theta_1, \dots, \theta_{k-1}$. En la etapa de entrenamiento, el algoritmo intenta optimizar los parámetros de las funciones de *split* a partir de las muestras de entrenamiento:

$$\theta_k^* = \arg \max_{\theta} I_j \quad (2.6)$$

Para ello se utiliza la siguiente función de ganancia de información:

$$I_j = H(j) - \sum_{i \in \{1,2\}} \frac{|S_j^i|}{|S_j|} H(S_j^i) \quad (2.7)$$

Donde S representa el conjunto de muestras que hay en el nodo por dividir, S^i son los dos conjuntos que se crean de la división, la función mide la entropía del conjunto donde su respuesta se obtiene a partir del resultado de una votación no ponderada de los distintos árboles (Breiman, 2001).

2.3.2.1. Criterio de convergencia

Dado un conjunto de clasificadores $h_1(x), h_2(x), \dots, h_k(x)$ y con el conjunto de entrenamiento extraído al azar de la distribución del vector aleatorio X, Y . Se define la función de

margen como:

$$mg(X, Y) = \frac{\sum_{k=1}^K I(h_k(X) = Y)}{K} - \max_{j \neq Y} \left[\frac{\sum_{k=1}^K I(h_k(X) = j)}{K} \right] \quad (2.8)$$

Donde, I es la función indicador.

Nota:

Si $mg(X, Y) > 0$, entonces el conjunto de clasificadores vota por la clasificación correcta.

Si $mg(X, Y) < 0$, entonces entonces el conjunto de clasificadores vota por la clasificación incorrecta

El margen mide la seguridad con la que el conjunto acierta o se equivoca en su predicción ya que es la diferencia entre la proporción de árboles que aciertan y los que se equivocan. Cuanto mayor sea el margen, mayor será el grado de confianza en la clasificación. El error de generalización esta dado por:

$$PE^* = P_{X,Y}(mg(X, Y) < 0) \quad (2.9)$$

Donde los subíndices X, Y indican que probabilidad está sobre el espacio X, Y . En **RF**, $h_k(X) = h(X, \theta_k)$ A medida que el número de árboles aumenta, PE^* tiende a converger a:

$$P_{X,Y} \left(P_{\theta}(h(X, \theta) = Y) - \max_{j \neq Y} (P_{\theta}(h(X, \theta) = j < 0)) \right) \quad (2.10)$$

Estimaciones internas supervisan el error, la fuerza y la correlación. Estos se utilizan para mostrar la respuesta al aumento del número de características utilizadas en la división y estimar la importancia de cada variable que alimenta el modelo.

2.3.2.2. Fuerza y correlación

Para evitar que muchos árboles tengan los mismos criterios de clasificación se inyecta aleatoriedad esperando maximizar la independencia de los árboles con una precisión razonable (Hidalgo, 2014). Así, la fuerza del conjunto se define como:

$$s = E_{X,Y} mg(X, Y) \quad (2.11)$$

Donde $mr(X, Y)$ es la función de margen; en el caso de dos clases que se expresa como:

$$mr(X, Y) = E_{\theta}[c(X, \theta) = Y] - E_{\theta}[c(X, \theta) \neq Y] = 2 * E_{\theta}[c(X, \theta) = Y] - 1 \quad (2.12)$$

Donde $E_{\theta}[c(X, \theta) = Y]$ es el límite de la proporción de árboles c_t que, dado un patrón x , lo clasifican correctamente (al aumentar T):

$$\frac{1}{T} \sum_{t=1}^T I(c_t(x, \theta_t) = y) \longrightarrow E_{\theta}[c(x, \theta) = y] \quad (2.13)$$

De esta forma, tal como lo presenta Medina y Ñique (2017), el margen es un valor definido en el intervalo $[-1, +1]$, siendo positivo cuando el ejemplo se ha clasificado correctamente y negativo en caso contrario. La razón de un clasificador se define como:

$$\frac{c}{s^2} = \frac{\bar{p}}{s^2} \quad (2.14)$$

Donde c es la correlación del conjunto, entre menor sea c y mayor sea s la independencia entre los distintos árboles del conjunto será mayor. La correlación de los árboles dentro del conjunto se mide de la siguiente forma:

$$\bar{p} = E_{\theta, \theta'}[\rho(c(\cdot, \theta), c(\cdot, \theta'))] \quad (2.15)$$

2.3 Métodos de clasificación

ρ corresponde al coeficiente de correlación entre dos variables aleatorias y (θ, θ') son independientes con la misma distribución. En la figura 2-2 se presenta una representación gráfica del entrenamiento y clasificación del un RF.

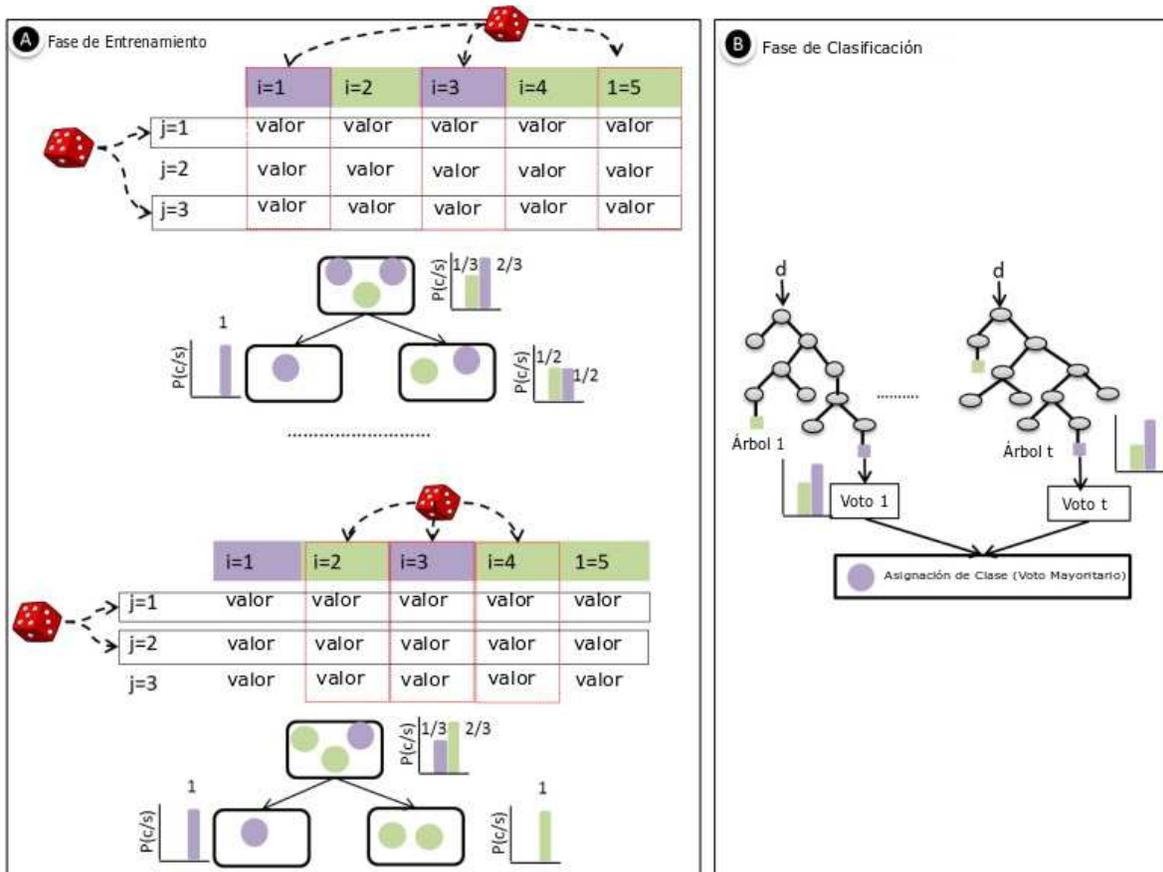


Figura 2-2: Proceso de entrenamiento y clasificación de RF: a) fase de entrenamiento, b) fase de clasificación.

Fuente: adaptado de (Belgiu y Drăguț, 2016).

2.3.3. Redes Neuronales Artificiales - RNA

Según (Caicedo y López, 2009) las redes neuronales artificiales son un modelo matemático que simula el funcionamiento de una neurona biológica que depende de unos datos iniciales para generar un proceso de aprendizaje adaptativo. El aprendizaje podría en una RNA

tener la capacidad de resolver problemas complejos como procesamiento de señales, reconocimiento de patrones, reconocimiento facial, clasificación de imágenes, entre otros.

Las redes neuronales tipo *Feed-Foward* han sido utilizadas para resolver problemas de identificación o clasificación (Olabe, 2008; Taravat, Frate, y cols., 2015). Este tipo de RNA esta compuesta principalmente de 3 secciones o capas: una capa de entrada, una capa oculta y otra de salida. Estas capas se componen de neuronas que están interconectadas en una relación 1 : n con la capa siguiente, donde la información fluye en un único sentido desde la entrada hasta llegar a la salida. Toda red neuronal inicia usando sus pesos sinápticos w y umbrales θ (*bias*) de manera aleatoria, donde se presenta un vector de entrada x_p :

$$x_p = [x_{p1}, x_{p2}, \dots, x_{pi}, \dots, x_{pN},]^T \quad (2.16)$$

Así se calcula la salida lineal de la red neuronal para la capa oculta, dando como resultado la salida:

$$z_{pk}^h = \sum_i^N w_{ji}^h x_{pi}^o + \theta_j^h \quad (2.17)$$

Posteriormente, se calcula la salida de la capa oculta, resultado de las funciones de activación $f(\cdot)$, las cuales podrían ser tangente hiperbólica, sigmoide, u otras variaciones.

$$a_{pj}^h = f_j^h(z_{pj}^h) \quad (2.18)$$

Después de calcular las salidas de la capa oculta, se debe calcular los valores netos de la entrada para la capa de salida:

$$z_{pk}^o = \sum_{j=1}^L w_k^o a_{pj}^o + \theta_k^o \quad (2.19)$$

Posteriormente se calcula las salidas de la red:

$$a_{pk}^o = f_k^o(z_{pk}^o) \quad (2.20)$$

2.4 Optimización de hiperparámetros

Con ello se calcula los términos de error para las unidades de salida.

$$\delta_{pk}^o = (d_{pk} - y_{pk}^o) f_k^{\prime}(a_{pk}^o) \quad (2.21)$$

La ecuación se utiliza para actualizar los pesos sinápticos de las redes neuronales en un procedimiento conocido como propagación hacia atrás. Los algoritmos de optimización son diversos, pero su gran mayoría buscan encontrar un buen factor de aprendizaje tal que garanticen una rápida convergencia que minimice la función de costo, que para este caso en particular sería el error cuadrático medio:

$$E_p = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^M (d_{pk} - y_{pk})^2 \quad (2.22)$$

2.4. Optimización de hiperparámetros

El rendimiento de un algoritmo puede depender en gran medida de la elección de los hiperparámetros, estos son parámetros ajustables que se eligen para entrenar un modelo y que rigen el propio proceso de entrenamiento. Por ejemplo, para entrenar un **RF** se debe decidir el número de variables muestreadas aleatoriamente como candidatos en cada división (*mtry*) y el número de árboles que componen el **RF** (*nree*). El objetivo de la optimización del hiperparámetro en el aprendizaje automático es encontrar los hiperparámetros de un determinado algoritmo de aprendizaje automático que permita el mejor rendimiento medido en un conjunto de validación (Koehrsen, 2018b).

La optimización de los hiperparámetros se representa en forma de ecuación como:

$$x^* = \operatorname{argmin} f(x) \quad (2.23)$$

2.4 Optimización de hiperparámetros

Donde, $f(x)$ representa una puntuación objetiva para minimizar, como una tasa de error o la **Raíz del Error Cuadrático Medio (RMSE)**, evaluada sobre el conjunto de validación; x^* es el conjunto de hiperparámetros que produce el valor más bajo de la puntuación y x puede tomar cualquier valor del dominio X ($x \in X$). Dentro de las técnicas de optimización se encuentran:

Búsqueda de cuadrícula: Consiste en una búsqueda exhaustiva a través de un subconjunto de hiperparámetros especificados manualmente, que comienza a iterar en el espacio definido y evalúa alguna métrica para conocer el rendimiento del modelo; este método no tiene en cuenta el resultado de validación anterior.

Búsqueda aleatoria: Similar al anterior se define un conjunto de hiperparámetros a optimizar, de este se selecciona un subconjunto de ellos aleatoriamente y se evalúa el rendimiento, luego se realiza una nueva selección aleatoria y se evalúa.

Optimización bayesiana: El concepto es limitar las evaluaciones de la función objetivo mediante la selección de los siguientes valores de entrada en función de los que han funcionado bien en el pasado. En el caso de la optimización del hiperparámetro, la función objetivo es el error de validación de un modelo de aprendizaje automático utilizando un conjunto de hiperparámetros. El objetivo es encontrar los hiperparámetros que producen el error más bajo en el conjunto de validación con la esperanza de que estos resultados se generalicen al conjunto de prueba. Idealmente, el método busca explorar el espacio de los hiperparámetros y al mismo tiempo, limitar las evaluaciones de las opciones deficientes; es decir, el ajuste bayesiano del hiperparámetro utiliza un modelo de probabilidad actualizado continuamente para “concentrarse” en los hiperparámetros prometedores al tener en cuenta los resultados anteriores. (Dewancker, McCourt, y Clark, 2015). Las cinco partes del problema son:

- Un dominio de hiperparámetros para buscar la mejor configuración posible.
- Una función objetivo que entrega los hiperparámetros que la minimizan.

- El modelo sustituto de la función objetivo.
- Una función de selección para evaluar los hiperpámetros para el modelo sustituto.
- Historial de resultados que almacena las evaluaciones para actualizar el modelo sustituto.

2.5. Gray-Level Co-Occurrence Matrix - GLCM

Haralick, Shanmugam, y Dinstein (1973) publicaron un trabajo que ha tenido un gran impacto en el análisis de imágenes, porque permite cuantificar las características texturales presentes en una imagen. Al método lo denominaron **GLCM**, lo aplicaron con el objetivo de discriminar distintos tipos de terreno en imágenes satelitales. Esta es una matriz de frecuencias con la que un píxel con un nivel de gris (i) aparece en una relación de espacio específica con otro píxel de nivel de gris (j). Las matrices de concurrencia son medidas de segundo orden porque consideran parejas de píxeles vecinos, separados una distancia d y en un determinado ángulo. Por tanto, las matrices de coocurrencia pueden revelar ciertas propiedades sobre la distribución espacial de los grises en la textura de la imagen.

Matemáticamente, se puede definir el **GLCM** como:

$$P_{(dx,dy)}P(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & \text{if } I(p, q) = i \text{ and } I(p + dx, q + dy) = j \\ 0, & \text{otherwise} \end{cases} \quad (2.24)$$

Donde, $P_{(dx,dy)}P(i, j)$ es una matriz cuadrada, que representa la frecuencias relativas con las que dos pixeles están separados por una distancia $d = (dx, dy)$ en la imagen I , el primero con tono gris i y el segundo píxel con tono gris j (Salhi y cols., 2018). Las características texturales para este trabajo son:

■ **Contraste**

$$Contrast = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i-j)^2 G(i,j) \quad (2.25)$$

■ **Correlación**

$$Correlation = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{(G(i,j)(i \times j) - (\mu_x \times \mu_y))}{\sigma_x \sigma_y} \quad (2.26)$$

■ **Energía**

$$Energy = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} G(i,j)^2 \quad (2.27)$$

■ **Entropía**

$$Entropy = - \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} G(i,j) \log_e(G(i,j)) \quad (2.28)$$

■ **Homogeneidad**

$$Homogeneity = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{G(i,j)}{1 + |i-j|} \quad (2.29)$$

■ **Media**

$$Media = \mu_i = \sum_{j=0}^{N-1} ij = 0^{N-1}(P_{ij})\mu_j = \sum_{j=0}^{N-1} ij = 0^{N-1}(P_{ij}) \quad (2.30)$$

■ **Varianza**

$$Varianza = \sigma^2 = \sum_{ij=0}^{N-1} P_{ij}(i - \mu_i)^2; \sigma^2 = \sum_{ij=0}^{N-1} P_{ij}(j - \mu_j)^2 \sigma_i = \sqrt{\sigma_i^2}; \sigma_j = \sqrt{\sigma_j^2} \quad (2.31)$$

2.6. Rational Unified Process - RUP

RUP es un marco de procesos integrales que proporciona prácticas probadas en la industria para la implementación de software y sistemas, y para la gestión eficaz de proyectos (Reyes-Delgado y cols., 2016). Además, proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es garanti-

zar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales (Kruchten, 2004).

