

Trabajo Fin de Grado

Ingeniería de las Tecnologías de Telecomunicación

Desarrollo de análisis de imagen de dron en el rango visible 4K.

Autor: María Auxiliadora Peña Moreno

Tutor: Prof. Dr. Emilio Gómez González

Dpto. de Física Aplicada III
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2022



Trabajo Fin de Grado
Ingeniería de las Tecnologías de Telecomunicación

Desarrollo de análisis de imagen de dron en el rango visible 4K.

Autor:

María Auxiliadora Peña Moreno

Tutor:

Prof. Dr. Emilio Gómez González

Catedrático

Dpto. de Física Aplicada III
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2022

Trabajo Fin de Grado: Desarrollo de análisis de imagen de dron en el rango visible 4K.

Autor: María Auxiliadora Peña Moreno

Tutor: Prof. Dr. Emilio Gómez González

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

A mi familia
A mis amigos

Agradecimientos

En primer lugar, quiero agradecer al que ha sido mi tutor en este trabajo fin de grado. Prof. Emilio Gómez, por los consejos que me ha transmitido a la hora de enfocar el proyecto.

Quiero agradecer a mi familia por apoyarme en continuar mis estudios en momentos difíciles de la carrera.

María Auxiliadora Peña Moreno

Sevilla, 2022

Con el paso del tiempo y acentuándose tras situaciones como las vividas tras la pandemia, surge un crecimiento exponencial de la digitalización para llegar a un aumento de la eficiencia. La digitalización de la agricultura está en un proceso de crecimiento en España, con la finalidad de ahorrar recursos y mejorar el manejo del terreno.

En la actualidad, tenemos dispositivos muy avanzados para poder explorar tanto los terrenos como recintos y extraer información sin apenas desplazamiento. Por ello, se han diseñado drones capaces de realizar todo esto ya sea de forma automatizada o con un posterior procesado de imágenes con programas externos.

El interés de este trabajo es extraer características del terreno, agrícolas y ambientales, tomando imágenes a dos alturas en diferentes puntos usando un dron comercial de “usuario”, DJI Mavic Mini, con su correspondiente procesado de imágenes con la interfaz creada mediante códigos en el programa Matlab® y realizando un análisis en el rango visible 4K.

Abstract

Over the years and accentuating after situations such as those experienced after the pandemic, an exponential growth of digitization arises to reach an increase in efficiency. The digitization of agriculture is in a process of growth in Spain, in order to save resources and improve land management.

Currently, we have very advanced devices to be able to explore both the land and the enclosures and extract information with hardly any movement. For this reason, drones capable of doing all this have been designed either automatically or with subsequent image processing with external programs.

The interest of this work is to extract characteristics of the land, agricultural and environmental, taking images at two heights at different points using a commercial "user" drone, DJI Mavic Mini, with its corresponding image processing with the interface created by means of codes in the Matlab® program and performing an analysis in the 4K visible range.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xiv
Índice de Tablas	xvi
Índice de Figuras	xviii
Notación	xxi
1 Introducción	1
1.1. <i>Objetivos</i>	2
1.2. <i>Requisitos del proyecto</i>	4
1.3. <i>Desarrollo del proyecto</i>	4
2 Material y métodos	7
2.1. <i>Descripción del Hardware</i>	7
2.2. <i>Descripción del Software</i>	7
2.3. <i>Dron DJI Mavic Mini</i>	7
2.3.1 <i>Especificaciones técnicas</i>	8
2.4. <i>Imagen RGB</i>	10
2.5. <i>Cálculo de la escala</i>	10
2.6. <i>Cálculo de los índices</i>	11
2.7. <i>Matlab</i>	11
2.8. <i>Fotografías JPEG</i>	12
2.9. <i>Ubicación de las tomas fotográficas</i>	13
3 Descripción de la interfaz AgropDrone	15
3.1 <i>Tratamiento de los datos en el entorno de Matlab®</i>	15
3.2 <i>Ficheros generados por AgropDrone</i>	16
3.3 <i>Diagramas de flujo</i>	16
	18
4 Guía de usuario y Ejemplos de uso	24
4.1 <i>Requisitos del Sistema</i>	24
4.2 <i>Inicio del programa</i>	24
4.3 <i>Procesado de las imágenes</i>	27
4.3.1 <i>Separación de los canales R, G y B</i>	30
4.3.2 <i>Operaciones Morfológicas</i>	31
4.3.3 <i>Ampliación sincronizada de las imágenes</i>	34
4.4 <i>Etiquetado de las imágenes</i>	34
4.5 <i>Cálculo de los índices</i>	36

