

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA AMBIENTAL**

Trabajo de titulación previo a la obtención del título de: Ingeniera Ambiental

**TEMA:
DESARROLLO DE UNA HERRAMIENTA MEDIANTE UN SISTEMA DE
INFORMACIÓN GEOGRÁFICA PARA EL CONTEO DE ESPECIES
AUTOMATIZADO**

**AUTORA:
DENISSE ELIZABETH LEÓN ERAZO**

**TUTOR:
CÉSAR IVÁN ÁLVAREZ MENDOZA**

Quito, febrero del 2019

CESIÓN DE DERECHOS DE AUTOR

Yo Denisse Elizabeth León Erazo, con documento de identificación N° 1721709762, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autora del trabajo de titulación intitulado: “DESARROLLO DE UNA HERRAMIENTA MEDIANTE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA EL CONTEO DE ESPECIES AUTOMATIZADO”, mismo que ha sido desarrollado para optar por el título de: INGENIERA AMBIENTAL, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autora me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
Denisse Elizabeth León Erazo
1721709762
Febrero de 2019

DECLARATORIA DE COAUTORIA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el presente trabajo de investigación, “DESARROLLO DE UNA HERRAMIENTA MEDIANTE UN SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA EL CONTEO DE ESPECIES AUTOMATIZADO”, realizado por: Denisse Elizabeth León Erazo, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, febrero del 2019



.....

César Iván Álvarez Mendoza
1720100922

DEDICATORIA

Este trabajo quiero dedicarlo principalmente a Dios, que ha sido mi fortaleza para no decaer a pesar de los tropiezos que han existido en el camino.

A mis papis que son unos seres maravillosos, tan comprensivos, amorosos, pacientes y son el pilar y ejemplo de mi vida.

A Mamá Esther por haber sido en vida una de las personas más importantes para mí, y que hoy desde el cielo sé que me cuida y bendice.

A mis abuelitos; Gigito, Mamita Norma, Mamita Magui y Papito Vichi, gracias por su amor y apoyo incondicional.

A mis hermanos que a pesar de cualquier diferencia son mis mejores amigos, cómplices y confidentes.

A Marquito por ser incondicional, ayudarme con mis trabajos y esforzarme a perseverar en la vida.

A mis hermanos de otra mamá; Anita y Héctor por siempre cuidarme y quererme como lo hacen.

A mi Matte hermoso y a mi Mia viejita por llegar a alegrar mis días.

A toda mi familia y amigos.

AGRADECIMIENTO

Expreso mi más profundo agradecimiento:

A la Universidad Politécnica Salesiana y a sus docentes por los conocimientos y consejos impartidos durante todos estos años.

Al personal de laboratorios de la Universidad, por su amabilidad y accesibilidad para la prestación de los equipos.

Al Ingeniero Alexis Libra, administrador del Parque de las Cuadras por la autorización de toma de imágenes aéreas, mediante dron.

A Saúl Noboa por su colaboración y guía para el manejo adecuado del dron.

A mi tutor César Iván Álvarez por el tiempo brindado para la realización del presente trabajo.

INDICE DE CONTENIDOS

1.	INTRODUCCIÓN	1
2.	OBJETIVOS.....	6
2.1.	Objetivo General	6
2.2.	Objetivos específicos.....	6
3.	MARCO TEÓRICO	7
3.1.	Metodologías de conteo de especies vegetales	7
3.2.	Sensores Remotos	9
3.3.	Drones.....	12
3.4.	Plan de vuelo.....	14
3.5.	Fotogrametría digital	15
3.6.	Procesador de imágenes	16
3.7.	Sistemas de Información Geográfica	17
3.8.	QGIS.....	20
3.9.	API	21
3.10.	GRASS.....	21
3.11.	OTB	22
3.12.	Python	23
3.13.	Metodología para desarrollar un script de conteo de especies en QGIS ...	23
3.14.	Definición de las funciones a utilizar	26

3.14.1.	Qgs Raster Calculator Entry	26
3.14.2.	Split Image.....	27
3.14.3.	i.segment.....	27
3.14.4.	Unsupervised kmeans image classification	27
4.	MATERIALES Y MÉTODOS.....	28
4.1.	Diseño.....	28
4.1.1.	Esquema de la modificación del script base y desarrollo del script de estudio 28	
4.1.2.	Esquema General.....	30
4.1.3.	Delimitación de áreas de estudio	31
4.1.4.	Plan de vuelo	31
4.1.5.	Procesamiento de imágenes	34
4.1.6.	Conteo manual de especies arbóreas	38
4.1.7.	Configuración de requerimientos previos al desarrollo del script.....	40
4.1.8.	Tipeo del código	42
4.1.9.	Ejecución del script.....	51
4.2.	Población y muestra:	63
4.3.	Consideraciones éticas:	63
5.	RESULTADOS Y DISCUSIÓN	64
5.1.	Resultados específicos.....	64

5.1.1.	Resultados Parque de las Cuadras	64
5.1.2.	Resultados UPS	70
5.1.3.	Resultados área del Volcán Chiles	71
5.2.	Resultados generales	79
6.	CONCLUSIONES Y RECOMENDACIONES	83
6.1.	Conclusiones	83
6.2.	Recomendaciones.....	86
7.	BIBLIOGRAFÍA	87
8.	ANEXOS.....	94

INDICE DE TABLAS

Tabla 1. Coordenadas Geográficas de las áreas de estudio	4
Tabla 2. Tipos de Muestreo de Vegetación	7
Tabla 3. Clasificación y funciones de los procesos fotogramétricos	15
Tabla 6. Resultados del Parque de las Cuadras.....	67
Tabla 4. Comparación de conteo automático y manual Parque de las Cuadras.....	68
Tabla 7. Resultados del Área del Volcán Chiles	76
Tabla 6. Comparación de conteo automático y manual Parque de las Cuadras.....	76

INDICE DE FIGURAS

Figura 1. Elementos de un sistema de teledetección	10
Figura 2. Clasificación de Tipos de Drones.....	13
Figura 3. Trayectoria de vuelo para ortoimágenes	14
Figura 4. Características de un SIG.....	18
Figura 5. Componentes de un SIG	20
Figura 6. Funcionalidades de GRASS.....	22
Figura 7. Metodología para script de estudio base	25
Figura 8. Interfaz gráfica del estudio base.....	26
Figura 9. Esquema script desarrollado	28
Figura 10. Procedimiento de la investigación.....	30
Figura 11. Plan de vuelo del área del Volcán Chiles	33
Figura 12. Plan de vuelo del Parque de las Cuadras	34
Figura 13. Interfaz de Pix4DMapper.....	36
Figura 14. Ortomosaico del Parque de las Cuadras	38
Figura 15. Nube de puntos del Parque de las Cuadras	38
Figura 16. Imagen de conteo manual en el Parque de las Cuadras mediante cuadrantes	39
Figura 17. Imagen con cuadrantes para conteo manual en el área de páramo Volcán Chiles.	39
Figura 18. Librerías del Sistema	42
Figura 19. Variables globales del sistema	43
Figura 20. Variables de la interfaz gráfica.....	43

Figura 21. Definición del botón de la interfaz gráfica	44
Figura 22. Descripción de todas las funciones del script	44
Figura 23. Función individual de selección de imagen	45
Figura 24. Función individual de Capa RGB	46
Figura 25. Función individual Extensión capa	47
Figura 26. Función individual de separacion de bandas.....	47
Figura 27. Función individual crear semilla	48
Figura 28. Función Individual segmentar imagen.....	48
Figura 29. Función individual clasificar imagen	49
Figura 30. Función individual Identificar copas	49
Figura 31. Función individual conteo de copas	50
Figura 32. Función individual archivo resultante	50
Figura 33. Generación de script en consola de Python.	51
Figura 34. Interfaz gráfica del script.	52
Figura 35. Seleccionador de archivos.....	52
Figura 36. Seleccionador de archivo en carpeta de prueba.	53
Figura 37. Avisos de ejecución del algoritmo.	54
Figura 38. Ejecución del algoritmo.	54
Figura 39. Comprobación de Ejecución del algoritmo.....	55
Figura 40. Documentos resultantes.	56
Figura 41. Imagen de prueba área de páramo del Volcán Chiles	57
Figura 42. Capa RGB	57
Figura 43. Multibanda 0	58
Figura 44. Multibanda 1	58

Figura 45. Multibanda 2	59
Figura 46. Multibanda 3	59
Figura 47. Semillas.....	60
Figura 48. Segmentación	60
Figura 49. Extracción	61
Figura 50. Clasificada.....	61
Figura 51. Identificación de copas	62
Figura 52. Documento de resultados.....	62
Figura 53. Imagen de Prueba Parque de las Cuadras.	64
Figura 54. Imagen con identificación de eucaliptos Parque de las Cuadras.....	65
Figura 55. Resultado final contrapuesto con la imagen de prueba	66
Figura 56. Resultado del primer conteo automático del Parque de las Cuadras.....	66
Figura 57. Resultado del segundo conteo automático del Parque de las Cuadras	67
Figura 58. Resultado del tercer conteo automático del Parque de las Cuadras	67
Figura 59. Imagen de Prueba de la UPS.....	70
Figura 60. Imagen RGB procesada de la UPS.....	71
Figura 61. Patrón de distribución y posicionamiento.....	72
Figura 62. Imagen completa de Prueba área de páramo Volcán Chiles.	73
Figura 63. Imagen seccionada de Prueba área de páramo Volcán Chiles.	73
Figura 64. Imagen de copas de frailejones área de páramo Volcán Chiles.	74
Figura 65. Resultado del primer conteo automático en el área de páramo del Volcán Chiles.	75
Figura 66. Resultado del segundo conteo automático en el área de páramo del Volcán Chiles.	75

Figura 67. Resultado del tercer conteo automático en el área de páramo del Volcán
Chiles.75

INDICE DE ANEXOS

Anexo 1. Ubicación del Parque de las Cuadras	94
Anexo 2. Ubicación de la Universidad Politécnica Salesiana.	95
Anexo 3. Ubicación del Páramo del Volcán Chiles	96
Anexo 4. Descripción del código desarrollado	97
Anexo 5. Procesamiento de Imágenes del Parque de las Cuadras	104
Anexo 6. Procesamiento de imágenes del área de Páramo del Volcán Chiles	108

RESUMEN

El caso de estudio busca aportar una herramienta de identificación y conteo automatizado de especies arbóreas, con la finalidad de optimizar procesos de recolección y análisis de muestras en áreas forestales. Para desarrollar esta herramienta, se ha modificado el script creado por Lia Duarte, Pedro Silva y Ana Claudia Teodoro, en su investigación de “Development of a QGIS Plugin to Obtain Parameters and Elements of Plantation Trees and Vineyards with Aerial Phorographs”.

La modificación fundamental del script sirve para trabajar sobre imágenes RGB, mientras que el script base trabaja en imágenes multiespectrales. Este aporte es de interés porque facilita usar la herramienta por cualquier usuario que posea una cámara fotogramétrica, las cuales son más accesibles que las cámaras multiespectrales. Para el cumplimiento de este objetivo, se desarrolló la herramienta en el SIG de QGIS, utilizando el lenguaje de programación Python.

Las imágenes se obtuvieron en tres diferentes áreas; en el Parque de las Cuadras, ubicado al sur de la ciudad de Quito, en la Universidad Politécnica Salesiana Sede Quito-Sur, y en el páramo del Volcán Chiles; ubicado en la parroquia Tufiño de la provincia del Carchi. La herramienta permitió una correcta identificación de las especies arbóreas en la imagen del área de páramo del Volcán Chiles, parcialmente en la imagen del Parque de las Cuadras y nula en la imagen de la Universidad Politécnica Salesiana. A pesar de ello, se puede concluir que esta investigación permite optimizar recursos al momento de obtención y procesamiento de información en el campo.

ABSTRACT

The case study seeks to provide a tool for the identification and automated counting of tree species, in order to optimize collection processes and analysis of samples in forest areas. To develop this tool, the script created by Lia Duarte, Pedro Silva and Ana Claudia Teodoro, has been modified in its investigation of "Development of a QGIS Plugin to Obtain Parameters and Elements of Plantation Trees and Vineyards with Aerial Photographs".

The fundamental modification of the script is used to work on RGB images, while the base script works on multispectral images. This contribution is of interest because it facilitates the use of the tool by any user that has a photogrammetric camera, which are more accessible than multispectral cameras. For the fulfillment of this objective, the tool was developed in the QGIS GIS, using the Python programming language.

The images were obtained in three different areas; in the Cuadras Park, located to the south of the city of Quito, in the Universidad Politécnica Salesiana Sede Quito-Sur, and in the páramo of the Chiles Volcano; located in the Tufiño parish of the province of Carchi. The tool allowed a correct identification of the arboreal species in the image of the páramo area of the Chiles Volcano, partially in the image of the Cuadras Park and null in the image of the Salesiana Polytechnic University. Despite this, it can be concluded that this research allows us to optimize resources when obtaining and processing information in the field.

Siglas y abreviaturas

API	Interfaz de programación de aplicaciones
CEFF	Coefficiente de eficiencia de Nash y Sutcliffe
GRASS	Sistema de soporte de análisis de recursos geográficos
MDT	Modelo Digital de Terreno
NDVI	Índice de Vegetación de Diferencia Normalizada
NIR	Infrarrojo cercano
OTB	Orfeo Toolbox
QGIS	Quantum GIS
RGB	Red, Green, Blue
RMSE	Error Cuadrático Medio
RPAS	Sistema de Aeronaves Tripuladas por Control Remoto
SIG	Sistema de Información Geográfica
UAV	Vehículo aéreo no tripulado
UPS	Universidad Politécnica Salesiana

1. INTRODUCCIÓN

La toma y análisis de información para generar inventarios forestales o para la planificación en la agricultura se ha realizado hace mucho tiempo de forma manual con investigadores directamente en el terreno. Esto ocasiona que los estudios sean extensos y costosos, por lo que se pretende establecer una metodología moderna, económica, automatizada y digital. Es de suma importancia generar control, monitoreo y seguimiento de diferentes áreas forestales y agrícolas; debido a la relevancia que tienen estos sectores, en cuanto a producción de recursos, prevención de riesgos, conservación medioambiental y preservación de la biodiversidad.

Los bosques representan un pilar fundamental para el bienestar de los ecosistemas, de los factores bióticos y abióticos que los componen, y de sus funciones ecológicas. (Departamento de Montes, 2004).

En la actualidad, los bosques se encuentran sometidos a una fuerte presión debido a la creciente demanda de productos y servicios agrícolas, lo cual a menudo produce la degradación y conversión de los bosques a formas insostenibles de uso de la tierra. Cuando los bosques se pierden o sufren de una degradación severa, su capacidad de funcionar como reguladores del medio ambiente también se pierde, aumentando así las inundaciones, el peligro de erosión, la reducción de la fertilidad del suelo, contribuyendo además a la pérdida de la vida vegetal y animal. (Departamento de Montes, 2004, pág. 1)

La agricultura tradicional es vulnerable a diferentes eventos de tipo climático, también a la presencia de enfermedades y plagas perjudiciales y resilientes a agroquímicos comunes. Por ello se implementan nuevas tecnologías como alternativas de manejo,

monitoreo y control de los cultivos agrícolas en diferentes etapas de su desarrollo, con el objetivo de disminuir costos y aumentar la productividad (Berrío Meneses, Mosquera, & Alzate, 2015).

Para la recolección de información sobre la flora, ya sea en áreas forestales o agrícolas, se pueden aplicar diferentes metodologías; ya sean bibliográficas, manuales en campo o mediante sensores remotos o vuelos aéreos no tripulado con equipo dron. La metodología bibliográfica consiste, en la revisión de estadísticas o inventarios previamente realizados. La metodología manual y la realizada mediante sensores remotos poseen diferentes procedimientos, los cuales se detallan más adelante.

Actualmente, para la identificación y contabilización de especies vegetales se han generado diferentes herramientas dentro de los SIG. QGIS permite ejecutar plug in o script a través de la consola de Python. Estos scripts son de código abierto y pueden ser modificadas por cualquier usuario. La generación de códigos en el lenguaje de Python, utiliza módulos y clases de la Interfaz de programación de aplicaciones (API) de QGIS. Esto permite automatizar los distintos parámetros, considerando la información recabada por el UAV.

Los objetivos de este tipo de herramientas son principalmente para el monitoreo de la vida silvestre en áreas protegidas/ cartografía de árboles, control y preservación medioambiental y para la planificación en agricultura de precisión. En cuanto al monitoreo de la vida silvestre/cartografía de árboles, estas herramientas permiten recopilar suficiente información para construir una base de datos en forma de inventarios, que se encuentren georreferenciados y los cuales puede tener un seguimiento con la finalidad de detectar y/o evitar la deforestación, tráfico ilegal de especies exóticas, cacería furtiva, entre otros.

Por otro lado, para el control medioambiental sirve fundamentalmente para realizar seguimiento de áreas que hayan sido contaminadas, degradadas o erosionadas; permitiendo generar planes de conservación y recuperación espacial. Y por último, la aplicación de estas herramientas para la planificación en agricultura de precisión, se enfoca en evaluar respuestas fisiológicas de las plantas a cambios climáticos, evaluar condiciones del terreno, evitar las plagas, establecer un sistema de irrigación; de tal forma que se puede controlar e incrementar la productividad de los cultivos.

Considerando estas premisas, el objetivo en el cual se enfoca la presente investigación es desarrollar y ejecutar una herramienta que permita el conteo automatizado de especies arbóreas, a través del análisis de imágenes obtenidas con cámaras fotogramétricas de UAV, con el fin de disminuir los tiempos y costos de toma y análisis de muestras, y de aumentar la precisión y exactitud en la identificación de estas especies. Debido a que la herramienta desarrollada es de código abierto, se abre la posibilidad de generar cambios en ella para diferentes propósitos; investigativos, de producción o de conservación.

Para la investigación y desarrollo del presente proyecto, se ha planteado realizar la toma de imágenes mediante UAV en el Parque de las Cuadras, el cual se encuentra ubicado en el sector de la Panga, en la Av. Rumichaca y Matilde Álvarez, al sur de la ciudad del Distrito Metropolitano de Quito, provincia de Pichincha, país Ecuador. Para visualizar el área de ubicación observar el ANEXO 1. El área de estudio, es de 16 hectáreas aproximadamente y la especie arbórea con la que se plantea trabajar es el eucalipto.

Por motivos de generar más pruebas en la herramienta a desarrollar; se ha previsto trabajar con dos imágenes procesadas de estudios realizados anteriormente; el primero

corresponde a un vuelo no tripulado llevado a cabo en la Universidad Politécnica Salesiana dirigido por el Ingeniero César Iván Álvarez, y el segundo correspondiente al vuelo hecho para la tesis del Programa de Maestría en Sistemas de Información Geográfica de UNIGIS América Latina; “Estimación de altura de frailejones (*Espeletia pycnophylla*) en el volcán Chiles mediante UAV (Carchi-Ecuador)”, del Ingeniero Saúl Noboa.

En cuanto a la Universidad Politécnica Salesiana, ésta se encuentra ubicada entre las calles Morán Valverde y Rumichaca, al sur de la ciudad de Quito, provincia Pichincha, país Ecuador. Dentro de esta área no se ha determinado una especie arbórea en específico. Para visualizar el área de ubicación observar ANEXO 2. En la Tesis de Maestría de Saúl Noboa, se realizó la toma de imágenes aéreas en el páramo del Volcán Chiles, ubicado en la Parroquia Tufiño, en la Provincia del Carchi. Se ha identificado como especie de estudio a la planta del frailejón y el área de estudio es de 1 hectárea. Para visualizar el área de ubicación observar la ANEXO 3.

Tabla 1. *Coordenadas Geográficas de las áreas de estudio*

	Latitud	Longitud
Parque de las Cuadras	0°17'15.85"S	78°32'51.34"O
Universidad Politécnica Salesiana Sede-Sur	0°16'55.01"S	78°32'58.64"O
Páramo del Volcán Chiles	0°49'30.90"N	77°56'6.50"O

Fuente: (Google Maps, 2019)
Elaborado por: D León

Los avances tecnológicos permiten simplificar varios aspectos de la vida. Para el caso de estudio el utilizar técnicas de detección remota, imágenes obtenidas a partir de vehículos aéreos no tripulados y la automatización de procesos; ofrece la ventaja de recabar información útil del campo sin la necesidad de trabajar directamente en él. En donde esta información puede ser digitalizada, almacenada y georreferenciada. Estos mecanismos pueden disminuir los lapsos de recopilación y análisis de datos. Además que permite facilitar el trabajo en zonas de difícil acceso o en áreas muy extensas. Al mismo tiempo que disminuyen los costos de viáticos del personal y equipamiento de muestreo.

El desarrollo de una herramienta en un software abierto permite que se pueda aplicar para varios propósitos porque puede ser modificada a la conveniencia del usuario. Por este motivo, la base fundamental de esta investigación es innovar en el área de detección de características arbóreas y de su cuantificación, mediante un script que realice esta función de manera automatizada. En estudios anteriores se han desarrollado herramientas similares, las cuales han estado enfocadas a trabajar con imágenes obtenidas por cámaras multispectrales de alta calidad, mientras que la presente investigación se dirige hacia imágenes obtenidas por cámaras fotogramétricas de drones.

2. OBJETIVOS

2.1. Objetivo General

- Desarrollar una herramienta de conteo de especies arbóreas a partir de imágenes obtenidas en un UAV.

2.2. Objetivos específicos

- Planificar mediante equipo Drone la adquisición de información de especies arbóreas.
- Generar un plug in o script en lenguaje de programación Python, ejecutándolo a través de QGIS.
- Obtener resultados visibles de la detección de conteo automatizado de especies arbóreas.

3. MARCO TEÓRICO

3.1. Metodologías de conteo de especies vegetales

Para realizar el conteo de especies arbóreas existen diferentes metodologías; estas pueden ser bibliográficas, manuales en campo o mediante sensores remotos o vuelos aéreos no tripulados mediante equipo drone. La metodología manual en campo posee diferentes tipos de muestreo, como menciona Mostacedo & Fredericksen (2000) y se detalla en la Tabla 2:

Tabla 2. *Tipos de Muestreo de Vegetación*

Tipo	Características
Transectos	Es un método rápido y heterogéneo. Se forma un rectángulo en el lugar de medición. Se miden parámetros como la altura de la planta, abundancia, diámetro de altura de pecho y frecuencia.
Transectos Variables	Este método permite muestrear un número determinado de individuos a lo largo de un transecto con un ancho determinado. Es utilizado para muestrear cualquier tipo de planta.
Cuadrantes	Es uno de los métodos más comunes. Se coloca un cuadrado sobre la vegetación con el fin de determinar la densidad, cobertura y frecuencia de las plantas. Es un método homogéneo.
Punto Centro Cuadrado	Se utiliza principalmente para muestreo de árboles. Tiene la ventaja de ser rápido, poco equipo y mano de obra. Se establece un punto central y a partir de ello se debe generar cuatro cuadrantes, en donde se

	ubica al árbol más cercano al punto central. Se obtienen los parámetros de especie, densidad diámetro de altura de pecho y frecuencia.
Puntos de intercepción	Se utiliza para la determinación de estructuras y composición vegetal. Es apto para muestrear vegetación graminoide y arbustiva. Este método utiliza una varilla delgada con escala graduada; ésta se coloca en forma vertical para registrar aquellas plantas que se interceptan en las diferentes alturas.
Fitosociológicos	Se utiliza para áreas homogéneas y se debe tener conocimiento previo de todas las especies que existen en la zona. Se asignan categorías de cobertura a cada especie. Es un método subjetivo.

Fuente: (Mostacedo & Fredericksen, 2000)
Elaborado por: D. León

En esta metodología, además de determinar el número adecuado de toma de muestras, el tiempo de recolección, la factibilidad de acceder al terreno; se debe tomar en cuenta el presupuesto destinado para viáticos, materiales y equipos.

La metodología mediante sensores remotos o UAV, tiene la ventaja de abarcar áreas extensas. La detección de árboles mediante procesamiento digital de imágenes puede realizarse mediante segmentación espectral, utilización de modelos de elevación y reconocimiento de formas geométricas. Se puede analizar imágenes a distintas escalas para obtener una mejor segmentación (Falk & Campos, 2014). Esta metodología permite recopilar información de forma eficiente y económica.

Al utilizar esta metodología de recopilación de datos en el terreno, se tiene como resultado imágenes digitalizadas con información geográfica. Para almacenar, administrar y analizar estas imágenes se utilizan los Sistemas de Información Geográfica. Estos son sumamente relevantes en la actualidad porque permiten resolver problemas complicados en cuanto a la planificación y gestión de un determinado territorio.

Es decir, que un SIG funciona como un modelo simplificado de características, “las cuales han sido obtenidas por el proceso de abstracción, de disección de la realidad en partes y de definición de las reglas y comportamiento de dichas partes y sus relaciones entre sí” (Garay & Agüero, 2016, pág. 4).

3.2. Sensores Remotos

La obtención de información ya sea de un objeto o de un proceso, a través de un instrumento el cual no está en contacto directo con los mismos, es llevada a cabo por los sensores remotos. (Pontificia Universidad Católica de Chile, s.f.) “La percepción remota de energía del espectro electromagnético se conoce como teledetección” (Pontificia Universidad Católica de Chile, s.f.).

La percepción remota o teledetección, es una técnica que no está en contacto directo o físico con el objeto, por lo que requiere un flujo de información entre el sensor y el objeto. Este flujo es conocido como radiación electromagnética. Se tiene conocimiento de que todos los seres y elementos en el universo emiten radiación electromagnética. En el sistema solar el mayor emisor de esta radiación es el Sol (Departamento de la Guajira, 2007).

La Teledetección tiene dos procesos. Primero, la recopilación de información de la superficie de la Tierra o de la atmósfera captando la radiación electromagnética emitida o reflejada, trabajando con sensores montados sobre satélites. Segundo, la información recopilada es transmitida a centros en la Tierra en donde se almacena para ser utilizada (Puente, 2007).

Cada cubierta terrestre como la vegetación, agua y suelo, reflejan la luz del sol en diferentes canales del espectro electromagnético, a partir de lo cual se puede clasificar los usos del suelo o conocer la vegetación predominante en un área específica (Puente, 2007).

Según el Departamento de la Guajira y en base al párrafo anterior se puede determinar los elementos que constituyen un sistema de teledetección, como se observa en la Figura 1.

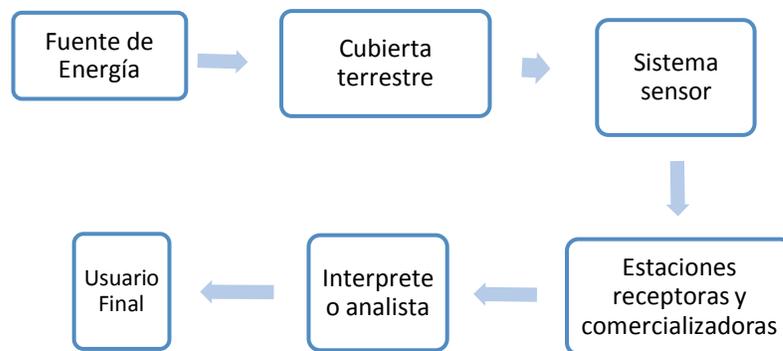


Figura 1. Elementos de un sistema de teledetección

Fuente: (Departamento de la Guajira, 2007).

Elaborado por: D. León

La clasificación de los sistemas sensores puede realizarse en base a los tipos de sensores, al tipo de resolución y la relación que existe entre distintos tipos de resolución. La clasificación por tipos de sensores tiene dos características fundamentales; los pasivos

y los activos. En donde los sistemas sensores pasivos tienen la capacidad limitada de sólo recolectar la energía electromagnética que proviene de la cubierta terrestre, ya sea por la reflexión de los rayos solares o por su propia temperatura. Para este tipo de sensor se conocen algunos aparatos como son: fotográficos, óptico electrónicos y de antena. Por otro lado los sistemas sensores activos tienen la capacidad de emitir un haz de energía, que es recogido posteriormente a su reflexión sobre la cubierta terrestre. Se tienen aparatos como son el Radar y el Lidar.

En cuanto al tipo de resolución se tiene conocimiento de que existen cuatro tipos, que son: espacial, espectral, temporal y radiométrica. “La resolución espacial es el mínimo detalle espacial, es decir un pixel, que puede registrar el sensor; este detalle depende del campo de visión instantáneo y de la altitud de la plataforma” (Departamento de la Guajira, 2007). La resolución espectral es la capacidad que posee el sensor para separar la radiación en diferentes longitudes de onda del espectro electromagnético, está determinada por el número de canales y el ancho de banda de cada canal.

La resolución temporal es la capacidad del sistema sensor para separar cambios temporales que hayan sucedido en la cubierta terrestre. En otras palabras corresponde a la periodicidad con la que el sensor puede adquirir una nueva imagen en el mismo punto de la cubierta (Tristan, Wainschenker, & Doorn). La resolución radiométrica es la capacidad del sensor para separar los niveles de intensidad de radiación espectral, se debe tener en cuenta que mientras mayor sea la precisión radiométrica se tendrá una mejor interpretación.

3.3. Drones

Para la presente investigación, se prevé trabajar con drones, también conocidos como vehículos aéreos no tripulados o Sistema de Aeronaves Tripuladas por Control Remoto. En primera instancia, se conoce que un drone es un equipo volador no tripulado el cual puede controlarse de manera remota. Se conoce que los UAV fueron concebidos con fines militares y posteriormente han tenido aplicaciones comerciales. Es importante considerar que cuando nos referimos a un UAV no estamos hablando solo del vehículo aéreo, sino también de los sensores que ayudan a controlar su vuelo, así como de sistemas que determinan su rumbo y trayectoria, y equipos que permiten recolectar información para ser transmitida a los centros de control.

Las áreas para las cuales se utilizan los UAV son varias; como por ejemplo: inspección y mantenimiento de infraestructuras, servicios de emergencias, cartografía 3D, transporte, ayuda humanitaria, vigilancia, agricultura y conservación de áreas protegidas. Debido a la gran cantidad de aplicaciones, es necesario conocer los tipos de drones que existen, sus características y posibles aplicaciones. Actualmente, hay una variedad extensa de estos aparatos, dependiendo el uso que se le vaya a dar. Según Santana (2017) escritor de Xdrones, se los puede clasificar en “base a los métodos de generación de sustentación, los cuales son mecanismos para generar y mantener la fuerza de sustentación”.

Dicha fuerza nace del principio de que a una altura constante en una capa de fluido, la sumatoria de la presión estática y la presión dinámica es constante; teniendo en cuenta que la presión estática y dinámica tienen una relación inversamente proporcional. Por lo tanto, la fuerza de sustentación “es el resultado de la diferencia de presión entre el extradós e intradós (parte superior e inferior del ala) (...) hay que considerar que en este fenómeno

influyen varios factores como son: ángulo, área y forma del ala, velocidad del aire y densidad del aire” (ANAC). La clasificación se observa en la Figura 2.