2.6.1. Fases del RUP

La metodología **RUP** divide el proceso de desarrollo en 4 fases (figura 2-3). En la fase de iniciación se encamina hacia el planteamiento del problema, la delimitación del alcance de proyecto y al establecimiento del punto de partida. El propósito de la fase de elaboración es seleccionar los casos de usos que definen la arquitectura base del sistema y dan una solución preliminar. En fase de construcción se completa la funcionalidad del sistema para que en la fase de transición se asegure que el proyecto está libre de errores y listo para usuario final.

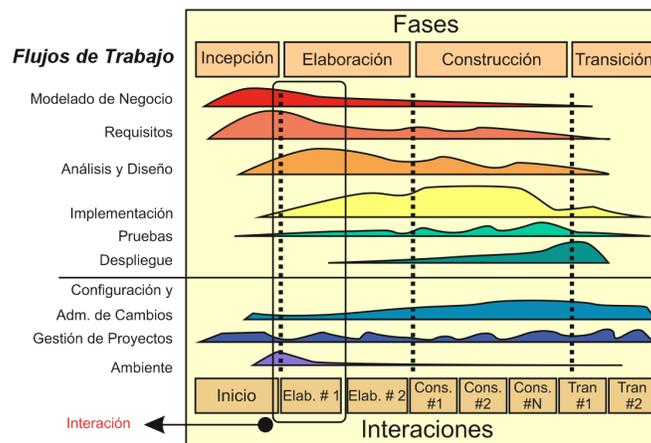


Figura 2-3: Fases de **RUP**.

2.7. Síntesis

Este capítulo aborda los principales conceptos que definen los componentes de la metodología. Cada uno de los conceptos presentados, serán un insumo importante en la obtención

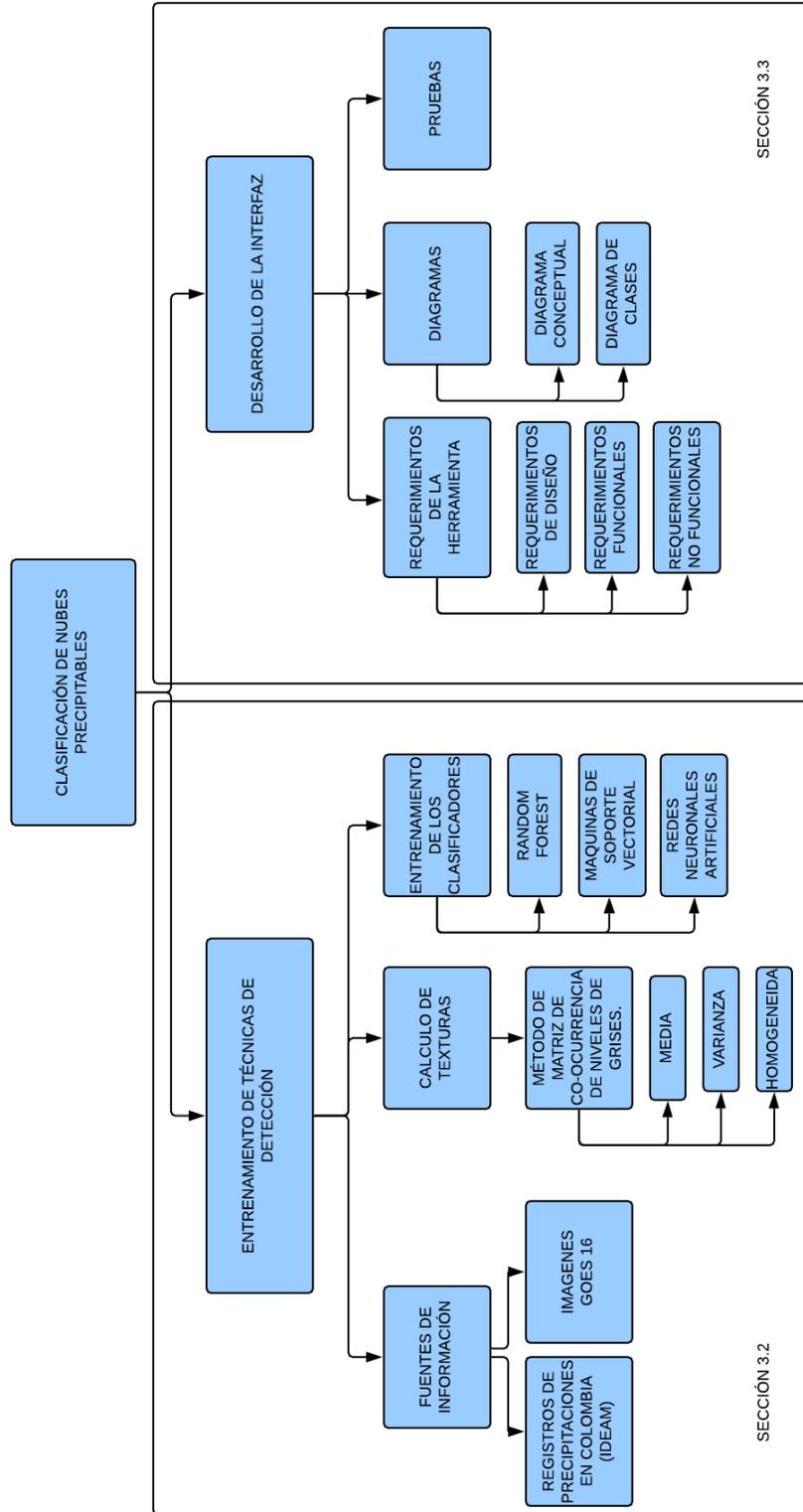
tanto de los clasificadores, de la forma de seleccionarlos, así como la manera de seleccionar el conjunto de características que representarán cada patrón. Finalmente, se especifica la construcción de la Interfaz de Usuario a través del marco RUP.

Capítulo 3

Determinación de nubes precipitables en Colombia

En este capítulo se describe la zona de estudio y la metodología propuesta para llevar a cabo el desarrollo de una herramienta para la identificación y clasificación de nubes precipitables, orientado a cumplir los objetivos propuestos en este trabajo. En el diagrama de la figura 3-1 se observa la metodología propuesta.

Figura 3-1: Metodología propuesta

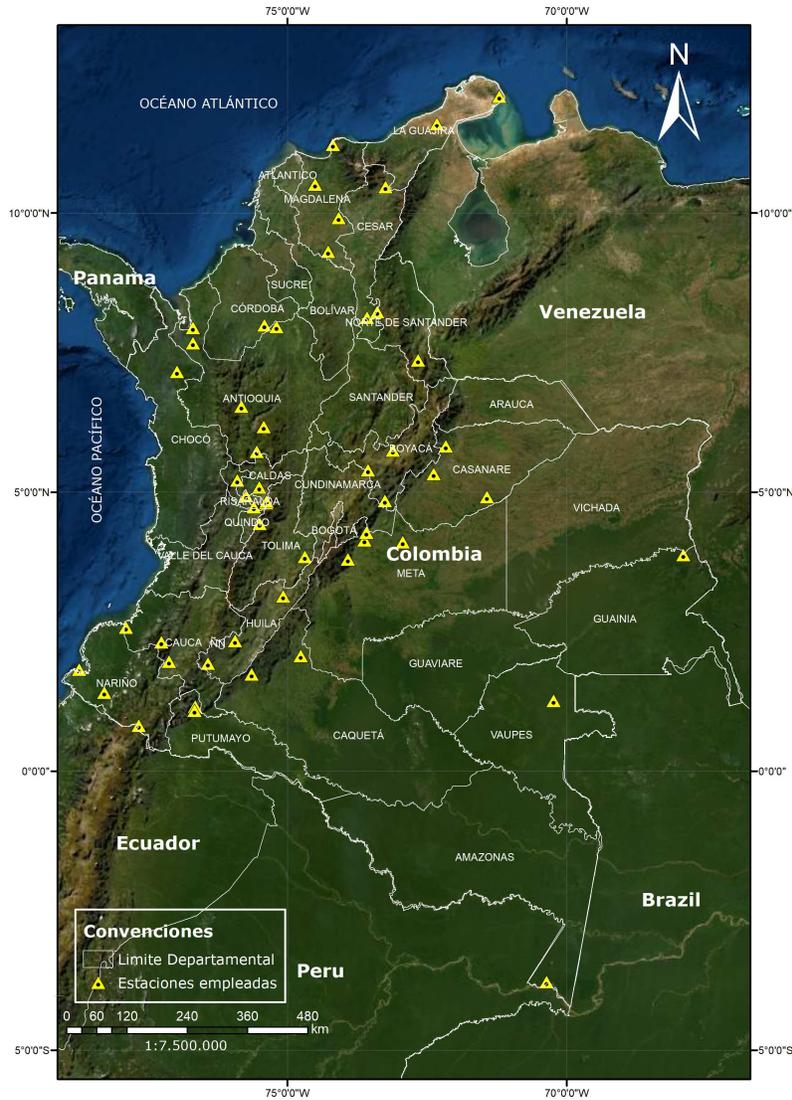


La metodología se desarrolla en dos partes principales: la primera consiste en el entrenamiento de los modelos ML (3.2.1) en esta se obtienen los dos datos tanto de entrenamiento como de validación que serán usado en las diferentes técnicas de ML, La segunda parte consiste en la implementación de la herramienta para poder visualizar y elegir los métodos de clasificación, esta consiste en la explicación detallada de cada módulo presente en la interfaz gráfica de usuario para la que se utilizó el programa Qt5 y un lenguaje de programación en Python (3.3)

3.1. Zona de estudio

Colombia (ver Fig. 3-2) se localiza al noroeste del continente de América del Sur, este recibe la influencia de la circulación atmosférica de la cuenca Amazónica, además del Mar Caribe y Óceano Pacífico (Guzmán, Ruiz, y Cadena, 2014). Debido a sus diferentes relieves, ecosistemas y biodiversidad, cuenta con un comportamiento climático variable dependiendo de la región en la que se encuentre. Sus cadenas montañosas son producto de la actividad sísmica en la zona de convergencia de las placas tectónicas, perteneciendo así al cinturón de fuego del Pacífico.

Figura 3-2: Zona de estudio.



El clima del país es generado a partir de aspectos geográficos y atmosféricos como la cantidad de radiación solar, la orografía, la humedad atmosférica, entre otros, con temperaturas desde bajo cero en las cumbres de las montañas de las diferentes cordilleras y climas calidos desde 35° en las costa, además de microclimas, como por ejemplo la Sierra Nevada de Santa Marta (Pabón y cols., 2001).

3.2. Metodología para la identificación de nubes precipitables y no precipitables

3.2.1. Entrenamiento de técnicas de detección

El proceso de entrenamiento se realizó en 3 fases: la primera (3.2.2) consta de la obtención de una base de registros de eventos de precipitación que posteriormente son usados para descargar las imágenes. En la fase 2 (3.2.3), se procesaron las imágenes obtenidas anteriormente, este procesamiento consistió en la obtención de las características de texturas que son usados para crear una base de datos de entrenamiento. En la fase final (3.2.4) se entrenan los modelos y se emplea la optimización de hiperparámetros para obtener los mejores resultados de clasificación.

3.2.2. Fuentes de información

Para el desarrollo del proyecto es necesario contar con algunas fuentes de información como lo es una base de registros de eventos de precipitación obtenidos a partir de estaciones meteorológicas del IDEAM¹. Se seleccionaron 52 estaciones que permiten representar las 17 regiones de precipitación descritas en (Guzmán y cols., 2014). Se eligieron 2 tipos de estaciones, climatológica ordinaria: que poseen un pluviómetro, pluviógrafo y psicrómetro para medir precipitación y humedad relativa; estaciones climatológicas principales: que son aquellas en la cual se hacen observaciones de visibilidad, tiempo atmosférico presente, cantidad, tipo y altura de las nubes, estado del suelo, precipitación, temperatura del aire, humedad, viento, radiación, solar, brillo solar, evaporación y fenómenos espaciales.

El segundo insumo son las imágenes satelitales del GOES-16 que se encuentran alma-

¹Información encontrada en <http://dhime.ideam.gov.co/atencionciudadano/>

3.2 Metodología para la identificación de nubes precipitables y no precipitables

cenadas en AWS ¹, estas son de acceso público.

3.2.3. Obtención de imágenes

Debido al volumen de información y los diferentes requerimientos del proyecto, todo se hizo con métodos automatizados en los lenguajes de programación **Python** y **R**, además, para la visualización de datos espaciales se empleó **Qgis**.

Para la extracción de las características inicialmente se procede a recolectar una base de registros de precipitación. Las estaciones automáticas poseen un producto con reportes cada 10 min, donde se tomaron registros entre el periodo comprendido del 3 de abril de 2019 hasta el 19 de diciembre del mismo año; esta fecha de inicio fue escogida conforme al satélite **GOES-16**, ya que cambió su modo de escaneo de modo 3 a modo 6, permitiendo obtener imágenes del disco completo cada 10 minutos (**NOAA y GOES-R, 2019**). Por lo tanto, se obtuvieron los registros de precipitación de las 52 estaciones seleccionadas con una resolución temporal de 10 minutos durante el periodo de tiempo descrito anteriormente. Teniendo estos datos a disposición se filtraron los registros inferiores a $1mm$ con el fin de eliminar posibles registros con ruido o anomalías de las estaciones. La base de datos de los registros meteorológicos de las estaciones cuentan con nombre de la estación, su ubicación geográfica (latitud, longitud) y la cantidad medida.

Para la descarga de imágenes se emplea la biblioteca **boto3** de **Python** que permite crear, configurar y acceder a servicios de AWS. Conociendo las fechas y horas de los registros recolectados anteriormente, se descargan las imágenes correspondientes. Para ello el nombre de los productos del **GOES-16** están estandarizados como se explica a continuación:

OR_ABI-Lxx-CMIPF-MxCxx_G16_

¹Información obtenida de: http://home.chpc.utah.edu/~u0553130/Brian_Blaylock/cgi-bin/generic_AWS_download.cgi?DATASET=noaa-goes16

3.2 Metodología para la identificación de nubes precipitables y no precipitables

sYYYYJJJHHMMSSs_eYYYYJJJHHMMSSs_cYYYYJJJHHMMSSs.nc

El nombre de los archivos esta dado bajo este formato donde todos los productos inician con OR que significa Sistema operativo de datos en tiempo Real, además:

ABI:	es el nombre del sensor
Lxx:	es el nivel de procesamiento de los datos, datos L1b o L2
CMIP:	nombre del producto
F:	cobertura de la imagen
Mx:	modo escaneo (ejemplo: <i>M3</i> , <i>M4</i>).
Cxx:	número de la banda espectral (ejemplo: 02, 13).
G16:	identificación del satélite
s:	es el inicio del tiempo de escaneo
e:	final de la observación
c:	tiempo de creación del archivo netCDF4
	formato de fecha especificado así:
	YYYY - Año de 4 dígitos
	JJJ - Día del año de 3 dígitos
YYYYJJJHHMMSSs:	HH - Hora de 2 dígitos
	MM - Minuto de 2 dígitos
	SS - Segundo de 2 dígitos
	s - 1 dígito décimo de segundo
.nc:	formato de la imagen

El tiempo está dado en *UTC* – 0; sin embargo, para nuestro caso el nombre se configura de la siguiente manera:

OR_ABI-L2-MCMIPF-M6_G16_s20192681700237_e20192681709545_c20192681710062.nc

Se usa el sensor ABI, con datos de procesamiento L2, además se emplea el producto MCMIP que consiste en un compilado de las 16 bandas a una resolución espacial de 2 km, este producto consiste de datos procesados a nivel de parámetros físicos. Las bandas ABI 1 a 6, indican la cantidad de factor de reflectancia adimensional que es normalizada por el ángulo cenital solar. Estas bandas apoyan la caracterización de nubes, vegetación, nieve/hielo y aerosoles. El valor de las imágenes para las bandas emisivas, bandas ABI 7 a 16, es la

3.2 Metodología para la identificación de nubes precipitables y no precipitables

temperatura de brillo en la parte superior de la atmósfera (TOA) en *Kelvin*. Estas bandas apoyan la caracterización de la superficie, las nubes, el vapor de agua, el ozono, las cenizas volcánicas y el polvo en función de las propiedades emisivas. Además, el modo de escaneo 6 con resolución temporal de 10 minutos, del satélite *G16*, donde, a partir de los registros de precipitación, se configura el inicio de escaneo en el que se transforma del formato mes/año a día del año. Esto requiere que se configure el tiempo a *UTC - 5* para que coincida con los registros. Con todas estas rutinas desarrolladas se realizan las peticiones al servidor y se procede a descargar las imágenes para un total de 2.916.