4.5.1	Por zonas	37
4.5.2	Por índices	48
4.6	<i>Procesado con escala con el Método “Con referencia”.</i>	52
4.7	<i>Guardado de los índices y las propiedades en Excel</i>	52
4.8	<i>Segmentación por color</i>	52
4.9	<i>Programas anexos para ver resultados finales</i>	52
4.10	<i>Ejemplos de uso</i>	53
5	Conclusiones	62
6	Líneas de desarrollo	63
	Bibliografía y referencias	64
	Anexos	65
	<i>Anexo A</i>	<i>67</i>
	<i>Ficheros de datos</i>	<i>67</i>
	<i>Anexo B</i>	<i>74</i>
	<i>Cálculo de índices</i>	<i>74</i>
	<i>Anexo C</i>	<i>80</i>
	<i>Códigos del programa</i>	<i>80</i>

ÍNDICE DE TABLAS

Tabla 1-1. Descripción y fórmulas	3
Tabla 2-1. Especificaciones de la aeronave	8
Tabla 2-2. Especificaciones control remoto y transmisión de vídeo	8
Tabla 2-3. Especificaciones de la cámara	9

Respecto al origen y procedencia de las tablas, la Tabla 1-1 procede de [1], mientras que Tabla 2-1, Tabla 2-2 y Tabla 2-3 proceden de [2]

ÍNDICE DE FIGURAS

Figura 1-1. Imagen con dron vs. Satélite	1
Figura 1-2. Rango visible	3
Figura 2-1. DJI Mavic Mini	7
Figura 2-2. Principales características geométricas de una cámara digital	10
Figura 2-3. Matlab R2022a	12
Figura 2-4. Terreno en barbecho	13
Figura 2-5. Terreno con cultivo y agua	14
Figura 2-6. Terreno con cultivo	14
Figura 3-1. Simbología utilizada	16
Figura 3-2. Diagrama ventana principal	17
Figura 3-3. Identificación ROI	18
Figura 3-4. Dividir imágenes	19
Figura 3-5. Identificar regiones	20
Figura 3-6. Examinar regiones	21
Figura 3-7. Cálculo de áreas	22
Figura 3-8. Cálculo de índices	23
Figura 4-1. Inicio del programa	24
Figura 4-2. Introducción de parámetros iniciales	25
Figura 4-3. Selección carpeta a 20 metros	25
Figura 4-4. Selección carpeta a 5 metros	26
Figura 4-5. Selección cálculo de la escala	27
Figura 4-6. Subcarpetas creadas en la carpeta de las imágenes a h1	28
Figura 4-7. Subcarpetas creadas en la carpeta de las imágenes a h2	28
Figura 4-8. Imagen actual y opción de continuar procesado	29
Figura 4-9. Imagen original con escala	29
Figura 4-10. Imagen separada por canales	30
Figura 4-11. Imagen con nitidez mejorada	30
Figura 4-12. Imagen con mejora de contraste	31
Figura 4-13. Imagen binarizada	32
Figura 4-14. Imagen con relleno de huecos	32

Figura 4-15. Imagen erosionada	33
Figura 4-16. Contorno de la imagen	33
Figura 4-17. Ampliación sincronizada	34
Figura 4-18. Imagen etiquetada	35
Figura 4-19. Imagen original etiquetada	35
Figura 4-20. Elección etiqueta	36
Figura 4-21. Elección del cálculo de los índices	37
Figura 4-22. Original y etiquetada	38
Figura 4-23. Índice 1 de la etiqueta elegida	39
Figura 4-24. Índice 2 de la etiqueta elegida	40
Figura 4-25. Índice 3 de la etiqueta elegida	41
Figura 4-26. Índice 4 de la etiqueta elegida	42
Figura 4-27. Índice 5 de la etiqueta elegida	43
Figura 4-28. Índice 6 de la etiqueta elegida	44
Figura 4-29. Índice 7 de la etiqueta elegida	45
Figura 4-30. Índice 8 de la etiqueta elegida	46
Figura 4-31. Índice 9 de la etiqueta elegida	47
Figura 4-32. Índices de la etiqueta elegida	48
Figura 4-33. Por índices	49
Figura 4-34. Índice elegido de la etiqueta seleccionada	50
Figura 4-35. Opción de cierre de figuras	51
Figura 4-36. Imagen de referencia para escala	52
Figura 4-37. Escenario 1 a 20 metros	53
Figura 4-38. Índices de la etiqueta	54
Figura 4-39. Escenario 1 a 5 metros	54
Figura 4-40. Índices de la etiqueta	55
Figura 4-41. Escenario 2 a 20 metros	56
Figura 4-42. Índices de la etiqueta	57
Figura 4-43. Escenario 2 a 5 metros	57
Figura 4-44. Índices de la etiqueta	58
Figura 4-45. Escenario 3 a 20 metros	59
Figura 4-46. Índices de la etiqueta	60
Figura 4-47. Escenario 3 a 5 metros	60
Figura 4-48. Índices de la etiqueta	61