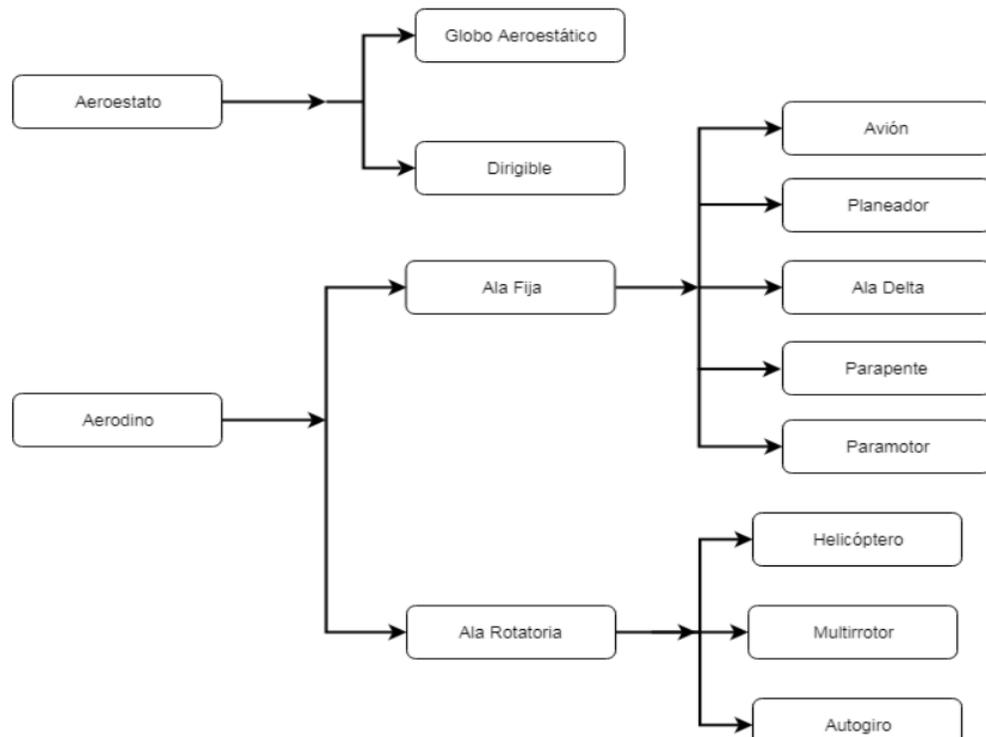


Figura 2. Clasificación de Tipos de Drones
Fuente: (Santana, 2017)

Para esta investigación se requiere trabajar en terrenos agrícolas y parques recreativos que poseen vastas áreas forestales, por lo tanto se ha optado trabajar con el dron DJI PHANTOM 4 en la Universidad Politécnica Salesiana y con el DJI MAVIC Pro en el Páramo del Volcán Chiles y en el Parque de las Cuadras, los cuales son equipos sumamente sofisticados y eficientes.

Se conoce que los drones son cada vez más utilizados para la gestión agrícola y protección de áreas biodiversas o frágiles en Europa y Estados Unidos. Esto sucede porque se trabaja con información de estas áreas de forma más eficiente, dándoles la posibilidad

a los agricultores, investigadores y a los gobiernos en sí, de planificar de mejor manera el entorno. Es necesario conocer que posterior a la realización del vuelo no tripulado se requiere procesar las imágenes obtenidas.

3.4. Plan de vuelo

Para realizar fotogrametrías con drones, se debe establecer un plan de vuelo. Este se refiere a la trayectoria que debe seguir el UAV. Lo primero que hay que tomar en cuenta es la localización, para saber si el área está controlada, si existen interferencias o son espacios restringidos o privados. Después es necesario conocer la orografía del lugar, para establecer un punto de despegue y de aterrizaje. Una vez determinado este punto se determina la altura a la que estará el equipo drone, tomando en cuenta los obstáculos naturales y artificiales (Quintero, 2017). Por último se planifica las trayectorias, identificando si se realiza un vuelo automático o controlado. Generalmente para obtener ortoimágenes la trayectoria a seguir es en forma de cuadrícula como se observa en la Figura 3.

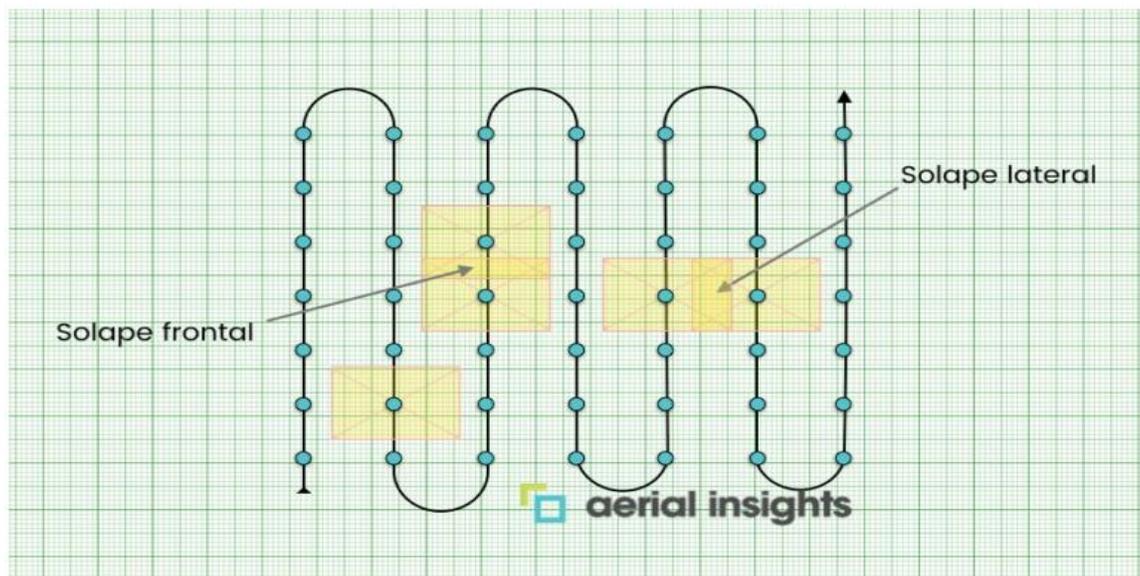


Figura 3. Trayectoria de vuelo para ortoimágenes
Fuente: (aerial insights, 2019)

Es necesario conocer que hoy en día existen aplicaciones móviles que permiten crear planes de vuelo, para el caso de estudio se trabaja con la aplicación de DroneDeploy. Esta aplicación se encuentra disponible para ser descargada desde cualquier Smartphone o Tablet. Se tiene conocimiento que la empresa de DroneDeploy se ha asociado con los principales fabricantes de drones, como es el caso de DJI, para que su software llegue a una gran variedad de usuarios y pueda utilizarse para diferentes propósitos.

3.5. Fotogrametría digital

La fotogrametría es una herramienta capaz de grabar, medir e interpretar información de objetos y su entorno mediante imágenes fotográficas, patrones de energía electromagnética radiante y otros fenómenos. Dentro de la fotogrametría se pueden identificar cuatro etapas: analógica, analítica, digital y utilización de cámaras de gran formato (Centro Europeo de Postgrado, s.f.). “La fotogrametría se puede considerar la tecnología precursora de la teledetección” (Villar, 2010).

La fotogrametría digital se basa en la restitución fotogramétrica digital a partir de imágenes satelitales o fotografías aéreas. Para crear ortoimágenes se deben utilizar todos los detalles planimétricos para ubicar correctamente los puntos y elementos de un plano y facilitar la interpretación (Topoequipos, 2015). En la Tabla 3, se detalla la clasificación y funciones de un sistema fotogramétrico digital.

Tabla 3. *Clasificación y funciones de los procesos fotogramétricos*

Categoría	Procesos, algoritmos	Funciones
Nivel básico	Almacenamiento. Acceso y presentación de imágenes	Manipulación de las imágenes digitales

Nivel bajo	Unión de imágenes. Extracción de características	Orientaciones, ortos digitales, MDT.
Nivel medio	Agrupación. Segmentación de imágenes	Reconstrucción de superficies y características
Nivel alto	Comprensión de imágenes	Reconocimiento e interpretación de imágenes

Fuente: (Cepeda, 2004)
Modificado por: D. León

La Ortofotografía Aérea es un producto cartográfico georreferenciado y corregido de deformaciones que afectan a las fotografías, en la que se pasa de la representación en perspectiva cónica del territorio a una perspectiva ortogonal, corrigiendo las deformaciones por los aspectos mencionados. Este proceso se denomina ortoproyección que aplicado a fotografías o imágenes (digitales) aéreas, mediante las herramientas adecuadas permite la obtención de la Ortofotografía digital. (Junta de Andalucía)

3.6. Procesador de imágenes

Como procesadores de imágenes se plantea utilizar el Pix4DMapper y el Agisoft PhotoScan. El primero se utiliza previamente para procesar las imágenes del área del Volcán Chiles, mientras que el segundo para el Parque de las Cuadras. Se usan dos procesadores porque el Pix4DMapper después 16 días de prueba gratuita requiere una actualización de la licencia.

El Pix4DMapper es un software de fotogrametría que permite generar ortomosaicos y convertir imágenes RGB obtenidas con dron en mapas 2D y modelados 3D. Los trabajos que realiza se encuentran georreferenciados y pueden utilizarse posteriormente en diferentes aplicaciones o softwares (ACRE, s.f.). De igual forma Agisoft PhotoScan permite elaborar mosaicos con las imágenes aéreas obtenidas mediante UAV. Es una herramienta con alto potencial de gestión espacial para construir ortomosaicos y MDT.

Este procesador es ideal para trabajar con imágenes obtenidas por cámaras fotogramétricas, es decir que posean valores RGB (Gis&Beers, 2016).

3.7. Sistemas de Información Geográfica

Los Sistemas de Información Geográfica permiten analizar y representar toda información que haya sido geolocalizada. El National Center for Geographic Information and Analysis de USA los define como “Sistema de hardware, software y procedimientos elaborados para facilitar la obtención gestión, manipulación, análisis, modelado, representación y salida de datos espacialmente referenciados para resolver problemas complejos de planificación y gestión” (NCGIA, 1990). El fundamento de un SIG, es poder representar en un mapa digital diferentes variables u objetos, que corresponden a entradas en una base de datos. (Murcia, 2017). Según Antoni Pérez Navarro y otros (2011), en su libro de “Introducción a los sistemas de información geográfica y geotelemática” menciona que las características que permiten identificar a un SIG son las que se presentan en la Figura 4:

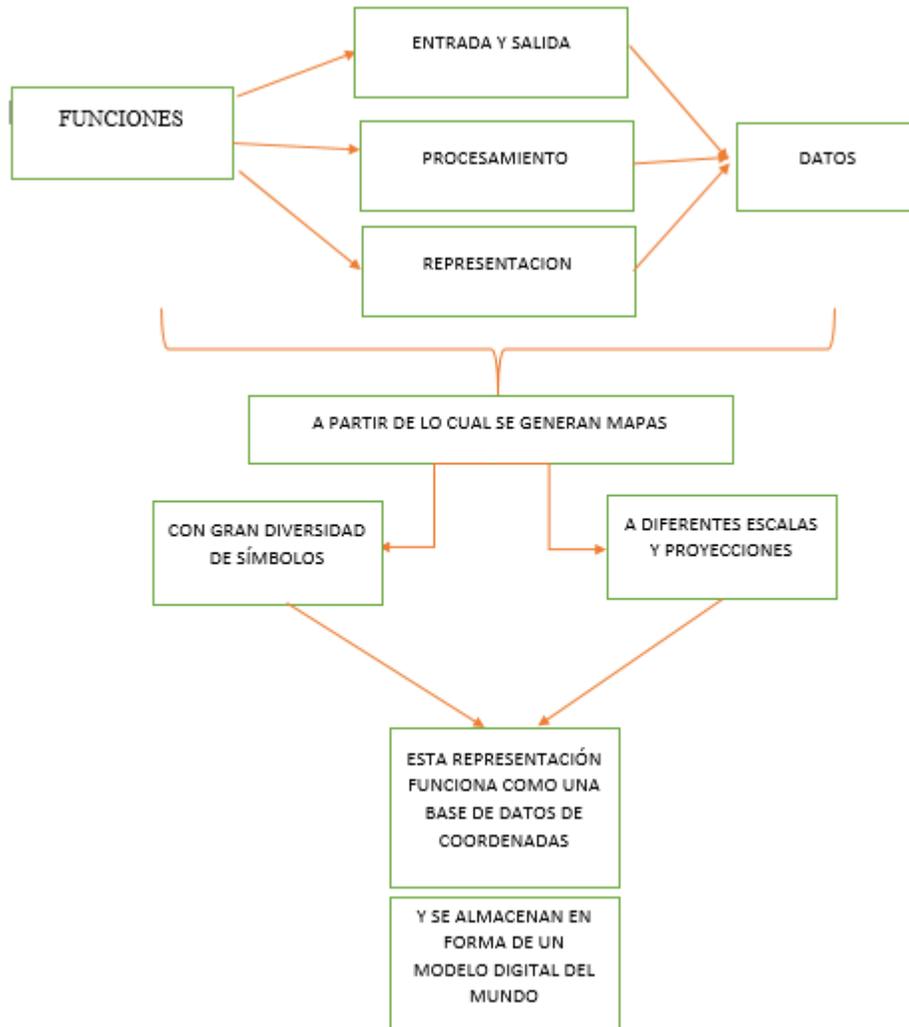


Figura 4. Características de un SIG
 Fuente: (Pérez Navarro, y otros, 2011)
 Elaborado por: D. León

En cuanto a los componentes que posee un SIG, se identifican cinco partes principalmente: software, hardware, datos geográficos, procesamientos y métodos, y las personas. El software corresponde a la parte que se puede visualizar de un SIG. El hardware se enfoca en cuanto a las necesidades de almacenamiento de información, la velocidad de procesamiento, los tipos de dispositivos de entrada y salida. “Los datos son

una abstracción de la realidad y se almacenan como códigos digitales en bases de datos geográficos” (Pérez Navarro, y otros, 2011).

Es necesario mencionar la naturaleza que tienen los datos geográficos de manera más amplia; puesto que con ellos se representan concretamente los hechos y antecedentes que constituyen un determinado fenómeno. A su vez, los datos geográficos poseen tres tipos de componentes:

- Espacial: Es la información referente a la localización, posición de objetos y uso de coordenadas. Ejemplo: área, longitud, volumen. Se puede representar mediante un modelo vectorial; ya sea a través de puntos, líneas o polígonos. También puede representarse mediante un modelo ráster; a través de filas y columnas, las cuales forman un mallado de celdas.
- De atributo o variables: Son las características asociadas a un tema específico en cuanto a un determinado objeto. En otras palabras, se puede entender como la forma de medir los atributos. Se conoce que existen variables de tipo discreta, continua, fundamental y derivada.
- Temporal: Fenómenos geográficos en un periodo de tiempo

Los procesamientos y métodos por su parte, hacen referencia a las funciones básicas que tiene un SIG, cabe recalcar que hay funciones que son más específicas, para lo cual es pertinente tener las herramientas que sistematicen los procesos. Y finalmente, las personas, que corresponden al último componente, las cuales son un pilar fundamental para manipular adecuadamente la información (Taboada González & Cotos Yáñez, 2005).

Observar Figura 5.

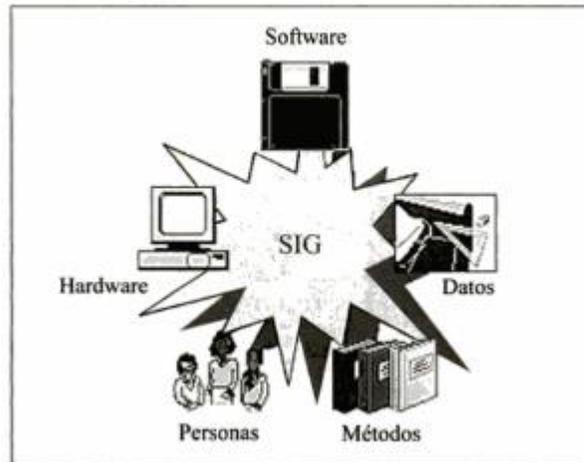


Figura 5. Componentes de un SIG
 Fuente: (Taboada González & Cotos Yáñez, 2005)

3.8. QGIS

En esta investigación se utiliza QGIS, porque es un Sistema de Información Geográfica de código abierto. QGIS admite diversos formatos de datos ráster y vectoriales, pudiendo añadir nuevos formatos usando la arquitectura de complementos. QGIS proporciona una creciente gama de capacidades a través de sus funciones básicas y complementos (CC BY-SA). Puede visualizar, gestionar, editar y analizar datos y diseñar mapas imprimibles. Según Pleiadesic (2014) algunas de sus características son:

- 1.- Visualización y vector de superposición y de datos de trama en diferentes formatos y proyecciones, sin conversión a un formato interno o común.
- 2.- Creación de mapas y explorar interactivamente los datos espaciales con una interfaz gráfica de usuario.
- 3.- Creación, edición y exportación de datos espaciales usando herramientas para la digitalización de GRASS y formatos shapefile, el plug in georeferencer, y herramientas de GPS

para la importación y exportación en formato GPX, convertir otros formatos de GPS para GPX, o bajar / subir directamente a una unidad de GPS

4.- Realizar el análisis espacial utilizando el plug in para fTools Shapefiles o el plug in de GRASS integrado.

3.9. API

API son las siglas en inglés de Application Programming Interface, que hace referencia a un “conjunto de códigos, reglas o especificaciones que las aplicaciones pueden seguir para comunicarse entre ellas: sirviendo de interfaz entre programas diferentes de la misma manera en que la interfaz de usuario facilita la interacción humano-software” (Merino, 2014). Para este estudio se utilizan dos tipos de API para crear nuevos complementos o funcionalidades dentro del QGIS, estas son: QGIS API y PyQT4 API.

3.10. GRASS

Debido al apogeo de ciertos SIG de código libre, muchos otros SIG han quedado relegados a aportar herramientas específicas, que sirven de complemento a los SIG más utilizados actualmente. Este es el caso de GRASS, el cual también es un software SIG, desarrollado por la Fundación OsGeo en 1982; sin embargo su utilidad hoy en día se basa en una combinación de sus herramientas o algoritmos, dentro de QGIS.

Según Alonso (2016) en su artículo publicado en el portal web de Mapping QGIS; el plugin de GRASS en QGIS permite el acceso a las bases de datos y funcionalidades que se detallan en la Figura 6.

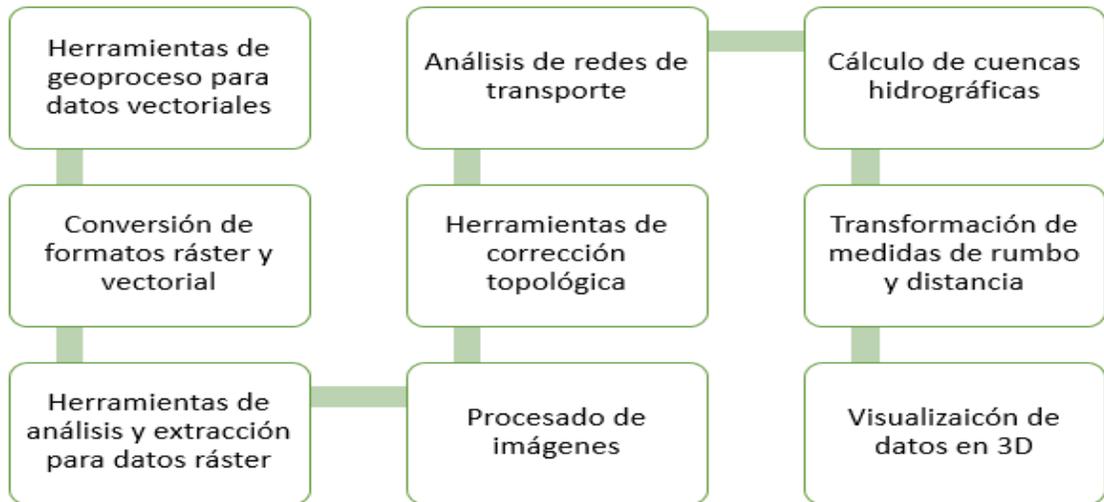


Figura 6. Funcionalidades de GRASS
 Fuente: (Alonso, Mapping Gis, 2016)
 Elaborado por: D. León

3.11. OTB

Orfeo Toolbox, es una biblioteca de código abierto en C para el procesamiento de imágenes, distribuido bajo la licencia CeCILL-v2. Este software tiene como objetivo permitir el procesamiento avanzado de imágenes de gran tamaño incluso en computadoras de recursos limitados. Este software suministra un conjunto de herramientas listas para la teledetección (Carita, 2016).

El OTB sirve para la detección remota, ocorectificación, procesamiento de imágenes ya sean multiespectrales o RGB, clasificación, segmentación y manipulación de las mismas. OTB proporciona varios algoritmos, para el presente trabajo se plantea utilizar los algoritmos de clasificación supervisada y de clasificación de imágenes no supervisada kmeans.

3.12. Python

Para el desarrollo de la herramienta de conteo de especies arbóreas de este trabajo, se requiere crear un script o un plug in, en el lenguaje de programación Python. Se conoce como Script, a un texto o documento el cual contiene instrucciones u órdenes escritos en códigos de un determinado lenguaje de programación. En cuanto a Python, se conoce que “es un lenguaje de programación con tipado dinámico, que posee estructuras de datos eficientes y un enfoque simple pero efectivo a la programación orientada a objetos” (Rossum, 2009).

Dadas las características de QGIS como las de Python, es que se conjugan íntegramente y se crea PyQGIS, con la finalidad de que sea factible desarrollar herramientas mediante el uso de un lenguaje sencillo. Una de las librerías con las que se trabaja frecuentemente es PyQt4 API; que es una herramienta que permite enlazar datos para el lenguaje de programación de Python, este proceso también es conocido como “binding”.

3.13. Metodología para desarrollar un script de conteo de especies en QGIS

En la actualidad se han desarrollado varias herramientas semi-automáticas y automáticas para la identificación y conteo de árboles. Muchas de estas herramientas se han creado en softwares de SIG que requieren licencias pagadas para acceder a todos sus productos, lo que dificulta su accesibilidad para varios usuarios. En tanto, otras de estas herramientas han sido desarrolladas en SIG de código abierto, sin embargo algunas suelen tener costo porque han sido creadas por empresas privadas.

Es por estas razones que se plantea una metodología para crear una herramienta accesible, replicable y que además es gratuita. Como se mencionó anteriormente, QGIS es un SIG de código libre que se usa comúnmente por los beneficios que brinda al usuario.

Uno de los beneficios que favorece a la presente investigación es la vinculación con Python API para poder desarrollar nuevas herramientas, y con múltiples librerías como GRASS, OTB, GDAL/OGR, entre otras. Estas librerías permiten acceder a comandos que ejecuten las funciones deseadas.

Basándose en el principio de que la herramienta sea accesible, replicable y gratuita; Lia Duarte, Pedro Silva y Ana Cláudia Teodoro, desarrollaron un plug in que permite obtener parámetros y elementos de diferentes plantaciones de árboles. El tipo de imágenes con las que decidieron trabajar son obtenidas de cámaras multiespectrales, lo que significa que la resolución espacial de estas imágenes es alta y de buena calidad. La metodología para el desarrollo de su herramienta es la siguiente:

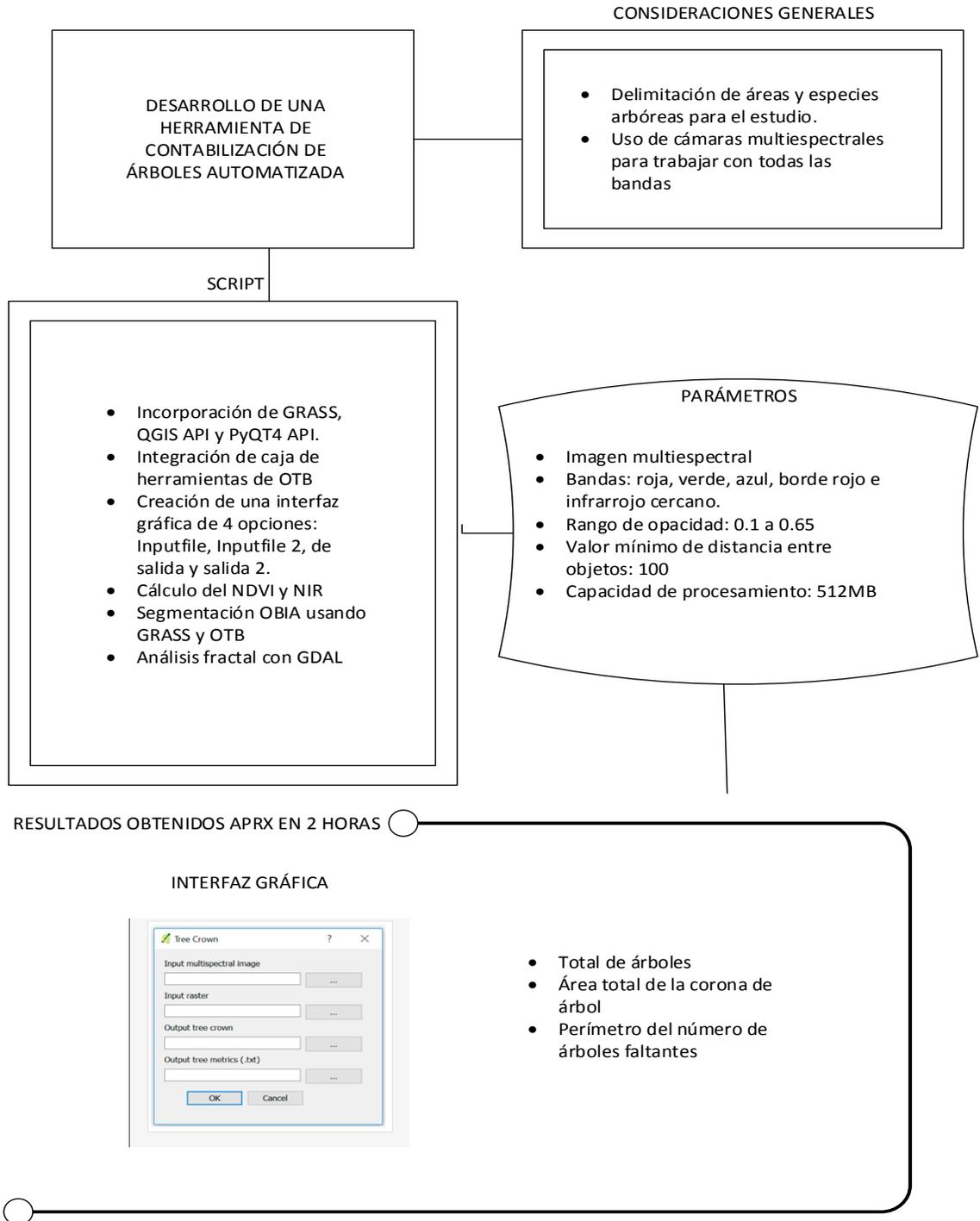


Figura 7. Metodología para script de estudio base
Elaborado por: D. León

Para una mejor apreciación se adjunta la imagen de la interfaz gráfica de su herramienta:

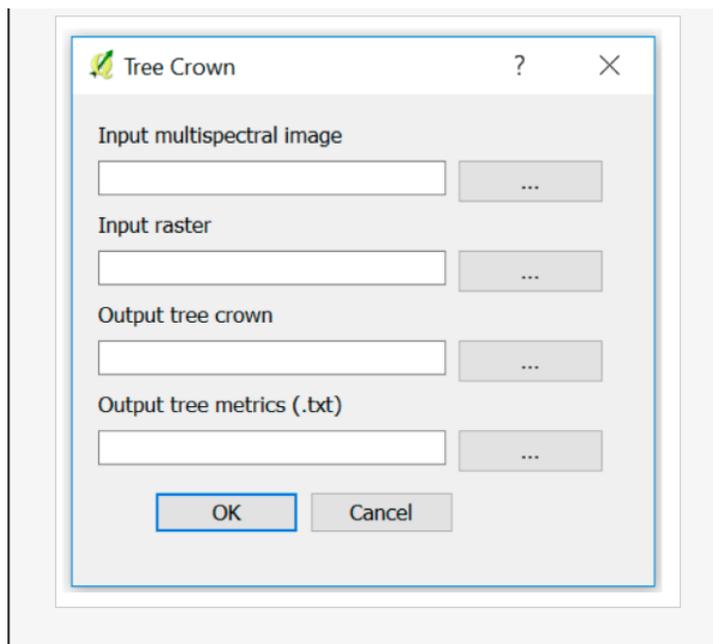


Figura 8. Interfaz gráfica del estudio base
Elaborado por: D. León

Para visualizar el plug in que han desarrollado se puede acceder mediante el siguiente link: <http://creativecommons.org/licenses/by/4.0/> (Duarte, Silva , & Teodoro, 2018).

3.14. Definición de las funciones a utilizar

3.14.1. Qgs Raster Calculator Entry

Raster Calculator Entry es una función propia de QGIS, que crea una lista de bandas ráster que contiene todas las capas ráster del proyecto. Si existen varias bandas se las debe enumerar. “Los resultados se escriben en una nueva capa ráster con un formato compatible con GDAL” (CC BY-SA). En esta investigación es de suma importancia que el archivo

resultante sea compatible con GDAL (un complemento de QGIS), porque será utilizado en el desarrollo del script en la etapa de análisis fractal.

3.14.2. Split Image

Existen diversas formas de dividir o seccionar una imagen multibanda de n bandas, en n imágenes; ya sea utilizando directamente herramientas de QGIS, sus complementos o a través de la consola de Python. Para este estudio se utiliza una herramienta de QGIS: “splitimage”, la cual proviene de la caja de herramientas de OTB.

3.14.3. i.segment

“i.segment” es una herramienta de la API de GRASS, que permite identificar diferentes objetos a partir de la información recopilada del procesamiento del archivo ráster. La información con la que se trabaja es con un reconocimiento automático entre segmentos, y para que se pueda llevar a cabo se recurre a establecer un valor de la distancia mínima entre objetos.

3.14.4. Unsupervised kmeans image classification

Esta función permite realizar una clasificación no supervisada mediante el algoritmo de agrupamiento de Kmeans. Este algoritmo “agrupa objetos en k grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo” (Departamento de matemáticas Universidad de Oviedo, 2014). Dicha función se obtiene de la caja de herramientas OTB. Además de la clasificación no supervisada, existen dos tipos más de algoritmos de clasificación; supervisada y semisupervisada.