3.2.4. Cálculo de texturas

Teniendo las imágenes ya disponibles de forma local se procederá a extraer las bandas necesarias y almacenarlas en formato *tif* con un sistema de referencia *EPSG* : 4326, para manipular los datos de manera sencilla. De las 16 bandas que nos proporciona este sensor se seleccionaron 7 bandas espectrales que se describen en la tabla 3.1. La selección de éstas se basó en las consultas bibliográficas como (Chen y cols., 1989; Tian y cols., 1999; Berendes y cols., 2008; Taravat, Proud, y cols., 2015; Nurtanio y cols., 2019). Estas imágenes son recortadas al cuadro de coordenada ($-81.72W, -66.87W, -4.24S, 12.59N$) que corresponde a la zona de estudio (Sección 3.1).

Con las imágenes ya depuradas se procede a calcular las texturas. Se aplica una conexión **Python-R** donde se envían las imágenes desde Python para que se realice el respectivo cálculo de las texturas en R. Las ecuaciones de las texturas se presentan en la tabla 3.2.

3.2 Metodología para la identificación de nubes precipitables y no precipitables

Tabla 3.1: Bandas seleccionadas para el procesamiento

Banda	Longitud de onda	Principal uso
Azul	0.47	Aerosol diurno sobre tierra, mapeo de aguas costeras
Rojo	0.86	vegetación diurna, cicatriz de quemaduras, aerosol sobre agua y vientos
Infrarrojo cercano	3.9	superficie y nube, niebla en la noche, fuego, vientos
WV Low	7.30	nivel inferior de vapor de agua, vientos y SO2
WV Medium	6.90	vapor de agua atmosférico de nivel medio, vientos, lluvia
WV high	6.20	vapor de agua atmosférico de alto nivel, vientos, lluvia
Banda de onda larga de IR limpio	10.3	superficie y nube

Tabla 3.2: Medidas de texturas, abreviaturas y ecuaciones

Abrev.	Nombre	Fórmula	Abrev.	Nombre	Fórmula
Hom	Homogeneidad	$\sum_{ij=0}^{N-1} P_{ij} / 1 + (i-j)^2$	Ent	Entropía	$\sum_{ij=0}^{N-1} P_{ij} (-\ln P_{ij})$
Con	Contraste	$\sum_{ij=0}^{N-1} P_{ij} (i-j)^2$	Asm	Segundo momento angular	$\sum_{ij=0}^{N-1} P_{ij}^2$
Mean	GLCM media	$\mu_i = \sum_{ij=0}^{N-1} (P_{ij}) \mu_j = \sum_{ij=0}^{N-1} (P_{ij})$	Dis	Disimilitud	$\sum_{ij=0}^{N-1} P_{ij} i-j $
Abrev.	Nombre	Fórmula			
Var	Variación de GLCM	$\sigma^2 = \sum_{ij=0}^{N-1} P_{ij} (i - \mu_i)^2; \sigma^2 = \sum_{ij=0}^{N-1} P_{ij} (j - \mu_j)^2 \sigma_i = \sqrt{\sigma_i^2}; \sigma_j = \sqrt{\sigma_j^2}$			
Cor	Correlación GLCM	$\sum_{ij=0}^{N-1} P_{ij} \left[(i - \mu_i)(i - \mu_j) / \sqrt{(\sigma_i^2)(\sigma_j^2)} \right]$			

3.2 Metodología para la identificación de nubes precipitables y no precipitables

En el análisis de información espacial, los patrones de textura de las coberturas pueden ser usados para técnicas de clasificación. La **GLCM** y los cálculos de características de textura asociados son técnicas de análisis de imágenes. Dada una imagen compuesta de píxeles cada uno con una intensidad (un nivel de gris específico), la **GLCM** tabula la frecuencia con la que coexisten diferentes combinaciones de niveles de gris en a una distancia en píxeles y una dirección en una imagen. Los cálculos de las características de textura usan el contenido del **GLCM** para medir la variación en la intensidad (también conocida como textura de la imagen) en el píxel de interés.

Las características usadas para este estudio fueron: la homogeneidad, la media y varianza. Estas texturas fueron seleccionadas basados en las propuestas de diferentes autores ya mencionados, en la observación manual que se hizo de cada patrón de textura y en la representatividad que tenía en el fenómeno estudiado, que para este caso es la detección de nubes precipitables. Debido a la resolución espacial de las imágenes se decidió calcular las texturas a una distancia de una vecindad, es decir para conformar una ventana de cálculo de 3×3 y rotación invariante obtenida a partir de la combinación de calcular las texturas para 8 direcciones ($0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$).

Una vez calculada las texturas para todas las imágenes se extraen los datos de entrenamiento, para ello se cuenta con los registros de precipitación y las coordenadas de cada estación, además con la resolución temporal de ambas fuentes de información, ellas se superponen para determinar si hubo o no precipitación en caso que el píxel muestre la presencia de nube. Con lo anterior, se cuenta con una base de datos de entrenamiento que cuenta con 21 características y 52 registros de precipitación, con ello se tienen los datos de entrenamiento para la clase nube precipitable. Para la clase nube no-precipitable y cielo despejado o suelo desnudo, se realiza una selección manual como se observa en las figuras 3-3 y 3-4 en las que se enmascaran pequeñas regiones que permiten representar las diferentes clases.

Finalmente la base de datos de entrenamiento cuenta con 42.000 registros, cada clase

3.2 Metodología para la identificación de nubes precipitables y no precipitables

Figura 3-3: Muestras de clase nubes no-precipitables

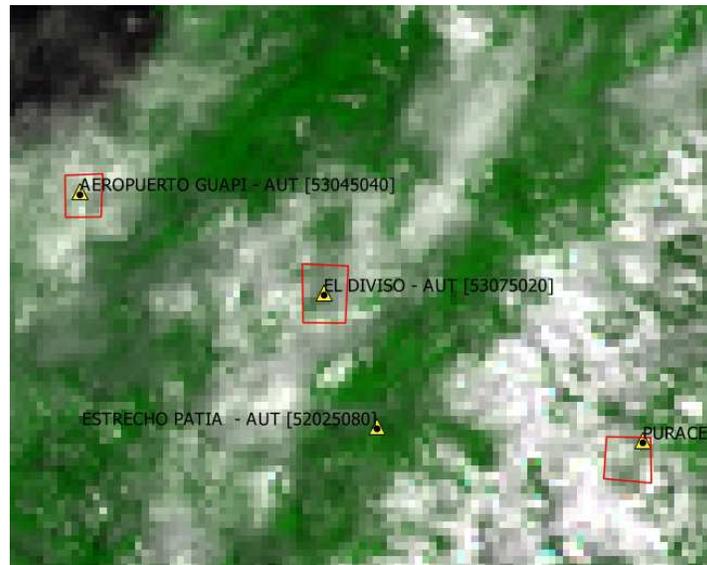
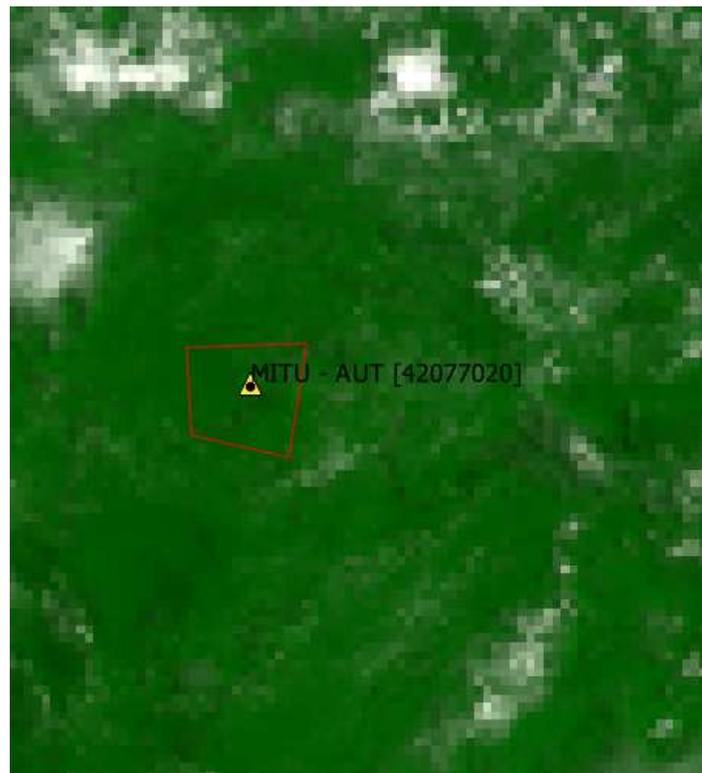


Figura 3-4: Ausencia de nubes desde la imagen satelital



3.2 Metodología para la identificación de nubes precipitables y no precipitables

de 14.000 con el fin de tener una base de entrenamiento balanceada.

3.2.5. Entrenamiento de los clasificadores

El proceso de entrenamiento de cada uno de los modelos de aprendizaje seleccionados, consistió en la sintonización de los hiperparámetros. **Scikit-Learn** implementa un conjunto de hiperparámetros predeterminados razonables para todos los modelos, pero no se garantiza que sean óptimos para un problema. El ajuste del hiperparámetro depende tanto de la teoría como de los resultados experimentales, por lo tanto, el mejor método para determinar la configuración óptima es probar muchas combinaciones diferentes para evaluar el rendimiento de cada modelo. Con el fin de obtener modelos con buenos desempeños y una capacidad de generalización evitando el sobre-ajuste se emplea una optimización de hiperparámetros con validación cruzada.

La validación cruzada consiste en dividir los datos de entrenamiento en un número de subconjuntos denominados pliegues, Luego se ajusta iterativamente el modelo K veces. Cada vez se entrena con $K - 1$ de los pliegues y se evalúa un pliegue en la validación. Como ejemplo, considere ajustar un modelo con $K = 5$. La primera iteración que entrenamos en los primeros cuatro pliegues y evaluamos en el quinto. La segunda vez se entrena en el primero, segundo, tercero y quinto y se evalúa en el cuarto. Se repite este procedimiento 3 veces más, evaluando cada vez en un pliegue diferente. Al final, se promedia el rendimiento en cada uno de los pliegues para obtener métricas de validación del modelo de la figura 3-5.

Para el ajuste de hiperparámetros, se realizan varias iteraciones, cada vez que se agrega alguna configuración específica del modelo. Estos modelos son comparados para seleccionar el mejor. Así, identificando un modelo, se realiza el proceso de validación cruzada (Koehrsen, 2018a); es decir, para cada modelo de aprendizaje primero se realiza una optimización aleatoria con el fin de explorar un mayor rango de combinaciones del espacio de hiperparámetros.

3.2 Metodología para la identificación de nubes precipitables y no precipitables



Figura 3-5: Validación cruzada.
Fuente: adaptado de (Koehrsen, 2018a)

Basado en los mejores resultados de esta optimización se procede a realizar un refinamiento de espacios por cuadrícula, con el fin de encontrar las regiones más efectivas.

3.2.5.1. Random Forest (RF)

Este método se entiende por un grupo de clasificadores basados en árboles de decisión, el cual se construye a partir de un vector aleatorio y el resultado se obtiene mediante votación no ponderada (Breiman, 2001). Los hiperparámetros más importantes son la cantidad de árboles ($n_{estimators}$) y la cantidad de características consideradas para la división en cada nodo ($maxfeatures$).

Para este método en la optimización aleatoria se configuran los hiperparámetros:

- $n_{estimadores}$ = número de árboles en el bosque, este se define en un rango desde 50 a 2000 cada 50 árboles.
- $maxfeatures$ = número máximo de características consideradas para dividir un nodo, hay dos opciones que se puede definirse ya sea automático o **sqrt**¹.
- $maxdepth$ = número máximo de niveles en cada árbol de decisión, rango (10, 110, 10)

¹<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

3.2 Metodología para la identificación de nubes precipitables y no precipitables

- *minsamples_split*= número mínimo de puntos de datos colocados en un nodo antes de que el nodo se divida [2, 5, 10]
- *minsamples_leaf*= número mínimo de puntos de datos permitidos en un nodo [1, 2, 4]
- *bootstrap*= método para muestrear puntos de datos (con o sin reemplazo) [*True, False*]

Posteriormente se realiza la optimización bayesiana usando los resultados obtenidos de la optimización aleatoria.

3.2.5.2. Máquinas de soporte Vectorial (SVM)

Las SVM es una técnica que caracteriza la superficie de dos clases de distintos puntos de entrada, esta está dada por los vectores de soporte que son capaces de crear una frontera de decisión alrededor de los datos de aprendizaje, estos datos son mapeados por medio de diferentes *kernels*. Este método busca la máxima separación entre las clases mediante la creación de hiperplanos. Para el entrenamiento de este algoritmo se tuvo en cuenta los diferentes tipos de *kernels* estos fueron:

I Lineal

II Polinomial

III Sigmoide

IV RBF

Los parámetros que se configuran en la optimización aleatoria son:

- $C = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 10, 100, 1000]$

3.2 Metodología para la identificación de nubes precipitables y no precipitables

- $kernel = ['linear', 'rbf', 'poly', 'sigmoid']$
- $gamma = [1, 0.1, 0.01, 0.001, 0.0001]$
- $poly\ degree = [3, 4, 5, 6]$

3.2.5.3. Redes Neuronales Artificiales (RNA)

Las RNA son un modelo que simula el funcionamiento de la neurona biológica que depende de datos iniciales para generar un proceso adaptativo que posteriormente se emplea en la capacidad de resolver problemas complejos como el reconocimiento de patrones y la clasificación de imágenes (Caicedo y López, 2009). En este proceso se utilizó una MLP de una capa con 48 entradas, de 10 a 50 neuronas en la capa oculta, y se utilizaron algoritmos de optimización: Adam y Adamdelta con un *dropout* que variaba de (0, 0.1 y 0.2). La función de costo de la red es la de entropía cruzada categórica (*categorical-crossentropy*) dada la naturaleza del problema, y se usaron funciones de activación *relu* en la capa oculta y *softmax* en la conexión de salida.

Los parámetros que se configuran en la optimización aleatoria son:

- $Hidden\ neurons = [10, 15, 20, 25, 30, 35, 40]$
- $batch\ size = [30, 60, 128]$
- $epochs = [10, 20, 30, 40, 50, 60]$
- $dropout = [0, 0.1, 0.2]$
- $Kernel\ inicial = ['uniform', 'normal']$
- $Optimizadores = ['Nadam', 'Adam', 'adadelta']$

- *Loss Function* = ['categorical_crossentropy', 'mean_squared_error']
- *Activation Function* = ['relu', 'elu']
- *Output Function* = ['softmax']

3.3. Desarrollo de la interfaz

El desarrollo de la herramienta para la detección de nubes precipitables se llevó a cabo a través de una metodología RUP, donde se presentan las diferentes etapas que tiene la herramienta, teniendo como fase de inicio el reconocimiento de los requerimientos funcionales y no funcionales, el diagrama conceptual, el diagrama de clase y un diagrama entidad-relación. Cada componente de este proceso y las pruebas de su funcionamiento se describirán de forma detallada la manera para dar cumplimiento a los requerimientos funcionales y no funcionales de los módulos que conforman la interfaz.

Los requerimientos funcionales y no funcionales, así como los diferentes diagramas de los módulos de la herramienta se explican en las siguientes secciones.

3.3.1. Requerimientos funcionales y no funcionales

Los requerimientos funcionales y no funcionales necesarios para el desarrollo de la herramienta se presentan en la tabla 3.3 y 3.4 respectivamente. En la columna "Función" describe el requerimiento brevemente y la columna final corresponde a la categoría en la que se encuentra, si es opcional (O) o esencial (E).