Respecto al origen y procedencia de las imágenes, la Figura 1-1 procede de [3], la Figura 1-2 se obtuvo de [4], la Figura 2-1 procede de [5].

Notación

e.o.c.	En cualquier otro caso
ROI	Región de interés (region of interest)
H	Altura
ω	Ángulo FOV

1 INTRODUCCIÓN

La agricultura se dice que tuvo su nacimiento en el Neolítico y revolucionó la historia, cambiando la supervivencia humana y el modo de vida al completo. Con el paso de los años, ha ido evolucionando con cambios en el cultivo y su tratamiento, y han surgido necesidades como la digitalización de la agricultura, con el fin de aumentar la eficiencia en la producción. La digitalización de la agricultura y la ganadería está en un proceso de crecimiento en España, con el fin de ahorrar recursos, mejorar el manejo y la gestión de las explotaciones agrícolas y por supuesto, la calidad de vida del agricultor.

Actualmente, tenemos dispositivos muy avanzados para poder controlar y gestionar los campos sin apenas desplazamiento y extraer información, siendo muchas las ventajas y formando parte de la agricultura de precisión. Por ello, se han utilizado sensores, satélites y drones capaces de realizar todo esto ya sea de forma automatizada o con un posterior procesamiento de imágenes con programas externos. La teledetección con drones da mejores resultados que la satelital en algunos aspectos a la hora de extraer información sobre varios aspectos y mejorar los rendimientos, ya sea en el rango visible o no, aunque un uso combinado de ambas podría dar resultados para el monitoreo del cultivo.



Figura 1-1. Imagen con dron vs. Satélite

[3]

La labor principal de los drones es la monitorización de las parcelas agrícolas, que captando imágenes y tratándolas es capaz de arrojar datos de manera precisa como puede ser: estado de las plantas, deficiencias nutricionales, estrés hídrico del cultivo, presencia de plagas y malas hierbas. En definitiva, el agricultor obtiene datos muy precisos y fiables del estado del cultivo.

En cuanto a la hora de aplicar productos fitosanitarios, la ventaja de aplicarlo con drones es el ahorro de material y de tiempo, ya que es capaz de identificar donde es precisa la aplicación del producto y es un beneficio para el medio ambiente.

A pesar del abanico de posibilidades que da el uso de drones en la agricultura, hay que tener en cuenta la legislación sobre el uso de ellos además de pilotarlos personas cualificadas, además de su elevado precio de adquisición y muchos agricultores aún son reacios en España a dar este salto a la hora de manejar sus terrenos.

1.1. Objetivos

El Objetivo General de este trabajo es diseñar y programar una herramienta (programa) de procesamiento de imágenes aéreas tomadas mediante un vehículo autónomo no tripulado (dron comercial) dotado de una cámara de imagen convencional en el rango visible de alta resolución (4K).

Las imágenes se tomaron a dos alturas diferentes, 20 metros y 5 metros, en diferentes puntos con un dron DJI Mavic Mini, para extraer características del terreno, agrícolas y ambientales, mediante una interfaz programada en el entorno Matlab®. Esta interfaz se ha denominado “AgropDrone”.

El proceso de obtención y análisis de las imágenes corresponde a los siguientes Objetivos Específicos (OE):

OE 1. Toma de imágenes a las alturas estipuladas para el análisis, teniendo en cuenta que contengan diferentes zonas.