4. MATERIALES Y MÉTODOS

4.1. Diseño

4.1.1. Esquema de la modificación del script base y desarrollo del script de estudio

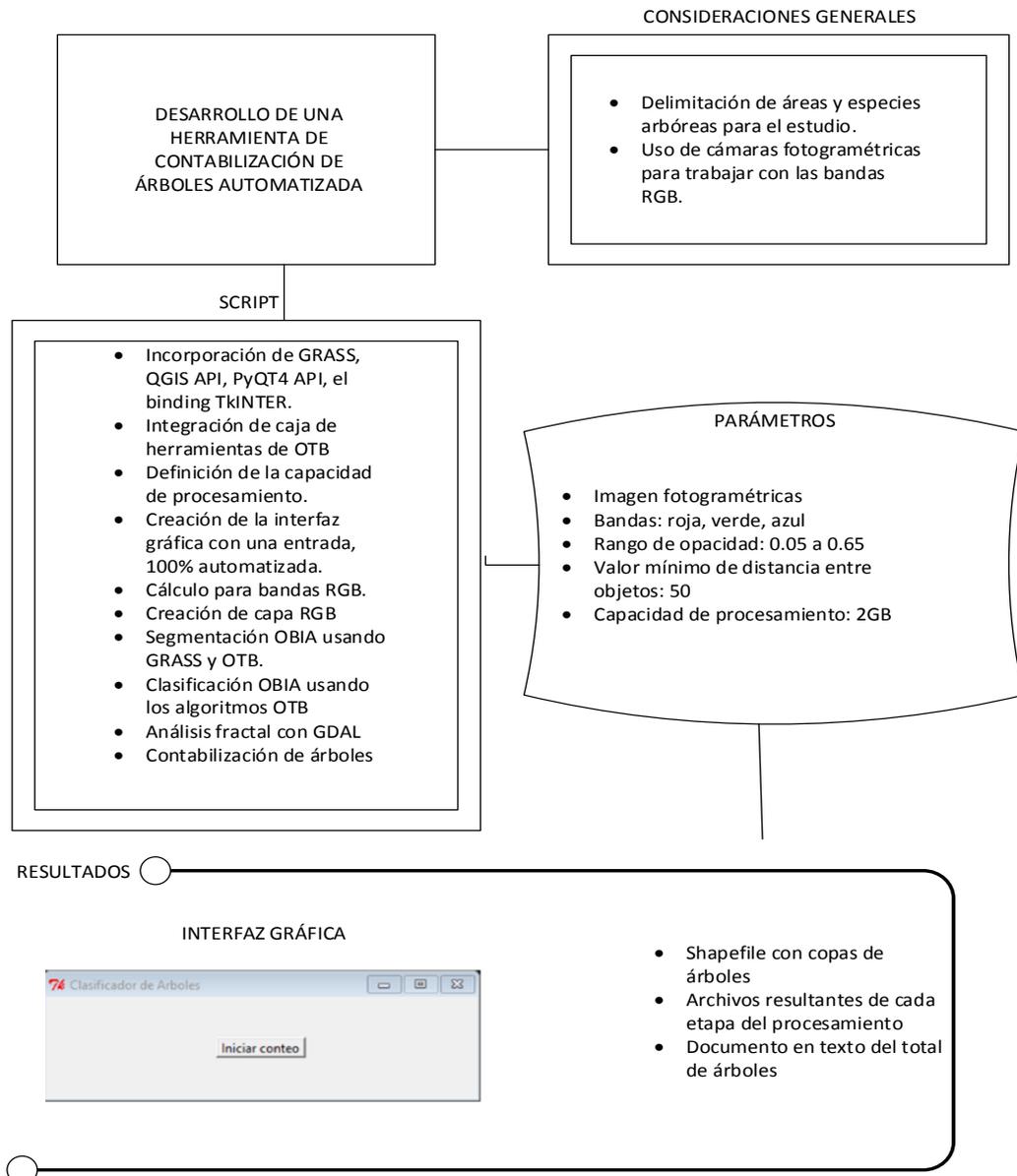


Figura 9. Esquema script desarrollado
Elaborado por: D. León

La principal modificación del script base para la presente investigación es en la fase del código para el cálculo de bandas RGB, porque se trabaja con imágenes fotogramétricas. El estudio de Lia Duarte, Pedro Silva y Ana Cláudia Teodoro; utiliza imágenes multiespectrales, las cuales debido a las longitudes de onda que emiten los objetos de la imagen posee varias bandas, mientras que las imágenes fotogramétricas solo tienen 3 bandas (rojo, verde y azul). Es decir, lo que pretende la modificación de la herramienta; es poder identificar objetos dentro de una imagen aérea de menor resolución espacial, por ende con menos información del terreno. La ventaja de esta herramienta está enfocada en que las cámaras fotogramétricas, son más accesibles en cuanto a costo que las cámaras multiespectrales; por lo que se puede realizar más pruebas por más usuarios.

Adicionalmente, se han realizado modificaciones en la forma de ejecución, procesamiento de la información e interfaz de la herramienta. Como primer punto, se ha incorporado el binding TkINTER para una mejor comunicación entre QGIS y Python. También se ha definido la capacidad de procesamiento de la herramienta, con un valor de 2GB, este valor depende la capacidad de cada computadora en la que se realice la prueba. La interfaz gráfica de la herramienta desarrollada difiere de la herramienta base, en que se tiene cómo parámetro de entrada únicamente el archivo de prueba, mientras que en la herramienta base se requieren cuatro archivos, dos de entrada y dos de salida. Esto representa una ventaja puesto que el procesamiento de la imagen es más automatizado.

Otra modificación que se ha realizado, es para encontrar el valor óptimo de minsize o valor mínimo de distancia entre objetos. Para el estudio base se obtuvo cómo valor ideal 100, mientras que para esta investigación es de 50. De igual forma se trabaja con diferentes rangos de opacidad para el entrenamiento de segmentación en la imagen, en el estudio

base es de 0.1 a 0.65 y en esta investigación de 0.05 a 0.65, aunque en ambos casos se trabajó con secuencias intercaladas por un valor de 0.05. Finalmente, la herramienta base genera los archivos resultantes del total de árboles, del área total de la corona de árbol y del perímetro del número de árboles faltantes. Por su lado, la herramienta de esta investigación permite obtener archivos de cada etapa del procesamiento, un shapefile con la identificación de las copas y un archivo de texto con el total de árboles.

4.1.2. Esquema General

A continuación, se esquematiza el procedimiento que se siguió para este trabajo:

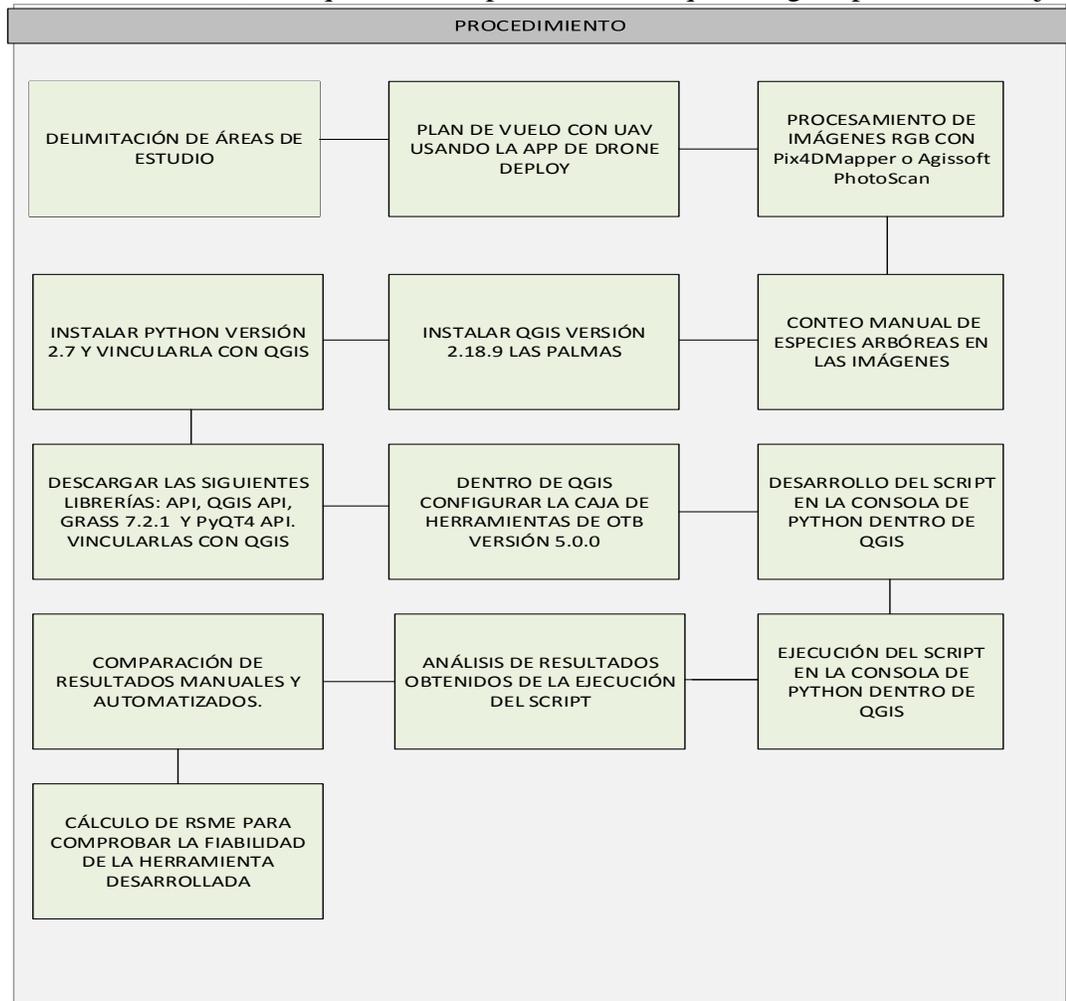


Figura 10. Procedimiento de la investigación
Elaborado por: D. León

4.1.3. Delimitación de áreas de estudio

Como primer paso se delimita las áreas de estudio y las especies arbóreas que se prevé identificar en cada área. En el Parque de las Cuadras se ha planteado realizar un vuelo a una altura de 100 metros, porque la especie arbórea que se quiere identificar y posteriormente contabilizar es el eucalipto. Esta especie de árbol tiene una altura promedio de 80 metros. El área de vuelo tiene una extensión aproximada de 16 hectáreas, sin embargo para el propósito de realizar la prueba de conteo de especies arbóreas tanto manual como automatizado por el script; se trabaja solo con una sección de la imagen para agilizar el procesamiento de la información. El drone utilizado en esta área es el DJI Mavic Pro.

Para la imagen aérea de la UPS, se utiliza el drone DJI Phantom 4, a una altura de 100 metros. En esta imagen se plantea identificar todas las especies arbóreas que existan, no se ha definido una especie en particular. En el área de páramo del Volcán Chiles, el vuelo se llevó a cabo a una altura de 40m, en donde los frailejones tienen una altura aproximada de 7m. El área del vuelo fue de 2 hectáreas, pero para el conteo de frailejones se trabaja con una sección de la imagen. Se utilizó el drone DJI Mavic Pro.

4.1.4. Plan de vuelo

Los planes de vuelo se establecen en la aplicación de DroneDeploy. Primero se tiene que descargar la aplicación y crear una cuenta en ella. Para la planeación del vuelo se requiere que la computadora, el celular o tablet con la que se vincula el drone, tenga señal de internet y activado el GPS. Una vez instalado el programa se procede a abrir la opción de configuraciones y seleccionar unidades métricas. Se escoge la altura máxima

dependiendo de las características del área delimitada, la altura máxima permitida por ley en el Ecuador es de 122 metros (Crónica, 2015).

Una vez establecidas estas características se debe salir de la opción de configuración. Se procede a crear el plan de vuelo, para lo cual se selecciona el símbolo de signo más y se escoge la primera opción “plan a map flight”. Se despliega una nueva lista de opciones a configurar y en la parte derecha de la pantalla se visualiza un mapa global similar al de Google Earth, en el que se puede ampliar las áreas en donde se desea realizar el vuelo. Se coloca un nombre al plan de vuelo.

La modificación de la zona de trabajo se realiza mediante la colocación de puntos de control alrededor de su perímetro, formándose un polígono. La información acerca del tiempo de vuelo, el área total en hectáreas, el número de imágenes a obtener y la cantidad de baterías que requiere el vuelo; dependen de la extensión del polígono. La cantidad de puntos se puede editar en base a las necesidades del vuelo. En esta etapa se define la altitud del vuelo. En la opción de “Advanced” se procede a configurar el porcentaje y dirección del solapamiento vertical y horizontal, recomendándose un valor del 75% y en cuanto a la dirección se debe ubicar el valor preciso para que el vuelo optimice el uso de baterías.

Se trabaja con la velocidad máxima del drone que es de 15 m/s. Después se escoge el punto en donde se requiere que inicie la toma de imágenes, es recomendable dejar que inicie en el vértice más lejano, para que el último punto registrado sea el de despegue y aterrizaje. Se selecciona la opción de “Preflight Checklist” para verificar que los requisitos de permisos, del drone, de la cámara, del controlador y del plan de vuelo estén activados. Finalmente se lleva a cabo el vuelo. El resultado del plan de vuelo del área del Volcán

Chiles y del Parque de las Cuadras se observa en la Figura 11 y Figura 12, respectivamente.

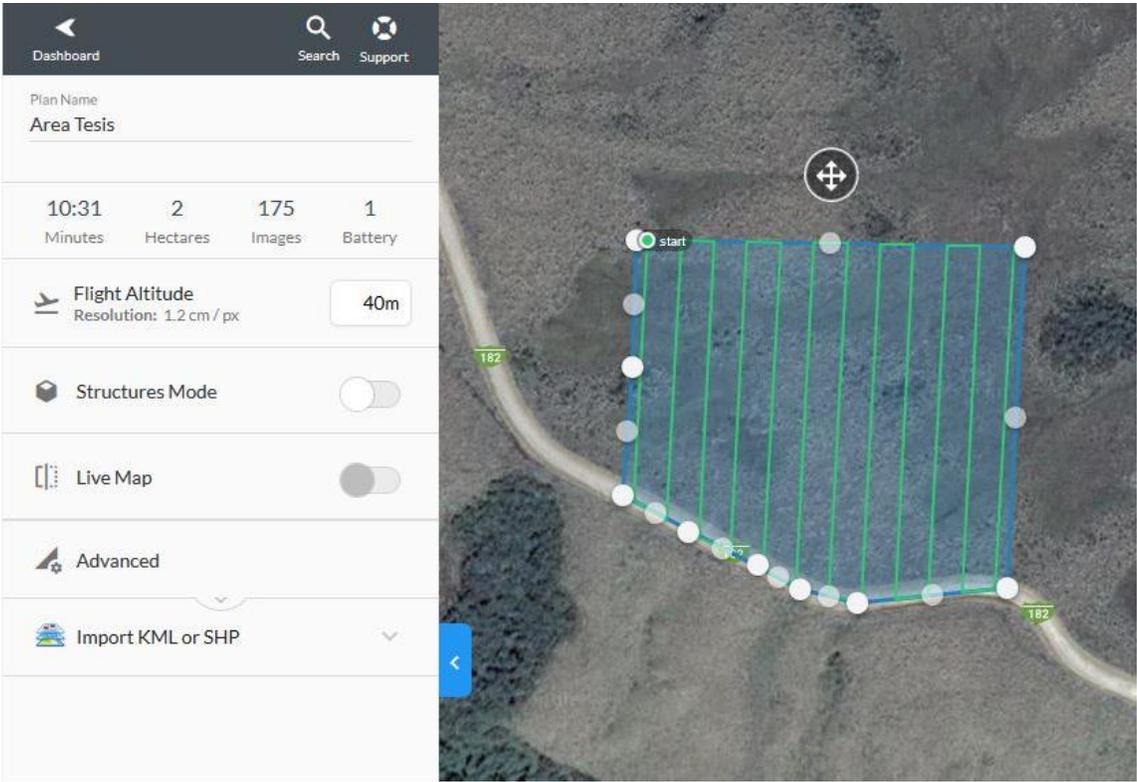


Figura 11. Plan de vuelo del área del Volcán Chiles
Fuente: (Noboa, Sin publicar)

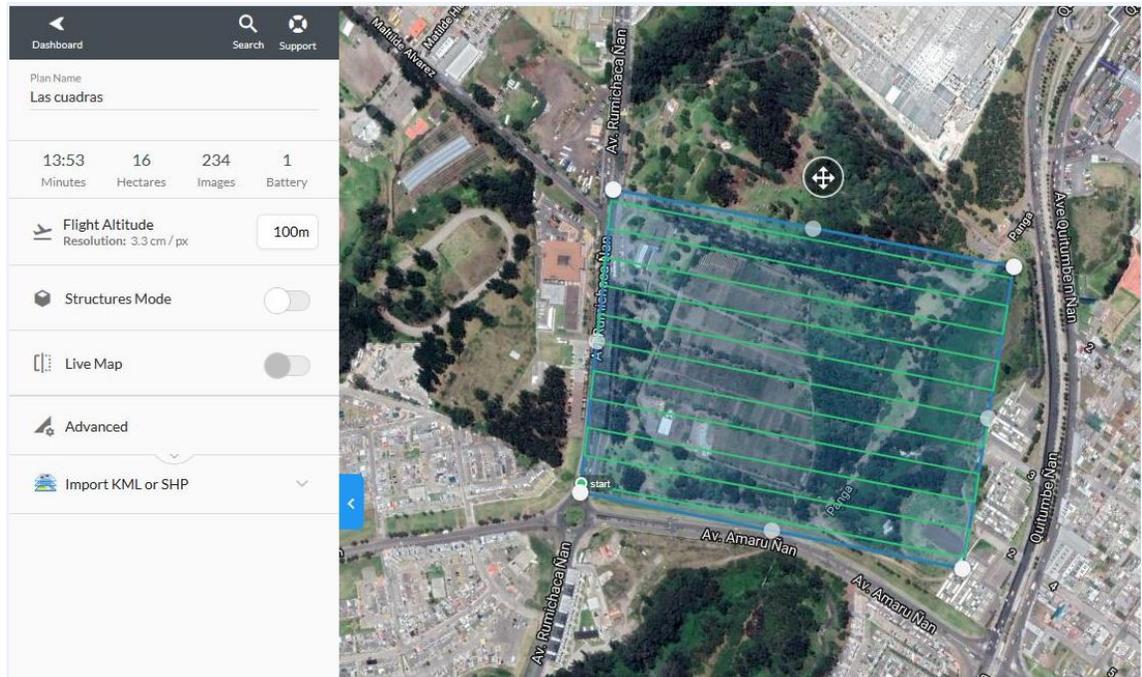


Figura 12. Plan de vuelo del Parque de las Cuadras
Elaborado por: D. León

4.1.5. Procesamiento de imágenes

Es necesario mencionar que las cámaras de los drones utilizados, no son cámaras multiespectrales sino cámaras fotogramétricas, por lo que las imágenes que se obtienen poseen la composición de color RGB. Esta composición sirve para establecer en base a la conversión del espacio de color, dónde existe un objeto y en dónde no. En el área del volcán Chiles se recopilaron aproximadamente 350 fotografías, mientras que en el Parque de las Cuadras 450.

Para llevar a cabo el procesamiento de las imágenes con el programa Pix4DMapper, como primer paso se descarga y ejecuta el programa en una computadora. Previamente se guarda las imágenes obtenidas con el drone al igual que los puntos de control establecidos en el polígono, los cuales tienen información de las coordenadas en donde se realiza el

plan de vuelo. Las imágenes y los puntos de control se guardan en carpetas separadas, pero ambas carpetas deben estar contenidas en una carpeta madre.

Una vez instalado el programa de Pix4DMapper se procede a abrirlo, en la opción Proyecto se selecciona nuevo proyecto, en donde se debe escoger la ubicación de la carpeta madre. Se coloca un nombre al proyecto. Seleccionar siguiente y se añaden todas las imágenes, avanzar hasta la opción de sistema de coordenadas de salida, escoger la unidad de medida de metros y activar la opción de auto detectado. En la plantilla de opciones de procesamiento, seleccionar el tipo de documento que se desea generar, para este caso se elige Ag RGB, verificar que antes de finalizar esté desactivada la opción de iniciar procesamiento ahora.

Se ejecuta el procesamiento inicial, para ello se abre la ventana de opciones de procesamiento. En el procesamiento inicial se escoge una escala de imagen para keypoints completa y aceptar. Se da clic en el botón iniciar. Después se abre la ventana de “Gestor GCP/MTP” para insertar los puntos de control. En los layers se desactivan los puntos automáticos para que solo se trabaje con los puntos GCP/MTPs. Seleccionar los puntos de control uno por uno. Una vez seleccionado un punto, se abre la ventana lateral derecha de propiedades, en donde se aprecian las imágenes que contienen el primer punto de control. En estas imágenes se debe marcar la ubicación del punto.

Se procede a realizar esta acción con todos puntos de control establecidos, para agilizar el proceso se selecciona las opciones automatic marking y aplicar. Es necesario definirlos como puntos de control, por lo que se les define en la opción tipo como tales. A pesar de que ya ha concluido el procesamiento inicial, se requiere realizar una reoptimización por las definiciones dadas. Para eso se selecciona la opción de procesar y reoptimizar.

Luego se trabaja con la nube de puntos, para lo cual se escoge la escala de imagen por defecto que tiene un valor 1/2, densidad de puntos óptima, número de emparejamiento mínimo de 3 y exportar malla de puntos en formato LAS. Se abre la ventana de procesamiento y se escoge la etapa de nube de puntos y malla. Se debe ejecutar la clasificación de la nube de puntos de terreno, esta función se ubica en la opción “procesar”.

Finalmente se abre la ventana de opciones de procesamiento, para esta etapa se deben colocar los valores por defecto de MDS y ortomosaico. Hay que tener en cuenta que este programa ofrece un tiempo de prueba gratuita pero después de 16 días pide una actualización de licencia con costo. En la Figura 13 se observa el procesador de Pix4DMapper.

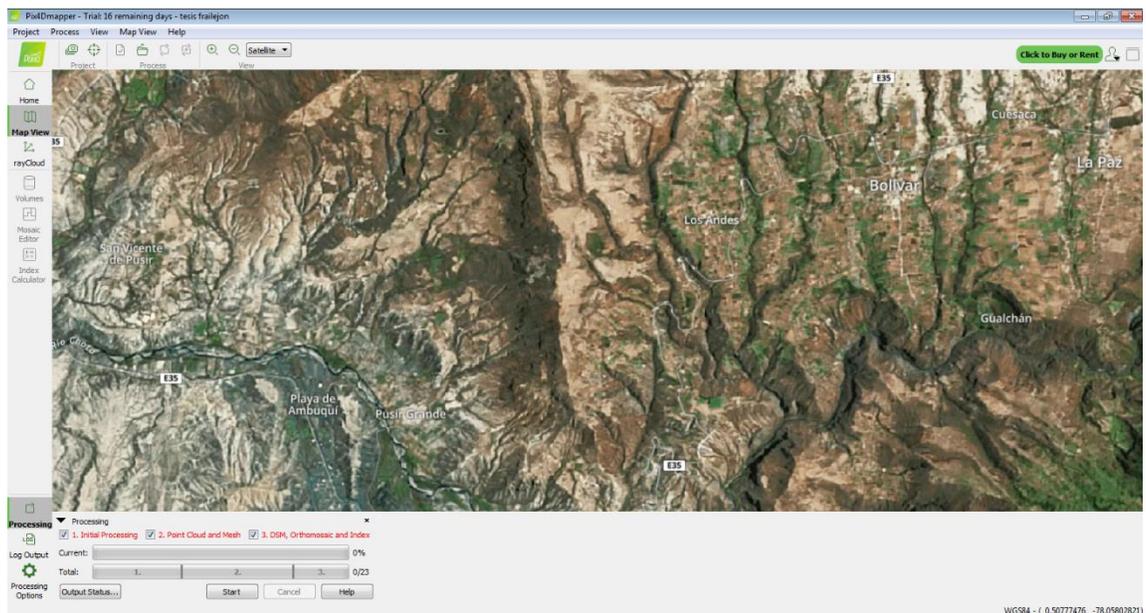


Figura 13. Interfaz de Pix4DMapper
Elaborado por: D. León

Por su parte Agisoft Photo Scan es un programa gratuito. Para el procesamiento de imágenes se utiliza el siguiente procedimiento; primero se debe descargar e instalar el programa. Dentro del programa se deben habilitar tres espacios: el área de trabajo, el de

las imágenes y el de las referencias. Se abre la carpeta en donde estén las imágenes obtenidas del UAV, se seleccionan todas y se arrastran al área de trabajo del procesador. Previamente debe estar habilitada la opción de “cámara”, puesto que permite la identificación de las fotogrametrías por puntos en el área de imágenes y con ello decidir si hay alguna imagen que se desea eliminar.

Se abre la opción de calibración de cámara, en donde los valores están parametrizados. En el área de trabajo se abre la opción de seleccionar sistema de coordenadas para verificar que sea el mismo que se tiene en las coordenadas de los puntos de control. Se procede a abrir el archivo de puntos de control en el área de trabajo. Se activa la opción de puntos de control para que se los pueda visualizar en el área de imágenes. Se guarda el archivo en la misma carpeta de las imágenes fotogramétricas. Se abre la ventana de flujo de trabajo para orientar las imágenes, se activan las opciones, se escoge la precisión máxima, se da clic en aceptar y se espera para que las imágenes se procesen. Inmediatamente se sobrepone la nube de puntos. En la Figura 14 y Figura 15 se visualiza el ortomosaico y la nube de puntos del Parque de las Cuadras, respectivamente.

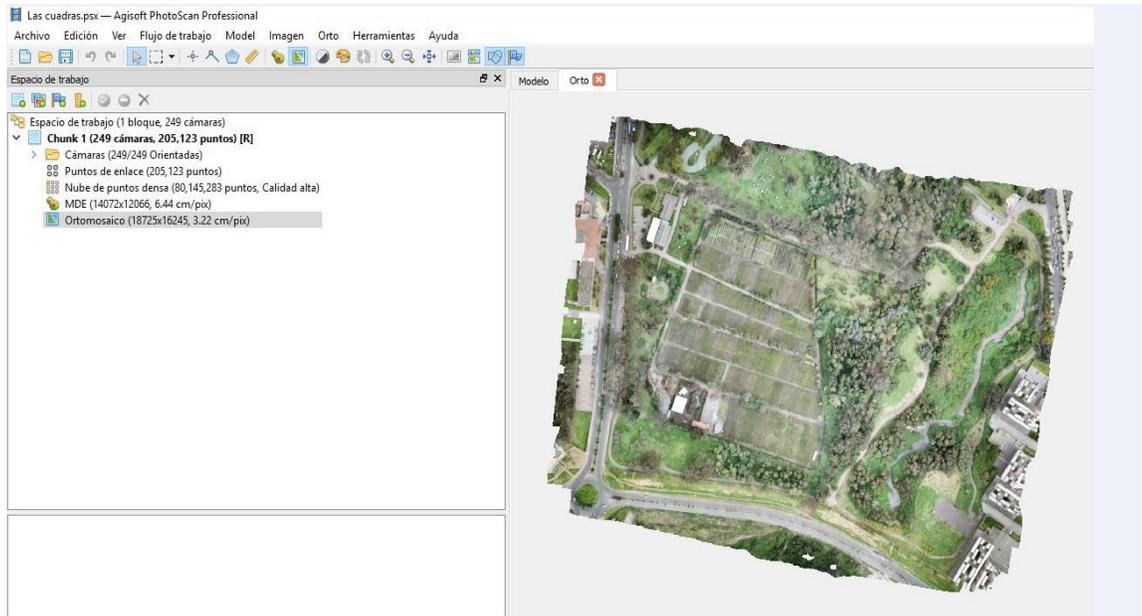


Figura 14. Ortomosaico del Parque de las Cuadras
Elaborado por: D. León

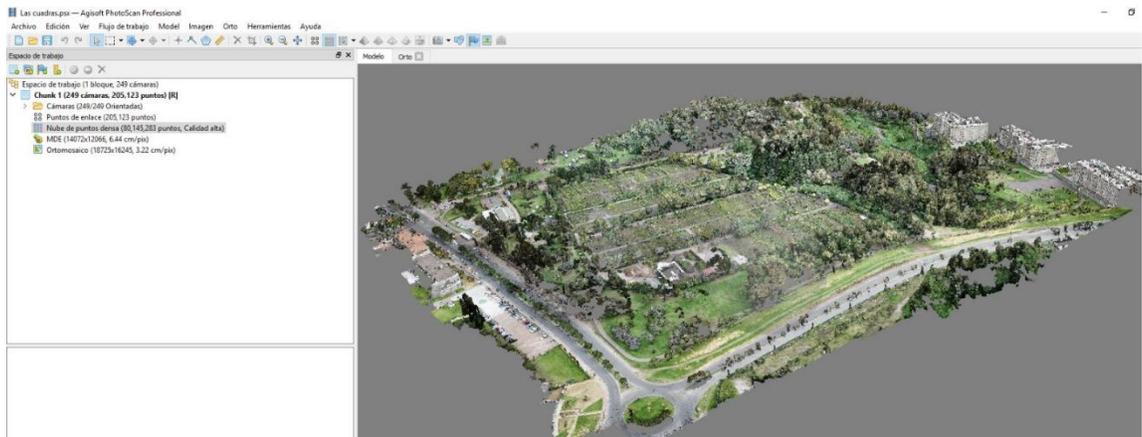


Figura 15. Nube de puntos del Parque de las Cuadras
Elaborado por: D. León

4.1.6. **Conteo manual de especies arbóreas**

Para poder determinar la fiabilidad de la herramienta desarrollada se debe contabilizar el número de especies arbóreas manualmente, para compararlo con el valor que se obtenga de forma automática tras la ejecución del script. Debido al resultado satisfactorio obtenido en las imágenes del área de Páramo del Volcán Chiles, y al resultado parcialmente

satisfactorio en el Parque de las Cuadras, se procede a realizar un conteo manual de los árboles de eucalipto y de frailejón, correspondientemente. Se aplica la metodología de cuadrantes como se observan en las Figuras 16 y 17; se utiliza este método porque permite identificar individuos de una misma especie en un área homogénea.

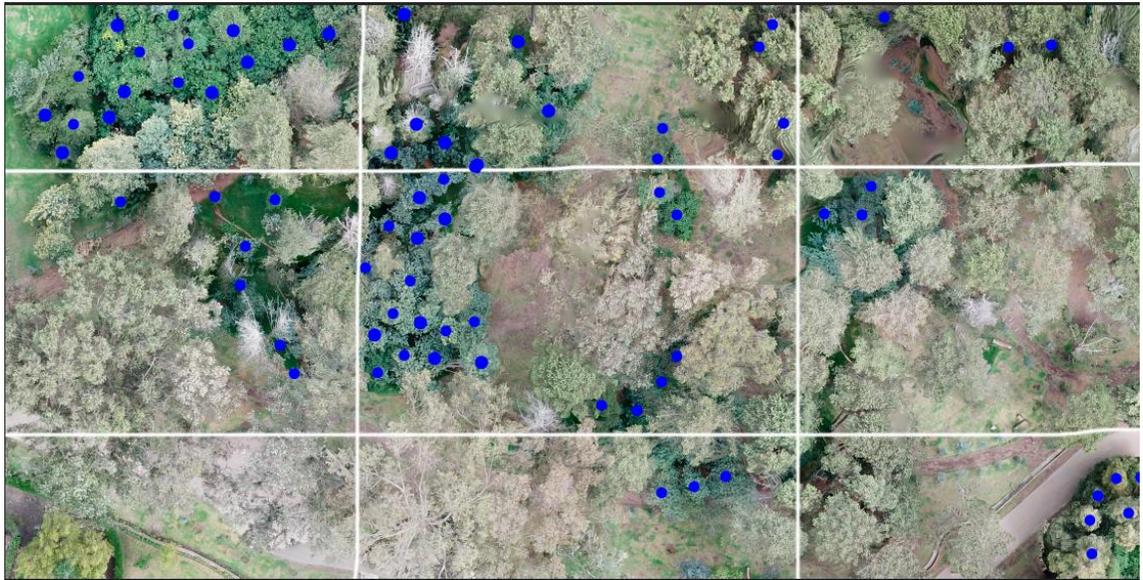


Figura 16. Imagen de conteo manual en el Parque de las Cuadras mediante cuadrantes
Elaborado por: D. León

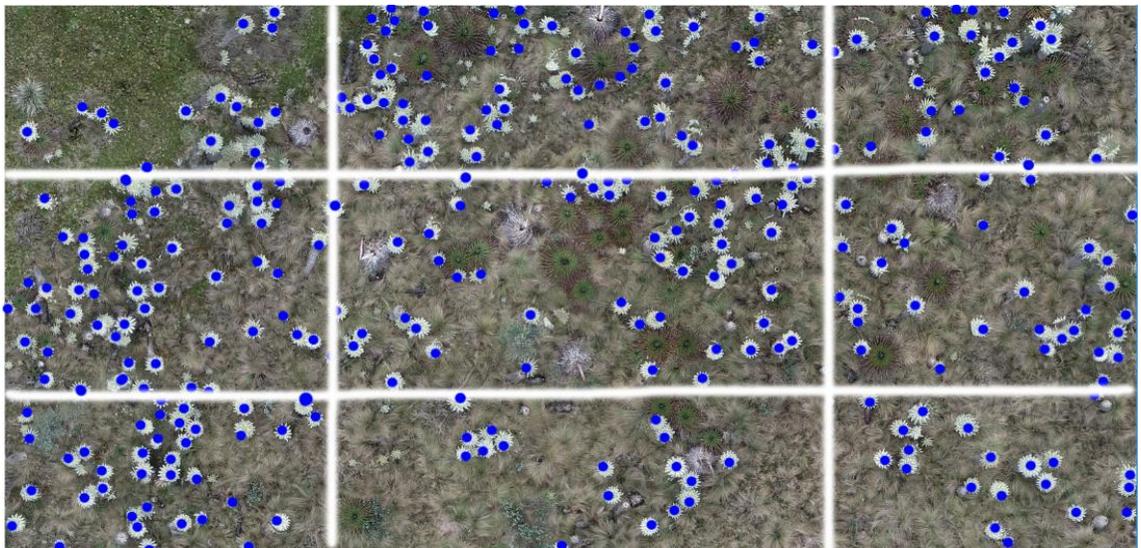


Figura 17. Imagen con cuadrantes para conteo manual en el área de páramo Volcán Chiles.
Elaborado por: D. León

Según el conteo manual de eucaliptos en el Parque de las Cuadras se obtuvo un valor de 67 individuos. En el conteo manual de los frailejones en el área del Volcán Chiles se obtuvo un valor de 316 individuos aproximadamente.