Tabla 3.3: Requerimientos funcionales

Nº	Función	Categoría
1.0	Cargar algoritmo de procesamiento de la imagen GOES (preprocesamiento)	E
2.0	Cargar la imagen GOES en formato <i>.nc</i>	E
3.0	Cargar algoritmos de machine learning (SVM, RF, RNA)	E
4.0	Ajustar la imagen a lo requerido por el algoritmo	E
5.0	Procesar las bandas requeridas para la detección de nubes (Rojo, azul, infrarojo WV...)	E
6.0	Cargar la zona de interés en formato <i>.shp</i>	O
7.0	Mostrar la zona elegida por el usuario	O
8.0	Procesar las imágenes recortadas a la zona de estudio mediante el algoritmo elegido por el usuario	E
9.0	Mostrar el resultado obtenido en la identificación de nubes precipitables según las bandas elegidas por el usuario	E
10.0	Guardas los resultados obtenidos en formato TIF	E

Tabla 3.4: Requerimientos no funcionales

Nº	Función	Categoría
1.0	Se desarrollará en Python 3.7	E
2.0	Se presentará un documento del funcionamiento del sistema	O
3.0	Se utilizarán framework como tensorflow y sklearn	E

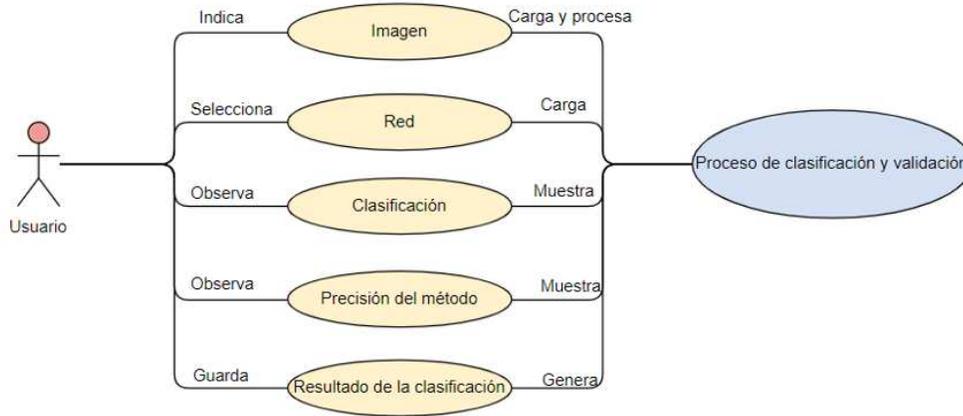
3.3.2. Diagrama conceptual

La figura 3-6 es un acercamiento a la herramienta, donde se puede observar el papel que ejerce cada modulo que conforma el sistema y cómo uno depende del otro.

Como se observa, la herramienta consta de varios módulos a saber:

1. Adquisición de la imagen (Imagen): esta diseñado para cargar la imagen desde el computador y que pueda ser guardada para continuar con el procesamiento.
2. Elección del método (Red): al cargar la imagen, se escoge el método de clasificación el cual es llamado en la interfaz para posteriormente ser observado.
3. Selección del Clasificador (Clasificación): se visualiza la clasificación obtenida por el

Figura 3-6: Diagrama conceptual.



método elegido.

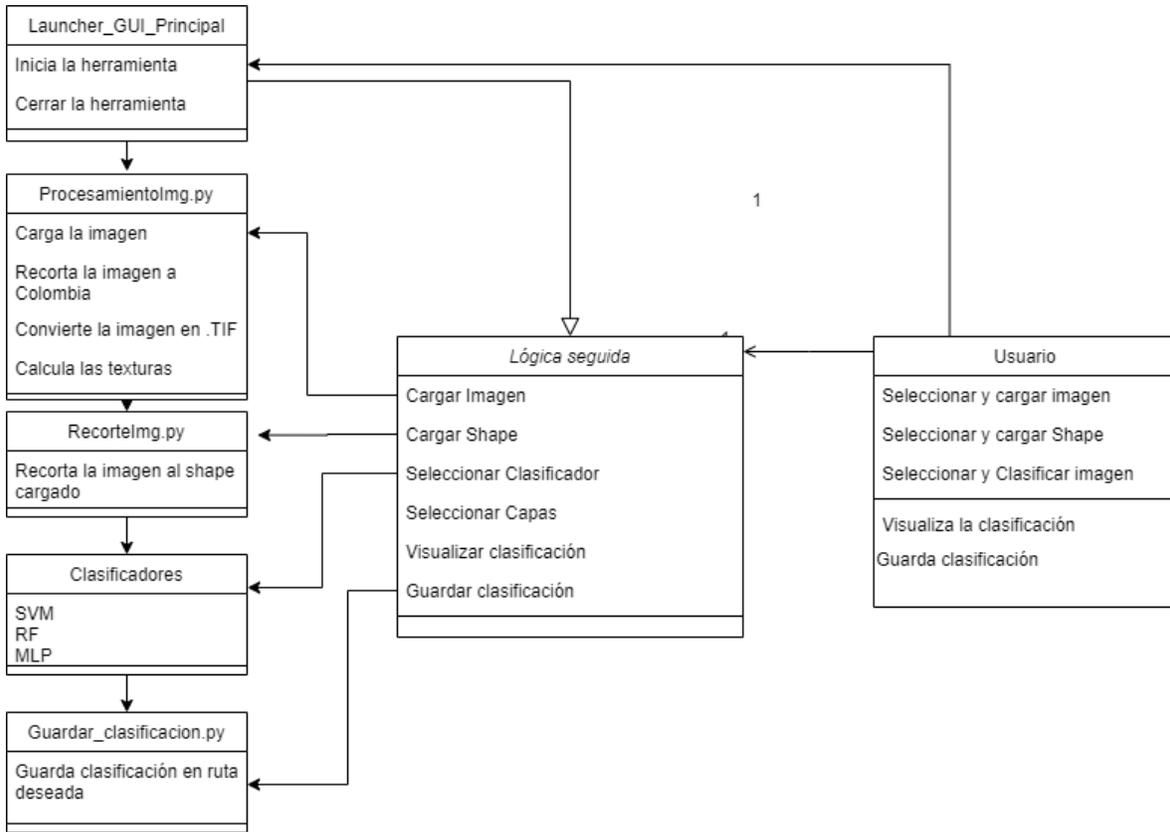
4. Precisión del Método: se observa la precisión del método
5. Resultado: en el último módulo se tiene el resultado de la clasificación para ser guardada en la ruta que elija el usuario.

3.3.3. Diagrama de clase

En el diagrama de clases observado en la figura 3-7, permite ver la forma en que se encuentra estructurada la herramienta. Se puede apreciar en el diagrama la interfaz gráfica *Launcher_GUI_Principal*, esta se encarga de que el usuario pueda interactuar con cada uno de los módulos, cargar imagen, cargar shape, seleccionar clasificador, seleccionar capas, visualizar y guardar la clasificación, esta se considera la interfaz gráfica principal.

Como se observa en la figura 3-7 el usuario interactúa con el *Launcher_GUI_Principal* ya que este es el encargado de iniciar la herramienta, posteriormente sigue la lógica de esta, donde cada uno de los procesos se empalman con los códigos observados en la figu-

Figura 3-7: Diagrama de clase

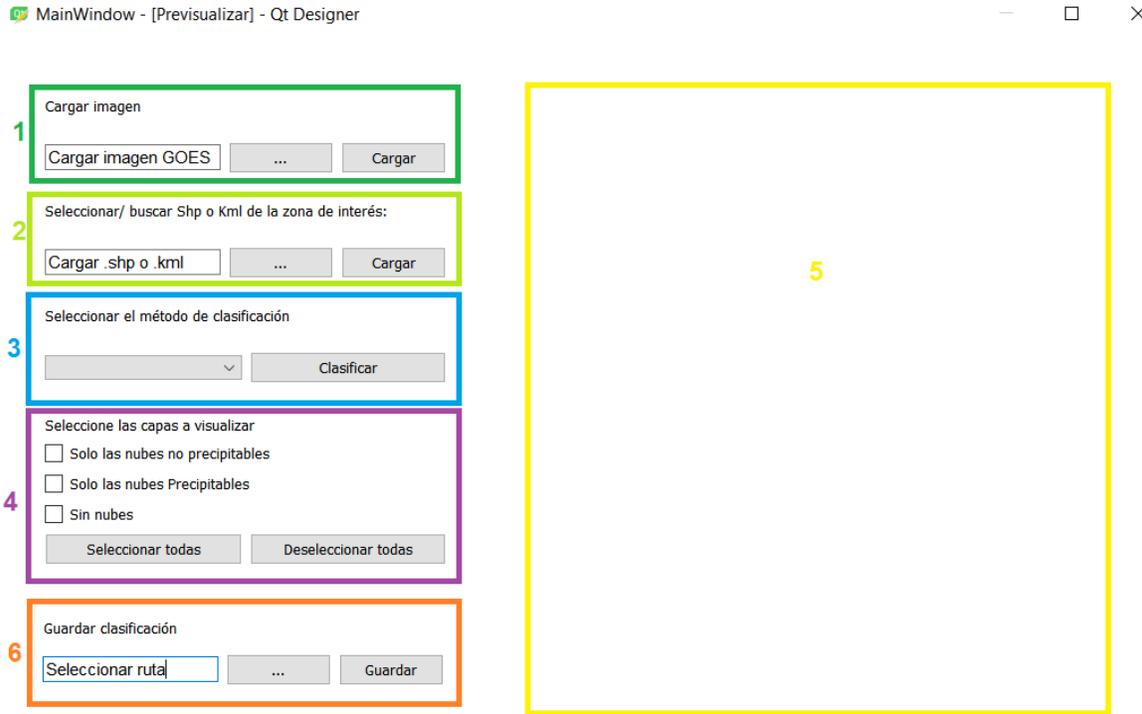


ra a la izquierda; los procesos que no tienen código externo, trabajan directamente en el *Launcher_GUI_Principal* como lo es la selección de capas a visualizar y la visualización.

3.3.4. Interfaz gráfica

El módulo de la interfaz gráfica es la ventana que se muestra inmediatamente después de ejecutar el aplicativo, esta interfaz cuenta con seis módulos, tal como se observa en la figura 3-8. Los módulos se dividen en :

Figura 3-8: Ventana principal de la interfaz.



1. Cargar la imagen: trae la imagen satelital.
2. Cargar *shape*: elemento para recortar la imagen y ser almacenada para uso posterior.
3. Seleccionar clasificador: eleecciona el método de clasificación con el que quiera procesar su imagen. En éste se usarán las imágenes de acuerdo a los pasos realizados anteriormente por el usuario.
4. Seleccionar las capas: permite al usuario seleccionar las capas para la visualización; si omite el paso no se mostrará la imagen.
5. Visualización: En el módulo de visualización en objetivo es presentar el producto del clasificador. Con ello, verificar la comprensión visual de las diferentes capas en varios tonos.

6. Guardar la clasificación: el usuario puede guardar la clasificación. Esta clasificación es guardada en formato .tif; de lo contrario, el archivo se borrará al terminar el proceso o cerrar la aplicación.

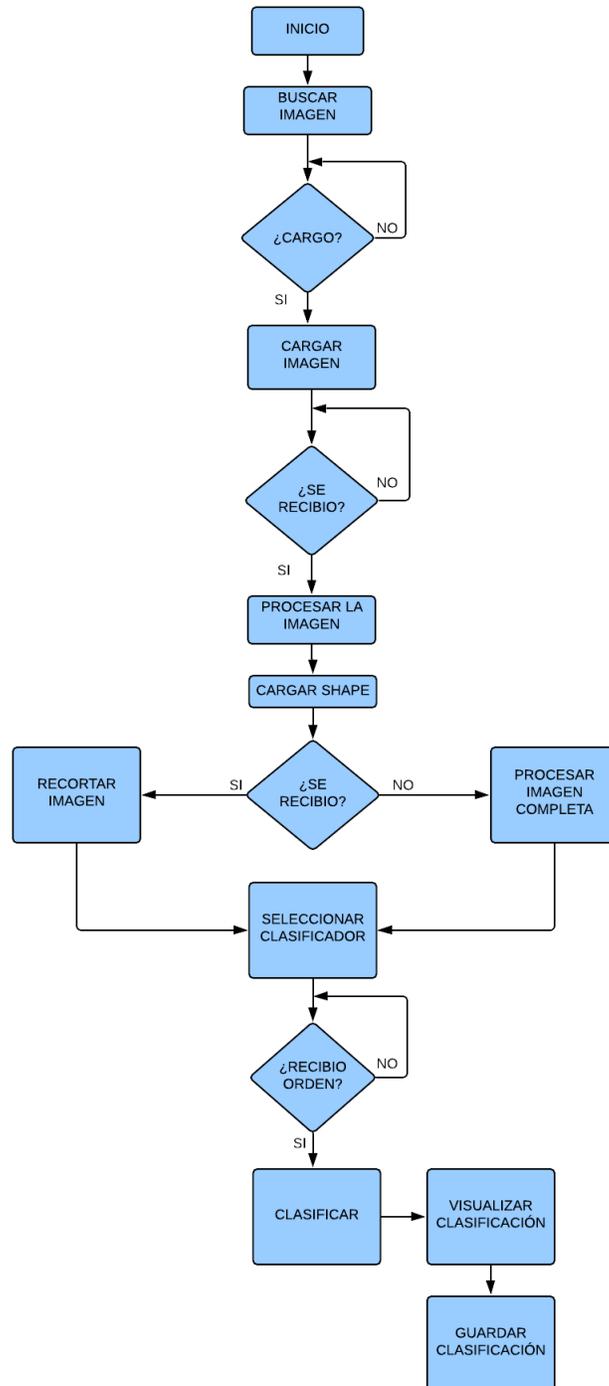
El procedimiento que realiza para hacer la clasificación se puede apreciar en la figura 3-9. Como primer paso al abrir la interfaz, se debe buscar la imagen **GOES-16**, la cual debe tener el nombre original para ser posteriormente cargada; al tenerla cargada esta internamente es procesada para obtener cada una de las capas y las texturas. El siguiente módulo permite al usuario, cargar un *shape* dentro del territorio colombiano; en este módulo la imagen es recortada a la zona especificada por el *shape*. Una vez se obtiene la imagen deseada, se procede a elegir los clasificadores, donde se despliega un menú y el usuario escoge entre ellos el que desee implementar en la clasificación. Posteriormente, se eligen las capas a visualizar, entre las que están: nubes no-precipitables, nubes precipitables y sin nubes. Finalmente, puede guardar la imagen clasificada.

3.4. Síntesis

En este capítulo se planteó el diseño y la implementación de la herramienta para la detección de nubes precipitables en Colombia, presentando cada uno de los pasos realizados para lograr una correcta detección. Se implementaron con software libre (Python, R-Studio, PyQt) cada uno de los algoritmos de Machine Learning así como los elementos de la interfaz de usuario.

Se realizaron pruebas exploratorias de sensibilidad de hiper-parámetros utilizando todas las características (contraste, correlación, energía, entropía, homogeneidad, media y varianza) obteniendo que los resultados con esta cantidad de características son deficientes, por lo tanto se realiza una reducción de características, seleccionando las características que

Figura 3-9: Procedimiento para realizar el proceso de clasificar la imagen.



más aportaban en el proceso de clasificación. Con estas características se realiza de nuevo sensibilidad de hiper-parámetros y un proceso de validación cruzada.

Al usar las estaciones meteorológicas, estas no son suficientes dado que la distribución espacial de ellas no es apropiada para tener una cobertura total de Colombia, adicionalmente, se eligieron las estaciones de acuerdo a la cantidad de precipitaciones reportadas que fueran mayores a *1mm*.

La interfaz gráfica se diseña para facilitar al usuario la comprensión del efecto de los algoritmos en el proceso de clasificación, donde, mediante diferentes capas, se observan las nubes no-precipitables y las nubes precipitables. El desarrollo metodológico será verificado en el siguiente capítulo, donde se verificará el efecto de entrenamiento de cada clasificador luego de un proceso de sintonización de hiper-parámetros. Adicionalmente, la interfaz se describe en cada uno de sus componentes con el ánimo de facilitar la comprensión de su funcionamiento.

Capítulo 4

Resultados y Discusión

En este capítulo se describen los resultados obtenidos al implementar la metodología propuesta en el capítulo 3. Como resultado general se tiene un sistema de detección de nubes precipitables en imágenes GOES-16 del sensor ABI.