OE 2. Diseño y programación en Matlab® de los algoritmos para

- a. Selección de las carpetas contenedoras de las imágenes.
- b. Obtención de los tres canales (formato RGB) correspondientes a los colores rojo (*red*, R), verde (*green*, G) y azul (*blue*, B).
- c. Segmentación interactiva de las imágenes, determinando las zonas de interés en las mismas.
- d. Análisis de las imágenes, mejorando su aspecto visual, y cálculo de parámetros (índices cuantitativos) en las zonas de interés de las mismas.

OE 3. Realización de pruebas de funcionamiento.

OE 4. Análisis de un conjunto de imágenes de prueba y cálculo de los índices referentes a características del terreno.

Los referidos índices han sido definidos a partir de los habitualmente utilizados para extraer características del terreno, agrícolas y ambientales mediante cámaras hiperespectrales en los rangos visible (VIS) e infrarrojo cercano (*near infrared*, NIR) mostradas en la Tabla 1-1. [6]. Debe tenerse en cuenta que la cámara del dron disponible registra imágenes únicamente en el referido rango visible (de 400 nm a 700 nm), en forma de “imagen en color” con los tres canales RGB. En este Trabajo se propone, por tanto, utilizar las versiones adaptadas de los referidos nueve índices (descritos en la sección 2.6) a los tres canales disponibles, desarrollando un programa que pueda ser actualizado con otras posibles versiones de tales parámetros que puedan extraer información específica de interés.

La interfaz desarrollada (AgropDron) ha sido utilizada para procesar un total de 124 fotografías tomadas con dron a diferentes alturas (5 m y 20 m). Todas las imágenes han sido procesadas y etiquetadas, habiéndose calculados los índices correspondientes.

Las fotografías han sido tomadas por la autora (piloto titulada de drones con subcategorías A1/A3) en diferentes zonas de Cádiz y Sevilla, previa consulta de la aplicación ENAIRE® para el permiso del vuelo, con características de terreno diferentes para así probar su funcionamiento. Asimismo, se han analizado imágenes de diferentes tamaños y resoluciones.

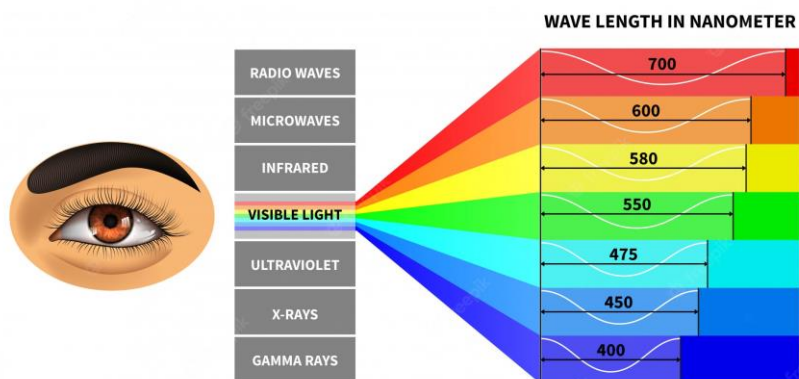


Figura 1-2. Rango visible

[4]

Tabla 1-1. Descripción y fórmulas

TABLE III
DESCRIPTIONS AND FORMULAS OF VIs INVESTIGATED IN THIS STUDY

Index	Description	Formula	Reference
Existing indices			
NDVI	NDVI	$\frac{\rho_{NIR} - \rho_{red}}{\rho_{NIR} + \rho_{red}}$	[15]
NDVI _{Red-edge}	Red-edge NDVI	$\frac{\rho_{NIR} - \rho_{RE}}{\rho_{NIR} + \rho_{RE}}$	[41]
MSR	MSR index	$\frac{\rho_{NIR}/\rho_{red} - 1}{\sqrt{\rho_{NIR}/\rho_{red} + 1}}$	[38]
MSR _{Red-edge}	Red-edge MSR index	$\frac{\rho_{NIR}/\rho_{RE} - 1}{\sqrt{\rho_{NIR}/\rho_{RE} + 1}}$	[34]
CI _{green}	Green CI	$\frac{\rho_{NIR}}{\rho_{green}} - 1$	[42]
CI _{Red-edge}	Red-edge CI	$\frac{\rho_{NIR}}{\rho_{RE}} - 1$	[43]
Improved indices			
NDVI _{red&RE}	Red and red-edge NDVI	$\frac{\rho_{NIR} - (a^* \rho_{red} + (1-a)^* \rho_{RE})}{\rho_{NIR} + (a^* \rho_{red} + (1-a)^* \rho_{RE})}$	–
MSR _{red&RE}	Red and red-edge MSR index	$\frac{\rho_{NIR} / (a^* \rho_{red} + (1-a)^* \rho_{RE}) - 1}{\sqrt{\rho_{NIR} / (a^* \rho_{red} + (1-a)^* \rho_{RE}) + 1}}$	–
CI _{red&RE}	Red and red-edge modified CI	$\frac{\rho_{NIR}}{a^* \rho_{red} + (1-a)^* \rho_{RE}} - 1$	–