4.1.7. Configuración de requerimientos previos al desarrollo del script

Se ha decidido trabajar con la versión de QGIS 2.18.9 Las Palmas y el lenguaje de programación de Python versión 2.7. Después de su instalación ha sido necesario descargar y vincular, las librerías de GRASS, QGIS API y PyQT4 API. Después de realizar pruebas con diferentes versiones de GRASS, se decide trabajar con la versión más óptima para los fines deseados; ésta es la versión 7.2.1. Se debe descargar la caja de herramientas de Orfeo OTB versión 5.0.0 para después anexarla y configurarla dentro de QGIS (Duarte, Silva , & Teodoro, 2018). Es necesario tomar en cuenta que se requieren utilizar funciones de la librería de GDAL/OGR. Esta librería existe por defecto en QGIS.

También es necesario descargar la versión de QGIS 2.18.27 para acceder al binding TkInter de la biblioteca gráfica Tcl/Tk de Python (Bryce, 2019). Esto se realiza porque la versión antigua de QGIS 2.18.9 no posee esta ligadura con Python. Para que se pueda trabajar con el binding, la carpeta de funciones de TkInter se debe colocar en la misma ruta de ubicación de la carpeta de QGIS 2.18.9, con la finalidad de que el binding trabaje como un espejo en el SIG. En este caso de estudio se utiliza TkInter porque es un módulo gráfico nativo de Python, lo que facilita su utilización para la interfaz gráfica de la herramienta a desarrollar.

Por otro lado, QGIS API y PyQT4 API permiten la vinculación de QGIS con extensiones del lenguaje de Python, para que los códigos de Python puedan ejecutarse

dentro del SIG. Debido a que PyQt4 API no es nativo de Python, no se la utiliza para la creación de la interfaz, se ha limitado su uso para llamar a funciones de manejo de archivos. Es decir, es más sencillo trabajar con el binding de TkInter que con PyQt4 API.

La vinculación de GRASS 7.2.1 con QGIS se utiliza como herramienta de análisis y extracción para datos ráster mediante el algoritmo “i.segment”. Este algoritmo permite realizar una segmentación de entrenamiento y una segmentación definitiva en las imágenes, como se detalla más adelante. En tanto OTB tiene algoritmos como “watershed segmentation” que permiten segmentar y clasificar imágenes RGB, además OTB tiene otras funcionalidades para procesar y separar bandas de colores, y generar ortorectificación (Duarte, Silva , & Teodoro, 2018).

Para configurar la plataforma de QGIS se debe administrar la caja de herramientas OTB, sus herramientas de líneas de órdenes y GRASS. Para configurar la caja de herramientas se debe abrir la opción de procesos y opciones, en donde se despliega una ventana con diferentes enunciados. Se selecciona el enunciado de proveedores, caja de herramientas de ORFEO (análisis de imágenes), carpeta de aplicaciones de OTB en este punto se debe seleccionar la ubicación en donde se haya guardado la descarga de información de la caja de herramientas.

Dentro de esta carpeta están contenidas tres carpetas más, con los nombres de bin, lib y share. Se selecciona la carpeta lib, dentro de ella está una nueva carpeta nombrada otb la cual debe abrirse, y finalmente se selecciona la carpeta de applications y se da clic en “seleccionar esta carpeta”. Para configurar las herramientas de línea de órdenes de OTB se debe seleccionar la ubicación, en donde se haya guardado la información correspondiente. Se abre la carpeta de OTB en donde están contenidas las carpetas de bin,

lib y share. Se selecciona la carpeta bin, se da clic “seleccionar esta carpeta” y finalmente en aceptar. En esta ventana de opciones de procesado se puede comprobar directamente si es que se ha vinculado con GRASS. Para ello se debe escoger el enunciado de proveedores, órdenes de GRASS GIS 7 y verificar que esté activo.

4.1.8. Típo del código

Una vez definidos todos estos requerimientos previos, se procede a escribir el script en la consola conjunta de Python dentro de QGIS o en un editor de código de preferencia del usuario, tomando como guía el script base. A continuación, se describe el script con explicación detallada en cada etapa o función. Con fines explicativos se colocan capturas del código, sin embargo el script redactado completamente se coloca en el ANEXO 4.

Librerías del Sistema

```
import Tkinter
import tkFileDialog
import os
import tkMessageBox
from qgis.core import *
import qgis.utils
from PyQt4.QtCore import QFile, QFileInfo, QVariant
from qgis.analysis import *
import os.path
from processing import Processing, QgsApplication
from qgis.analysis import QgsRasterCalculatorEntry, QgsRasterCalculator
```

Figura 18. Librerías del Sistema
Elaborado por: D. León

En esta etapa se importa el binding de la biblioteca gráfica Tcl/Tk nativa de Python, esta función permite generar una interfaz gráfica capaz de cargar archivos. A su vez se importan variables propias del sistema operativo; sea este Windows, Linux o Mac. También se importa la función para abrir mensajes de confirmación y se llaman a las funciones principales y utilidades del QGIS. Se vincula la API de PyQt4 para el manejo

de archivos desde la interfaz gráfica. Se importa la función del QGIS de análisis de información, la ubicación de archivos y los procesos del GRASS y OTB.

Variables globales del sistema

```
baseLayerName = "baseLayer"  
rgbLayerName = "rgbLayer"  
minsize = 50  
ramSize = 2048  
usrPath = ""  
fileName = ""  
totalBandas = 0  
maxvalClas = 0  
totalPixel = 0
```

Figura 19. Variables globales del sistema
Elaborado por: D. León

Esta etapa establece las variables globales del sistema, es decir variables cuyo valor estarán disponibles para cualquier función del sistema. Todas estas variables son las que inicializan el programa. Se define el nombre con el que la imagen principal o de prueba se va a cargar en el QGIS. En este punto se redefine el valor mínimo entre objetos, conocido como minsize, al cual se le asigna un valor de 50 debido a que las imágenes procesadas; tienen los árboles muy cercanos unos con otros. En cuanto a la memoria de RAM, se ha delimitado un valor de 2048, esto significa que el proceso ocupará aproximadamente 2GB.

Variables de la interfaz gráfica

```
top = Tkinter.Tk()  
top.title('Clasificador de Arboles')  
top.geometry("400x100")  
top.resizable(0, 0)
```

Figura 20. Variables de la interfaz gráfica
Elaborado por: D. León

Se inicializa la ventana gráfica de “Clasificador de árboles”, en donde se ha establecido un área de la pantalla de 40*100 y que permanece inalterable en cuanto a su tamaño.

Definición del botón de la interfaz gráfica

```
B = Tkinter.Button(top, text = "Iniciar conteo" , command = mainFunction)
B.pack(pady = 40)
top.mainloop()
```

Figura 21. Definición del botón de la interfaz gráfica
Elaborado por: D. León

A pesar de que este paso se ubica al final del código se lo debe generar antes de desarrollar las funciones del programa, porque Python lee secuencialmente y es necesario que la función del botón esté previamente definida. Se agrega el botón en la ventana gráfica de la herramienta y se establece que la aplicación se ejecute 100% en la ventana de QGIS.

Descripción de todas las funciones del programa

```
def mainFunction() :
    baseLayer = seleccionarImagen()
    rgb = crearCapaRGB(baseLayer)
    extentRaster = obtenerExtencionCapa(baseLayer)
    Processing.initialize()
    bandas = separarBandas()
    seed = crearSemilla(bandas, extentRaster)
    segmentada = segmentarImagen(bandas, seed, extentRaster)
    classLayer = clasificarImagen(rgb, segmentada)
    copasLayer = identificarCopas(classLayer)
    totalArboles = contarCopas(copasLayer)
    crearArchivoResultados(totalArboles)
```

Figura 22. Descripción de todas las funciones del script
Elaborado por: D. León

Esta parte del código, detalla todas las funciones que se van a realizar. Se usan valores de retorno y se capturan en variables. Se inicia cargando únicamente la imagen de prueba. Esta imagen posee bandas, las cuales se convierten en una sola y se crea la capa RGB. Además se obtienen los píxeles más cercanos a la cámara para usarlos posteriormente. Se obtiene una extensión de la capa principal. Se notifica al QGIS que se van a utilizar procesos “i.segment” y “watershed segmentation” de GRASS y OTB, respectivamente.

Se utiliza la imagen principal para separarla en bandas y se crea la primera imagen para empezar a segmentar.

En base a la primera segmentación, se procede a realizar una segunda segmentación que utiliza los parámetros anteriores de forma recursiva. Se utiliza el algoritmo de inteligencia artificial “kmeans” de OTB para comparar la imagen RGB y la segmentada, con la finalidad de identificar puntos en común, los cuales probablemente podrían ser árboles. Toma en cuenta los puntos más sobresalientes. Después se genera una imagen clasificada que sirve para separar los píxeles y realizar el conteo de árboles. Con el valor de árboles contados se crea un archivo con el resultado obtenido.

Funciones individuales

Selección de imágenes

```
def seleccionarImagen() :
    global fileName
    fileName = tkFileDialog.askopenfilename(initialdir = "/",title = "select file",
    global usrPath
    usrPath = QFileInfo(fileName).path() + '/'
    baseLayer = QgsRasterLayer(fileName, baseLayerName)
    QgsMapLayerRegistry.instance().addMapLayer(baseLayer)
    return baseLayer
```

Figura 23. Función individual de selección de imagen
Elaborado por: D. León

La selección de imagen de prueba es la primera función individual. En ella se delimita que se pueden abrir los archivos tipo ráster (.tif). Se define un path, es decir una ubicación en donde se van a almacenar los archivos resultantes. Es recomendable que estos archivos se guarden en la misma ubicación de la imagen de prueba. En esta función se crea una capa ráster y luego se agrega automáticamente al QGIS. Para la generación de este archivo se utiliza la herramienta QgsRaster propia de QGIS. El archivo resultante de la capa ráster sirve para dar inicio a la siguiente función.

Capa RGB

```
def crearCapaRGB(baseLayer) :
    print("Inicio calculo capa RGB..")
    entries = []
    rasBand4 = QgsRasterCalculatorEntry()
    rasBand4.ref = baseLayerName + '@4'
    rasBand4.raster = baseLayer
    rasBand4.bandNumber = 4
    entries.append( rasBand4 )
    rasBand3 = QgsRasterCalculatorEntry()
    rasBand3.ref = baseLayerName + '@3'
    rasBand3.raster = baseLayer
    rasBand3.bandNumber = 3
    entries.append( rasBand3 )
    rgb = usrPath + 'rgb.tif'
    calc = QgsRasterCalculator( '(' + baseLayerName + '@4' + ' - '
    calc.processCalculation()
    rgbLayer = QgsRasterLayer( rgb, rgbLayerName)
    QgsMapLayerRegistry.instance().addMapLayer( rgbLayer)
    print("Fin calculo capa RGB..")
    return rgb
```

Figura 24. Función individual de Capa RGB
Elaborado por: D. León

Esta parte del código es fundamental para el presente trabajo, porque la aplicabilidad de este script está enfocado en procesar imágenes obtenidas por cámaras fotogramétricas que son de menor definición y resolución espacial que las imágenes obtenidas por cámaras multiespectrales. En esta función se inicia indicando que la calculadora ráster es un objeto tipo entrada, con la cual se trabaja el arreglo de bandas RGB de la imagen de prueba. La herramienta que se utiliza es `QgsRasterCalculatorEntry` propia de QGIS.

Con la herramienta `QgsRasterCalculator` se calcula el número de bandas para la capa RGB. El número de bandas con el que se trabaja al ser una imagen RGB son tres; roja, verde y azul. Al ejecutarse esta función se obtiene una cuarta banda, la cual se agrega como objeto de entrada y sirve de base para la generación de la capa RGB. Esta función del código difiere notablemente del script base, porque en él no existe y se utiliza directamente la función de extensión de capa para el cálculo del NDVI y el NIR.

Extensión capa

```
def obtenerExtencionCapa(baseLayer) :
    global totalBandas
    totalBandas = baseLayer.bandCount() # numero total de bandas
    xPixel = baseLayer.width()
    yPixel = baseLayer.height()
    global totalPixel
    totalPixel = xPixel*yPixel
    minx = baseLayer.extent().xMinimum()
    maxx = baseLayer.extent().xMaximum()
    miny = baseLayer.extent().yMinimum()
    maxy = baseLayer.extent().yMaximum()
    return str(minx) + ',' + str(maxx) + ',' + str(miny) + ',' + str(maxy)
```

Figura 25. Función individual Extensión capa
Elaborado por: D. León

En esta función se obtiene el número total de bandas y se crea una cuadrícula de la capa base mediante el total de píxeles que se calcula con los valores de ancho y alto de los píxeles. Además se utilizan los valores mínimos y máximos del ancho y alto.

Separar bandas

```
def separarBandas() :
    print("Inicio separacion bandas..")
    split = usrPath + 'split.tif'
    Processing.runAlgorithm("otb:splitimage", None, fileName, ramSize, split)
    bandas=[usrPath + 'split_' + str(nband) + '.tif' for nband in range(totalBandas - 1)]
    print("Fin separacion bandas..")
    return bandas
```

Figura 26. Función individual de separacion de bandas
Elaborado por: D. León

Para la separación de bandas se utiliza la herramienta Processing.runAlgorithm de QGIS, la cual llama a la función SplitImage del OTB. Esta función segrega dinámicamente los nombres de los archivos de las bandas. Su funcionamiento se basa en una secuencia repetitiva de un valor a otro. A esta característica se la conoce como bucle repetitivo, para lo cual se utiliza un for, para que el procesamiento se realice por cada banda; para este caso en específico es 3 veces. Cabe mencionar que esta función es

importante porque permite almacenar la información de cada banda por separado, es decir que se obtienen 3 archivos resultantes, cada uno de ellos correspondientes a una capa.

Crear semilla

```
def crearSemilla(bandas, extentRaster) :
    print("Inicio creacion semilla..")
    seed = usrPath + 'semilla.tif'
    # Imagen inicial con puntos semilla
    Processing.runAlgorithm("grass7:i.segment", None, bandas, 0.05, 0, 0, minsize, ramSize, 20, NoI
    print("Fin creacion semilla..")
```

Figura 27. Función individual crear semilla
Elaborado por: D. León

En esta función se entrena a la herramienta para segmentar la imagen. Para el entrenamiento se utiliza un valor inicial de opacidad de 0.05. El rango de opacidad es de 0 a 1, considerando a 0 como blanco y 1 como negro; es decir totalmente opaco. Se utiliza la herramienta Processing.runAlgorithm de QGIS para llamar a la función “i.segment” de GRASS. Se genera un archivo de salida de primera segmentación.

Segmentar imagen

```
def segmentarImagen(bandas, seed, extentRaster) :
    print("Inicio segmentacion recursiva..")
    segmentada = usrPath + 'segmentada.tif'
    for thresh in [0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65]:
        Processing.runAlgorithm("grass7:i.segment", None, bandas, thresh, 0, 0, minsize, ramSi
        seed = segmentada
    QgsMapLayerRegistry.instance().addMapLayer( QgsRasterLayer(segmentada, "seedTest"))
    print("Fin segmentacion recursiva..")
    return segmentada
```

Figura 28. Función Individual segmentar imagen
Elaborado por: D. León

Esta función utiliza la primera segmentación “semilla” como base para generar las siguientes segmentaciones. Es una función recursiva porque realiza varias segmentaciones. Para ello fue necesario utilizar un for, porque es un bucle repetitivo. El rango de valores de opacidad se prueba desde 0.1 a 0.65, no se utiliza el rango de 0 a 1 porque genera errores. Se utiliza la herramienta Processing.runAlgorithm de QGIS para

llamar a la función “i.segment” de GRASS. Se genera un archivo resultante de segmentación.

Clasificar imagen

```
def clasificarImagen(rgb, segmentada) :
    print("Inicio clasificacion..")
    trainingSet = int(totalPixel*0.00705)
    clasificada = usrPath + 'clasificada.tif'
    Processing.runAlgorithm("otb:unsupervisedkmeansimageclassification", None, rgb,
        ramsize, segmentada, trainingSet, 5, 1000, 0.0001, clasificada, None)
    print("Fin clasificacion..")
    classLayer = QgsRasterLayer(clasificada, 'class_rc')
    QgsMapLayerRegistry.instance().addMapLayer(classLayer)
    global maxValClas
    maxValClas = classLayer.dataProvider().bandStatistics(1, QgsRasterBandStats.All, classLayer)
    return classLayer
```

Figura 29. Función individual clasificar imagen
Elaborado por: D. León

Para la clasificación en la imagen se trabaja con los archivos RGB y el de segmentación. Se utiliza la herramienta Processing.runAlgorithm de QGIS para llamar al algoritmo de OTB “unsupervised kmeans image classification”. Este algoritmo permite trabajar con un archivo de clasificación existente (imagen de prueba) y con una aplicación de entrenamiento (segmentación). Se genera un archivo con la clasificación de árboles.

Identificar copas

```
def identificarCopas(classLayer) :
    print("Inicio extraccion copas..")
    extraida = usrPath + 'extraida.tif'
    Processing.runAlgorithm("gdalgr:rastercalculator", None, classLayer, 1, None, None, None, None, None)
    # conversion de imagen clasificada en fichero vectorial
    copas = usrPath + 'copas.shp' # output shapefile
    Processing.runAlgorithm("gdalgr:polygonize", None, extraida, 'DN', copas)
    layer_ = QgsVectorLayer(copas, "copasLayer", "ogr")
```

Figura 30. Función individual Identificar copas
Elaborado por: D. León

Para la identificación de copas se utiliza la herramienta Processing.runAlgorithm de QGIS para llamar a las herramientas de la librería GDAL/OGR; “raster calculator” y “polygonize” para que se puedan señalar los árboles.

Conteo de copas

```
def contarCopas(layer) :
    # Conteo copas
    print("Inicio conteo de copas..")
    if layer.dataProvider().capabilities() & QgsVectorDataProvider.AddAttributes:
        remover = layer.dataProvider().deleteAttributes([0])
        adicionar = layer.dataProvider().addAttributes([QgsField("Arboles-ID", QVariant.Int)])
    layer.updateFields()
    totalArboles = 0 # Contador para el total de arboles
    centroides = {}
    # Se realiza el conteo automatico de los arboles obtenidos
    with edit(layer):
        for f in layer.getFeatures():
            area = f.geometry().area() # area
            perimetro = f.geometry().length() # perimetro
            centroid_x = f.geometry().boundingBox().center().x() # coordenada x del centroide
            centroid_y = f.geometry().boundingBox().center().y() # coordenada y del centroide
            # Dicionario con los centroides
            centroides[int(f.id())] = QgsPoint(centroid_x, centroid_y)
            # Asignar id al arbol
            f[0] = f.id()
            layer.updateFeature(f)
            totalArboles += 1
    print("Fin conteo de copas..")
    return totalArboles
```

Figura 31. Función individual conteo de copas
Elaborado por: D. León

Para la función de conteo de árboles se utiliza la herramienta de Qgs Vector Data Provider del QGIS para determinar los centroides de los polígonos y poder contabilizarlos.

Archivo resultante

```
def crearArchivoResultados(totalArboles):
    print("Inicio creacion archivo resultados..")
    info = usrPath + 'resultados.txt'
    archivo = open(info, 'w') # archivo de resultados
    # Se escriben los resultados en el archivo
    line = '\n-----#####-----\n\n'
    archivo.write(line)
    line = 'Numero total de arboles = ' + str(totalArboles) + '\n'
    archivo.write(line)
    archivo.close()
    print("Fin creacion archivo resultados..")
```

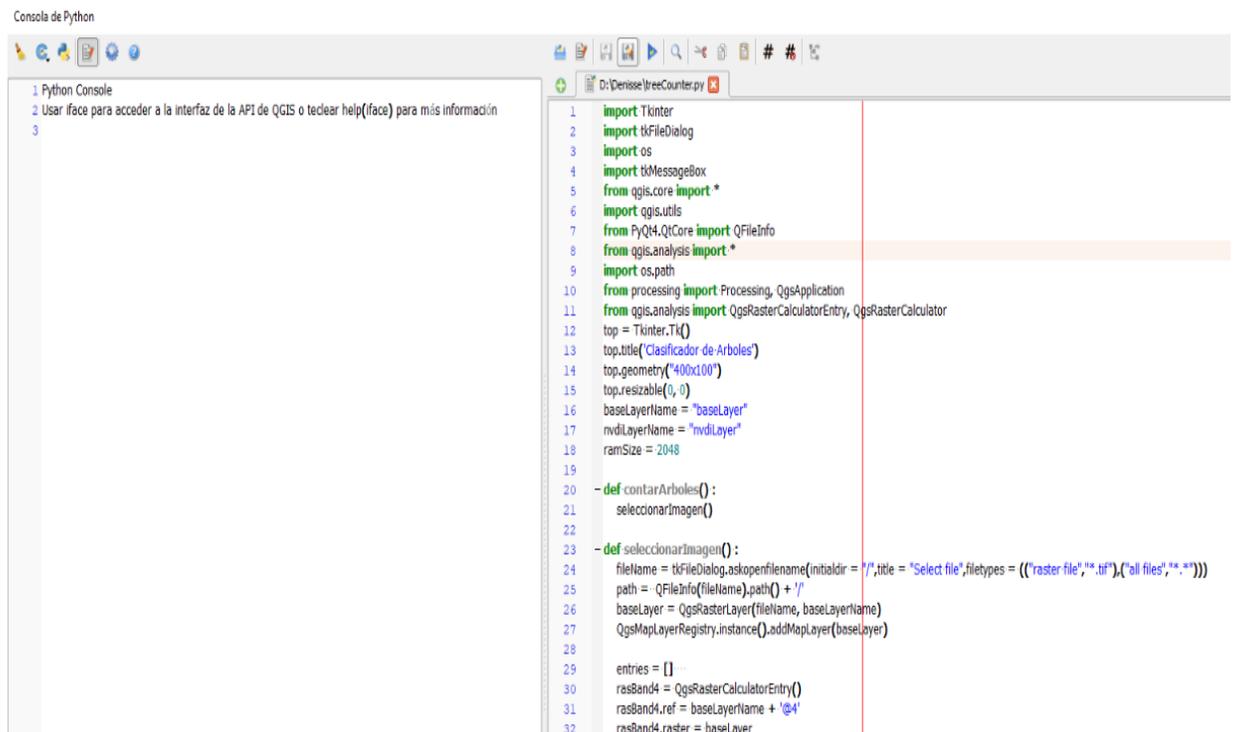
Figura 32. Función individual archivo resultante
Elaborado por: D. León

En esta función se crea un archivo con el resultado obtenido en la contabilización de especies arbóreas.

4.1.9. Ejecución del script

Para la verificación de la fiabilidad de la herramienta se la debe ejecutar por lo menos tres veces por cada imagen de prueba, con la finalidad de obtener varios resultados que permitan realizar un cálculo estadístico.

A continuación, se presenta de forma gráfica la ejecución de la herramienta. En la Figura 33 se visualiza el script ejecutado en la consola de Python.



```
Python Console
1 Python Console
2 Usar iface para acceder a la interfaz de la API de QGIS o teclear help(iface) para más información
3

D:\Denisse\treeCounter.py
1 import Tkinter
2 import tkFileDialog
3 import os
4 import tkMessageBox
5 from qgis.core import *
6 import qgis.utils
7 from PyQt4.QtCore import QFileInfo
8 from qgis.analysis import *
9 import os.path
10 from processing import Processing, QgsApplication
11 from qgis.analysis import QgsRasterCalculatorEntry, QgsRasterCalculator
12 top = Tkinter.Tk()
13 top.title("Clasificador de Arboles")
14 top.geometry("400x100")
15 top.resizable(0, 0)
16 baseLayerName = "baseLayer"
17 rvdilayerName = "rvdilayer"
18 ramSize = 2048
19
20 -def contarArboles():
21     seleccionarImagen()
22
23 -def seleccionarImagen():
24     fileName = tkFileDialog.askopenfilename(initialdir = "/", title = "Select file", filetypes = (("raster file", "*.tif"), ("all files", "*.*")))
25     path = QFileInfo(fileName).path() + '/'
26     baseLayer = QgsRasterLayer(fileName, baseLayerName)
27     QgsMapLayerRegistry.instance().addMapLayer(baseLayer)
28
29     entries = []
30     resBand4 = QgsRasterCalculatorEntry()
31     resBand4.ref = baseLayerName + '@4'
32     resBand4.raster = baseLayer
```

Figura 33. Generación de script en consola de Python.
Elaborado por: D. León

Al ejecutarse el código, aparece una pantalla con el nombre de “Clasificador de árboles”, a partir del cual se puede dar inicio al conteo de los mismos. Es decir, el script se ha generado para permitir trabajar al usuario, a través de una interfaz gráfica como se observa en la Figura 34.



Figura 34. Interfaz gráfica del script.
Elaborado por: D. León

Se procede a dar clic en el botón de “Iniciar conteo” en donde se abre una pantalla de los documentos existentes en la computadora, se debe seleccionar la imagen con la cual se va a realiza la prueba. Es necesario tomar en cuenta que se debe crear una carpeta que contenga la imagen de prueba, puesto que dentro de esta carpeta se almacenan los resultados. Observe las Figuras 35 y 36.

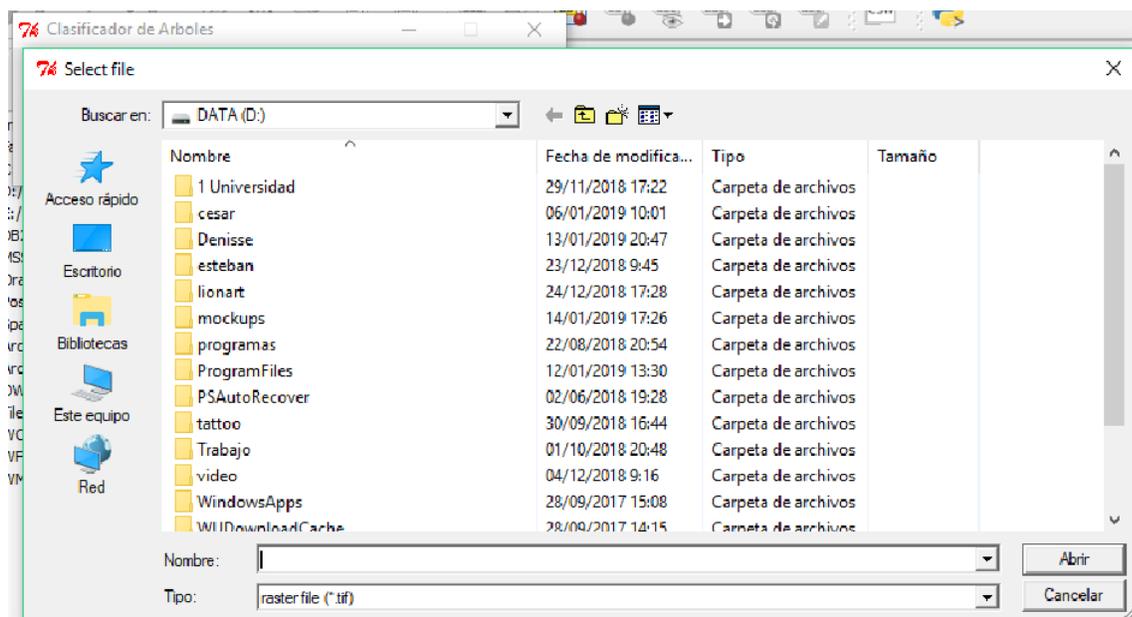


Figura 35. Seleccionador de archivos.
Elaborado por: D. León

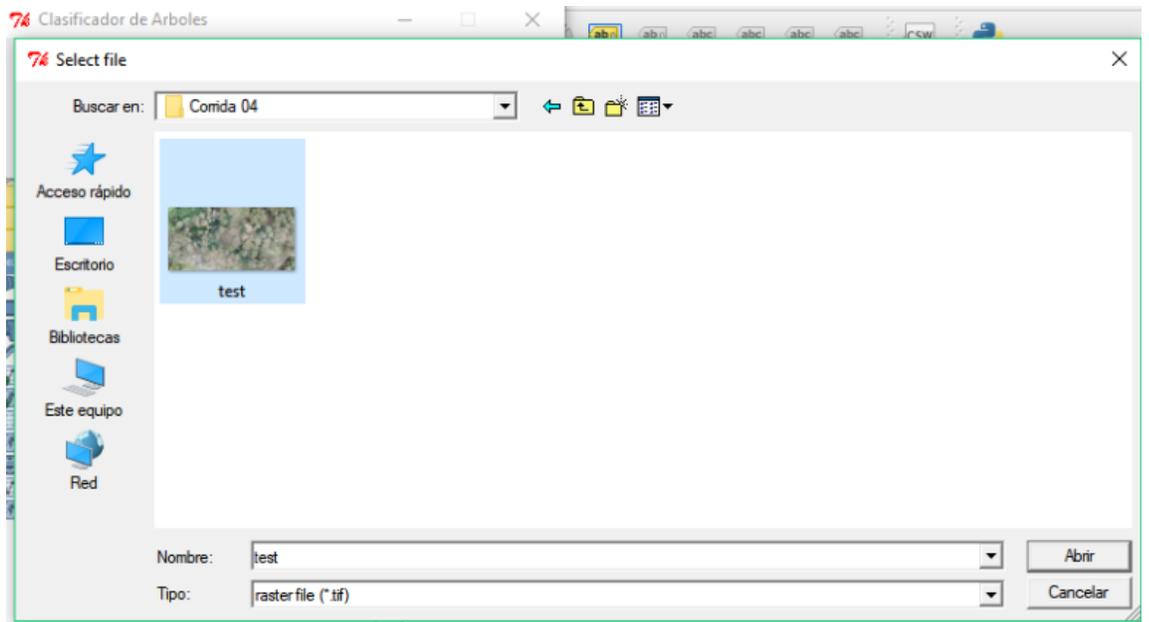
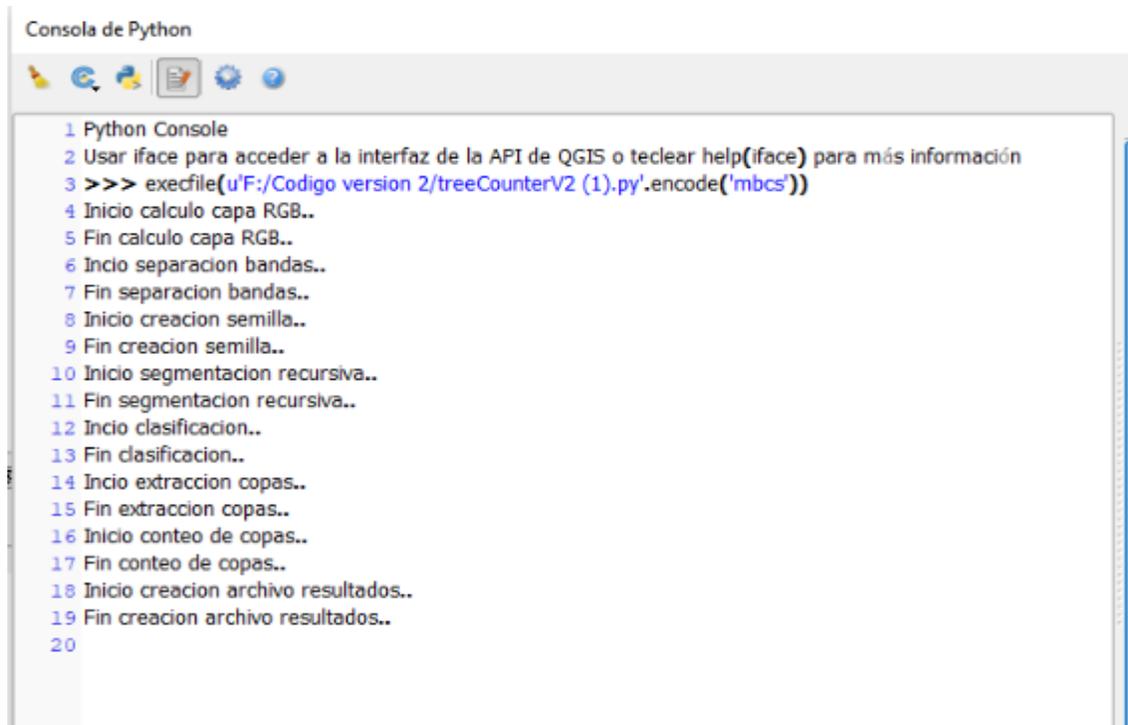


Figura 36. Seleccionador de archivo en carpeta de prueba.
Elaborado por: D. León

A partir de este momento, el script trabaja automáticamente. La herramienta genera un aviso de las tareas que se están ejecutando en la consola de Python. Se visualiza como está en la Figura 37.



```
1 Python Console
2 Usar iface para acceder a la interfaz de la API de QGIS o teclear help(iface) para más información
3 >>> execfile(u'F:/Codigo version 2/treeCounterV2 (1).py'.encode('mbcs'))
4 Inicio calculo capa RGB..
5 Fin calculo capa RGB..
6 Inicio separacion bandas..
7 Fin separacion bandas..
8 Inicio creacion semilla..
9 Fin creacion semilla..
10 Inicio segmentacion recursiva..
11 Fin segmentacion recursiva..
12 Inicio clasificacion..
13 Fin clasificacion..
14 Inicio extraccion copas..
15 Fin extraccion copas..
16 Inicio conteo de copas..
17 Fin conteo de copas..
18 Inicio creacion archivo resultados..
19 Fin creacion archivo resultados..
20
```

Figura 37. Avisos de ejecución del algoritmo.
Elaborado por: D. León

En ocasiones el script aparentemente tiene un fallo en el momento de la ejecución, puesto que pareciera que no está trabajando, incluso el QGIS llega a ralentizarse y a pararse en su totalidad; sin embargo sí está funcionando, por lo cual se recomienda dejar que la aplicación se continúe ejecutando por debajo del programa, se podría visualizar como en la Figura 38.