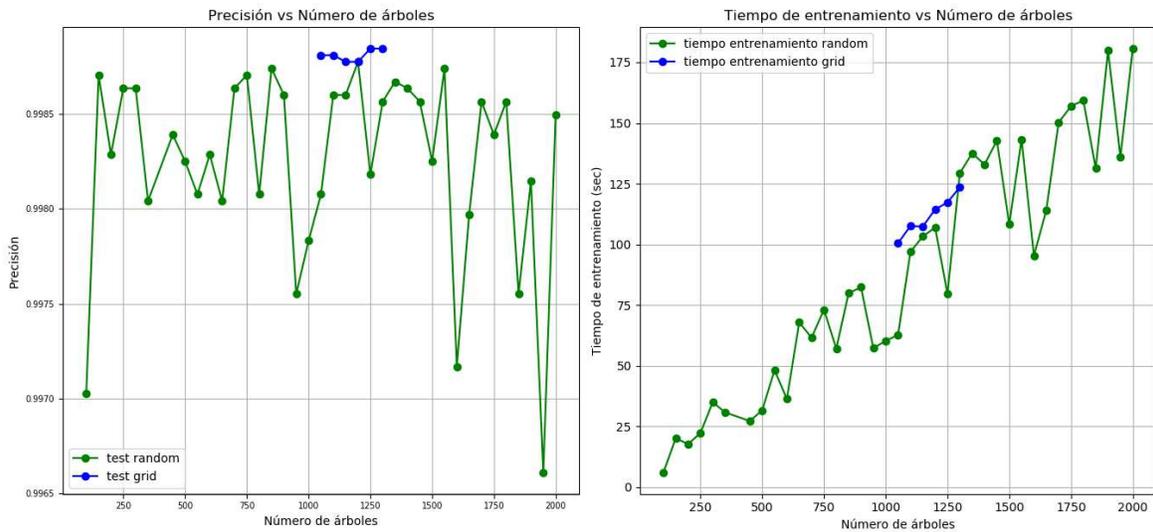
4.1. Entrenamiento de los modelos de aprendizaje

4.1.1. Entrenamiento Random Forest

En la figura 4-1 se presenta los resultados obtenidos en el proceso de entrenamiento para el método RF. En éste se realizó la optimización de los hiper-parámetros por el método de búsqueda aleatoria y búsqueda por cuadrícula. Como se describe en la Sección 3.2.5, primero se realiza la optimización aleatoria con el fin de explorar un conjunto mayor de hiper-parámetros, que después se enfoca en un conjunto de hiper-parámetros en una región más pequeña, la cual se optimiza por una búsqueda en cuadrícula.

4.1 Entrenamiento de los modelos de aprendizaje

Figura 4-1: N°. Árboles vs Precisión.



Del total de combinaciones posibles se seleccionaron aleatoriamente 100 de ellas. Éstas se presentan en la figura 4-1 en color verde. La figura presenta uno de los hiper-parámetros más importantes en RF, el cual corresponde al número de árboles de decisión respecto a la precisión en el *test* de validación. Tal como se observa, se obtienen excelentes desempeños en el espacio de búsqueda, obteniendo el mejor desempeño alrededor de los 1200 árboles con 99.87% de precisión; además, se puede observar en la figura a la derecha cómo aumenta el tiempo de entrenamiento a medida que aumenta el número de arboles. Los hiper-parámetros obtenidos de esta proceso de optimización aleatoria fueron:

n_estimators	1200
min samples split	2
min samples leaf	1
max features	auto
max depth	90
bootstrap	False

Con el espacio de hiper-parámetros identificado, se procede a hacer una búsqueda sobre

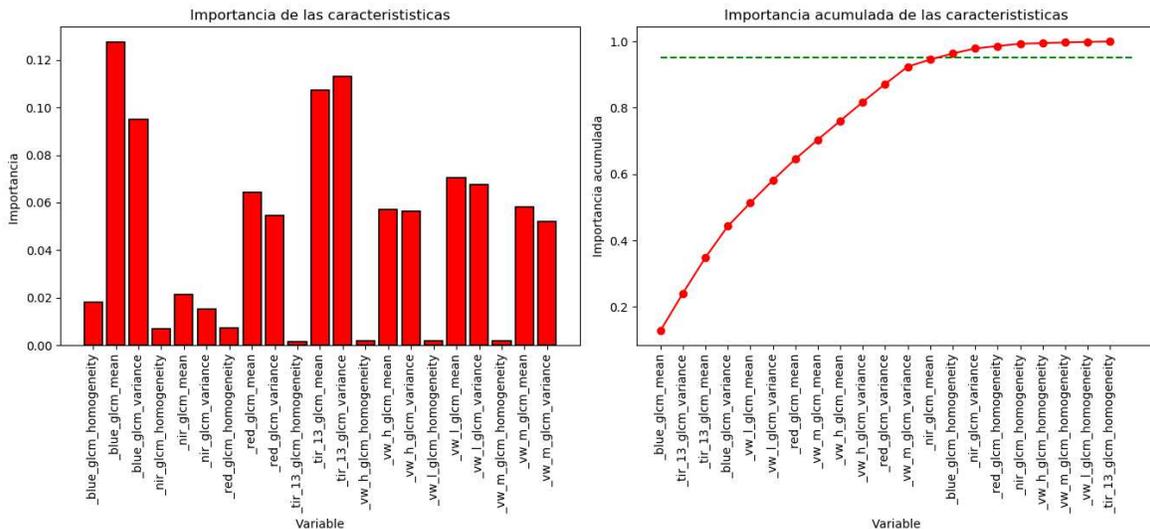
4.1 Entrenamiento de los modelos de aprendizaje

una pequeña región. Esta región se central alrededor del mejor conjunto de hiperparámetros obtenidos anteriormente y se procede a realizar una optimización por cuadrícula. La figura 4-1 presenta los resultados en azul, obteniendo un buen desempeño en la configuración de 1.300 árboles con una precisión de 99.88%, esto representa una mejora de 0.1% en nuestro modelo final de RF. El mejor conjunto de hiperparámetros resultantes quedaría así:

bootstrap	False
max_depth	95
max_features	auto
min_samples_leaf	1
min_samples_split	2
n_estimators	1.300

Del mejor modelo RF, la figura 4-2 muestra la importancia de las características calculadas a partir de la disminución de la impureza del nodo ponderada por la probabilidad de llegar a dicho nodo. La probabilidad del nodo se puede calcular por el número de muestras que llegan a este, dividido por el número total de muestras (Ronaghan, 2018).

Figura 4-2: Importancia de las características.



Como se observa las bandas espectrales que más aportan a este problema en este caso ha sido la banda azul y el infrarrojo térmico, junto con las texturas de varianza y media. Por

4.1 Entrenamiento de los modelos de aprendizaje

otro lado, se identifica que las texturas de homogeneidad son las que menos aportan. La línea punteada representa el 95 % de importancia total acumulada ordenada de mayor a menor.

4.1.2. Entrenamiento SVM

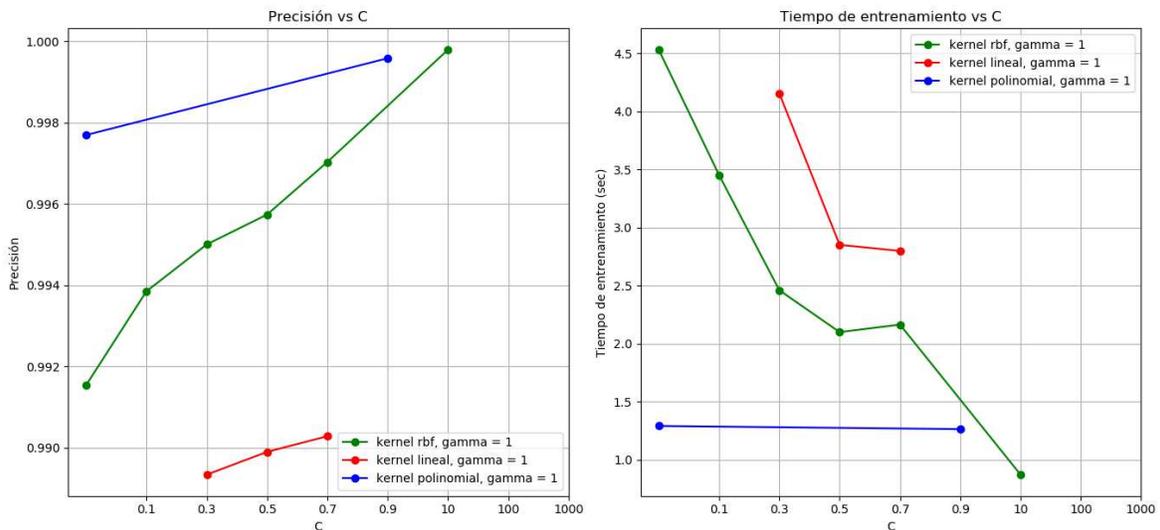
Para el entrenamiento de **SVM** se realiza el mismo procedimiento de **RF**; es decir, primero una optimización aleatoria para definir un espacio de hiperparámetros alrededor del mejor resultado anterior, junto con una optimización por cuadrícula buscando obtener el clasificador **SVM** con mejor desempeño.

En la figura 4-3 se presenta los resultados de las diferentes combinaciones aleatorias obtenidas, destacando el parámetro de regularización (C) y el *kernel* empleado respecto a la precisión en el *test* de validación. Dentro de los resultados se omite la presentación del *kernel* sigmooidal debido a que su desempeño es muy bajo frente a los demás. Se evidencia que existe un mejor desempeño en el *kernel* **rbf** y **polinomial** de grado 6. También se hace evidente que a medida que C aumenta, el desempeño de los modelos tiende a mejorar; adicionalmente, al aumentar este parámetro el tiempo de entrenamiento disminuye. El mejor conjunto de hiperparámetros en esta optimización aleatoria se presenta a continuación con una precisión de 99.97 % en el *test* de validación:

Kernel	rbf
gamma	1
'C'	100

4.1 Entrenamiento de los modelos de aprendizaje

Figura 4-3: Parámetro de regularización vs precisión para diferentes kernels.



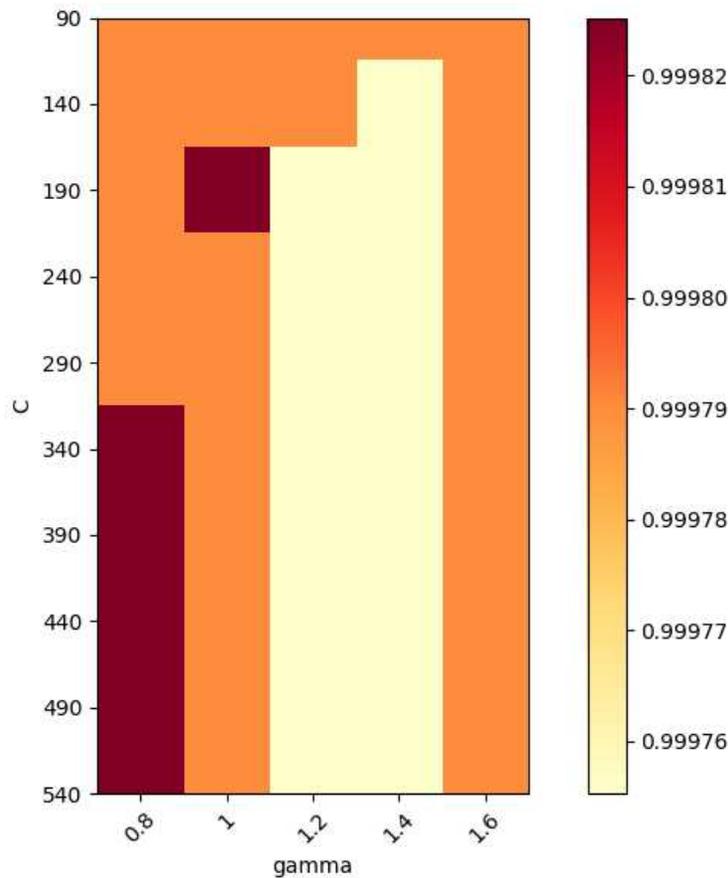
De los resultados de la optimización aleatoria, se obtiene que el mejor rendimiento se logra con el *kernel rbf*, por lo que en la optimización por cuadrícula se trata de ajustar los hiperparámetros C y $gamma$. El resultado de esta optimización se puede apreciar en la Figura 4-4. El mejor resultado con el modelo de aprendizaje SVM se obtiene con una configuración de ' C ': 340, ' $gamma$ ': 0.8, ' $kernel$ ': ' rbf ' el cual muestra una precisión del 99.98%, es decir, una mejora del 0.01% respecto al mejor modelo de la optimización aleatoria.

4.1.3. Entrenamiento MLP

Del espacio de hiperparámetros para MLP se obtuvieron 15.500 posibles combinaciones de hiperparámetros, de ellos se seleccionó aleatoriamente el 2% de combinaciones para un total de 310 modelos de entrenamiento. En la Figura 4-5 se presenta un *boxplot* en el que están agrupados los resultados inicialmente por el número de iteraciones y por el optimizador; de aquí, se puede observar que a medida que aumenta el número de épocas o iteraciones,

4.1 Entrenamiento de los modelos de aprendizaje

Figura 4-4: Precisión de validación determinada a partir del parámetro de regularización (C) y gamma.



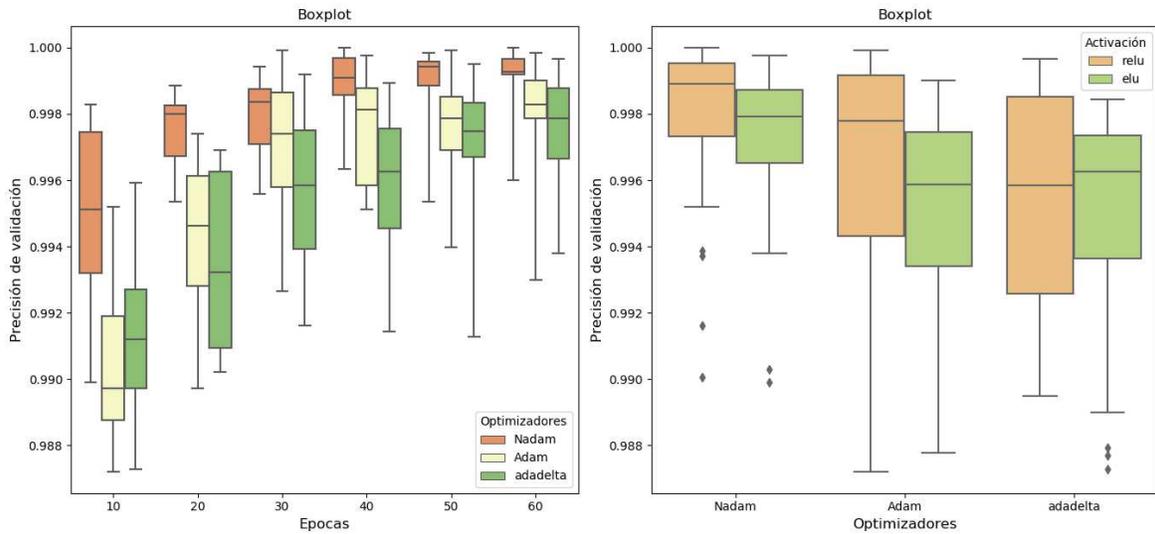
aumenta la precisión de los modelos en los *test* de validación. Además, el optimizador *Nadam* presenta mejores resultados a los demás optimizadores. El optimizador *Nadam* mantiene una menor dispersión entre las épocas 20 – 60 respecto a los demás optimizadores, siendo así el mejor optimizador para este problema.

En el *boxplot* de la derecha, se presentan los optimizadores junto con las funciones de activación, demostrando el activador *relu* tiene un mejor desempeño sobre *elu* y se reafirma un mejor desempeño en el optimizador *Nadam*. Aunque los resultados muestran mayor nivel de correspondencia para *Nadam*, los otros algoritmos también presentaron buenos resultados.