*NIR refers to near infrared; RE refers to red-edge; $a \in [0, 1]$

[1]

1.2. Requisitos del proyecto

Requisitos y funcionalidades

- Tener dos carpetas con las imágenes a 20 metros y a 5 metros de altura.
- Análisis individuales por separado de cada imagen de ambas carpetas.
- Posibilidad de abrir archivos jpg.
- Posibilidad de abrir archivos.xlsx.

Requisitos de equipamiento

Serán necesarios los componentes siguientes:

Ordenador compatible con capacidad de soportar el software necesario para el correcto funcionamiento del programa. Como mínimo deberá estar dotado de 8 GB de memoria RAM.

Sistema Operativo: Windows 10® Profesional.

Software matemático: Matlab® R2022a.

Sistema Ofimático: Microsoft Office®.

1.3. Desarrollo del proyecto

El proyecto se ha desarrollado siguiendo las siguientes fases relativas a los Objetivos Específicos indicados:

Fase de diseño

- Definición de los valores iniciales del procesado en entorno Matlab®.
- Diseño de la función de cálculo del Umbral Otsu en Matlab®.
- Diseño de los algoritmos de Tratamiento de Imágenes en Matlab®.
- Diseño del flujo del programa.

Fase de programación

- Lectura de las carpetas que contienen las imágenes a ambas alturas en Matlab®.
- Cálculo de la escala.
- Creación de los algoritmos de tratamiento de datos en Matlab®.

- Programación de la interfaz gráfica del programa.
- Introducción de los algoritmos desarrollados en la interfaz del programa.
- Creación del algoritmo de extracción de Regiones de Interés (ROI).
- Guardado de valores en Excel.

Fase de cálculos

- Calculo de los índices adaptados a la cámara del dron en referencia a los índices usados en cámaras VIS NIR.

Fase de pruebas y validación

- Prueba del tratamiento de las imágenes con dos nuevas interfaces a las distintas alturas.
- Pruebas de los algoritmos de cálculo de índices adaptados a la cámara usada.
- Prueba de salida de datos.

2 MATERIAL Y MÉTODOS

2.1. Descripción del Hardware

Procesador: Intel® Core i3 CPU 3.20 GHz

RAM: 8GHz

2.2. Descripción del Software

Sistema Operativo: Windows 10® Profesional

Software Matemático: Matlab® R2022a.

Software Ofimático: Microsoft Office®.

2.3. Dron DJI Mavic Mini

Este dron es de los considerados como drones de “usuario” o lúdicos con el que podremos volar como máximo a 120 metros de altura y permitirá grabaciones en imagen estática (fotografía) y video en alta resolución (formato 4K, 4000 px × 2250 px, con relación de aspecto 16:9, como se detalla en la Tabla 2-3).



Figura 2-1. DJI Mavic Mini

[5]

2.3.1 Especificaciones técnicas

Las especificaciones de este equipo han sido obtenidas en [2].

Tabla 2-1. Especificaciones de la aeronave

Peso de despegue (con la batería, las hélices y una tarjeta SD)	249 gramos
Dimensiones	Plegada: 140 × 81 × 57 mm Desplegada: 159 × 202 × 55 mm Desplegada (con hélices): 245 × 289 × 55
Velocidad máxima de ascenso	4 m/s (modo S) 2 m/s (modo P) 1.5 m/s (modo C)
Velocidad máxima de descenso	3 m/s (modo S) 1.8 m/s (modo P) 1 m/s (modo C)
Tiempo máximo de vuelo	30 minutos (medición realizada sin viento a 14 km/h)
Resistencia máxima al viento	8 m/s (escala 4)
Rango de temperatura de funcionamiento	De 0 a 40 °C (de 32° a 104°F)
Rango de precisión en vuelo estacionario	Vertical: ±0.1 m (con posicionamiento visual), ±0.5 m (con posicionamiento por GPS) Horizontal: ±0.3 m (con posicionamiento visual), ±1.5 m (con posicionamiento por GPS)