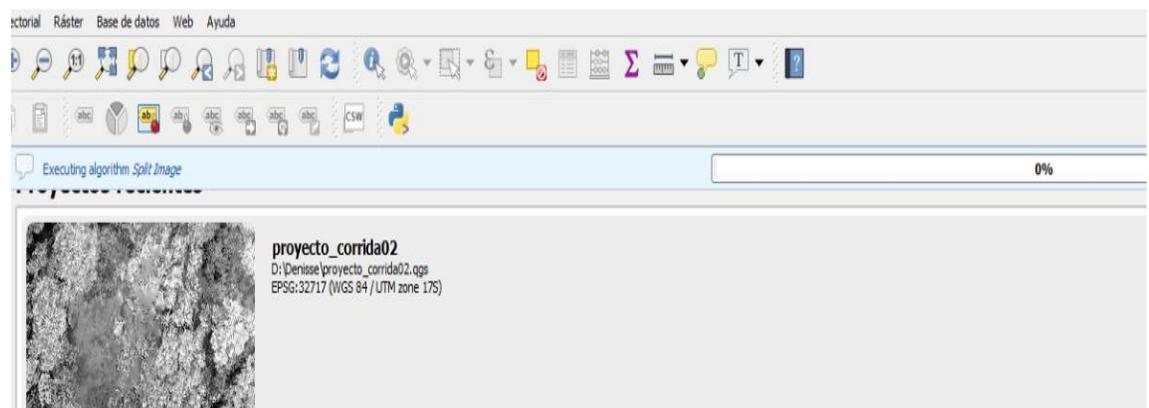


Figura 38. Ejecución del algoritmo.
Elaborado por: D. León

Para comprobar que se está ejecutando correctamente, se puede abrir el administrador de tareas mediante el comando de teclas (Ctrl+shift+Esc) y observar el porcentaje de uso por el CPU. La pantalla que se visualiza es como se muestra en la Figura 39. El tiempo aproximado de análisis y generación de resultados es de treinta minutos a 1 hora dependiendo de la computadora con la que se trabaje.

The screenshot shows the Windows Task Manager Performance tab. The 'Procesos' tab is selected, and the 'Rendimiento' sub-tab is active. The table below shows the resource usage for the 'qgis-bin (5)' process.

Nombre	Estado	CPU	Memoria	Disco	Red	GPU	Mot
> qgis-bin (5)		18,1%	666,8 MB	0 MB/s	0 Mbps	0%	

Figura 39. Comprobación de Ejecución del algoritmo.
Elaborado por: D. León

Al finalizar la ejecución de la herramienta, se guardan todos los archivos resultantes en la carpeta en donde se encuentra la imagen de prueba, de la siguiente forma:

Nombre	Fecha de modifica...	Tipo
clasificada	03/02/2019 22:38	Archivo TIF
clasificada.tif.aux	03/02/2019 22:47	Documento XM
copas.dbf	03/02/2019 22:47	Archivo DBF
copas.prj	03/02/2019 22:38	Archivo PRJ
copas	03/02/2019 22:47	Recurso de form
copas	03/02/2019 22:47	Forma compila
corrida 01.qgs	03/02/2019 22:47	Archivo QGS
extraida	03/02/2019 22:38	Archivo TIF
extraida.tif.aux	03/02/2019 22:48	Documento XM
resultados	03/02/2019 22:42	Documento de
rgb	03/02/2019 22:15	Archivo TIF
rgb.tif.aux	03/02/2019 22:47	Documento XM
segmentada.tfw	03/02/2019 22:36	Archivo TFW
segmentada	03/02/2019 22:36	Archivo TIF
segmentada.tif.aux	03/02/2019 22:48	Documento XM
semilla.tfw	03/02/2019 22:19	Archivo TFW
semilla	03/02/2019 22:19	Archivo TIF
semilla.tif.aux	03/02/2019 22:19	Documento XM
split_0	03/02/2019 22:15	Archivo TIF
split_0.tif.aux	03/02/2019 22:48	Documento XM
split_1	03/02/2019 22:15	Archivo TIF
split_1.tif.aux	03/02/2019 22:48	Documento XM
split_2	03/02/2019 22:15	Archivo TIF
split_2.tif.aux	03/02/2019 22:48	Documento XM
split_3	03/02/2019 22:15	Archivo TIF
test	03/01/2019 14:01	Archivo TIF
treeCounterV2 (1)	03/02/2019 22:12	Archivo PY

Figura 40. Documentos resultantes.
Elaborado por: D. León

Finalmente, se abren los archivos formato TIF y el Shapefile de “copas” en el QGIS, para una correcta visualización de la información analizada. Para ejemplificación del orden en cómo se debe abrir los archivos; se muestran los archivos resultantes del procesamiento de la imagen del área del Volcán Chiles.

- Test: Corresponde a la imagen fotogramétrica de prueba.



Figura 41. Imagen de prueba área de páramo del Volcán Chiles
Elaborado por: D. León

- RGB: Es el archivo resultante con las bandas identificadas por capas.

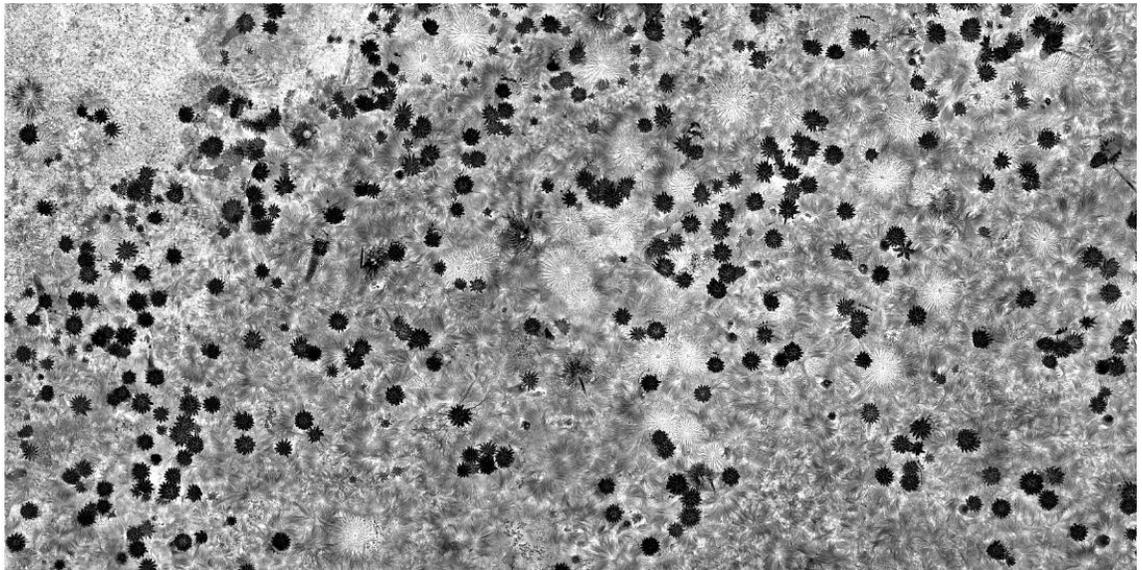


Figura 42. Capa RGB
Elaborado por: D. León

- Split_0: Corresponde a la banda 0

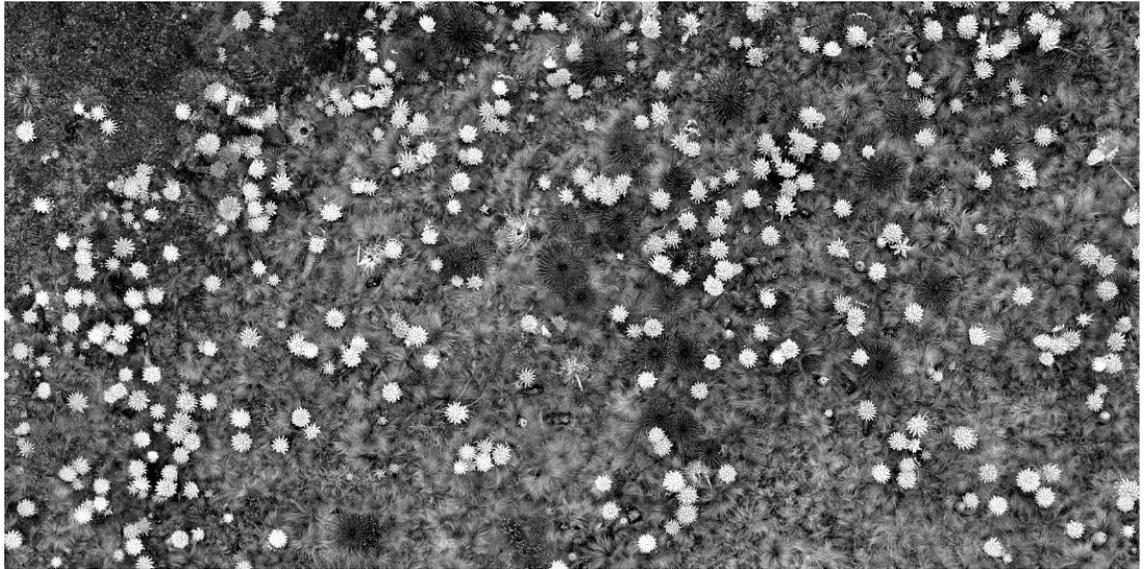


Figura 43. Multibanda 0
Elaborado por: D. León

- Split_1: Corresponde a la banda 1

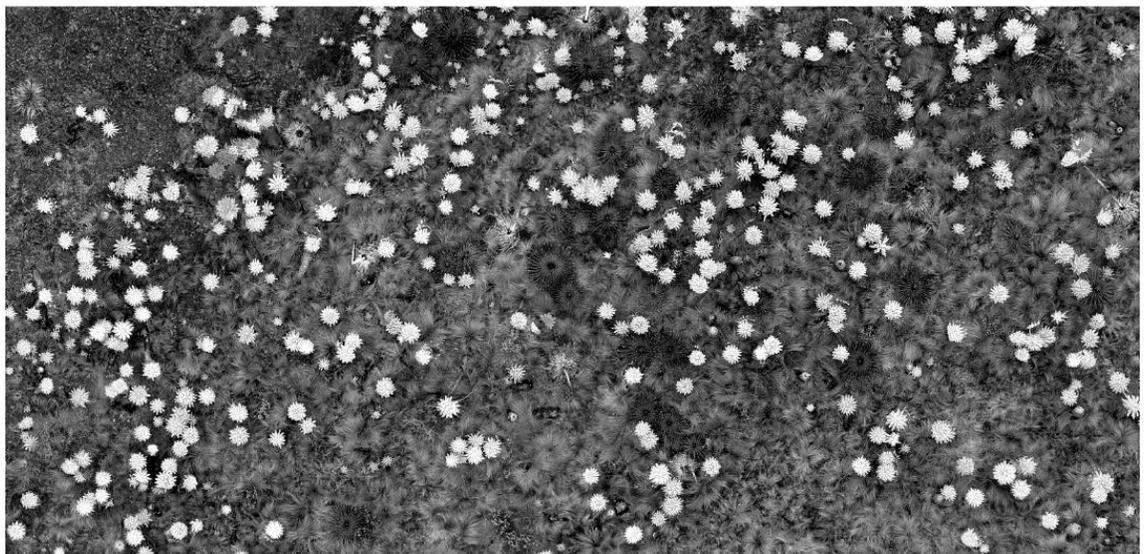


Figura 44. Multibanda 1
Elaborado por: D. León

- Split_2: Corresponde a la banda 2

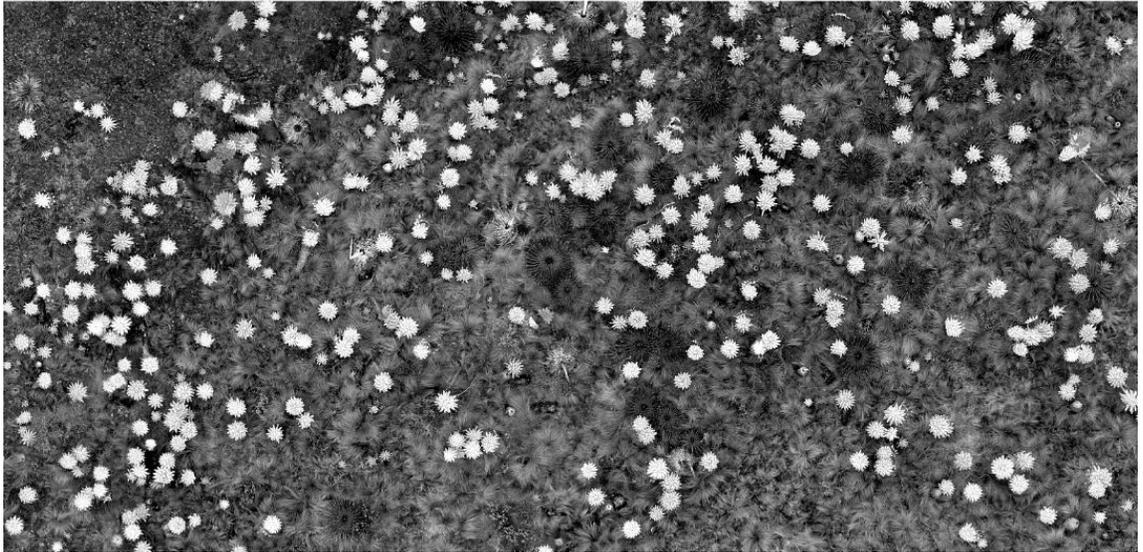


Figura 45. Multibanda 2
Elaborado por: D. León

- Split_3: Corresponde a la banda 3

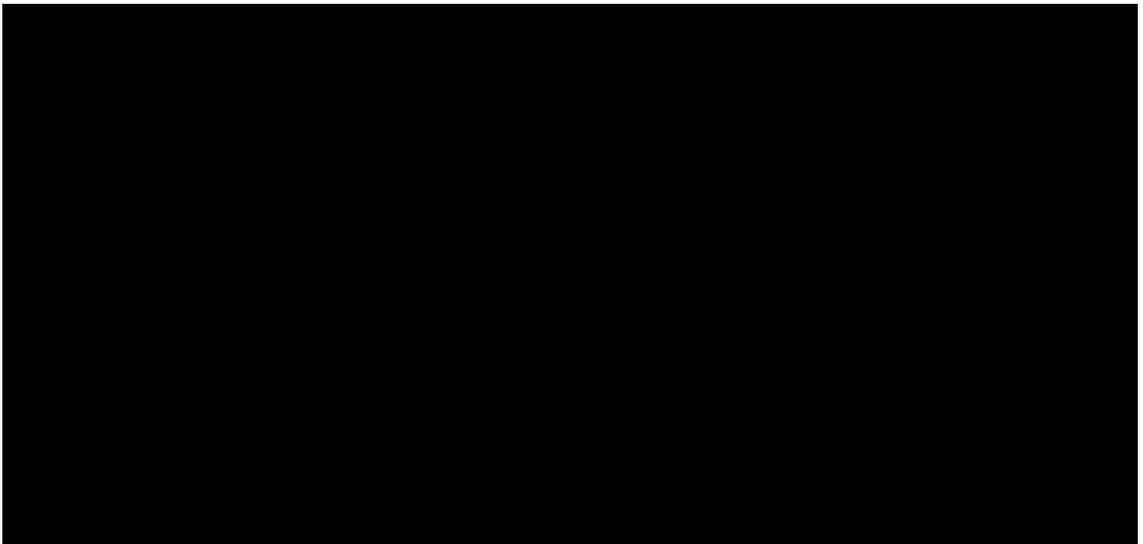


Figura 46. Multibanda 3
Elaborado por: D. León

- Semillas: Es el archivo resultante de la primera prueba de segmentación



Figura 47. Semillas
Elaborado por: D. León

- Segmentada: Es el archivo resultante de las segmentaciones recursivas realizadas con identificación de objetos.



Figura 48. Segmentación
Elaborado por: D. León

- Extraída: Es el archivo que posee el reconocimiento e identificación de los posibles árboles.

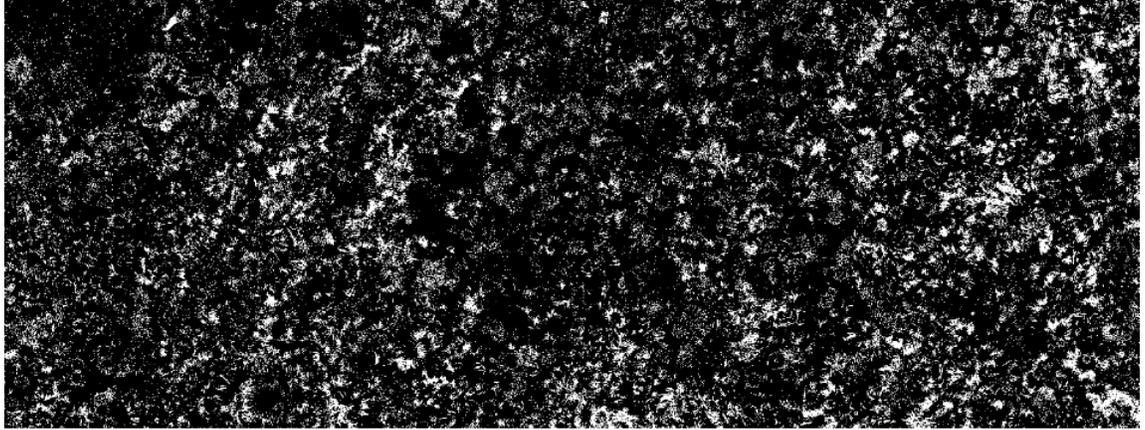


Figura 49. Extracción
Elaborado por: D. León

- Clasificada: Es el archivo de la clasificación de árboles

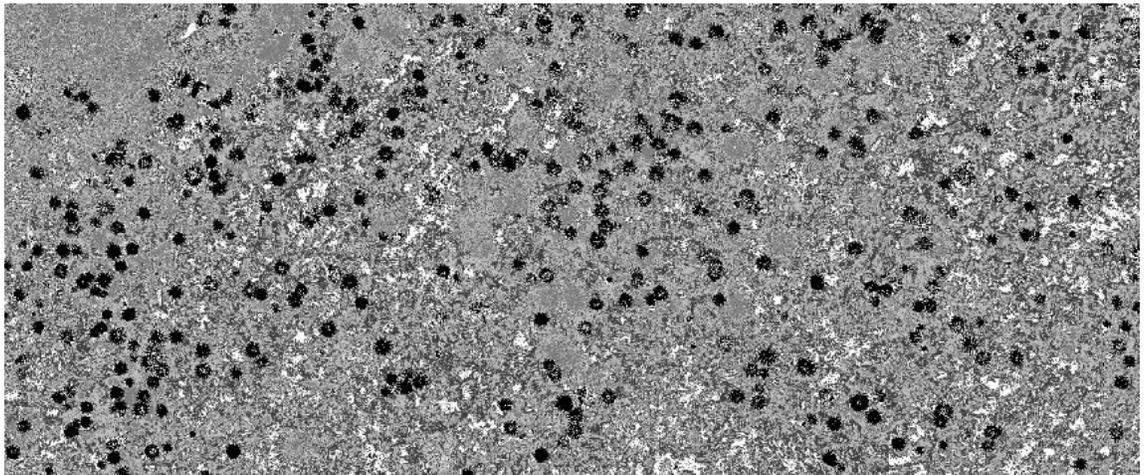


Figura 50. Clasificada
Elaborado por: D. León

- Copas: Es el archivo que indica mediante polígonos la ubicación de los árboles.

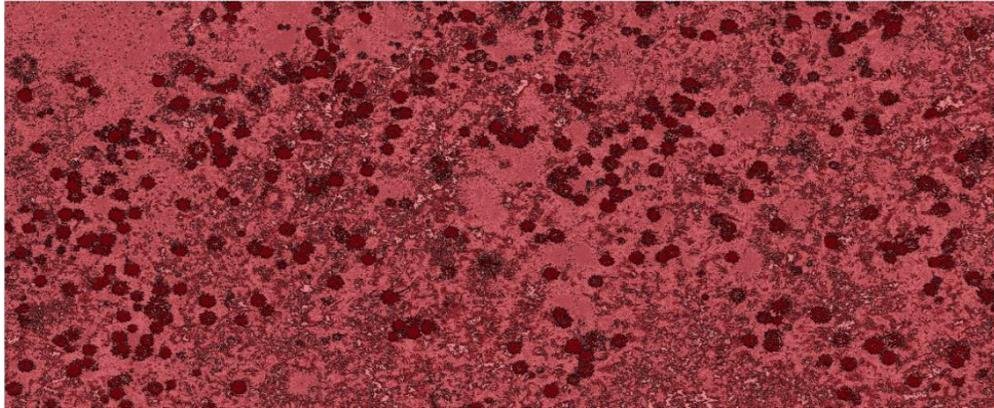


Figura 51. Identificación de copas
Elaborado por: D. León

Adicionalmente se genera un archivo de bloc de notas resultante con el número total de especies arbóreas contadas, se visualiza como está en la Figura 52.

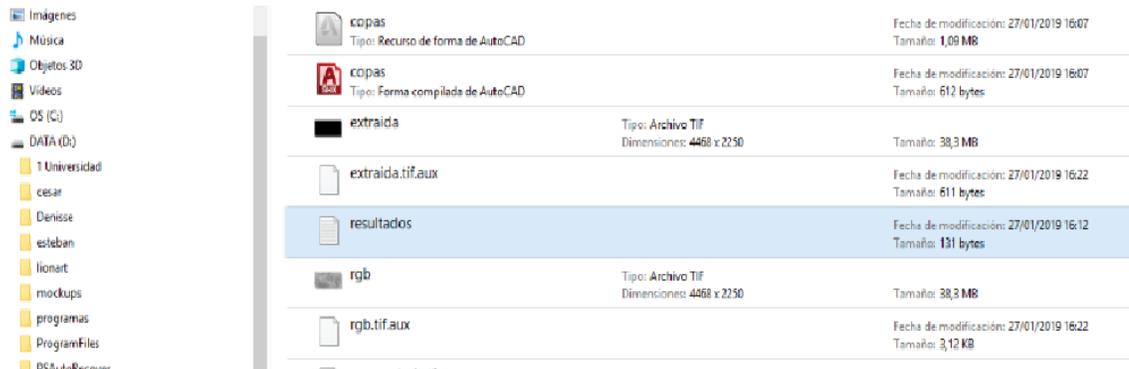


Figura 52. Documento de resultados.
Elaborado por: D. León

4.2. Población y muestra:

Las poblaciones de especies arbóreas con las que se ha decidido trabajar son diferentes para cada área de estudio. Para el Parque de las Cuadras se ha escogido al eucalipto y una extensión de terreno de 16 hectáreas. En la UPS no se ha determinado una especie arbórea en específico. Y para el área del volcán Chiles se ha escogido el frailejón y una extensión de terreno de 2 hectáreas.

4.3. Consideraciones éticas:

Para la realización de toma de imágenes aéreas en el Parque de las Cuadras, fue necesario pedir un permiso escrito tanto al administrador del parque como al encargado de parques y zonas recreacionales del Distrito Metropolitano de Quito.

5. RESULTADOS Y DISCUSIÓN

5.1. Resultados específicos

5.1.1. Resultados Parque de las Cuadras

Para todas las áreas de estudio se realizaron tres ejecuciones del script para cada imagen de prueba. A continuación, se muestra paso a paso el procesamiento generado por el script en la imagen del Parque de las Cuadras, con figuras de la imagen de prueba, archivo final de las copas y el resultado final contrapuesto con la imagen de prueba. Para observar los resultados del procesamiento completo de forma gráfica ver ANEXO 5.

1. Test: Figura 53



Figura 53. Imagen de Prueba Parque de las Cuadras.
Elaborado por: D. León

2. RGB
3. Split_0
4. Split_1
5. Split_2
6. Split_3

7. Semilla
8. Segmentada
9. Extraída
10. Clasificada
11. Copas: Figura 54



Figura 54. Imagen con identificación de eucaliptos Parque de las Cuadras.
Elaborado por: D. León

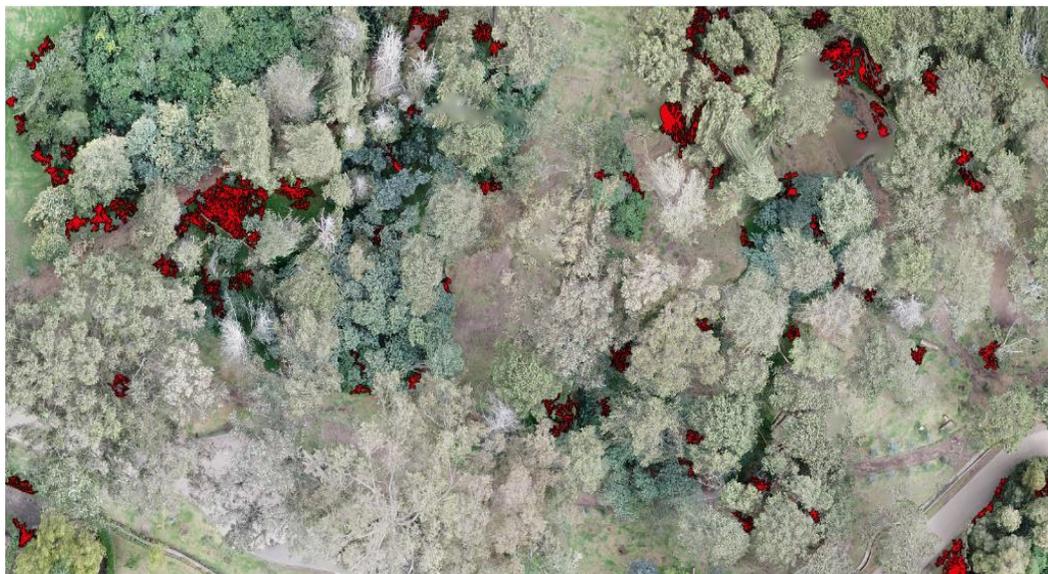


Figura 55. Resultado final contrapuesto con la imagen de prueba
Elaborado por: D. León

También se generó un archivo en bloc de notas con el resultado del conteo de especies arbóreas. En las siguientes figuras se observan los resultados obtenidos:

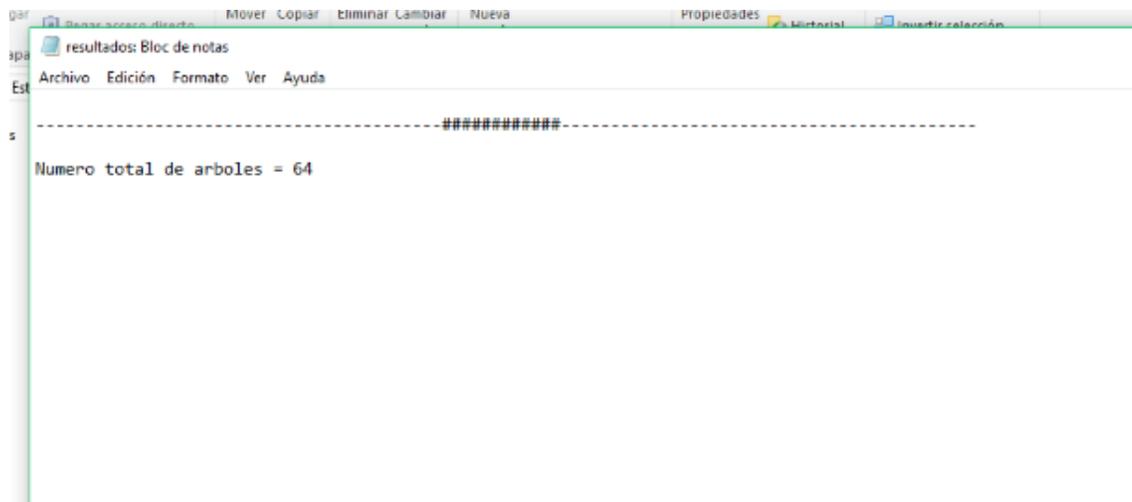


Figura 56. Resultado del primer conteo automático del Parque de las Cuadras.
Elaborado por: D. León