En la Figura 4-6 se presenta un gráfico de barras del número de neuronas en la capa

4.2 Comparación de los clasificadores

Figura 4-5: Boxplot optimizadores, función de activación, épocas vs Precisión



oculta, *dropout* y función de costo respecto a la precisión de validación. Aunque existen pequeñas diferencias al verificar la sensibilidad de los modelos, el mejor modelo **MLP** obtiene una precisión de 100% en el *test* de validación con los siguientes hiperparámetros:

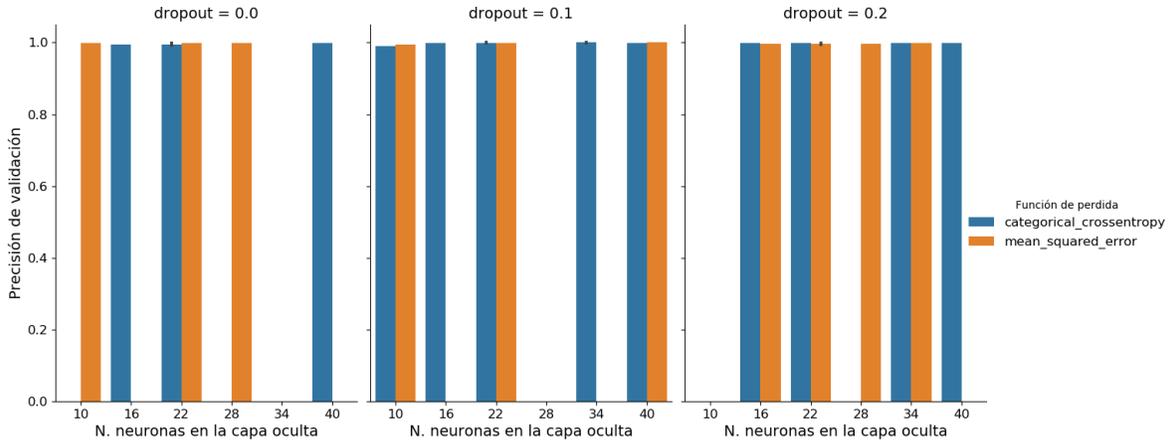
Función de activación	Relu
Batch size	128
Dropout	0.1
Épocas	60
Neuronas en la capa oculta	40
Kernel de iniciación	Uniform
Last activación	Softmax
Función de pérdida	Error medio cuadrático
Optimizador	Nadam

4.2. Comparación de los clasificadores

Los resultados anteriores indican que los clasificadores tienen la capacidad para separar las diferentes clases de nubes; sin embargo, se verifica su desempeño usando una imagen

4.2 Comparación de los clasificadores

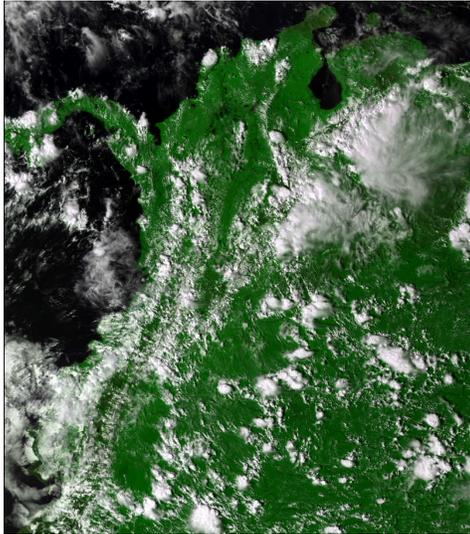
Figura 4-6: Gráfico de barras número de neuronas capa oculta, dropout, función de pérdida vs precisión de validación.



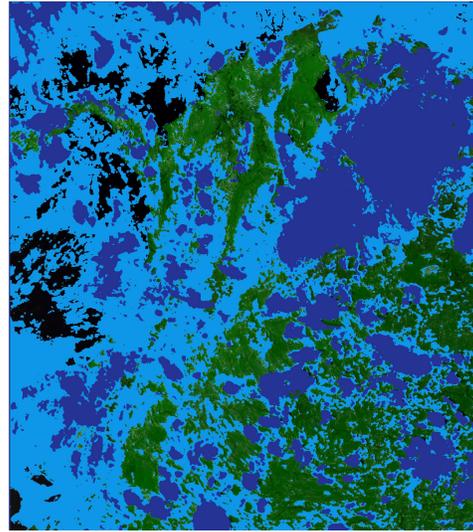
con nubes para comparar (Ver Fig. 4-7).

En la Fig. 4-7 con fecha del 2019 – 10 – 25 a las 19 : 00 : 00UTC se observa el resultado de la clasificación por los tres métodos empleados. La imagen (a) presenta la imagen en color verdadero (RGB) del territorio colombiano; la imagen (b) presenta la clasificación realizada por el método de RNA con una **MLP**, donde se muestra la clasificación de las nubes (precipitables/no-precipitables) y la ausencia de nubes. Las nubes no-precipitables se presentan en color azul claro, las nubes precipitables en azul oscuro y la ausencia no presenta alguna máscara sobre la imagen original. Para la **MLP** se observa que el clasificador identifica parte del territorio como nubes, donde la mayoría corresponden a nubes no-precipitables. El método destaca la presencia de nubes que son menos densas.

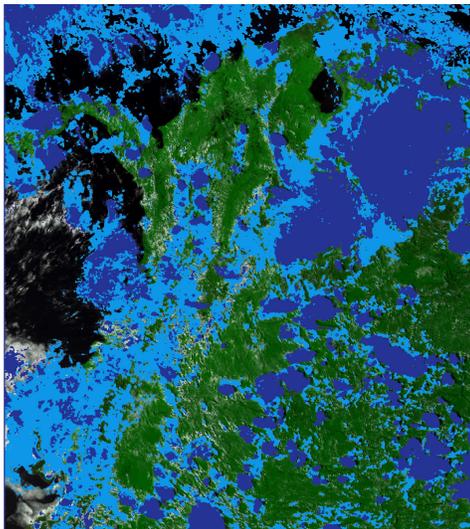
La imagen (c) presenta un resultado con el método **SVM**, en el se identifican las nubes más densas en comparación con el resultado obtenido con **MLP**. Además, se observa que las nubes precipitables son en proporción similares entre las escenas (b) y (c). Para el método de **RF** se muestra que el método clasifica solamente las nubes más densas que se pueden percibir de la imagen original (a); en comparación con los métodos anteriores, **RF** es el que



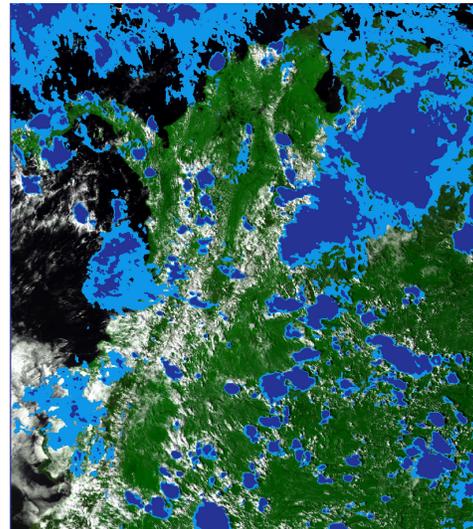
(a) Imagen a color verdadero



(b) Mascara MLP.



(c) Mascara SVM.



(d) Mascara RF.

Figura 4-7: Comparación máscaras de detección de nubes precipitables y no precipitables.

tiene una menor clasificación de nubes que sean poco densas, pero, identifica las mismas nubes precipitables de los dos métodos anteriores, variando sólo en algunos píxeles que no son muy perceptibles a la vista.

En cuanto a los cuerpos de agua observados en cada una de las imágenes, se puede observar que en la clasificación por el métodos de **MLP** presenta falencias en esta identificación, ya que no los diferencian de las nubes. En cuanto a los métodos de **RF** y **SVM** diferencian más estos cuerpos con pequeños errores en píxeles. Se considera que el método más preciso es **SVM** debido a que identifica las nubes tanto precipitables, como las que no lo son e identifica de mejor forma los cuerpos de agua.

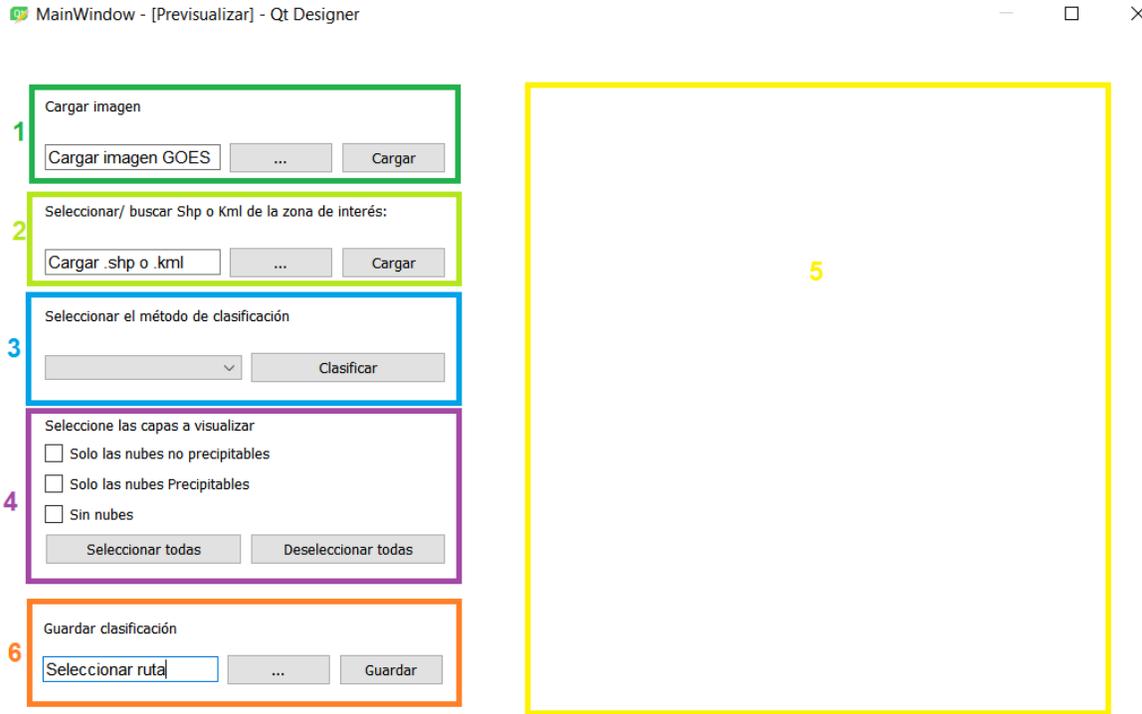
4.3. Desarrollo de la interfaz

4.3.1. Ventana principal

Al ejecutar el aplicativo, se abre la ventana principal la cual tiene las siguientes funcionalidades (explicadas por módulos como se observa en la figura 4-8):

1. En el primer modulo (verde oscuro) se carga la imagen desde el ordenador, donde se ejecuta la aplicación. La imagen debe ser cargada con el nombre original en formato *.nc*. En cuyo caso, donde la imagen no posea su nombre original, saldrá una alerta y no permitirá procesarla. Este paso solo almacena la ruta o el *path* de la imagen a utilizar. El proceso que sigue al cargarse el *path* es el mencionado en la sección 3.2.1, donde el proceso que realiza intermamente el cual consiste en: seleccionar las 7 bandas a utilizar, las cuales se convierten en formato *.TIF*; se recorta a Colombia, para posteriormente, crea una carpeta temporal que almacena estas imágenes para ser procesadas en el software **R Studio**; y se calculan las texturas para determinar el

Figura 4-8: Modulos de la interfaz.



vector de características.

2. En el módulo 2 (verde claro) se determina la zona de interés. En esta parte se busca en el ordenador el archivo en formato *shape* o *kml*, para que la imagen sea recortada en la zona. Esto acelera el procesamiento llevado a cabo por el clasificador.
3. Posteriormente en el módulo 3 (azul) se encuentran los métodos de clasificación implementados para detectar las nubes precipitables de sus opuestas. En ella encontramos la elección del método que se quiere utilizar, para ello, se cargan los archivos de la carpeta interna "Clasificadores" y dependiendo de los archivos en esta carpeta se despliegan los métodos para la clasificación. Para efectos de este trabajo se encuentran disponibles los métodos **RF**, **SVM** y **RNA**.
4. En el módulo 4 (morado) se procede a seleccionar las capas de la clasificación que se quieren visualizar. Aquí, se puede elegir mostrar las nubes precipitables, las que no lo

son y la ausencia de nubes, o todas, si así lo requiere el usuario. El módulo permite elegir las capas que se quieren visualizar.

5. En el siguiente módulo (amarillo) se visualiza la imagen agregada en sus bandas RGB para ser observadas en color verdadero, encima de esta imagen, donde existen nubes no-precipitables y nubes precipitables, el color de ellas se vera alterado mostrando un falso color: para nubes precipitables (azul oscuro), para nubes no-precipitables (azul claro) y para ausencia de nubes (sin color), donde solo se verá alterada la capa que el usuario ha elegido.
6. En el último módulo (naranja) se da la elección de guardar la clasificación. En este, se puede guardar las nubes precipitables o las nubes no-precipitables en general. El módulo permite elegir las capas de la clasificación que serán almacenadas. Se encuentra un botón de guardado, el cual guarda en formato *.tif* la imagen clasificada.

4.3.2. Pruebas de integración

Las pruebas de integración tienen el objetivo de evaluar por conjunto las funcionalidades del sistema. Estas pruebas se encuentran separadas según lo expuesto en los requerimientos funcionales. Para la exposición de cada prueba se utiliza una tabla estructurada de acuerdo a lo exigido por la metodología RUP, de la siguiente manera:

- Requerimientos: nombres de los requerimientos funcionales evaluados en la prueba.
- Tipo de prueba: tipo de prueba implementada.
- Hardware requerido: hardware implementado en la prueba.
- Software requerido: software implementado en la prueba.

- Objetivo del requerimiento: objetivo de los requerimientos implementados.
- Objetivo de la prueba: objetivo de la prueba en el sistema.
- Datos necesarios para la prueba: datos necesarios para la prueba del sistema.
- Procedimiento de la prueba: procedimiento para la realización de la prueba.
- Resultado esperado: resultado esperado de la prueba.
- Resultado obtenido: resultado obtenido de la prueba.
- Comentarios: aspectos a tener en cuenta.

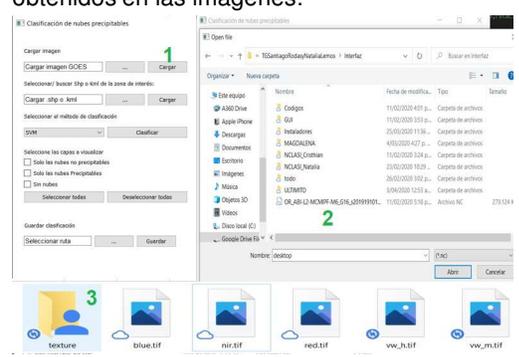
Finalmente, después de presentarlos se pasará a discutir los resultados obtenidos.

4.3.2.1. Búsqueda y carga de imagen

La prueba de integración correspondiente a la carga de la imagen, está destinada a comprobar que el usuario pueda buscar y cargar la imagen en el formato *.nc*. Esta prueba y sus resultados se evidencian en la tabla 4.1

En la tabla 4.1 se pueden visualizar las capturas de pantalla respectivas al procedimiento realizado para la búsqueda y carga de imágenes y los resultados obtenidos. En la captura (1) se evidencia que el usuario pueda iniciar la aplicación y posteriormente, en la captura (2) pueda buscar la imagen en el formato *nc*. Al cargar la imagen, la herramienta crea una carpeta temporal donde se irán almacenando todos los procesos de la interfaz. La carpeta temporal es un contenedor con las bandas utilizadas y una carpeta con el nombre **texturas**, tal como se observa en la captura (3). Este proceso evidencia el cumplimiento del requerimiento planteado en la prueba.

Tabla 4.1: Prueba de integración: Búsqueda y carga de la imagen

PRUEBA DE INTEGRACIÓN - CARGA DE IMAGEN	
Requerimientos: 1. Cargar la imagen GOES-16 en formato .nc 2. Cargar el algoritmo de pre-procesamiento de imágenes	
Tipo de prueba	Integración
Hardware requerido	Computador portátil
Software requerido	Python 3.7, Rstudio, PyQt
Objetivo del requerimiento	Probar que la carga de la imagen sea exitoso
Objetivo de la prueba	Verificar que el usuario pueda agregar las imágenes del sensor GOES-16 en formato .nc
Datos de la prueba	Imagen GOES-16 en formato .nc
Procedimiento	1. Inicio de la herramienta 2. Buscar la imagen GOES-16 3. Cargar la imagen
Resultado Esperado	1. Buscar la imagen GOES en formato .nc con éxito 2. Realizar el procesamiento interno con éxito 3. Cargar la imagen con éxito
Resultado obtenido	Prueba satisfactoria. Se presenta en forma detallada los resultados obtenidos en las imágenes: 
Comentarios	El usuario al agregar la imagen debe verificar que esta tenga el nombre en el formato de descarga.

4.3.2.2. Búsqueda y carga de shape

La prueba de integración correspondiente a la carga del *shape*, está destinada a comprobar que el usuario pueda buscar y cargar el *shape* en el formato *.shp*. Esta prueba y sus resultados se evidencian en la tabla 4.2.

En la tabla 4.2 se pueden visualizar las capturas de pantalla respectivas al procedimiento realizado para la búsqueda y carga del *shape* y los resultados obtenidos. En la captura (1) el usuario busca que el *shape* se abra en una ventana emergente dentro del equipo en un archivo de extensión *.shp*. Posteriormente, en la captura (2) se carga la imagen lo cual hará que se cree una carpeta nueva llamada recorte, donde se almacenan las imágenes recortadas en bandas y texturas, tal como se muestra en la captura (3).

4.3.2.3. Carga y selección de clasificadores

La prueba de integración correspondiente a la carga y selección de clasificadores, está orientada a comprobar que el usuario pueda seleccionar y clasificar la imagen. Esta prueba y sus resultados se evidencian en la tabla 4.3.