[2]

Tabla 2-2. Especificaciones control remoto y transmisión de vídeo

Capacidad de la batería	2600 mAh
Corriente de funcionamiento	1200 mA 3.6 V (Android) 450 mA 3.6 V (iOS)
Sistema de transmisión de vídeo	Wi-Fi mejorado

Latencia (dependiendo de las condiciones del entorno y del dispositivo móvil)	170-240 ms
---	------------

[2]

Tabla 2-3. Especificaciones de la cámara

Sensor	1/2.3" CMOS Píxeles efectivos: 12 MP
Objetivo	FOV: 83° Formato equivalente a 35 mm: 24 mm Apertura: f/2.8 Distancia de enfoque: 1 m a ∞
Rango ISO	Vídeo: 100-3200 (automático) 100-3200 (manual) Foto: 100-1600 (automático) 100-3200 (manual)
Velocidad de obturación	Obturador electrónico: 4-1/8000s
Tamaño de fotografía	4:3: 4000×3000 16:9: 4000×2250
Modos de fotografía	Disparo único Intervalo: 2/3/5/7/10/15/20/30/60 s
Resolución de vídeo	2.7K : 2720×1530 24/25/30 p FHD : 1920×1080 24/25/30/48/50/60 p
Formatos de fotografía	JPEG
Formatos de vídeo	MP4 (H.264/MPEG-4 AVC)

[2]

2.4. Imagen RGB

La codificación RGB (rojo-verde-azul) es una composición espectral -y, por tanto, del color- de la luz emitida por una fuente (mezcla aditiva) mediante la cuantificación de los pesos relativos de cada una de las tres contribuciones rojo, verde y azul en un formato de señal digital. Cada uno de los tres canales R, G y B tiene un valor comprendido entre 0 (negro) y 255 (el color más intenso) en cada punto de la imagen [7].

2.5. Cálculo de la escala

El cálculo de la escala de las imágenes es necesario para la correcta determinación de las distancias y áreas en las mismas. Para determinarla, usaremos la relación geométrica entre el campo de visión lineal y angular registrados por la cámara del dron a la altura de vuelo del mismo.

El campo de visión lineal (*field of view*, FOV) es una medida del tamaño de la región del espacio (escena) captada por un instrumento óptico. Corresponde a la diagonal del rectángulo imaginario que enmarca la escena y se expresa en unidades de longitud (metros) [7]. Su valor depende de la distancia de trabajo (*working distance*, WD), que se puede considerar del orden de la distancia de enfoque (DE), $WD \approx DE$ a la que se mide. Siendo ω la mitad del ángulo correspondiente al campo de visión real (*real field of view*, RFOV).

$$tg(2\omega) = tg(RFOV) = \frac{FOV}{2 \cdot WD}$$

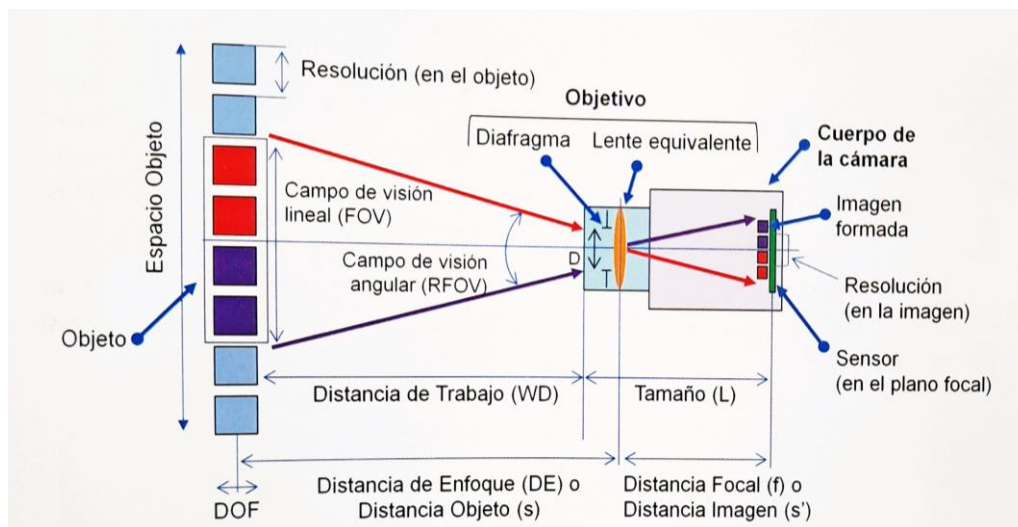


Figura 2-2. Principales características geométricas de una cámara digital

[7]