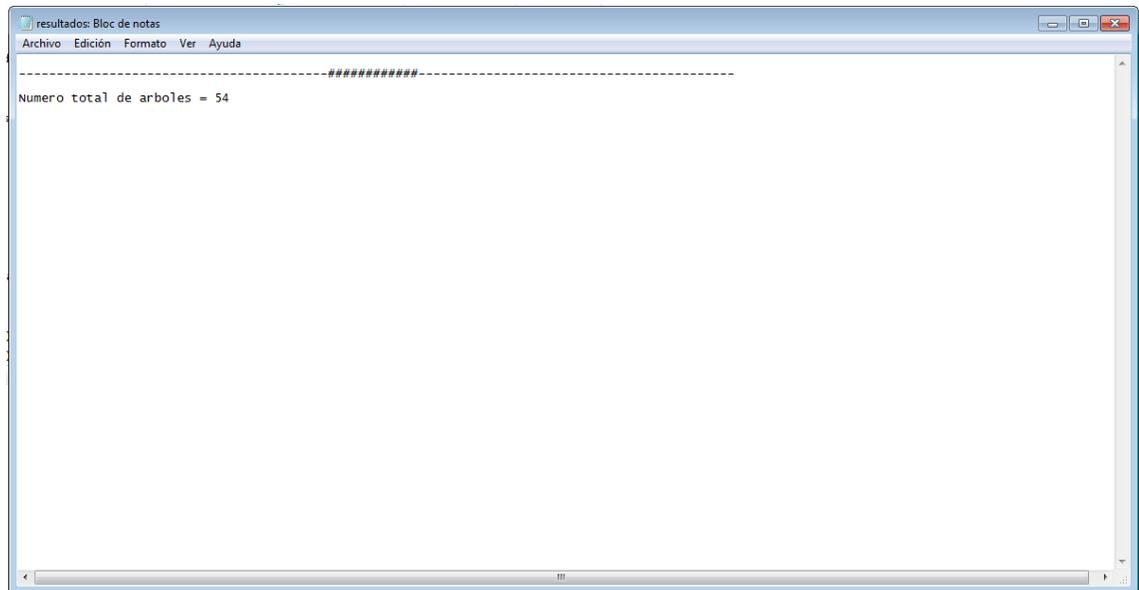


Figura 57. Resultado del segundo conteo automático del Parque de las Cuadras

Elaborado por: D. León

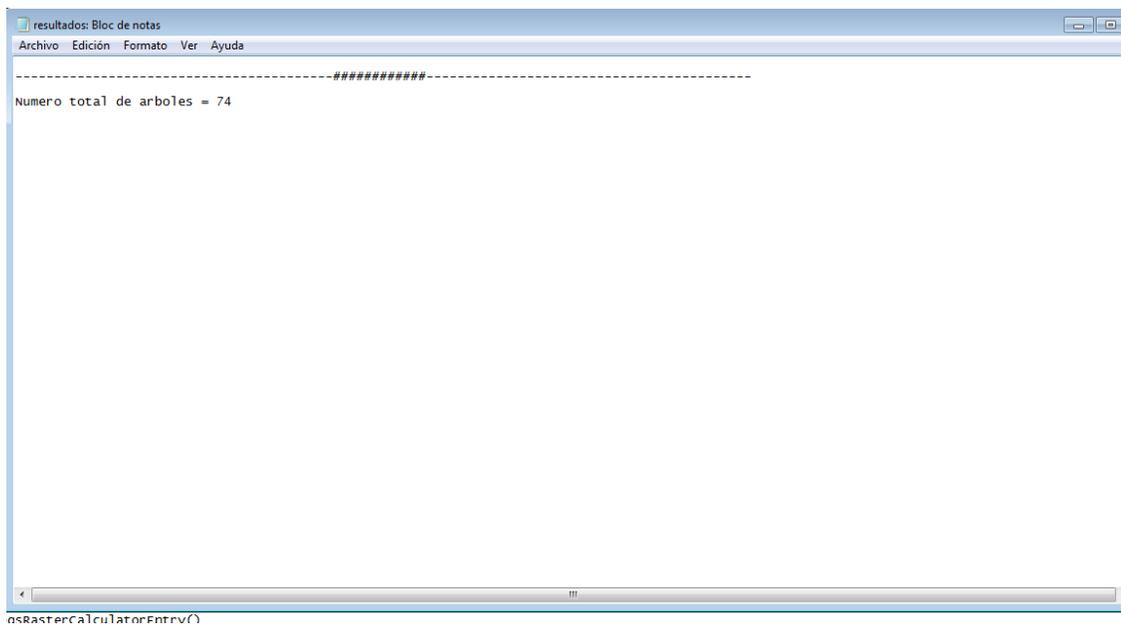


Figura 58. Resultado del tercer conteo automático del Parque de las Cuadras

Elaborado por: D. León

Para comparar los valores obtenidos se ha generado la siguiente tabla:

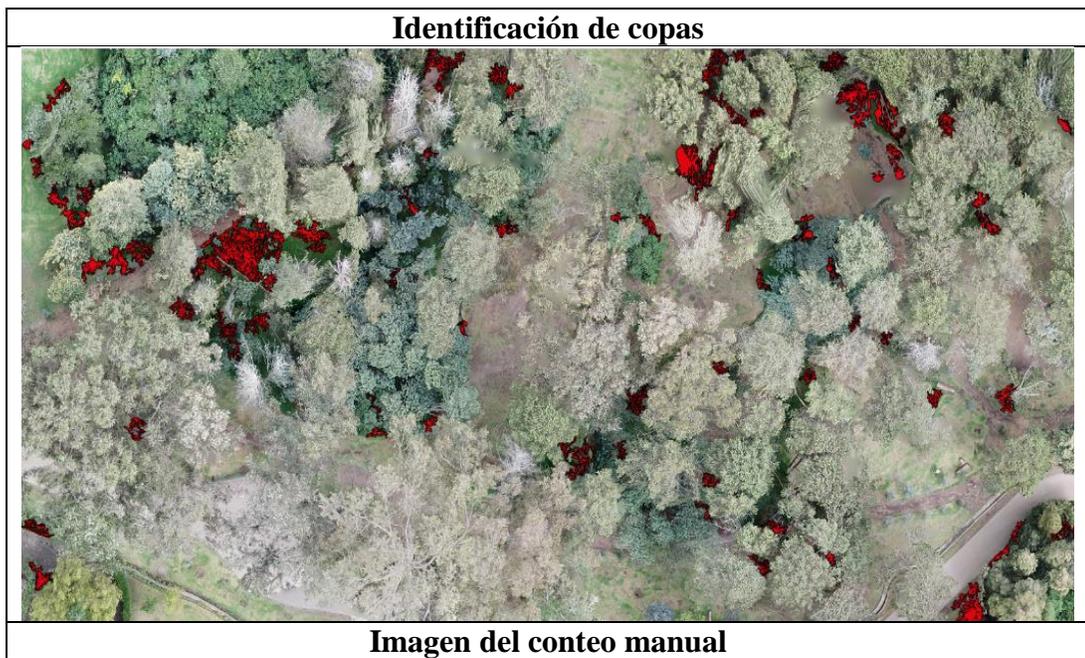
Tabla 4. *Resultados del Parque de las Cuadras*

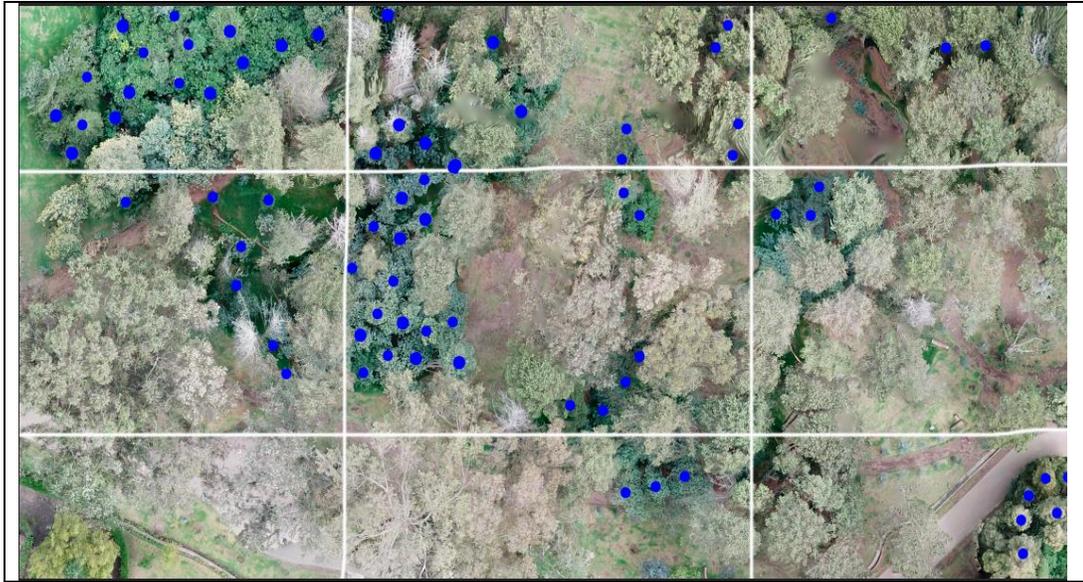
Número de toma de muestras	Conteo Manual	Conteo por herramienta automatizada
1	67	64
2		54
3		74

Elaborado por: D León

Para apreciar la similitud sobre el conteo de las copas y por ende corroborar la efectividad de la herramienta, se colocan de forma contigua la imagen obtenida de la identificación de copas automáticamente y la imagen del conteo manual.

Tabla 5. *Comparación de conteo automático y manual Parque de las Cuadras*





Elaborado por: D. León

A pesar de que los valores calculados por la herramienta son similares a los obtenidos de forma manual, se deben eliminar aquellas áreas que no concuerdan. Es evidente que son varias áreas que la herramienta identifica erróneamente, por lo que los resultados obtenidos son falsos positivos. Esto sucede porque la distancia entre los árboles es muy pequeña y existe solapamiento entre las copas de los mismos.

El siguiente paso a considerar, sería realizar un cálculo del error de la herramienta versus el conteo manual, sin embargo por los resultados obtenidos en la comparación de las imágenes resultantes, en donde no coinciden los puntos, se puede afirmar que existe una identificación de objetos mas no una correcta identificación y conteo de eucaliptos. Es decir, que la herramienta funciona parcialmente y tiene un alto error en la interpretación de datos.

5.1.2. Resultados UPS

El procesamiento de la imagen de la Universidad Politécnica Salesiana, presentó inconvenientes puesto que no logra identificar correctamente a los árboles, de hecho confunde otro tipo de vegetación como si fuesen árboles, en este caso específicamente sucede con el césped. Debido a esta situación, la herramienta no puede continuar el procesamiento después de haber procesado la imagen RGB. A pesar de ello, se adjuntan las imágenes procesadas y el resultado obtenido.

1. Test: Figura 59



Figura 59. Imagen de Prueba de la UPS.
Elaborado por: D. León

2. RGB: Figura 60

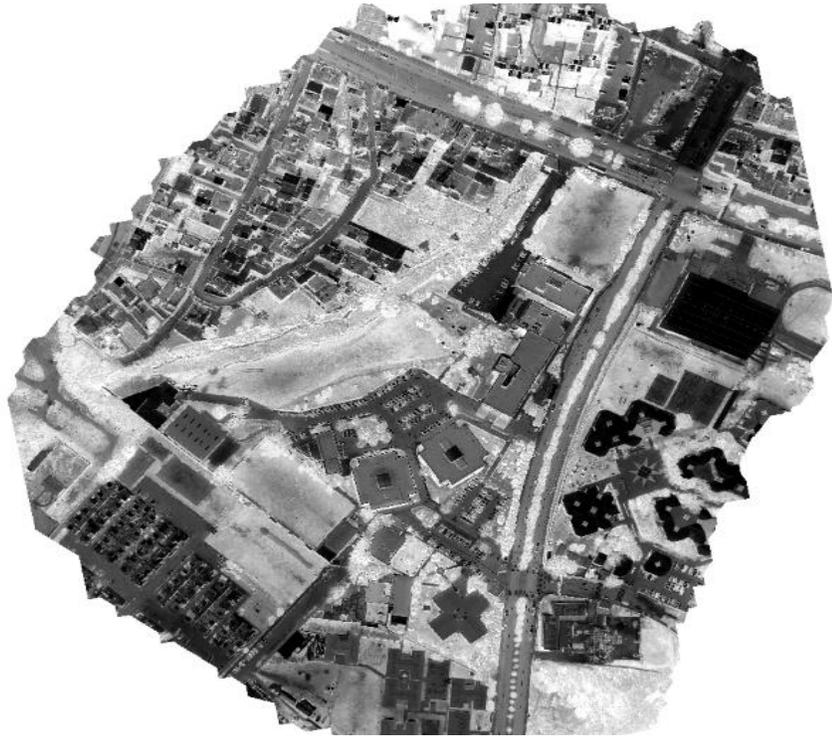


Figura 60. Imagen RGB procesada de la UPS.

Elaborado por: D. León

5.1.3. Resultados área del Volcán Chiles

Finalmente, los resultados obtenidos en la imagen procesada del área de páramo del Volcán Chiles, permiten una correcta identificación de los individuos de frailejones porque poseen cierto patrón en su posición y distanciamiento con otros, además de que no existe solapamiento entre ellos. Un patrón ideal entre los árboles sería el siguiente:

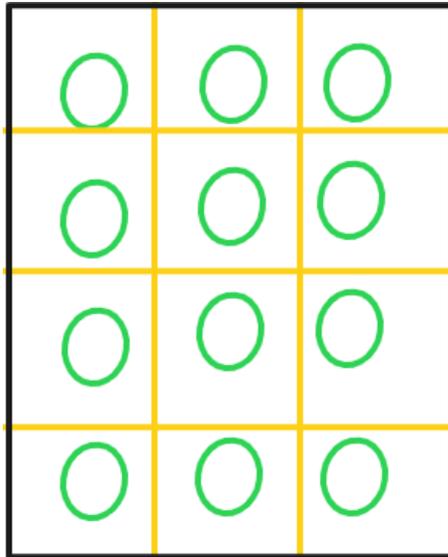


Figura 61. Patrón de distribución y posicionamiento
Elaborado por: D. León

Es decir, que en el área de estudio los árboles sean equidistantes entre sí, se posicionen secuencialmente, que el tamaño de las copas sea similar y no exista sobreposición de un árbol con otro.

A continuación, se indica el procesamiento de la imagen paso a paso, detallando las imágenes de prueba y la imagen resultante. Para una mejor visualización del proceso ver ANEXO 6. Cabe mencionar que la imagen inicial de esta área no permitía una correcta visualización del frailejón por lo que se procedió a trabajar con una parte de la misma.

1. Test completa: Figura 62



Figura 62. Imagen completa de Prueba área de páramo Volcán Chiles.
Elaborado por: D. León

2. Test seccionada: Figura 63



Figura 63. Imagen seccionada de Prueba área de páramo Volcán Chiles.
Elaborado por: D. León

3. RGB
4. Split_0
5. Split_1
6. Split_2
7. Split_3
8. Semilla
9. Segmentada
10. Extraída
11. Clasificada
12. Identificación de frailejones
13. Copas de frailejones: Figura 64

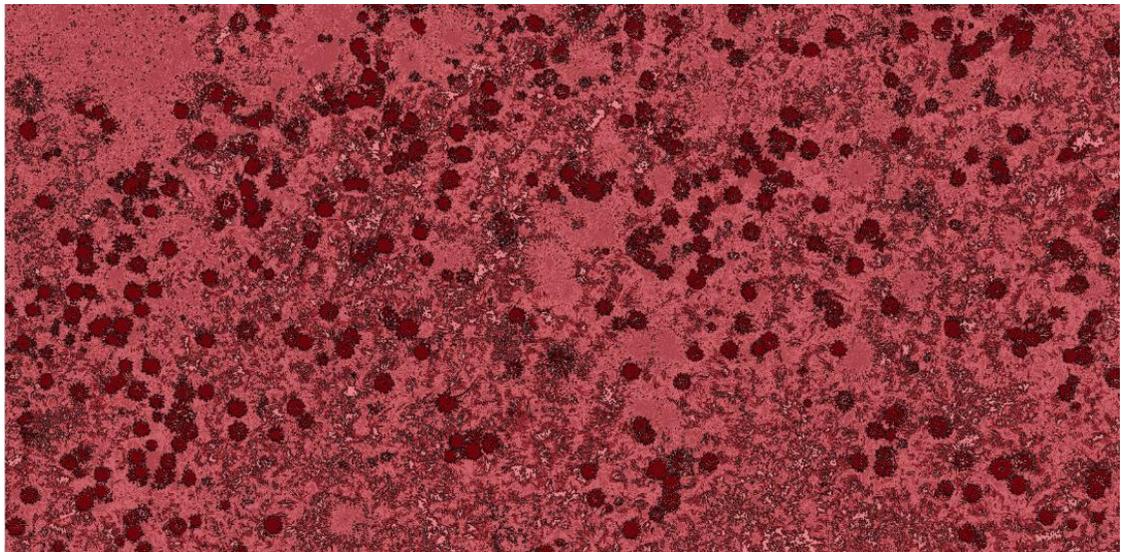


Figura 64. Imagen de copas de frailejones área de páramo Volcán Chiles.
Elaborado por: D. León

Los resultados del conteo automático de los frailejones se muestran en las siguientes

Figuras.

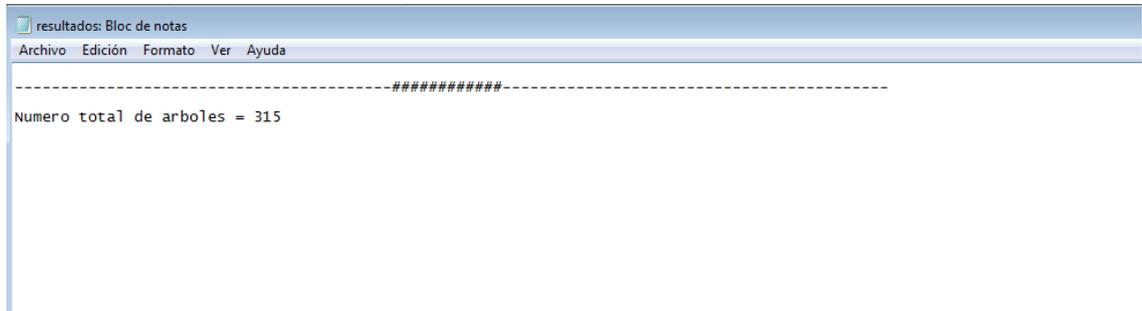


Figura 65. Resultado del primer conteo automático en el área de páramo del Volcán Chiles.
Elaborado por: D. León

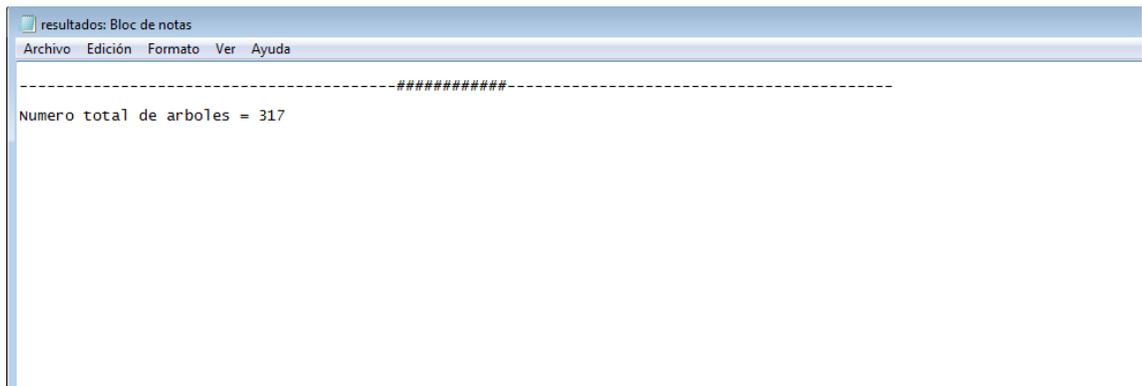


Figura 66. Resultado del segundo conteo automático en el área de páramo del Volcán Chiles.
Elaborado por: D. León

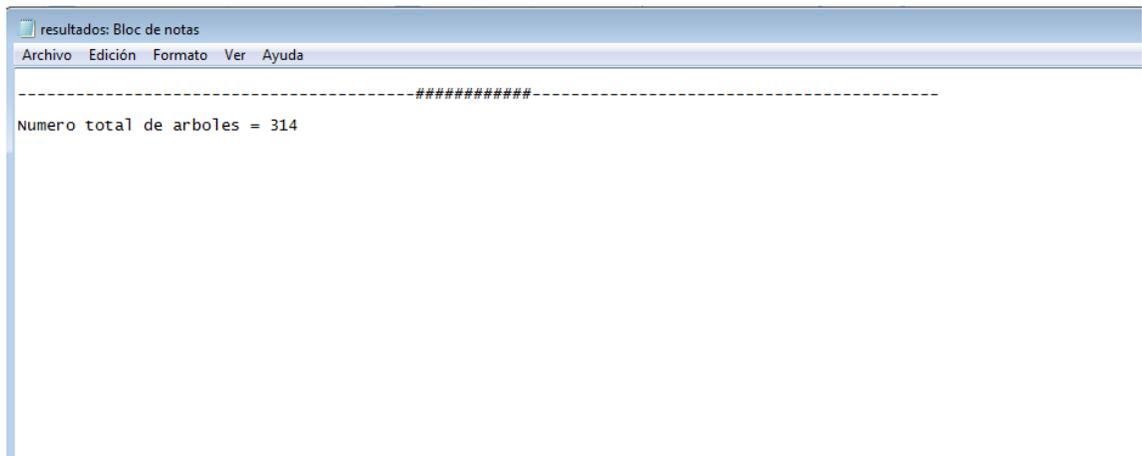


Figura 67. Resultado del tercer conteo automático en el área de páramo del Volcán Chiles.
Elaborado por: D. León

Para una mejor visualización de los resultados se ha creado la siguiente tabla:

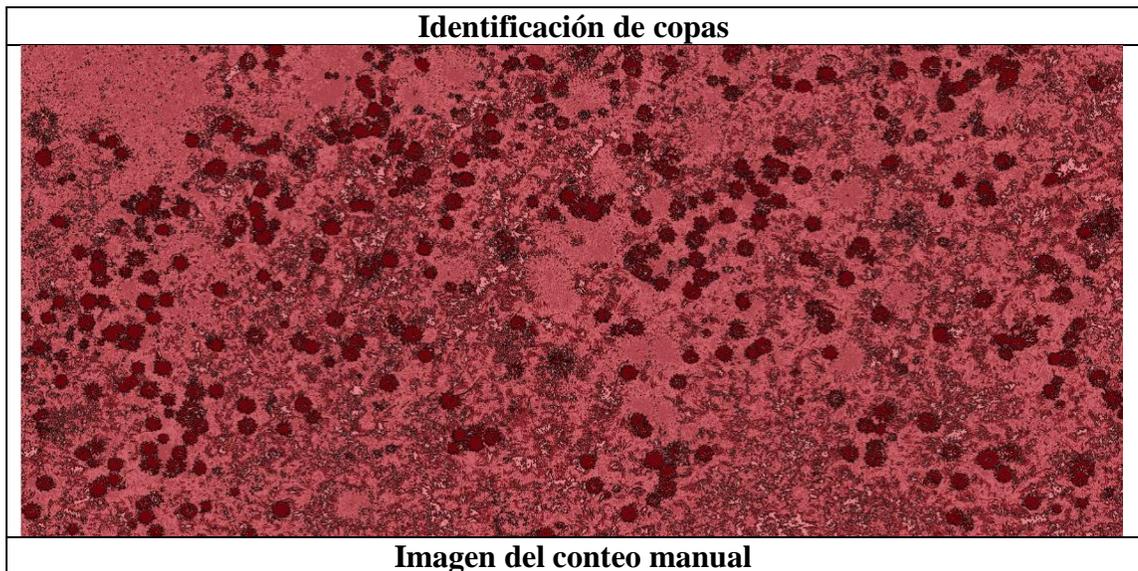
Tabla 6. *Resultados del Área del Volcán Chiles*

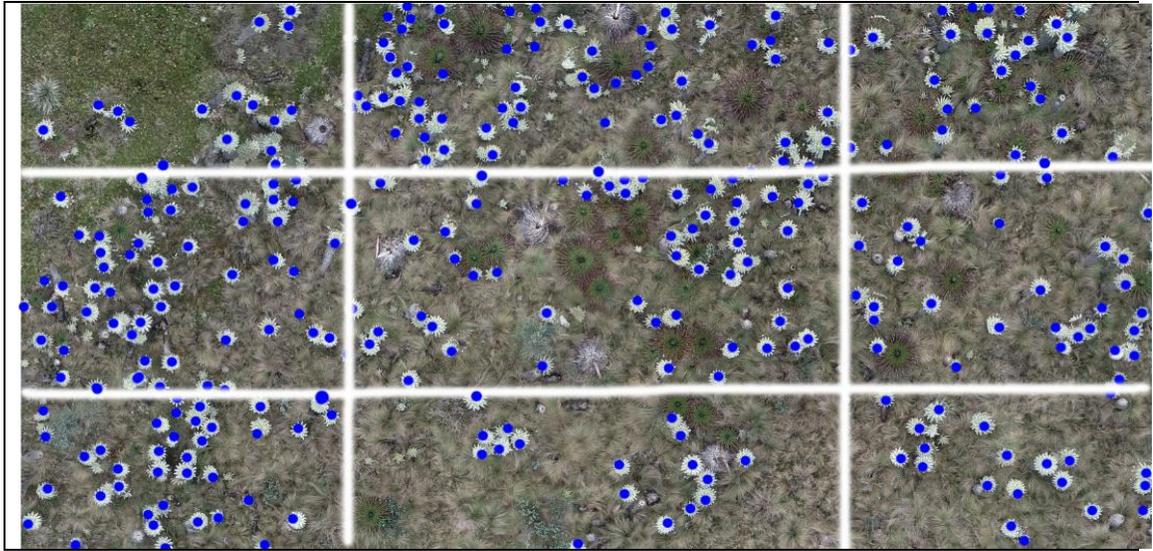
Número de toma de muestras	Conteo Manual	Conteo por herramienta automatizada
1	316	315
2		317
3		314

Elaborado por: D León

Para apreciar la similitud en el área de páramo del Volcán Chiles sobre el conteo de las copas y por ende corroborar la efectividad de la herramienta, se colocan de forma contigua la imagen obtenida de la identificación de copas automáticamente y la imagen del conteo manual.

Tabla 7. *Comparación de conteo automático y manual Parque de las Cuadras*





Elaborado por: D. León

En esta tabla comparativa se puede apreciar alta similitud entre la imagen resultante de la identificación de copas automáticamente por la herramienta, con la imagen del conteo manual.

Una vez comparadas las imágenes resultantes, se procede a calcular el error medio cuadrático (RMSE), que mide la cantidad de error que existe entre dos conjuntos de datos; es decir que compara un valor predicho y un valor observado o conocido. También se lo conoce como Raíz de Desviación Cuadrática Media y es una de las estadísticas más utilizadas en SIG (Gabri, 2018).

Para los resultados obtenidos la interpretación es que el valor calculado automáticamente por la herramienta corresponde al valor previsto, el valor calculado manualmente es el valor conocido y el tamaño de la muestra de valores observados y calculados para cada área de estudio es 4.

La fórmula con la que se trabaja es:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (P_i - C_i)^2}{n}}$$

En donde:

P_i Es el valor previsto

C_i Es el valor conocido

n Es el número de observaciones

Ahora se procede a realizar el RMSE para los resultados obtenidos en el área del Volcán Chiles.

$$RMSE = \sqrt{\frac{(315 - 316)^2 + (317 - 316)^2 + (314 - 316)^2}{4}} = 1.22$$

El valor obtenido de RSME del área del volcán Chiles es bajo, por lo que se puede considerar que la herramienta es fiable. Para una mejor interpretación del resultado, se utiliza un indicador adimensional de la bondad de ajuste; el coeficiente de eficiencia de Nash y Sutcliffe (*Ceff*). Este coeficiente representa el complemento a la unidad de la relación entre el RSME y la varianza de los valores observados (Ritter, Muñoz-Carpena, & Regalado).

La fórmula con la que se trabaja es:

$$Ceff = 1 - \frac{\sum_{i=1}^n (P_i - C_i)^2}{\sum_{i=1}^n (P_i - \bar{P})^2}$$

En donde:

P_i Es el valor previsto

C_i Es el valor conocido

\bar{P} Es el valor promedio entre los valores previstos y observados

Se calcula C_{eff} para el área del volcán Chiles.

Reemplazando los valores se tiene que:

$$C_{eff} = 1 - \frac{(315 - 316)^2 + (317 - 316)^2 + (314 - 316)^2}{(315 - 315.5)^2 + (317 - 315.5)^2 + (314 - 315.5)^2} = -0.263$$

C_{eff} puede tomar valores entre $-\infty \leq C_{eff} \leq 1$. Un $C_{eff} = 1$ indica un ajuste perfecto, mientras que un $C_{eff} \leq 0$ sugiere que el promedio de los valores previstos, son mejores predictores que los valores conocidos. Es decir, que los valores calculados automáticamente se obtienen con un rango de error bajo y que son más confiables que los valores calculados manualmente. Por lo tanto, se corrobora matemáticamente que la herramienta desarrollada es fiable para la imagen del área de páramo del volcán Chiles.

5.2. Resultados generales

Los resultados son diversos tanto en comparación de las tres imágenes utilizadas, como con los resultados obtenidos en el estudio base. Primero, cabe mencionar que el dimensionamiento del área de vuelo del estudio base es de 10m*10m, lo que genera un área de 100 metros cuadrados, esto significa que es un área pequeña, lo que facilita el procesamiento y análisis de la información.

Mientras que las áreas tomadas en el Parque de las Cuadras y en el páramo del Volcán Chiles, son más representativas en cuanto a tamaño, lo cual inicialmente generó una demora en el procesamiento de la información, por lo tanto se optó por trabajar con secciones de las imágenes procesadas. Por otro lado, se trabajó con la imagen aérea completa de la Universidad Politécnica Salesiana, puesto que no es tan extensa, sin embargo se evidenciaron otros inconvenientes que se detallan más adelante.

Para el desarrollo y ejecución del script, se realizaron varias pruebas con diferentes versiones de GRASS; estas fueron: 7.4.1, 7.4.2, 7.4.3 y 7.2.1, en donde la versión 7.4.1 no funcionó de ninguna manera, mientras que las versiones 7.4.2 y 7.4.3 si funcionaron pero no generaron los resultados deseados. Por lo tanto, la versión ideal para la investigación realizada fue la 7.2.1. En el caso del script del estudio base, se trabajó con la versión de 7.2.2. Además que en la metodología de investigación se incorporó el binding de TkINTER para que la vinculación de Python con QGIS fuera más simple.

Como se ha mencionado anteriormente, la diferencia sustancial entre el estudio base y el presente estudio ha sido el tipo de imágenes con las que se trabaja, porque el estudio base utiliza cámaras multiespectrales de alta resolución espacial. En tanto que en este estudio se utilizaron cámaras fotogramétricas de resolución media. Debido a esta razón se incorporaron líneas de código para calcular las bandas RGB, separarlas y trabajar con cada una de ellas. Esto representa un cambio notable en el código porque en el estudio base no existe el cálculo de RGB, lo que se calcula es el NDVI utilizando todas las bandas incluyendo el NIR, mediante la función de extensión de capa.

El script del estudio base genera una interfaz gráfica de cuatro características; dos de entrada y dos de salida. Las de entrada son: la imagen multiespectral y el ráster, y las de

salida son: un archivo de corona de árbol y las métricas de los árboles. El script desarrollado en esta investigación emite una interfaz gráfica de una opción, la de iniciar la contabilización de árboles, en donde se selecciona el archivo de prueba y a partir de ello la herramienta trabaja automáticamente.

En cuanto a los archivos resultantes, en el caso del estudio base, los que permitieron una mejor visualización son las que se trabajaron con un umbral de 0.95 y un tamaño mínimo de objeto de 100. Por otro lado para las tres imágenes procesadas, se evidenciaron mejores resultados con un umbral de 0.65 y un tamaño mínimo de objeto de 50, habiéndose realizado pruebas con valores mínimos de 100, 50 y 10. Cabe mencionar que utilizando los valores de minsize de 100 y 10 la herramienta desarrollada no se ejecuta. Una diferencia primordial es que la herramienta investigada genera múltiples archivos en los cuales se puede apreciar cuáles son los cambios que se generan paso a paso.

En síntesis, se evidencia que el procesamiento de las imágenes en la UPS no generó ningún resultado satisfactorio. Por otro lado, los resultados obtenidos en el procesamiento de la imagen del Parque de las Cuadras, no son concluyentes puesto que a pesar de que se han logrado identificar algunos polígonos y una posterior contabilización de los mismos, la herramienta está reconociendo zonas que no pertenecen a la especie arbórea en estudio, el eucalipto. Es decir, que la herramienta arroja resultados falsos positivos, esto sucede posiblemente por el solape que existe entre los árboles del parque.