En la tabla 4.3 se muestran las capturas de pantalla respectivas al procedimiento realizado para la carga y selección de los clasificadores y los resultados obtenidos en ella. En la captura (1) se observa la carpeta “clasificadores” donde esta cada uno de los métodos de clasificación empleados en este trabajo, los clasificadores en esta carpeta serán los que se visualicen en el despegable de la captura (2) donde se observa que el usuario pueda seleccionar el clasificador entre los que se encuentran **RF**, **SVM**, y **RNA**. Posteriormente en la captura (3) se observa una carpeta donde la clasificación es almacenada para ser mostrada posteriormente

Tabla 4.2: Prueba de integración: Búsqueda y carga del shape

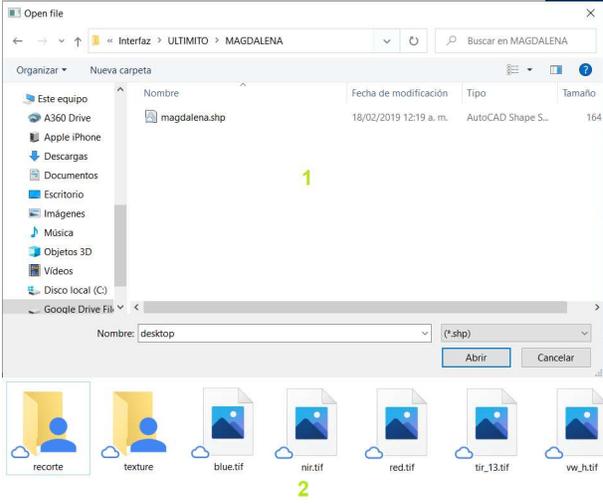
PRUEBA DE INTEGRACIÓN - CARGA DE IMAGEN	
Requerimientos: 1. Cargar el shape del área que quiere observar en Colombia 2. Cargar el algoritmo que recorta las imágenes de acuerdo al área cargada por el usuario	
Tipo de prueba	Integración
Hardware requerido	Computador portátil
Software requerido	Python 3.7, R Studio, PyQt
Objetivo del requerimiento	Cargar la zona de estudio deseada por el usuario dentro de Colombia
Objetivo de la prueba	Verificar que el usuario pueda cargar el shape y este sea usado para recortar el área de estudio
Datos de la prueba	Archivo en formato <i>.shp</i>
Procedimiento	1. Dirigirse al módulo “Cargar Shape” 2. Buscar el archivo en formato <i>.shp</i> 3. Cargar el archivo
Resultado Esperado	1. Al buscar el shape se filtre por los formato <i>.shp</i> 2. Cargar el shape 3. Que el archivo recorte la imagen
Resultado obtenido	Se presenta en forma detallada los resultados obtenidos en las imágenes: 
Comentarios	Si el usuario no desea agregar un shape, puede pasar al siguiente paso sin ningún problema.

Tabla 4.3: Prueba de integración: Carga y selección de los clasificadores

PRUEBA DE INTEGRACIÓN - CARGA Y SELECCIÓN DEL MÉTODO DE CLASIFICACIÓN	
Requerimientos: 1. Cargar los algoritmos 2. Procesar los algoritmos de acuerdo a las condiciones dadas por el usuario.	
Tipo de prueba	Integración
Hardware requerido	Computador portátil
Software requerido	Python 3.7, Rstudio, PyQt
Objetivo del requerimiento	Cargar los algoritmos de machine learning
Objetivo de la prueba	Verificar que los algoritmos carguen y procesen correctamente
Datos de la prueba	Algoritmos de machine learning RF, SVM y RNA
Procedimiento	1. Cargar los algoritmos que se encuentran almacenados en la carpeta “clasificadores” 2. Seleccionar el algoritmo que se requiera para hacer la clasificación 3. Cargar la clasificación
Resultado Esperado	1. Que el usuario pueda seleccionar los métodos de clasificación 2. Que al cargar el método elegido sea guardado en una carpeta temporal
Resultado obtenido	Prueba satisfactoria: Se presenta en forma detallada los resultados obtenidos en las imagenes:
Comentarios	El proceso se realiza a la imagen completa o recortada por el usuario, dependiendo de ello, es el tiempo de carga.

4.3.2.4. Selección de capas

La tabla 4.4 correspondiente a la selección de capas a visualizar, está destinada a comprobar que el usuario pueda seleccionar las capas que se verán en la visualización. En este caso se puede seleccionar las nubes no-precipitables, nubes precipitables o no nube (ausencia de nubes). Es posible seleccionar todas las capas, así como no seleccionadas.

Como se observa en la tabla 4.4, en la captura (1) se muestra la selección de las capas para realizar la visualización de ellas y en la captura (2) se observa la visualización de acuerdo a las capas seleccionadas.

4.3.2.5. Visualización

La prueba de integración correspondiente a la visualización de la clasificación está destinada a comprobar que el usuario pueda observar la clasificación obtenida por el método elegido, adicionalmente, que observe las capas elegidas en el paso anterior en este caso eligiendo entre nube, nube precipitable o no nube, como se muestra en la tabla 4.5

En la tabla 4.5 se evidencia el procedimiento realizado para la visualización de las nubes en sus diferentes clases. En la captura (1) se ve la clasificación obtenida dentro de la carpeta temporal, de esta forma en la captura (2) se muestra como se visualiza dependiendo del clasificador seleccionado.

4.3.2.6. Guardar clasificación

En la tabla 4.6 corresponde a la prueba de integración que permite guardar el archivo generado en la clasificación de acuerdo a las capas elegidas por el usuario, las pruebas y resultados se evidencian en ella.

Tabla 4.4: Prueba de integración: Selección de capas a visualizar

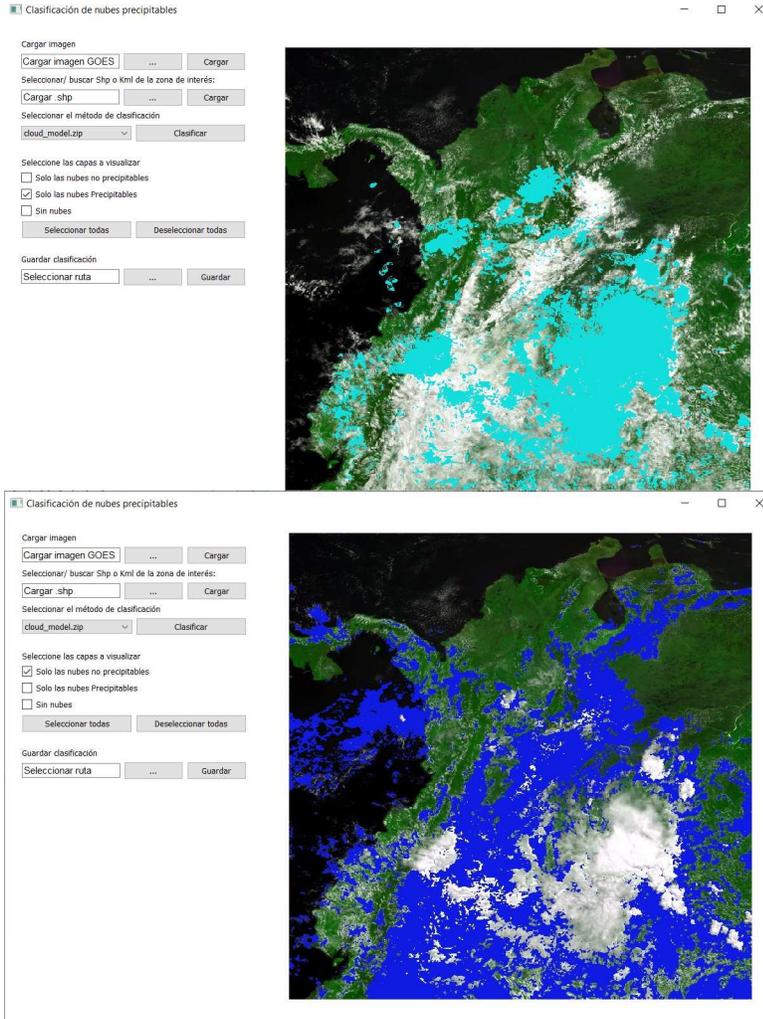
PRUEBA DE INTEGRACIÓN - SELECCIÓN DE CAPAS	
Requerimientos: 1. Seleccionar las capas que el usuario quiera observar	
Tipo de prueba	Integración
Hardware requerido	Computador portátil
Software requerido	Python 3.7, Rstudio, PyQt
Objetivo del requerimiento	Seleccionar las capas que se quieren visualizar
Objetivo de la prueba	Que el usuario pueda elegir que capas visualizar
Datos de la prueba	Imágenes clasificadas
Procedimiento	1. Seleccionar las capas que desee visualizar 2. Que pueda seleccionar todas o una sola
Resultado Esperado	1. Que al usuario seleccionar las o la capa se visualice la selección
Resultado obtenido	<p>Prueba satisfactoria: Se presenta en forma detallada los resultados obtenidos en las imágenes:</p> 
Comentarios	La visualización será de acuerdo a lo que el usuario elija, si es nube precipitable, nube o no nube.

Tabla 4.5: Prueba de integración: Visualización de la clasificación

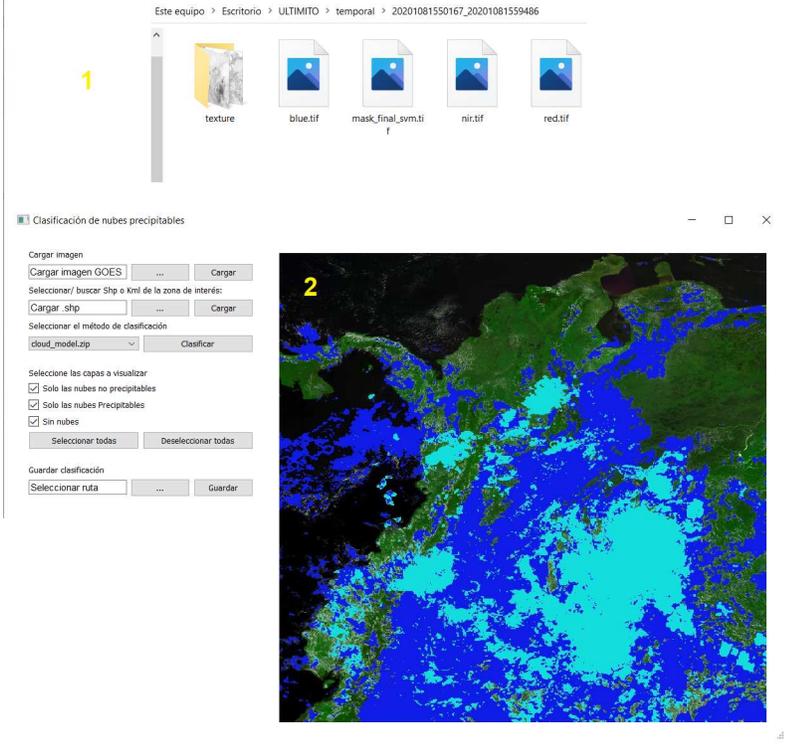
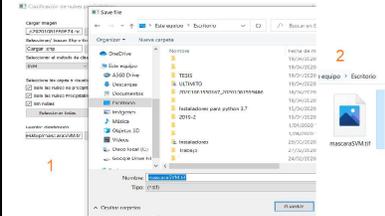
PRUEBA DE INTEGRACIÓN - VISUALIZACIÓN	
Requerimientos: 1. Visualizar la clasificación realizada por los diferentes métodos	
Tipo de prueba	Integración
Hardware requerido	Computador portátil
Software requerido	Python 3.7, Rstudio, PyQt
Objetivo del requerimiento	Mostrar el resultado obtenido en la detección de nubes precipitables de acuerdo al método elegido.
Objetivo de la prueba	Que el usuario en el campo de visualización pueda observar la clasificación por el método elegido
Datos de la prueba	Imágenes clasificadas
Procedimiento	1. Realizar la clasificación de nubes 2. Seleccionar las capas a visualizar
Resultado Esperado	1. Se espera que el usuario pueda observar la clasificación de nubes de acuerdo al método de clasificación y las capas seleccionadas
Resultado obtenido	<p>Prueba satisfactoria Se presenta en forma detallada los resultados obtenidos en las imágenes:</p>  <p>The screenshot shows a file explorer window with a path: 'Este equipo > Escritorio > ULTIMITO > temporal > 20201081550167_20201081559486'. It contains files: 'texture', 'blue.tif', 'mask_final_svm.tif', 'nir.tif', and 'red.tif'. Below the files is a software window titled 'Clasificación de nubes precipitables'. The interface includes sections for 'Cargar imagen' (with 'GOES' and 'shp' buttons), 'Seleccionar el método de clasificación' (with a dropdown set to 'cloud_model.zip' and a 'Clasificar' button), 'Selección de capas a visualizar' (with checkboxes for 'Solo las nubes no precipitables', 'Solo las nubes Precipitables', and 'Sin nubes', and 'Seleccionar todas'/'Deseleccionar todas' buttons), and 'Guardar clasificación' (with a 'Seleccionar ruta' field and a 'Guardar' button). To the right of the controls is a large image showing a satellite view of a region with blue and cyan overlays representing cloud classification, marked with a yellow '2'.</p>
Comentarios	La visualización es posible después de utilizar alguno de los métodos de clasificación

Tabla 4.6: Prueba de integración: Guardar la clasificación

PRUEBA DE INTEGRACIÓN - GUARDAR CLASIFICACIÓN	
Requerimientos: 1. Permitir exportar la clasificación en la ruta elegida por el usuario	
Tipo de prueba	Integración
Hardware requerido	Computador portátil
Software requerido	Python 3.7, Rstudio, PyQt
Objetivo del requerimiento	Guardar los resultados en formato .tif
Objetivo de la prueba	Comprobar que la herramienta guarde la clasificación en .tif
Datos de la prueba	Imágenes clasificadas
Procedimiento	1. Oprimir el botón guardar "Guardar" 2. Elegir la carpeta y el nombre del archivo.
Resultado Esperado	1. Despliegue ventada para buscar la carpeta y el archivo generado 2. Desbloqueo de pantalla que indique que ocurrió con éxito
Resultado obtenido	Prube satisfactoria: Se presenta en forma detallada los resultados obtenidos en las imagenes: 
Comentarios	El usuario puede elegir el nombre y la ubicación para el archivo generado

En la tabla 4.6 mostrada anteriormente, se puede observar en la captura (1) el usuario decide la ruta donde desee guardar el archivo, en esta se abre una ventana emergente para que el localice donde sera guardado el archivo en formato .tif, en la caputra (2) se observa en la carpeta que se elija como queda guardado el archivo. Este proceso evidencia el funcionamiento de los requerimientos planteados en la prueba.

4.4. Síntesis

En este capítulo se mostraron los resultados de cada uno de los procesos realizados. Estas pruebas y resultados fueron mostrados a través de tablas, imágenes y gráficas, donde se observó el funcionamiento de cada uno de los clasificadores empleados y las pruebas de integración de la interfaz que hacen parte de la metodología RUP, donde se visualizó el cumplimiento de cada uno de los requerimientos funcionales de este trabajo, mostrado mediante capturas de pantalla cada uno de los módulos de la herramienta.

En el proceso de este trabajo, se encontraron diferentes ventajas y desventajas, entre las desventajas, se encontró que es complicado conseguir información de forma rápida de las estaciones con datos de precipitación que contengan una resolución temporal de *10 minutos*, dado que las imágenes del satélite *GOES – 16* nos proporciona esa temporalidad, además, aunque el trabajo se realizó de forma local, se presenta un volumen de datos grande, lo que hace que el procesamiento sea largo, debido a que se deben realizar varios procesos como lo son, la descargar las imágenes, el filtrado los datos y el cálculo de los diferentes procesos expuestos en el capítulo de metodología. Sin embargo, como se trabaja por medido de códigos y los procesos son repetitivos, estos se automatizan para hacer más fácil el procesamiento de cada uno de los pasos.