La escala también se puede determinar mediante la inserción de una referencia de tamaño conocido en el campo de visión registrado por la cámara y la correspondiente medida de los píxeles a los que corresponden sus lados. En este proyecto se propone el uso de una hoja de papel (tamaño A4, 29,7 cm x 21,0 cm) por su sencillez y disponibilidad generalizada.

2.6. Cálculo de los índices

El cálculo de los índices se realiza con una adaptación para la cámara del dron VIS RGB usado de las correspondientes a las realizadas con cámaras VIS NIR [1]. Cada uno se visualizará usando el mapa de colores (*colormap*) indicado.

$$BR-R = (B - R)/R$$

$$GR-R = (G - R)/R$$

$$BG-R = (B - G)/R$$

Colormap Hot

$$BG-G = (B - G)/G$$

$$RG-G = (R - G)/G$$

$$BR-G = (B - R)/G$$

Colormap Winter

$$RB-B = (R - B)/B$$

$$GB-B = (G - B)/B$$

$$RG-B = (R - G)/B$$

Colormap Cool

2.7. Matlab

La plataforma de programación usada para el procesamiento de las imágenes es Matlab®, en concreto la versión R2022a.

Matlab® (abreviatura de **MAT**rix **LAB**oratory, «laboratorio de matrices») es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, macOS y GNU/Linux [8]. Combina un entorno de escritorio perfeccionado para el análisis iterativo y los procesos de diseño con un lenguaje de programación que expresa las matemáticas de matrices y arrays directamente.

También contiene Apps interactivas, que permiten ver cómo funcionan diferentes algoritmos con sus datos. Realizar iteraciones hasta obtener los resultados deseados y, después, genera automáticamente un programa de MATLAB para reproducir o automatizar su trabajo [9].

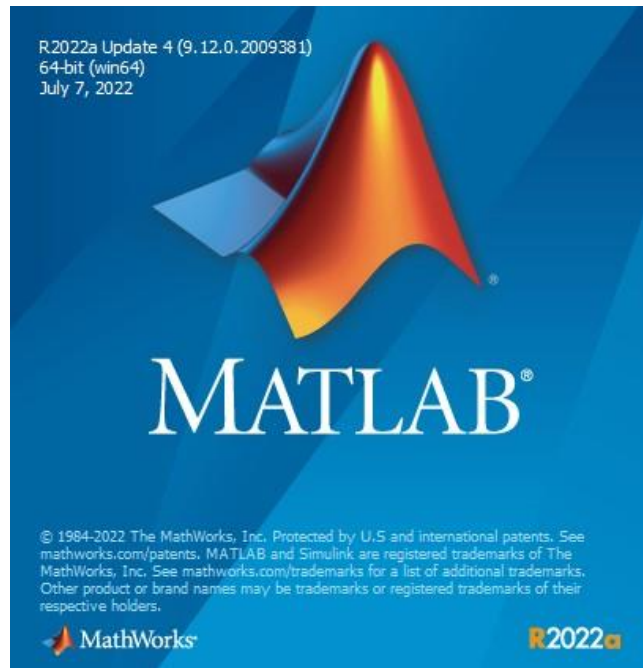


Figura 2-3. Matlab R2022a

Este entorno de desarrollo no llega a ser el óptimo para realizar una computación eficiente, pero si ofrece una manera rápida y fiable de evaluar complejos cálculos numéricos sin tener la necesidad de añadir librerías o de desarrollar funciones adicionales para completar ciertos cálculos, reduciendo el tiempo de desarrollo de resolución del problema.

Finalmente tenemos en cuenta el potente sistema de representación de gráficas con la capacidad de comparación entre ellas y la extracción de valores fácilmente.

2.8. Fotografías JPEG

El formato de las fotografías tomadas con el dron es JPEG, también llamado JPG, y significa Joint Photographic Experts Group, que es el nombre del subcomité que ayudó a crear el estándar JPEG, así como otros estándares de codificación de imágenes fijas bajo el grupo más amplio llamado ISO [10]; es un formato estándar para comprimir fotografías, tanto en color como en escala de grises, con alta calidad.