Finalmente, el procesamiento en las imágenes del área de páramo del Volcán Chiles es sumamente satisfactoria, puesto que alcanza el objetivo planteado de identificación y conteo de la especie arbórea de estudio, el frailejón. Tanto en la corroboración de los valores calculados automática y manualmente, como en la comparación visual de los

resultados obtenidos en la imagen de identificación de copas y de la imagen seccionada por cuadrantes; se logra verificar la similitud en los resultados. Por lo que se calculó el RSME, obteniendo un valor de 1.22 y del *Ceff* de -0.263. Estos valores sugieren que la herramienta funciona correctamente, proporcionando valores fiables para la imagen fotogramétrica del área de páramo del Volcan Chiles.

6. CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Gracias a la integración del QGIS con Python, se ha podido generar una herramienta de código abierto con la posibilidad de ser estudiada y modificada acorde a las necesidades del investigador. De esta forma, a partir de la investigación base, se ha desarrollado una herramienta que posee un enfoque similar pero con atenuaciones en el área de aplicación, en su creación, presentación, diseño, forma de obtención y muestra de resultados.

Los cambios de la herramienta base para su creación fueron los siguientes:

- La capacidad de procesamiento de 512 MB a 20 GB
- Tiempo de procesamiento de 2 hora a 1 hora aproximadamente
- Incorporación del binding TkInter para la comunicación de QGIS con Python
- Rango de opacidad de 0.1 -0.65 a 0.05-0.65
- Tamaño mínimo de objetos “minsize” de 100 a 50

En cuanto a su presentación y diseño, se creó una interfaz gráfica de una sola entrada para el conteo de árboles, puesto que la interfaz gráfica de la herramienta base tiene cuatro parámetros; dos de entrada y dos de salida. La interfaz gráfica desarrollada es más simple de utilizar además que permite el procesamiento de la información de forma 100% automatizada. Por su parte, la obtención de datos, se realiza mediante la generación automática de archivos en cada una de las etapas o funciones que se detallan en el código. Y la muestra de resultados se evidencia en un shapefile con las copas identificadas y con

un archivo de texto con el número total de árboles. Es necesario mencionar que la diferencia más importante es el área de aplicación de la herramienta, puesto que el código desarrollado se ha descrito para la identificación de especies arbóreas en imágenes RGB obtenidas mediante cámaras fotogramétricas.

Para llegar a definir los parámetros mencionados con los que el script se ejecutó en totalidad, se realizaron varias pruebas con otros valores, sin embargo con ninguno de ellos la herramienta se logró ejecutar completamente.

El desarrollo de esta herramienta ha permitido la identificación de diferentes tipos de especies arbóreas, esto significa un avance en el área de investigación geográfica y ambiental, puesto que facilita el trabajo que se realiza en campo, además que permite generar a futuro un registro digital de la información de diferentes terrenos para tener una mejor planificación, gestión y monitoreo de los mismos.

El script desarrollado para la identificación y conteo de especies arbóreas, no significó un trabajo de gran complejidad, sin embargo al ejecutarlo presentó errores en ciertas imágenes. Por lo que se podrían implementar nuevos complementos de visualización, manipulación, procesamiento y análisis de datos, para obtener resultados más precisos en cualquier tipo de imagen, además de nuevas funcionalidades como área total de árboles, distancia entre individuos y el total de fallas.

Se lograron obtener resultados visibles de la detección de identificación y conteo de especies arbóreas, en donde se puede concluir que las imágenes con las cuales se debe trabajar, son aquellas que:

- Tengan un área pequeña; recomendable de 1 hectárea como máximo.

- Especies arbóreas que posean un patrón de distanciamiento y distribución, es decir que tengan un valor mínimo de distancia entre objetos considerable. Valor óptimo de 100.
- Que el área de estudio no tenga árboles frondosos contiguos, porque en el momento de la identificación de los mismos se genera solapamiento. Por lo tanto, se dificultaría la identificación en un bosque.

De forma visual en el área del Volcán Chiles, la herramienta desarrollada cumple con los objetivos planteados de identificación y conteo del frailejón, por lo cual se procedió a corroborar matemáticamente que la información generada sea confiable. Para ello se trabajó con dos modelos estadísticos; el RMSE y *Ceff*, para determinar la cantidad de error y grado de eficiencia que existe entre los valores calculados de forma manual y automática. El valor de RMSE que se obtuvo es de 1.22, lo que indica que es un error bajo. El valor de *Ceff* es -0.263, lo que sugiere que los valores calculados automáticamente, no solo son fiables si no que son más certeros que los valores calculados manualmente. De tal forma se concluye que la herramienta desarrollada es fiable para la imagen del Volcán Chiles.

En tanto que en las imágenes procesadas del Parque de las Cuadras y de la UPS, no se obtuvieron resultados satisfactorios. En el Parque de las Cuadras se lograron identificar objetos, clasificarlos mediante polígonos y contabilizarlos, sin embargo se evidenció que estos valores son falsos positivos, porque los valores numéricos de la contabilización automática coincidió con la de la contabilización manual, pero al comparar los polígonos identificados con los puntos contados estos no coincidieron. Mientras que en la imagen de la UPS, no se pudo generar ningún procesamiento después de que se generó la capa RGB.

6.2. Recomendaciones

En cuanto a la planificación mediante equipo drone, es necesario tener en cuenta el área y la altura del terreno sobre el cual se va a realizar el vuelo con el drone, la clase y tipo de drone y la altura de la especie arbórea; esto con la finalidad de tener una mejor definición de las imágenes que se van a procesar, y por ende un análisis más preciso de los resultados.

Es recomendable trabajar con áreas pequeñas y con especies arbóreas que presenten determinado patrón en su distanciamiento y distribución espacial, para que sea más sencilla su identificación y conteo.

Se sugiere trabajar con la versión de Python 2.7 y de QGIS 2.18.9, ya que con estas versiones aunque son antiguas; se pueden habilitar correctamente las librerías de API, QGIS API, GRASS Y PyQT4 API.

Se podrían realizar variaciones en el código en la función de la clasificación, para probar los algoritmos de clasificación supervisada y semisupervisada.

7. BIBLIOGRAFÍA

- ACRE. (s.f.). *ACRE Surveying Solutions*. Obtenido de Software de procesamiento de imágenes Pix4DMapper: <https://grupoacre.es/catalogo-productos/pix4dmapper/>
- aerial insights. (2019). *Cómo planificar capturas de drone*. Obtenido de <http://www.aerial-insights.co/blog/como-planificar-capturas-de-dron/>
- Agencia Espacial del Perú. (2011). *Procesamiento Digital de Imagenes Satelitales*. Peru: Ministerio de Defensa. Conida.
- Alonso, D. (20 de Mayo de 2016). *Mapping Gis*. Recuperado el 2 de Enero de 2019, de Qué podemos hacer con GRASS GIS 7: <https://mappinggis.com/2016/05/puedo-grass-gis-7/>
- Alonso, D. (5 de Septiembre de 2017). *Mapping Gis*. Recuperado el 2 de Enero de 2019, de NDVI: Qué es y cómo calcularlo con SAGA desde QGIS: <https://mappinggis.com/2015/06/ndvi-que-es-y-como-calcularlo-con-saga-desde-qgis/>
- Álvarez, M. Á. (20 de Octubre de 2014). *desarrolloweb.com*. Recuperado el 2 de Enero de 2019, de Binding en Angular JS, y doble binding: <https://desarrolloweb.com/articulos/binding-angularjs-doble-binding.html>
- ANAC. (s.f.). *Capítulo 1: Aerodinámica Básica*. Argentina: Dirección Nacional de Seguridad Operacional.

- Bejerano, P. (22 de Febrero de 2015). *ToDrone. El uso de Drones en la Agricultura*. Recuperado el 17 de Diciembre de 2018, de <https://www.todrone.com/uso-drones-agricultura/>
- Berrío Meneses, V. A., Mosquera, J., & Alzate, D. F. (2015). Uso de drones para el análisis de imágenes multiespectrales en agricultura de precisión. *ResearchGate*, 29.
- Bryce, C. (05 de 01 de 2019). *Wiki. Python*. Obtenido de TkInter: <https://wiki.python.org/moin/TkInter>
- Carita, G. (20 de Septiembre de 2016). *Gidahatari*. Recuperado el 2 de Enero de 2019, de Tutorial de instalación de OTB para procesamiento de imágenes satelitales en QGIS : <http://gidahatari.com/ih-es/instalacion-de-otb-para-el-procesamiento-de-imagenes-satelitales-en-qgis>
- CC BY-SA. (s.f.). *Documentación de QGIS 2.8*. Obtenido de https://docs.qgis.org/2.8/es/docs/user_manual/working_with_raster/raster_calculator.html
- Centro Europeo de Postgrado. (s.f.). *¿Qué es la fotogrametría analítica y digital?* Obtenido de <https://www.ceupe.com/blog/fotogrametria-analitica-y-digital.html>
- Cepeda, F. G. (2004). *Fotogrametría Digital*. Obtenido de http://pdi.topografia.upm.es/cepeda/Curso_2004/InstrumentacionEFD1.pdf
- Copters, A. (10 de Diciembre de 2015). *Drones by OpenOpin*. Recuperado el 18 de Diciembre de 2018, de <http://www.drones.openopin.com/dronedeploy-mapas/>

Crónica. (19 de Septiembre de 2015). DGAC EMITIÓ REGLAMENTACIÓN PARA EL USO DE DRONES EN ECUADOR. Loja, Loja, Ecuador.

Departamento de la Guajira. (2007). *Mejora de los Sistemas de Cartografía del Territorio Colombiano. Teledetección*. Rioacha.

Departamento de matemáticas Universidad de Oviedo. (05 de 06 de 2014). *Computación Numérica*. Obtenido de Grado de Ingeniería Informática del Software: https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html#clasifica

Departamento de Montes. (2004). *Inventario Forestal Nacional. Manual de Campo. Modelo*. Guatemala: Programa de Evaluación de los Recursos Forestales.

DJI. (2016). *Phantom 4*. Recuperado el 2018 de Diciembre de 2018, de <https://www.dji.com/phantom-4/info#specs>

Duarte, L., Silva , P., & Teodoro, A. C. (2018). *Development of a QGIS Plugin to Obtain Parameters and Elements of Plantation Trees and Vineyards with Aerial Photographs*. Portugal: Department of Geosciences, Environment and Land Planning, Faculty of Sciences, University of Porto.

Falk, D., & Campos, A. N. (2014). Algoritmo semiautomático para el conteo de árboles en plantaciones forestales mediante el uso de imágenes aéreas. *6° Congreso Argentino de Agroinformática* (pág. 2). Argentina: Universidad Tecnológica Nacional.

Gabri. (22 de Mayo de 2018). *ArcGeek*. Obtenido de ¿Qué es el error cuadrático medio RMSE?: <https://acolita.com/que-es-el-error-cuadratico-medio-rmse/>

Garay, D., & Agüero, J. (2016). *Introducción a los SIG Software QGIS*. La Rioja: INTA Ediciones.

Gis&Beers. (2 de Octubre de 2016). *Gis and Beers*. Obtenido de <http://www.gisandbeers.com/agisoft-photoscan-ortomosaicos-para-drones/>

Google Maps. (2019).

Institut Cartografic u Geologic de Catalunya. (s.f.). *Diferencias entre fotografia aérea y ortofoto*. Obtenido de <http://www.icgc.cat/es/Ayuda/Preguntas-frecuentes/Diferencias-entre-fotografia-aerea-y-ortofoto>

Junta de Andalucía. (s.f.). *Rediam*. Obtenido de Qué es y cómo se hace una Ortofoto: http://www.juntadeandalucia.es/medioambiente/site/rediam/menuitem.04dc44281e5d53cf8ca78ca731525ea0/?vgnnextoid=c6fd05464ea09110VgnVCM1000000624e50aRCRD&vgnnextchannel=a76090a63670f210VgnVCM2000000624e50aRCRD&vgnnextfmt=rediam&lr=lang_es

Lahoz, J. G. (s.f.). 4. Introducción a la fotogrametría digital: El método general de la fotogrametría digital. En J. G. Lahoz, *Universidad de Salamanca Master de Geotecnologías Cartográficas en Ingeniería y Arquitectura* (pág. 4). Salamanca: Departamento de Ingeniería Cartográfica y del Terreno. Escuela Politécnica Superior de Ávila.

Marínez Vega, J., & Martín Isabel, M. (2010). *Guía Didáctica de Teldetección y Medio Ambiente*. Madrid: Centro de Ciencias Humanas y Sociales (CSIC).

Merino, M. (12 de Julio de 2014). *Tic Beat*. Recuperado el 2 de Enero de 2019, de ¿Qué es una API y para qué sirve?: <https://www.ticbeat.com/tecnologias/que-es-una-api-para-que-sirve/>

Mora, J. (23 de Mayo de 2018). *Instituto Español de Estudios Estratégicos*. Recuperado el 17 de Diciembre de 2018, de http://www.ieee.es/Galerias/fichero/docs_analisis/2018/DIEEEA19-2018_Drones-Clave_Desarrollo_SeguridadAfrica_JAMT.pdf

Mostacedo, B., & Fredericksen, T. S. (2000). *Manual de Métodos Básicos de Muestreo y Análisis en Ecología Vegetal*. Santa Cruz de la Sierra: BOLFOR.

Murcia, U. d. (2017). *Universidad de Murcia*. Obtenido de <https://www.um.es/geograf/sigmur/teledet/tema04.pdf>

Noboa, S. (Sin publicar). *Estimación de altura de frailejones (Espeletia pycnophylla) en el volcán Chiles mediante UAV (Carchi-Ecuador)*.

Pérez Navarro, A., Botella Plana, A., Muñoz Bollas, A., Olivella González, R., Olmedillas Hernández, J. C., & Rodríguez Lloret, J. (2011). *Introducción a los sistemas de información geográfica y geotelemática*. Barcelona: UOC.

Pinilla, W. (2008). *El futuro del vuelo no convencional*. Colombia: Colombia aprende.

Pleiadestic. (4 de Noviembre de 2014). *Qué es Quantum GIS y por qué utilizarlo*. Obtenido de <https://pleiadestic.com/es/que-es-quantum-gis-y-por-que-utilizarlo/>

Pontificia Universidad Católica de Chile. (s.f.). *Curso Geografía del Mar*. Obtenido de 5
Percepción Remota Satelital Aplicada al Mar:
http://www7.uc.cl/sw_educ/geo_mar/html/h52.html

Puente, V. (15 de 08 de 2007). *Xataka Ciencia*. Obtenido de
<https://www.xatakaciencia.com/tecnologia/teledeteccion-la-tecnica-de-la-observacion-remota>

Quintero, J. L. (3 de Febrero de 2017). *DronerStudio*. Obtenido de DronerBlog:
<http://dronerstudio.com/planificar-un-vuelo-dron/>

Ribeiro, P. (25-26 de Julio de 2016). *Fundación Biodiversa Colombia*. Recuperado el 17
de Diciembre de 2018, de http://www.fundacionbiodiversa.org/wordpress/wp-content/uploads/2016/05/Curso-drones-Convocatoria_FINAL.pdf

Ritter, A., Muñoz-Carpena, R., & Regalado, C. M. (s.f.). *Capacidad de predicción de modelos aplicados a la ZNS: Herramienta informática para la adecuada evaluación de la bondad de ajuste con significación estadística*. Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria; Ministerio de Ciencia e Innovación.

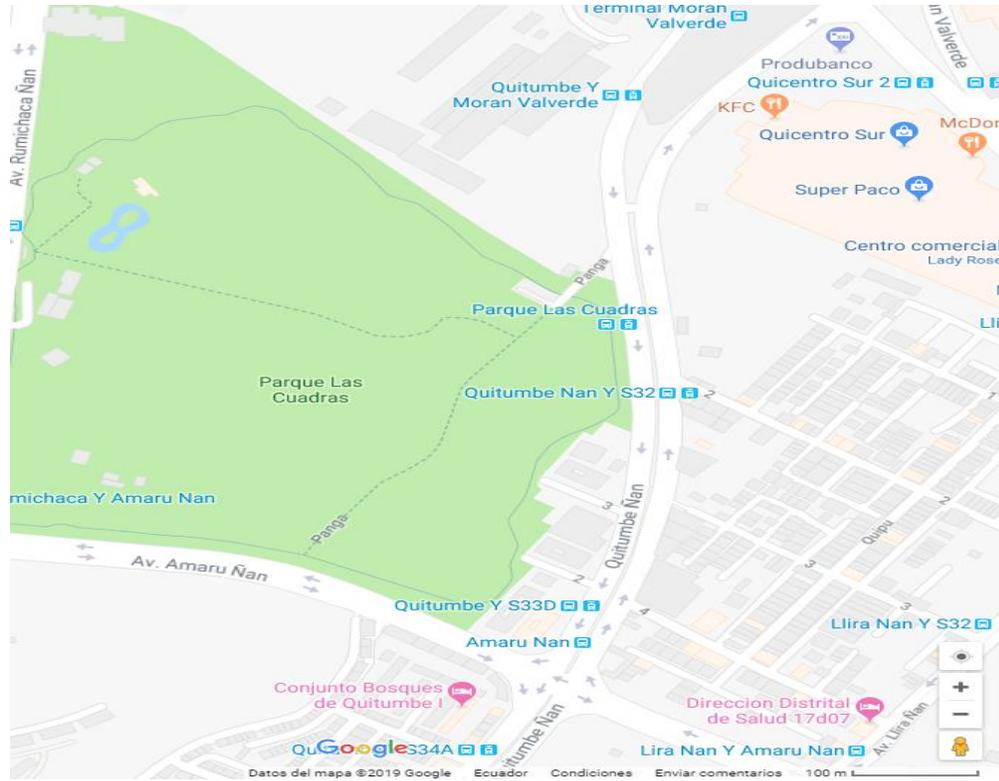
Rodriguez Chavez, O. E., & Arredondo Bautista, H. A. (2005). *Manual para el manejo y procesamiento de imágenes satelitales del sensor remoto MODIS de la NADA, aplicado en estudios de ingeniería civil*. Bogotá: Pontificia Universidad Javeriana.

Rossum, G. v. (Septiembre de 2009). *El Tutorial de Python*. Recuperado el 17 de
Diciembre de 2018, de <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>

- Ruiz, C. P. (s.f.). *Universidad de Jaén*. Recuperado el 17 de Diciembre de 2018, de Teledetección. Ingeniería Técnica en Topografía|: <https://www.ujaen.es/huesped/pidoceps/tel/archivos/7.pdf>
- Santana, E. (2017). *Xdrones*. Recuperado el 18 de Diciembre de 2018, de <https://www.xdrones.es/tipos-de-drones-clasificacion-de-drones-categorias-de-drones/>
- Taboada González, J. Á., & Cotos Yáñez, J. M. (2005). *Sistemas de Información Medioambiental*. España: Netbiblo.
- Topoequipos. (2015). *Fotogrametría digital*. Obtenido de ¿Qué es fotogrametría digital?: <http://www.topoequipos.com/topoequipos2.0/fotogrametraa-digital/fotogrametria-digital>
- Tristan, P. M., Wainschenker, R. S., & Doorn, J. H. (s.f.). *Normalización de Imágenes Satelitales en el Análisis Multi-temporal*. Buenos Aires: INTIA.
- Villar, A. A. (03 de 03 de 2010). *OpenCourseWare de la Universidad Politécnica de Madrid*. Obtenido de Fotogrametría y teledetección: http://ocw.upm.es/ingenieria-cartografica-geodesica-y-fotogrametria/topografia-cartografia-y-geodesia/contenidos/TEMA_11_FOTOGAMETRIA_Y_TELEDETECCION/Teledeteccion/microsoft-word-teledeteccion_y_sist_tratamiento_digital_imagenes.pdf

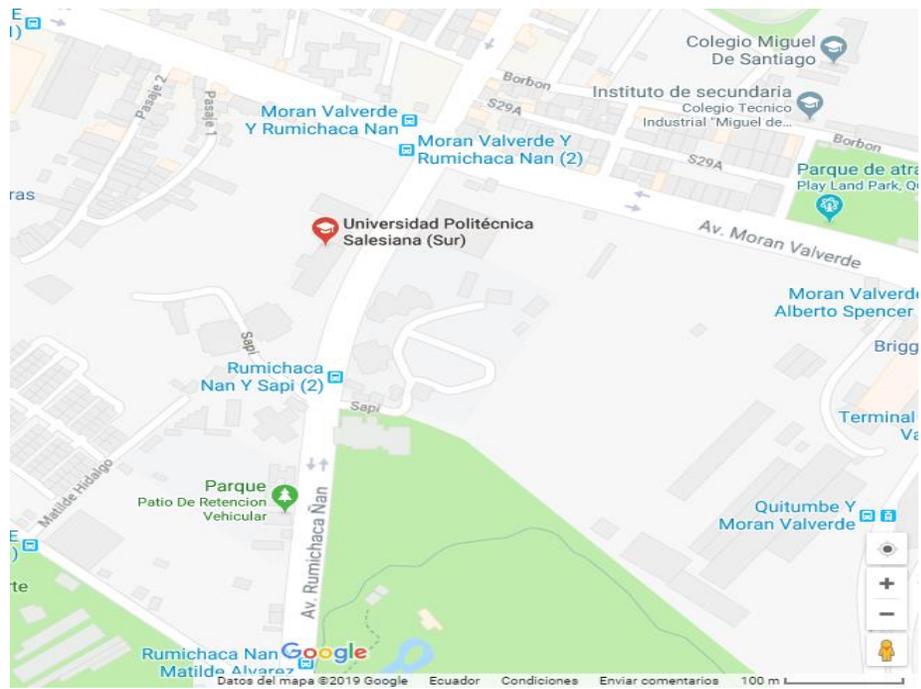
8. ANEXOS

Anexo 1. Ubicación del Parque de las Cuadras



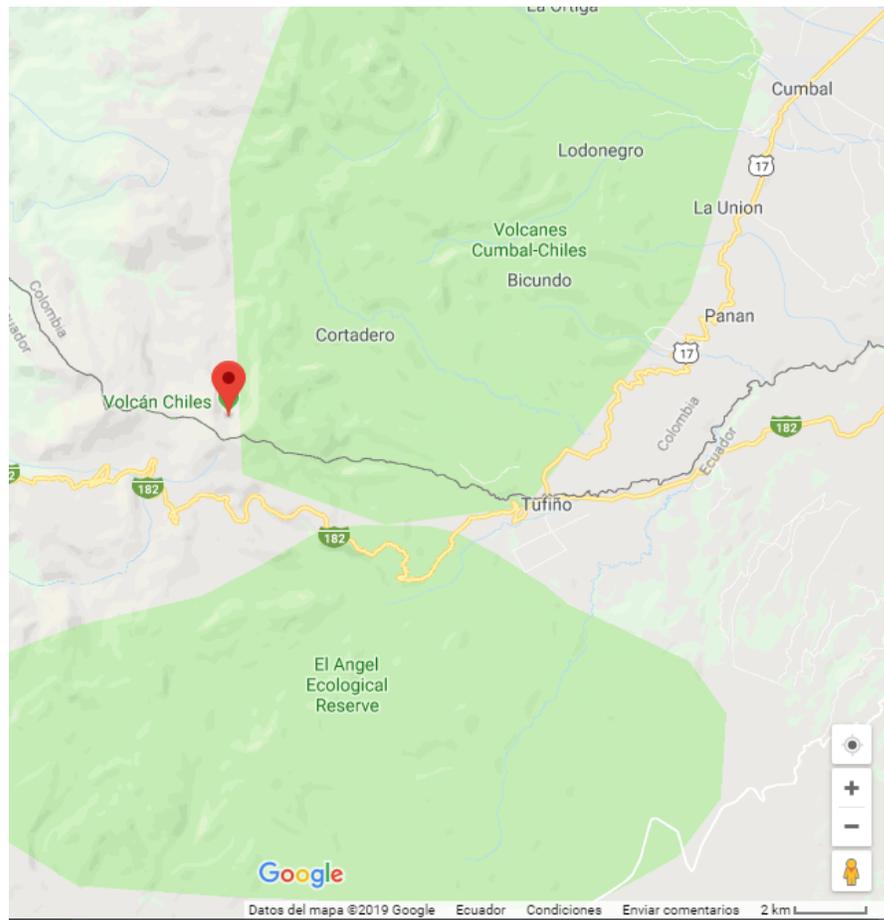
Fuente: (Google Maps, 2019)

Anexo 2. Ubicación de la Universidad Politécnica Salesiana.



Fuente: (Google Maps, 2019)

Anexo 3. Ubicación del Páramo del Volcán Chiles



Fuente: (Google Maps, 2019)

Anexo 4. Descripción del código desarrollado

INICIO

```
import Tkinter
import tkFileDialog
import os
import tkMessageBox
from qgis.core import *
import qgis.utils
from PyQt4.QtCore import QFileInfo, QVariant
from qgis.analysis import *
import os.path
from processing import Processing, QgsApplication
from qgis.analysis import QgsRasterCalculatorEntry,
QgsRasterCalculator

baseLayerName = "baseLayer"
rgbLayerName = "rgbLayer"
minsize = 50
ramSize = 2048
usrPath = ""
fileName = ""
totalBandas = 0
maxValClas = 0
totalPixel = 0

top = Tkinter.Tk()
top.title('Clasificador de Arboles')
top.geometry('400x100')
top.resizable(0, 0)
```

```
def mainFunction() :
```

```
    baseLayer = seleccionarImagen()  
    rgb = crearCapaRGB(baseLayer)  
    extentRaster = obtenerExtencionCapa(baseLayer)  
    Processing.initialize()  
    bandas = separarBandas()  
    seed = crearSemilla(bandas, extentRaster)  
    segmentada = segmentarImagen(bandas, seed, extentRaster)  
    classLayer = clasificarImagen(rgb, segmentada)  
    copasLayer = identificarCopas(classLayer)  
    totalArboles = contarCopas(copasLayer)  
    crearArchivoResultados(totalArboles)
```

```
def seleccionarImagen() :
```

```
    global fileName  
    fileName = tkFileDialog.askopenfilename(initialdir = '/',title = "Select  
file",filetypes = (('raster file','*.tif'),('all files','*.*')))  
    global usrPath  
    usrPath = QFileInfo(fileName).path() + '/'  
    baseLayer = QgsRasterLayer(fileName, baseLayerName)  
    QgsMapLayerRegistry.instance().addMapLayer(baseLayer)  
    return baseLayer
```

```
def crearCapaRGB(baseLayer) :
```

```
    print("Inicio calculo capa RGB..")  
    entries = []  
    rasBand4 = QgsRasterCalculatorEntry()  
    rasBand4.ref = baseLayerName + '@4'  
    rasBand4.raster = baseLayer
```

```

rasBand4.bandNumber = 4
entries.append( rasBand4 )
rasBand3 = QgsRasterCalculatorEntry()
rasBand3.ref = baseLayerName + '@3'
rasBand3.raster = baseLayer
rasBand3.bandNumber = 3
entries.append( rasBand3 )
rgb = usrPath + 'rgb.tif'

calc = QgsRasterCalculator( '(' + baseLayerName + '@4' + ' - ' +
baseLayerName + '@3' + ' )/( ' + baseLayerName + '@4' + ' + ' +
baseLayerName + '@3' + ' )', rgb, 'GTiff', baseLayer.extent(),
baseLayer.width(), baseLayer.height(), entries )

calc.processCalculation()
rgbLayer = QgsRasterLayer(rgb, rgbLayerName)
QgsMapLayerRegistry.instance().addMapLayer(rgbLayer)
print("Fin calculo capa RGB..")
return rgb

```

```

def obtenerExtencionCapa(baseLayer) :
    global totalBandas
    totalBandas = baseLayer.bandCount() # numero total de bandas
    xPixel = baseLayer.width()
    yPixel = baseLayer.height()
    global totalPixel
    totalPixel = xPixel*yPixel
    minx = baseLayer.extent().xMinimum()
    maxx = baseLayer.extent().xMaximum()
    miny = baseLayer.extent().yMinimum()
    maxy = baseLayer.extent().yMaximum()
    return str(minx) + ',' + str(maxx) + ',' + str(miny) + ',' + str(maxy)

```

```

def separarBandas() :
    print("Inicio separacion bandas..")
    split = usrPath + 'split.tif'
    Processing.runAlgorithm("otb:splitimage", None, fileName, ramSize,
split)
    bandas=[usrPath + 'split_' + str(nband) + '.tif' for nband in
range(totalBandas - 1)]
    print("Fin separacion bandas..")
    return bandas

def crearSemilla(bandas, extentRaster) :
    print("Inicio creacion semilla..")
    seed = usrPath + 'semilla.tif'
    # Imagen inicial con puntos semilla
    Processing.runAlgorithm("grass7:i.segment", None, bandas, 0.05, 0, 0,
minsize, ramSize, 20, None, None, False, False, str(extentRaster), 0, seed,
None)
    print("Fin creacion semilla..")

def segmentarImagen(bandas, seed, extentRaster) :
    print("Inicio segmentacion recursiva..")
    segmentada = usrPath + 'segmentada.tif'
    for thresh in [0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65]:
        Processing.runAlgorithm("grass7:i.segment", None, bandas, thresh,
0, 0, minsize, ramSize, 20, seed, None, False, False, str(extentRaster), 0,
segmentada, None)
        seed = segmentada
    QgsMapLayerRegistry.instance().addMapLayer(
QgsRasterLayer(segmentada, "seedTest"))
    print("Fin segmentacion recursiva..")
    return segmentada

```

```

def clasificarImagen(rgb, segmentada) :
    print("Inicio clasificacion..")
    trainingSet = int(totalPixel*0.00705)
    clasificada = usrPath + 'clasificada.tif'

Processing.runAlgorithm("otb:unsupervisedkmeansimageclassification",
None, rgb,
    ramSize, segmentada, trainingSet, 5, 1000, 0.0001, clasificada, None)
    print("Fin clasificacion..")
    classLayer = QgsRasterLayer(classificada, 'class_rc')
    QgsMapLayerRegistry.instance().addMapLayer(classLayer)
    global maxValClas
    maxValClas = classLayer.dataProvider().bandStatistics(1,
QgsRasterBandStats.All, classLayer.extent()).maximumValue
    return classLayer

def identificarCopas(classLayer) :
    print("Inicio extraccion copas..")
    extraida = usrPath + 'extraida.tif'

    Processing.runAlgorithm("gdalogr:rastercalculator", None,
classLayer, 1, None, None, None, None, None, None, None, None,
None, "A==" + str(maxValClas), 0, 4, None, extraida)

    # Conversion de imagen clasificada en fichero vectorial
    copas = usrPath + 'copas.shp' # output shapefile

    Processing.runAlgorithm("gdalogr:polygonize", None, extraida, 'DN',
copas)

    layer = QgsVectorLayer(copas,"copasLayer","ogr")

    # Excluir objetos da shapefile, a traves del valor del area, atributo y el
radio del bounding box
    difRadio=0.55

    with edit(layer):
        for f in layer.getFeatures():

```

```

    area = f.geometry().area()
    height = f.geometry().boundingBox().height()
    width = f.geometry().boundingBox().width()
    ratio = height/width
    if area<=1 or area>=50 or f['DN']!=1 or ratio<1-difRadio or
ratio>1+difRadio:
        layer.deleteFeature(f.id())
    QgsMapLayerRegistry.instance().addMapLayer(layer)
    print("Fin extraccion copas..")
    return layer

def contarCopas(layer) :
    # Conteo copas
    print("Inicio conteo de copas..")
    if layer.dataProvider().capabilities() &
QgsVectorDataProvider.AddAttributes:
        remover = layer.dataProvider().deleteAttributes([0])
        adicionar = layer.dataProvider().addAttributes([QgsField("Arboles-
ID", QVariant.Int)])
    layer.updateFields()
    totalArboles = 0 # Contador para el total de arboles
    centroides = {}
    # Se realiza el conteo automatico de los arboles obtenidos
    with edit(layer):
        for f in layer.getFeatures():
            area = f.geometry().area() # area
            perimetro = f.geometry().length() # perimetro
            centroid_x = f.geometry().boundingBox().center().x() # coordenada
x del centroide
            centroid_y = f.geometry().boundingBox().center().y() # coordenada
y del centroide
        # Diccionario con los centroides

```

```

centroides[int(f.id())] = QgsPoint(centroid_x, centroid_y)

# Asignar id al arbol
f[0] = f.id()
layer.updateFeature(f)

totalArboles += 1

print('Fin conteo de copas..')
return totalArboles

```

```

def crearArchivoResultados(totalArboles):

```

```

    print('Inicio creacion archivo resultados..')