En cuanto a la interfaz, se tuvo en cuenta cada uno de los requerimientos funcionales y la lógica de la interfaz para su correcto funcionamiento, estas pruebas se presentaron y cumplieron satisfactoriamente cada uno de los procesos. Las pruebas de integración realizadas confirman el cumplimiento de cada uno de los módulos como lo es: la carga de la imagen, la carga del shape, la clasificación, la selección de capas las a visualizar, la visualización del proceso de clasificación y el poder guardar la imagen clasificada de acuerdo con las capas visualizadas.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

Para la caracterización de las nubes de manera remota es necesario identificar las regiones del espectro en donde sus propiedades se destacan, así como estudiar su comportamiento espacial. Respecto a la revisión bibliográfica, se indica que la banda azul ($0.47\mu m$), roja ($0.64\mu m$) e infrarrojo cercano ($0.86\mu m$) permiten discriminar entre suelo y nubes por su alta reflectividad; la región del infrarrojo comprendida entre longitudes de onda de $6.2\mu m$ hasta $7.3\mu m$ permiten identificar características atmosféricas como la temperatura media de una capa de humedad (lower, mid, upper); y el infrarrojo térmico ($10.3\mu m$) determina la temperatura de brillo en función de la radiación absorbida y emitida por parte de las nubes. Se concluye entonces que transformar las características distintivas de los rangos de los espectros ayudan a detectar, con los patrones de textura como la homogeneidad, la varianza y la media, con una mayor precisión la clasificación de las nubes no-precipitables, precipitables e incluso la ausencia de las mismas.

Dentro de las técnicas de clasificación de nubes se emplearon 3 métodos de aprendizaje automatizado (*Machine Learning*), seleccionados a partir de un análisis de trabajos similares demostrando que este tipo de métodos tiene ventajas en el campo de la meteorología y la percepción remota, debido a que se presentan procesos complejos en una multidimensionalidad tal como fue desarrollado en esta investigación. Los 3 métodos seleccionados requirieron un proceso de optimización de hiperparámetros con el fin de obtener el mejor desempeño, destacando un conjunto de hiper-parámetros específicos para cada modelo.

Todos los clasificadores usaron el mismo conjunto de datos, el cual dividió en dos grupos: el primero, compuesto por el 60% de las muestras, empleadas en el entrenamiento y el segundo, compuesto por el 40% restante, para determinar la precisión en que clasificaban las 3 clases definidas: nubes precipitables, nubes no-precipitables y suelo. Para cada modelo se obtuvo un conjunto de hiper-parámetros destacados por su alta precisión en dicho proceso. En *Random Forest* su principal hiper-parámetro que perfecciona el rendimiento fue el número de árboles de decisión, obteniendo un número de 1.300. Así mismo, en las *Máquinas de Soporte Vectorial* se destacó el hiper-parámetro *kernel*, donde se encontró que el tipo *rbf* era aquel que perfeccionaba el rendimiento. En MLP la función de optimización *Nadam* y la función de activación *Relu* fueron los hiper-parámetros que se destacaron.

Adicionalmente, basados en el análisis de los resultados se observó que en el método *Random Forest*, las bandas azul y *tir*, en combinación con las texturas media y varianza son las que más aportan en la distinción de clases.

Finalizando con los clasificadores, para determinar la precisión de cada uno de los métodos, se realiza el proceso de validación con el 40% de los registros, comparando las salidas de los modelos con las salidas reales. Los resultados demuestran que los métodos son eficaces a la hora de detectar las nubes, pero presenta una capacidad pobre a la hora de clasificar los cuerpos de agua, además, entre los métodos usados no existen diferencias significativas ya que sus porcentajes de precisión fueron 99.87%, 99.97% y el 100% para *Random Forest*,

SVM y *MLP* respectivamente. Este trabajo se enfocó en la información de textura proveniente del conjunto de bandas de las imágenes **GOES-16** para determinar la precisión de cada uno de los modelos de clasificación.

El diseño de la herramienta para la clasificación de nubes precipitables usando imágenes del satélite **GOES-16** funciona a través de una interfaz gráfica de fácil uso, donde el usuario puede elegir entre cada uno de los métodos propuestos. Además, brinda al usuario de forma rápida una visualización de la detección de nubes de acuerdo a cada capa seleccionada. Esta herramienta proporciona ciertas ventajas como lo son: el fácil uso, la clasificación de nubes por tres métodos diferentes los cuales son efectivos y la obtención de resultados de manera rápida. Las pruebas de integración de la herramienta resultaron exitosas, mostrando así la correcta ejecución de cada uno de los requerimientos que tiene la herramienta para la detección de nubes precipitables.

Finalmente, se concluye que esta herramienta demuestra ser de gran utilidad para la detección de nubes precipitables a partir de imágenes satelitales; además, permite al público que no tiene conocimientos previos en *Machine Learning* y manejo de imágenes satelitales, poder generar de forma eficaz el procesamiento de imágenes mediante diferentes métodos de clasificación automáticos.

5.2. Trabajos futuros

Con el objetivo de querer mejorar lo realizado en este documento y poderlo integrar en diferentes trabajos o proyectos futuros, se plantean los siguientes trabajos futuros:

- Realizar la herramienta para diferentes partes del mundo, teniendo en cuenta que el trabajo realizado se hizo solo para el territorio Colombiano, debido a las diferentes características que él presenta por su topografía.

5.2 Trabajos futuros

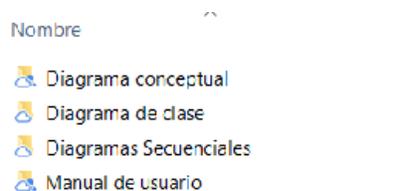
- Generar una herramienta ejecutable en cualquier medio de trabajo, sea computador o teléfono inteligente.
- Debido a la poca capacidad de clasificación de los cuerpos de agua, se podría perfeccionar a partir de la extracción de características de estos cuerpos de tal manera que se mejore la clasificación de las nubes precipitables y no precipitables y poder visualizar los resultados sin que existan estos errores entre los distintos elementos que componen la imagen.

Anexos

Anexo A

A continuación se presenta el esquema en el que están almacenados los diferentes archivos y programas citados en el documento, ese material se encuentra almacenado en una carpeta drive accesible mediante el link: https://drive.google.com/drive/folders/1KmUpaH09NgEt-XowKLx_eEu2piXoN8eL?usp=sharing. Esta carpeta cuenta con los diferentes diagramas como: conceptual, de clase, secuenciales.

Figura 5-1: Distribución de anexos



Nombre
 Diagrama conceptual
 Diagrama de clase
 Diagramas Secuenciales
 Manual de usuario

Anexo B

Con el fin de poner a disposición de toda la comunidad la herramienta desarrollada en el presente trabajo, se crea un repositorio GitHub el cual se puede acceder mediante la url: https://github.com/lapyrodas/Precipitation_cloud. En este se encuentra contenido todos los archivos necesarios para utilizar la herramienta, acompañado de un archivo Readme.md en el que se especifica los recursos necesarios para hacer uso de la misma.

Referencias

- Arab, D., Ameer, S., y Ameer, Z. (2014, Nov). Cloud classification using local multiresolution texture descriptors. En *International image processing, applications and systems conference* (p. 1-8). doi: 10.1109/IPAS.2014.7043266
- Bai, T., Li, D., Sun, K., Chen, Y., y Li, W. (2016). Cloud detection for high-resolution satellite imagery using machine learning and multi-feature fusion. *Remote Sensing*, 8(9). Descargado de <https://www.mdpi.com/2072-4292/8/9/715> doi: 10.3390/rs8090715
- Belgiu, M., y Drăguț, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24–31. Descargado de <http://www.sciencedirect.com/science/article/pii/S0924271616000265> doi: <https://doi.org/10.1016/j.isprsjprs.2016.01.011>
- Berendes, T. A., Mecikalski, J. R., MacKenzie Jr., W. M., Bedka, K. M., y Nair, U. S. (2008). Convective cloud identification and classification in daytime satellite imagery using standard deviation limited adaptive clustering. *Journal of Geophysical Research: Atmospheres*, 113(D20). Descargado de <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JD010287> doi: 10.1029/2008JD010287
- Breiman, L. (2001). Random forests. *Machine learning*, 45(1), 5–32.
- Cai, K., y Wang, H. (2017, Nov). Cloud classification of satellite image based on convolutional neural networks. En *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (p. 874-877). doi: 10.1109/ICSESS.2017.8343049
- Caicedo, E. F., y López, J. A. (2009). *Una aproximación práctica a las redes neuronales artificiales* (U. del Valle, Ed.).
- Chen, D. W., Sengupta, S. K., y Welch, R. M. (1989, oct). Cloud field classification based upon high spatial resolution textural features: 2. simplified vector approaches. *Journal of Geophysical Research: Atmospheres*, 94(D12), 14749–14765. doi: 10.1029/jd094id12p14749

- Cheng, H.-Y., y Yu, C.-C. (2015). Block-based cloud classification with statistical features and distribution of local texture features. *Atmospheric Measurement Techniques*, 8(3), 1173–1182.
- Dewancker, I., McCourt, M., y Clark, S. (2015). *Bayesian optimization primer* (Inf. Téc.). SIGOPT. Descargado 2020-04-08, de https://app.sigopt.com/static/pdf/SigOpt_Bayesian_Optimization_Primer.pdf
- Funk, C., Peterson, P., Landsfeld, M., Pedreros, D., Verdin, J., Shukla, S., ... Michaelsen, J. (2015). The climate hazards infrared precipitation with stations - A new environmental record for monitoring extremes. *Scientific Data*, 2, 1–21.
- Glickman, T. S., y Zenk, W. (2000). *Glossary of meteorology*. American Meteorological Society.
- Guzmán, D., Ruiz, J., y Cadena, M. (2014). Regionalización de Colombia según la estacionalidad de la precipitación media mensual, a través análisis de componentes principales (acp). *Informe Técnico. IDEAM. Bogotá DC*.
- Haralick, R. M., Shanmugam, K., y Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6), 610–621.
- HARRIS. (2018). *Goes-r series product definition and users' guide*.
- Hastie, T., Tibshirani, R., y Friedman, J. (2009). *The elements of statistical learning*. Springer New York.
- Hidalgo, S. (2014). *Random forests para detección de fraude en medios de pago*. Departamento de Ingeniería Informática, Universidad Autónoma de Madrid, Madrid.
- Jin, W., Gong, F., Tang, B., y Wang, S. (2019). Cloud types identification for meteorological satellite image using multiple sparse representation classifiers via decision fusion. *IEEE Access*, 7, 8675–8688.
- Koehrsen, W. (2018a). *A conceptual explanation of bayesian hyperparameter optimization for machine learning*. Descargado de <https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-f>

- [or-machine-learning-b8172278050f](#)
- Koehrsen, W. (2018b). *Hyperparameter tuning the random forest in python*. Descargado de <https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74>
- Kruchten, P. (2004). *The rational unified process: an introduction*. Addison-Wesley Professional.
- Liu, K., Shi, W., y Zhang, H. (2011). A fuzzy topology-based maximum likelihood classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(1), 103–114.
- Manoj, S. O., Kavitha, V., y Arul, J. S. (2010). Recognising the cloud type and estimation of rainfall. En *2010 international conference on communication and computational intelligence (incocci)* (pp. 178–183).
- Mecikalski, J. R., y Bedka, K. M. (2006). Forecasting convective initiation by monitoring the evolution of moving cumulus in daytime goes imagery. *Monthly Weather Review*, 134(1), 49-78.
- Medina, R. F., y Nique, C. I. (2017). Bosques aleatorios como extensión de los árboles de clasificación con los programas r y python. *Interfases*(010), 165–189.
- Menzel, W. P. (2001). Cloud tracking with satellite imagery: From the pioneering work of ted Fujita to the present. *Bulletin of the American Meteorological Society*, 82(1), 33–47.
- Michaelides, S., Levizzani, V., Anagnostou, E., Bauer, P., Kasparis, T., y Lane, J. (2009, dec). Precipitation: Measurement, remote sensing, climatology and modeling. *Atmospheric Research*, 94(4), 512–533. doi: 10.1016/j.atmosres.2009.08.017
- NOAA, y GOES-R. (2019). *Advanced baseline imager scan mode information*. Descargado de <https://www.goes-r.gov/users/abiScanModeInfo.html#scanMode6A>
- NOAA, y NASA. (2018). *Ground system: Overview*. Descargado de <https://www.goes-r.gov/ground/overview.html>
- Nurtanio, I., Zainuddin, Z., y Setiadi, B. H. (2019, dec). Cloud classification based on images texture features. *IOP Conference Series: Materials Science and Engineering*, 676,

012015. doi: 10.1088/1757-899x/676/1/012015
- Olabe, X. B. (2008). *Redes neuronales artificiales y sus aplicaciones* (P. de la Escuela de Ingenieros, Ed.).
- OMM. (s.f.). *International cloud atlas*.
- OMM. (2011). *Organización meteorológica mundial*. Descargado de <https://public.wmo.int/es>
- OMM. (2017). *Día meteorológico mundial 2017 — organización meteorológica mundial*. Descargado 2019-01-28, de <https://public.wmo.int/es/Dia-Meteorologico-Mundial-2017>
- Orun, A. (2000). A comparative study of meteosat, ECMWF, and radiosonde wind vectors at istanbul. *Remote Sensing of Environment*, 72(3), 309–316.
- Pabón, J., Zea, J., León, G., Hurtado, G., González, C., y Montealegre, J. (2001). El medio ambiente en colombia: La atmósfera, el tiempo y el clima. *Instituto de Hidrología, Meteorología y Estudios Ambientales de Colombia. Bogotá, Colombia*.
- Pyszczyk, L., y Snajder, P. (2017). Identificación de procesos atmosféricos mediante imágenes de satélite. *Geográfica digital*, 13(26), 1–59.
- Ramírez, S. E., y Lizarazo, I. A. (2014). Clasificación digital de masas nubosas a partir de imágenes meteorológicas usando algoritmos de aprendizaje de máquina. *Revista Facultad de Ingeniería Universidad de Antioquia*, 73(0120-6230), 43–57.
- Reyes-Delgado, P. Y., Mora, M., Duran-Limon, H. A., Rodríguez-Martínez, L. C., O'Connor, R. V., y Mendoza-Gonzalez, R. (2016). The strengths and weaknesses of software architecture design in the RUP, MSF, MBASE and RUP-SOA methodologies: A conceptual review. *Computer Standards & Interfaces*, 47, 24–41.
- Rochas, M. (1996). Látlas international des nuages. *La Météorologie*.
- Rodríguez, R. J., Montilla, Y., Méndez, Y., y Palma, W. (2014). Alternativas pedagógicas de neofanálisis con imágenes satelitales a través de herramientas manuales y digitales para la estimación de áreas nubosas en Venezuela. *Gaceta Técnica*, 12(1), 51–60.

- Ronaghan, S. (2018). *The mathematics of decision trees, random forest and feature importance in scikit-learn and spark*. Descargado de <https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3>
- Salhi, K., Jaara, E. M., Alaoui, M. T., y Alaoui, Y. T. (2018). GPU implementation of haralick texture features extraction algorithm for a neuro-morphological texture image segmentation approach. En *2018 international conference on electronics, control, optimization and computer science (ICECOCS)*. IEEE.
- Szantai, A., Desalmand, F., y Desbois, M. (2002). A method for the construction of cloud trajectories from series of satellite images. *International Journal of Remote Sensing*, 23(8), 1699–1732.
- Tapakis, R., y Charalambides, A. G. (2013). Equipment and methodologies for cloud detection and classification : A review. *Solar Energy*, 95, 392–430.
- Taravat, A., Frate, F. D., Cornaro, C., y Vergari, S. (2015). Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images. *IEEE Geoscience and Remote Sensing Letters*, 12(3), 666–670.
- Taravat, A., Proud, S., Peronaci, S., Frate, F. D., y Oppelt, N. (2015). Multilayer perceptron neural networks model for meteosat second generation SEVIRI daytime cloud masking. *Remote Sensing*, 7(2), 1529–1539.
- Tian, B., Shaikh, M., Azimi-Sadjadi, M., Haar, T., y Reinke, D. (1999). A study of cloud classification with neural networks using spectral and textural features. *IEEE Transactions on Neural Networks*, 10(1), 138–151. doi: 10.1109/72.737500
- Tsonis, A. A. (1988). Single thresholding and rain area delineation from satellite imagery. *Journal of Applied Meteorology*, 27(11), 1302–1306.
- Wang, Y., Shi, C., Wang, C., y Xiao, B. (2018). Ground-based cloud classification by learning stable local binary patterns. *Atmospheric Research*, 207, 74 - 89.
- Wohlfarth, K., Schroer, C., Klab, M., Hakenes, S., Venhaus, M., Kauffmann, S., . . . Wohler, C.

- (2018). Dense cloud classification on multispectral satellite imagery. En *2018 10th IAPR workshop on pattern recognition in remote sensing (PRRS)*. IEEE.
- Wu, D., y Li, B. (2018). Cloud feature extraction and classification of meteorological satellite cloud imagery. En *2018 international conference on sensor networks and signal processing (SNSP)*. IEEE.