```

```

    info = usrPath + 'resultados.txt'

```

```

    archivo = open(info, 'w') # archivo de resultados

```

```

    # Se escriben los resultados en el archivo

```

```

    line = '\n-----#####-----\n\n'

```

```

    archivo.write(line)

```

```

    line = 'Numero total de arboles = ' + str(totalArboles) + '\n'

```

```

    archivo.write(line)

```

```

    archivo.close()

```

```

    print('Fin creacion archivo resultados..')

```

```

    B = Tkinter.Button(top, text = "Iniciar conteo" , command =
mainFunction)

```

```

    B.pack(pady = 40)

```

```

    top.mainloop()

```

FIN

Anexo 5. Procesamiento de Imágenes del Parque de las Cuadras

Imagen de Prueba



RGB

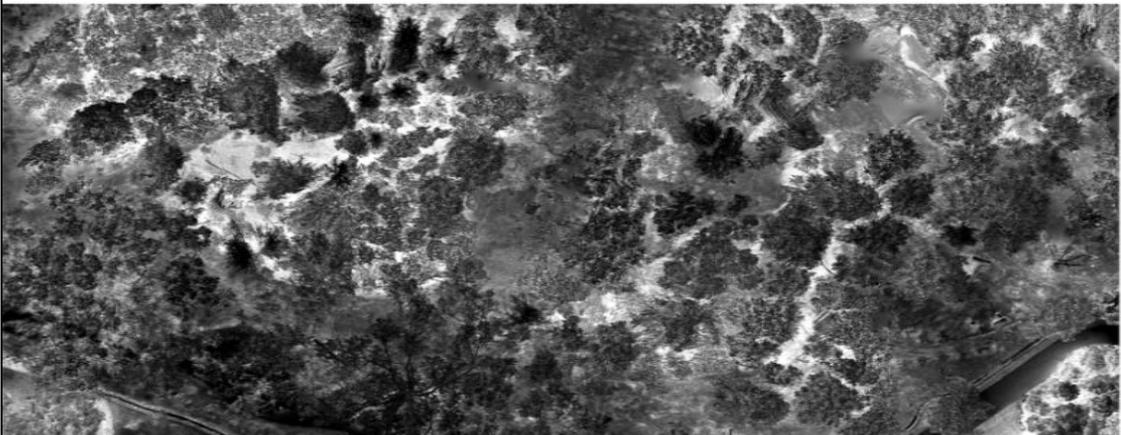


Imagen Multibanda 0

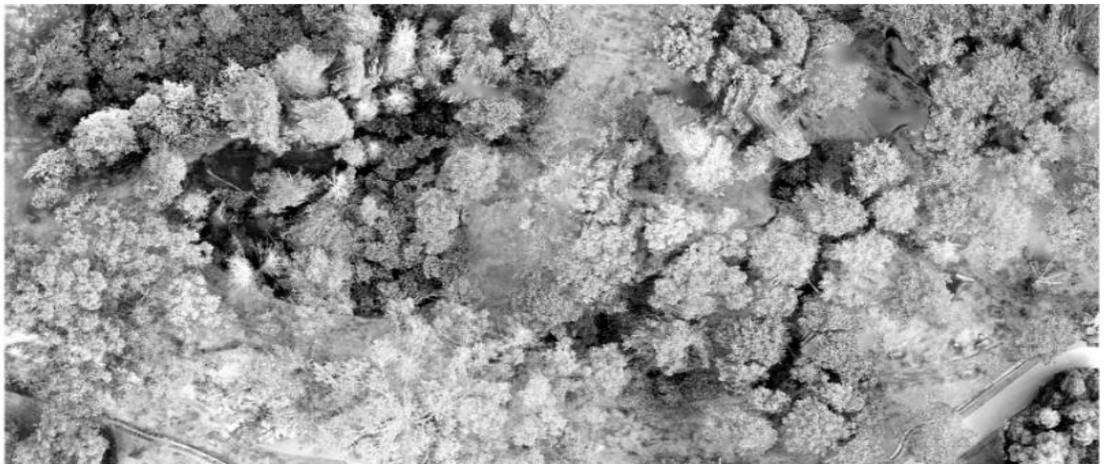


Imagen Multibanda 1

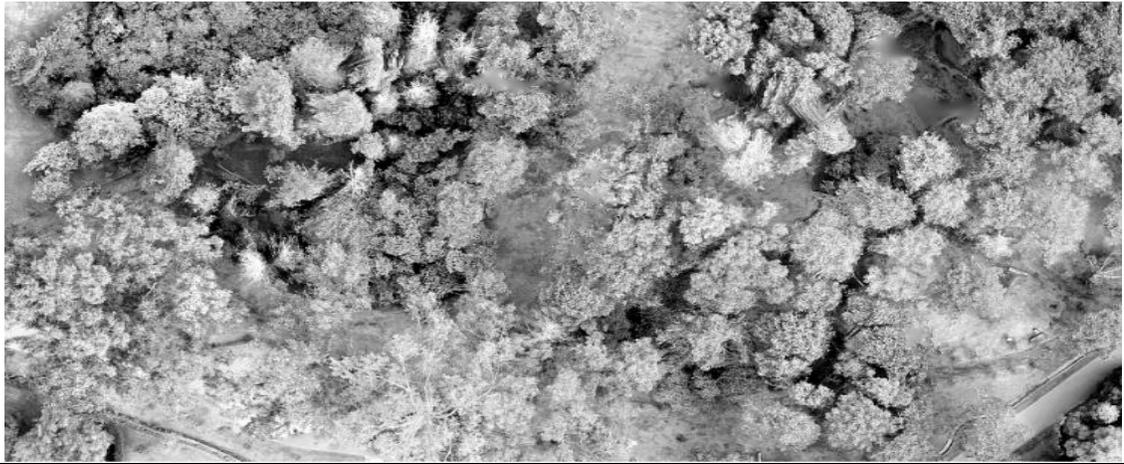


Imagen Multibanda 2

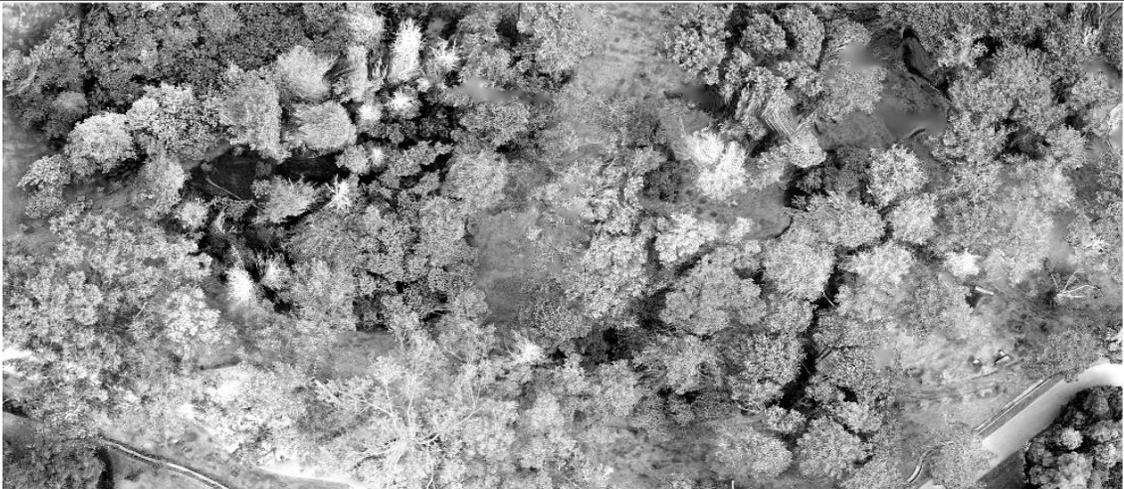


Imagen Multibanda 3

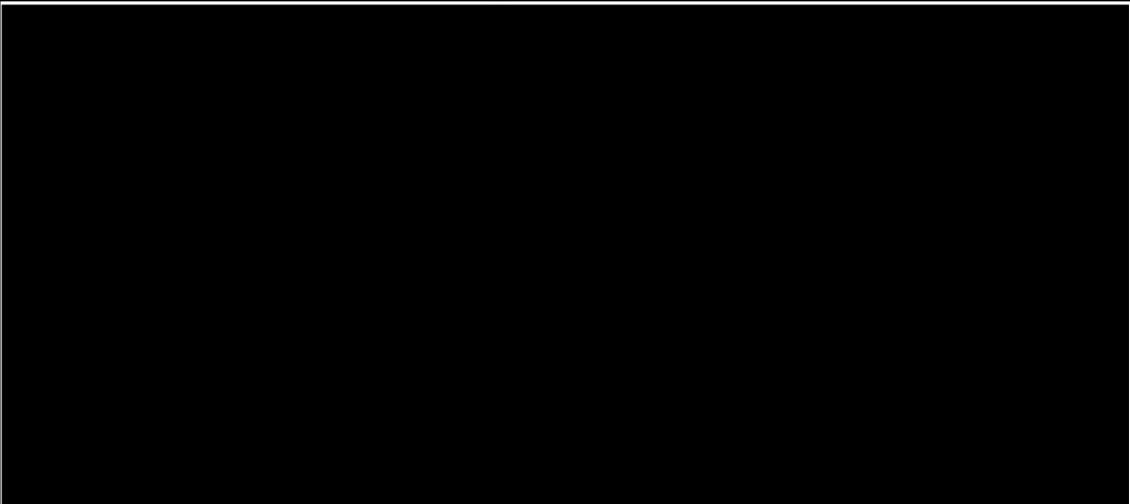


Imagen de segmentación de primera prueba



Imagen de segmentada final



Imagen extraída

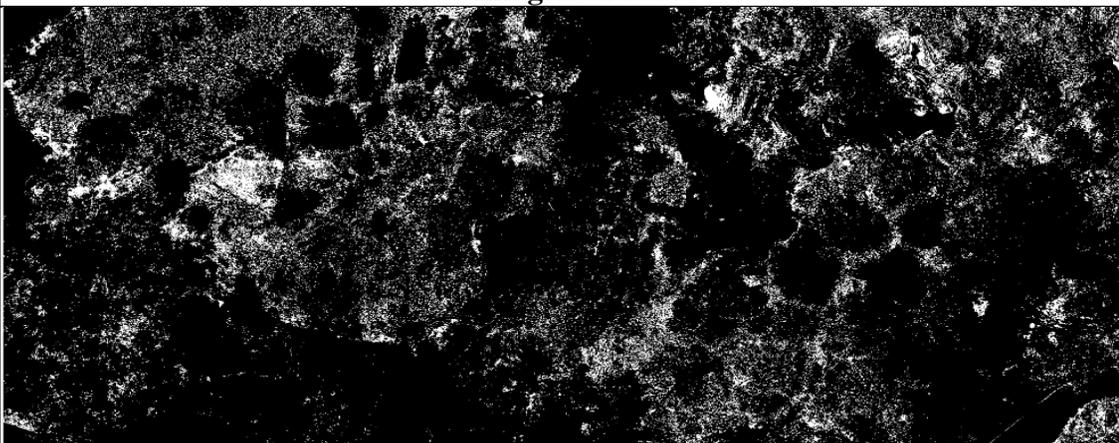


Imagen clasificada

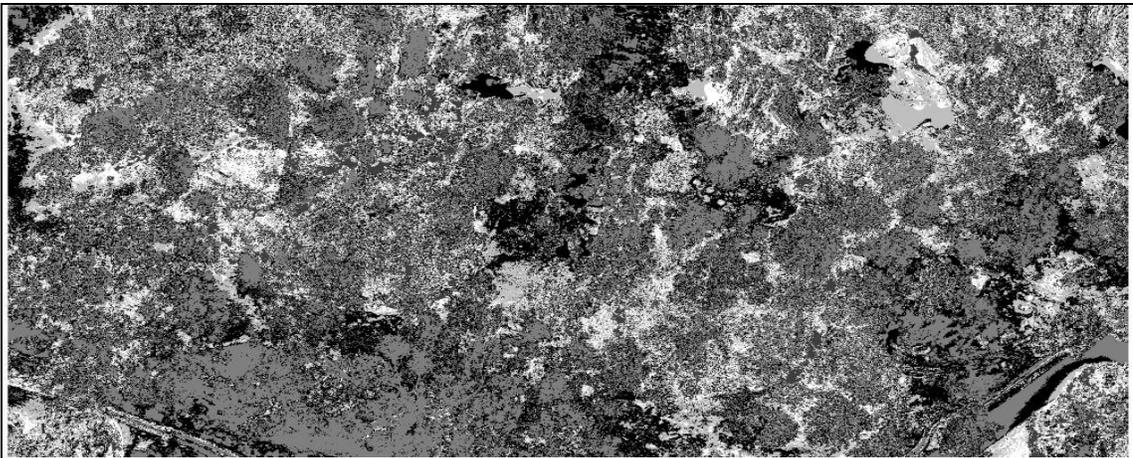


Imagen clasificada con identificación de eucaliptos

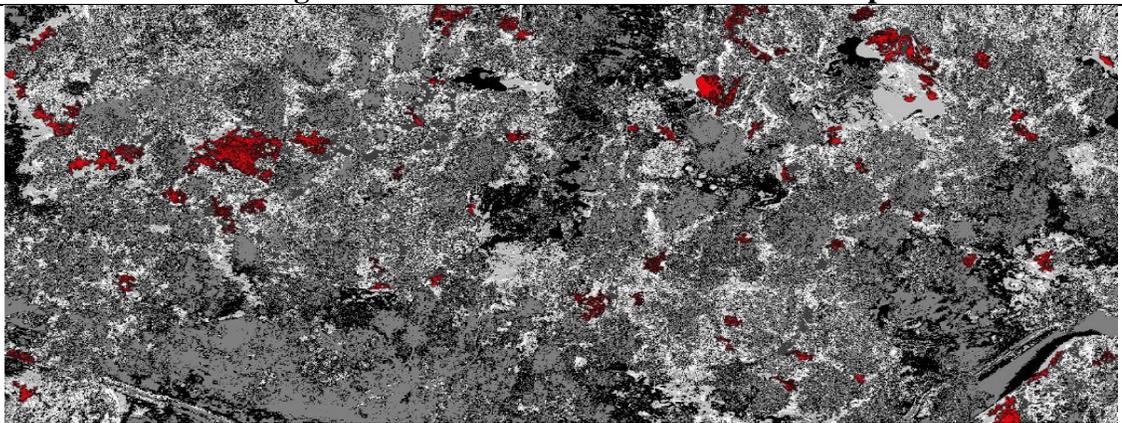
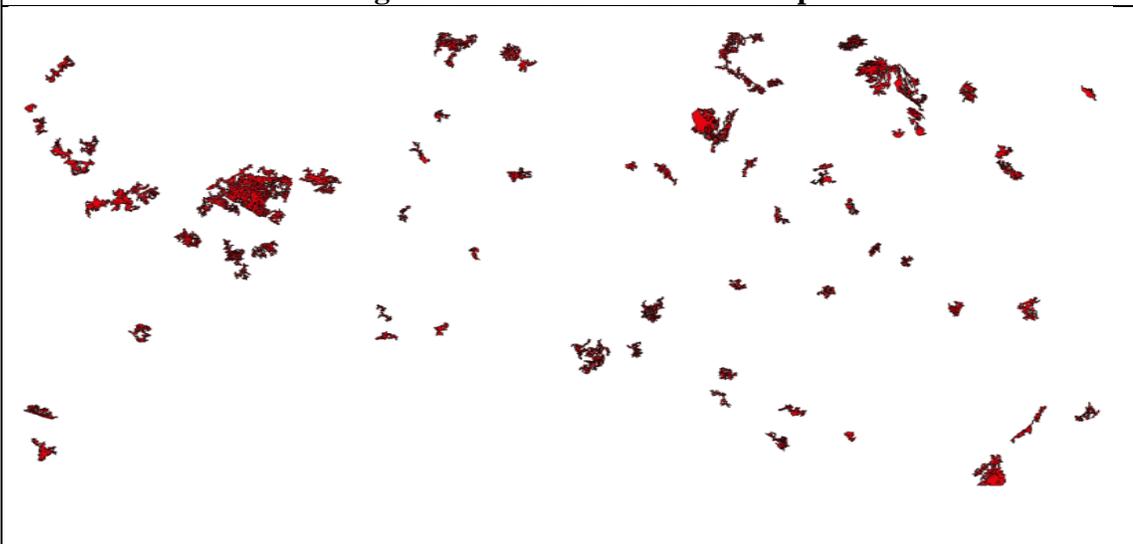


Imagen con identificación de eucaliptos



Elaborado por: D. León

Anexo 6. Procesamiento de imágenes del área de Páramo del Volcán Chiles

Imagen de Prueba



RGB

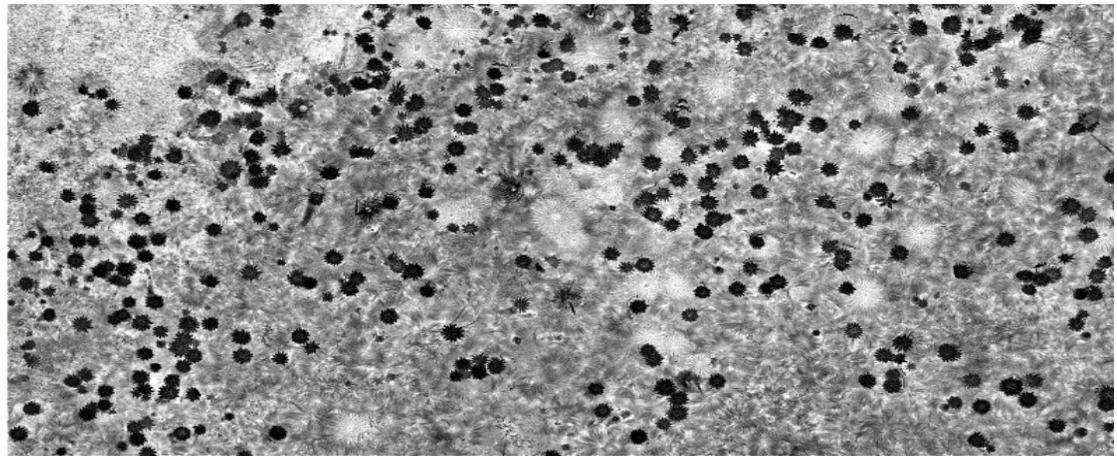


Imagen Multibanda 0

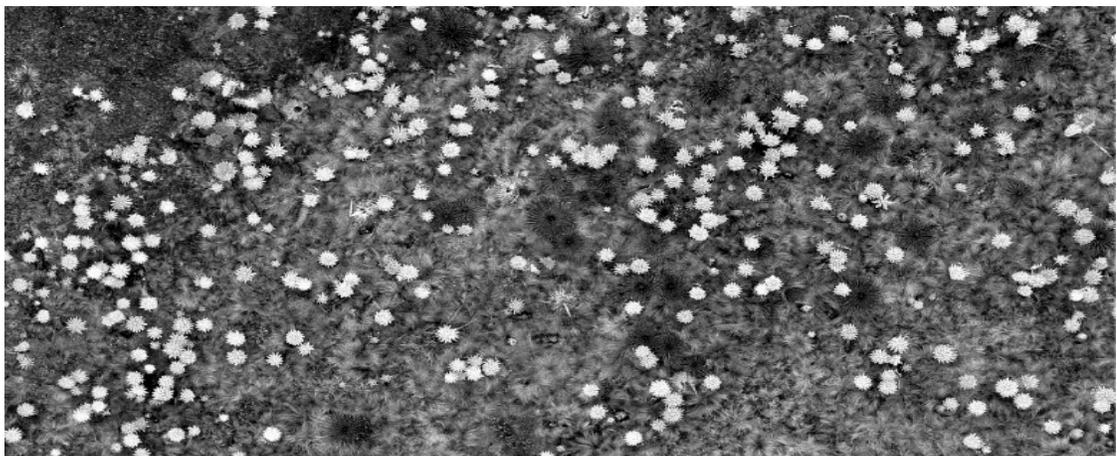


Imagen Multibanda 1

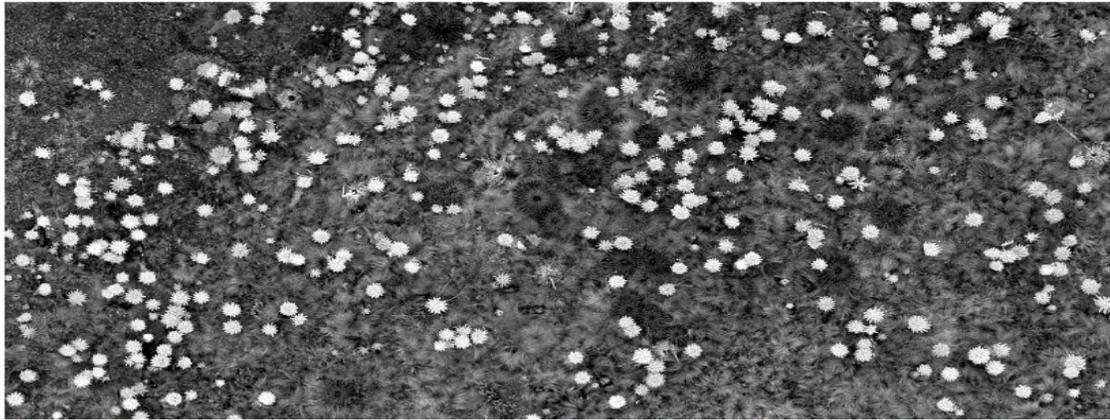


Imagen Multibanda 2

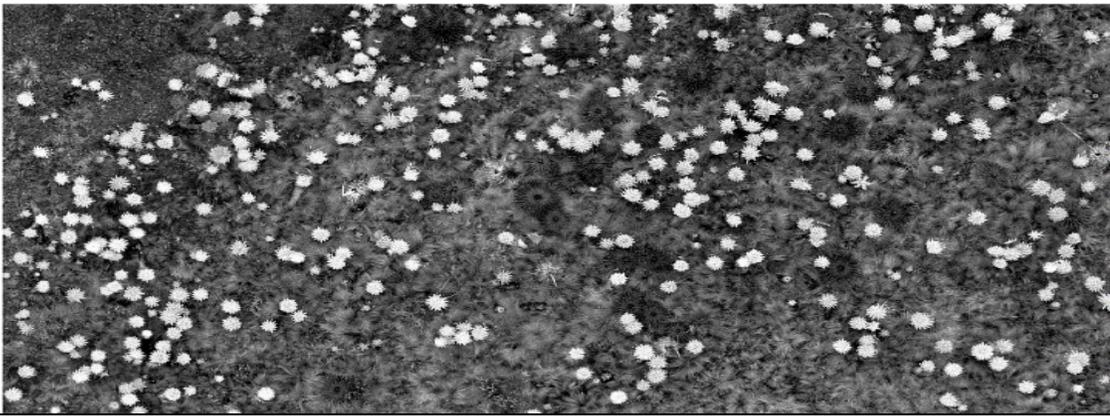


Imagen Multibanda 3

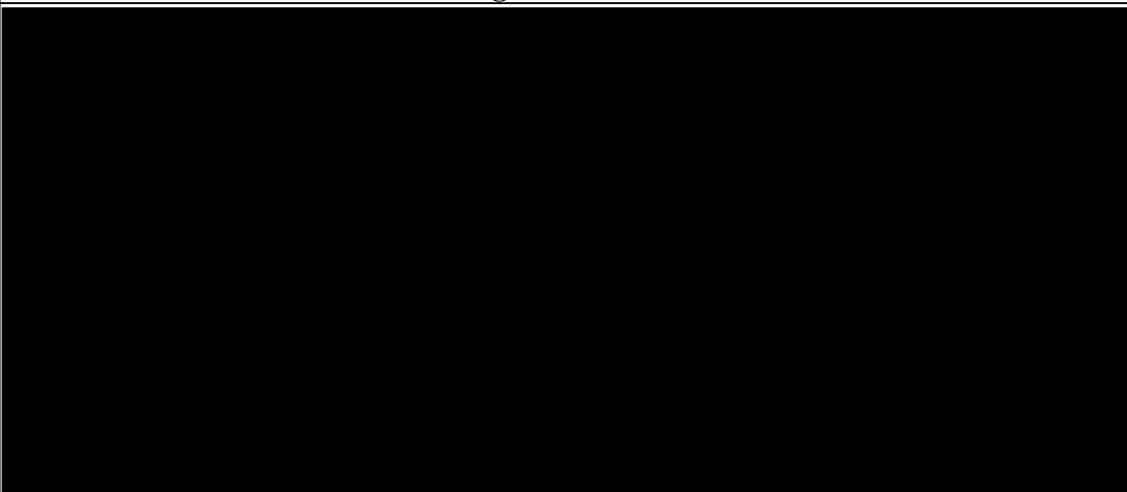


Imagen de segmentación de primera prueba



Imagen de segmentada final



Imagen extraída

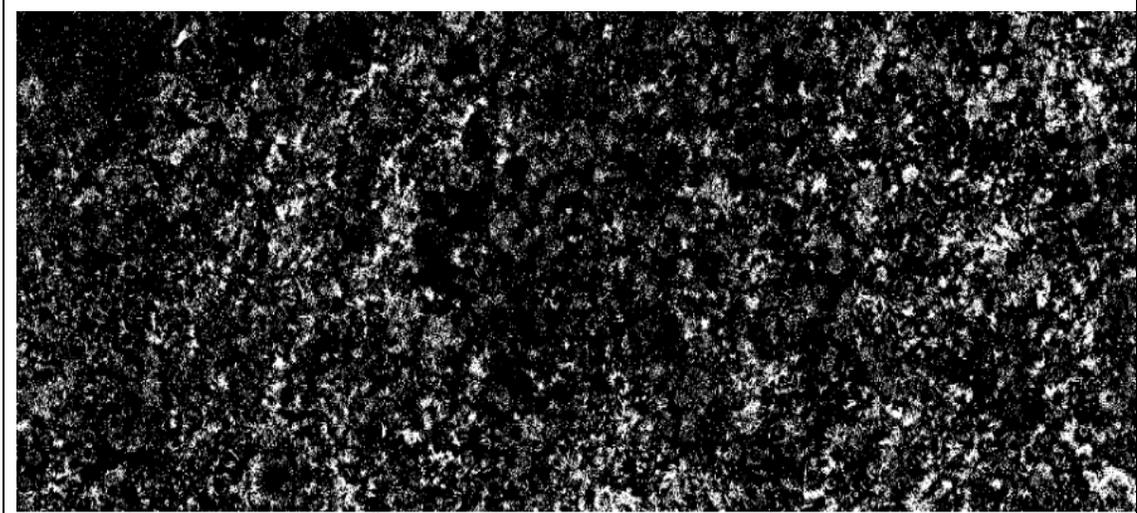


Imagen clasificada

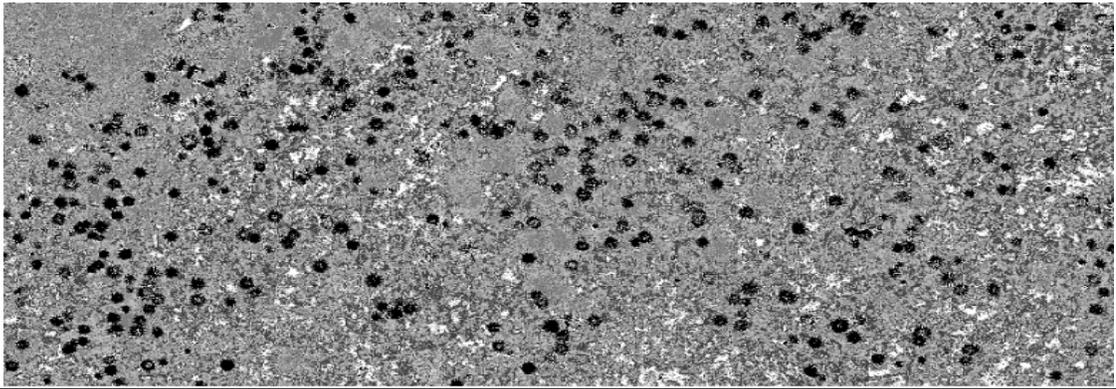


Imagen clasificada con identificación de frailejones

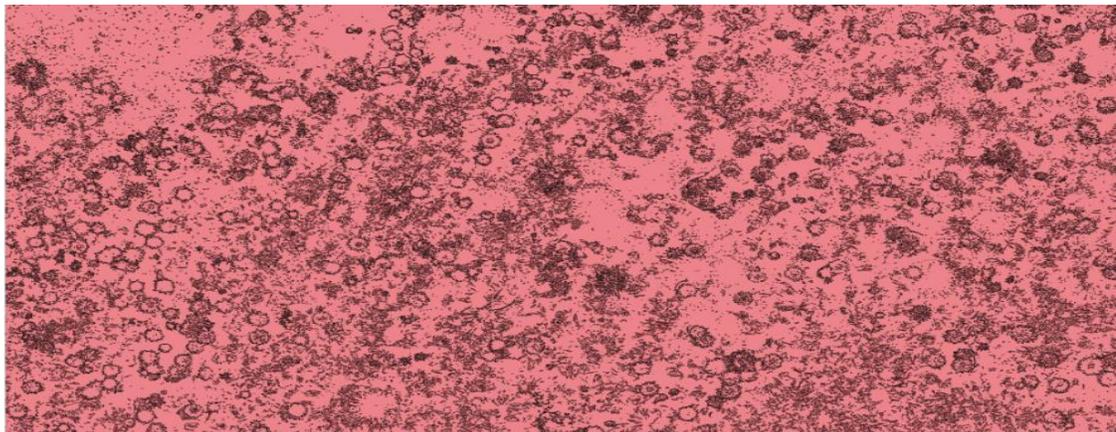
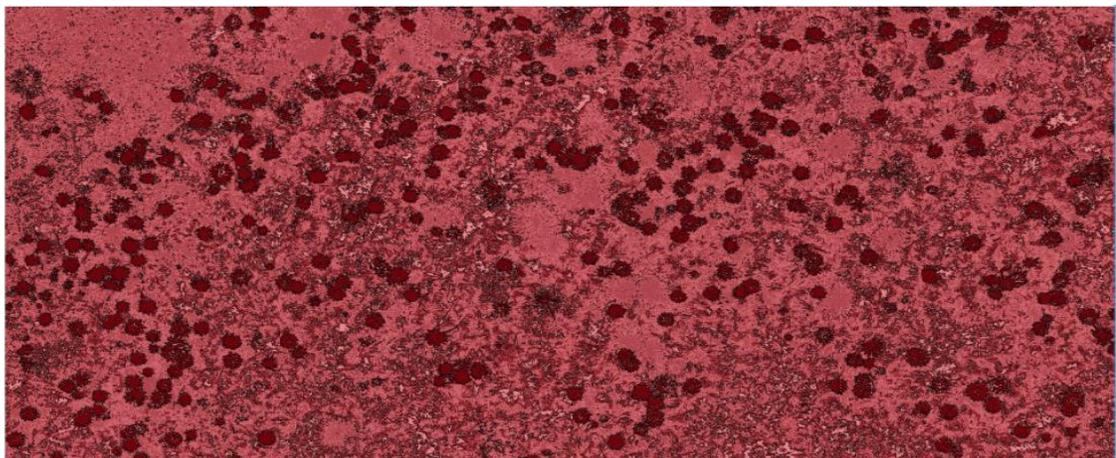


Imagen de copas de frailejones



Elaborado por: D. León