El archivo JPEG reduce el tamaño de la foto mediante un sistema de compresión que elimina la información innecesaria mediante un procedimiento de comparación de píxeles cercanos. De este modo, los píxeles iguales o similares se eliminan, reduciendo la cantidad de píxeles y por tanto el peso de la imagen [11].

2.9. Ubicación de las tomas fotográficas

Las fotografías fueron tomadas en diferentes puntos en terreno agrícola con vuelo de dron permitido, teniendo en cuenta varios escenarios:

- Terreno en barbecho¹.
- Terreno con cultivo y agua.
- Terreno con cultivo.



Figura 2-4. Terreno en barbecho

¹ El **barbecho** es una técnica de la agricultura en la que la tierra de cultivo se deja sin sembrar durante uno o varios ciclos vegetativos.



Figura 2-5. Terreno con cultivo y agua



Figura 2-6. Terreno con cultivo

3 DESCRIPCIÓN DE LA INTERFAZ AGROPDRONE

La interfaz gráfica AgropDrone con la que se van a procesar las imágenes, a ambas alturas, en cuestión puede ser usada por cualquier usuario sin apenas conocimiento de programación básica, ya que no es necesario el acceso al código de programación de ésta.

Se ha prevalecto la usabilidad frente a la inclusión de complicadas opciones y se ha trabajado para que la entrada inicial de datos sea de los más sencillo posible, sin disminuir la capacidad de cálculo y análisis del programa.

En estas secciones se van a definir las instrucciones de uso de la aplicación diseñada en Matlab® R2022a.

3.1 Tratamiento de los datos en el entorno de Matlab®

Con el fin de facilitar el trabajo con los datos dentro del entorno de programación, al inicio se posibilita la introducción de parámetros iniciales para el posterior tratamiento de las imágenes y posterior cálculo de los índices de características del terreno. Del mismo modo, es posible elegir las carpetas contenedoras de las fotografías tomadas a ambas alturas y son almacenadas en dos variables para facilitar el procesado.

El programa está estructurado de la siguiente forma:

1. Selección de las carpetas que contienen las fotografías a 20 metros y 5 metros.
2. Creación de las subcarpetas donde serán almacenadas las imágenes en sus procesados intermedios, así como en procesado final con etiquetas de las ROI; además de la subcarpeta donde se guardarán las propiedades de las etiquetas y los valores de los índices calculados.
3. Cálculo de la escala, con la posibilidad de realizarlo con “Cámara y altura” o “Con referencia”.
4. Separación de las imágenes en sus tres canales: R, G y B.
5. Filtrado de ruido de las imágenes.
6. Mejora de contraste de las imágenes.
7. Binarización de las imágenes y cálculo del umbral Otsu.
8. Aplicación de Operaciones Morfológicas, dejando la posibilidad al usuario de aplicar otras a las definidas en el inicio del programa.
9. Aplicación a elección del usuario de aplicar la Transformada de Hough.
10. Extracción de las ROI y etiquetado de las imágenes.
11. Cálculo de índices de la etiqueta seleccionada.
12. Segmentación de las imágenes por colores a elección del usuario.

3.2 Archivos generados por AgropDrone

Para la realización de todo el análisis de las imágenes, genera dos “bancos de imágenes” con las fotografías a cada altura definida, así como imágenes intermedias de procesamiento y archivos Excel, con las propiedades de las etiquetas de cada foto y con los valores calculados de los índices de características del terreno.

Formato Matlab® (.m): Este es uno de los formatos propios de Matlab® y que permite ser almacenado y usado posteriormente de forma rápida y eficiente, aunque no es fácil su uso fuera del entorno de Matlab®.

Formato compresión de imágenes (.jpg): Es el tipo de archivo de fotografía más común en la industria digital y en nuestro día a día, tanto en color como en escala de grises en alta calidad.

Formato Microsoft Excel® (.xlsx): Ha sido elegido este formato pensando en la facilidad de trabajo con este tipo de archivos fuera del entorno de Matlab®.

3.3 Diagramas de flujo

En el apartado presente se describen las distintas funciones mediante diagramas de flujo; se puede ver cuál es el proceso que sigue el programa en el avance de la estructura definida (Ver Sección 3.1).

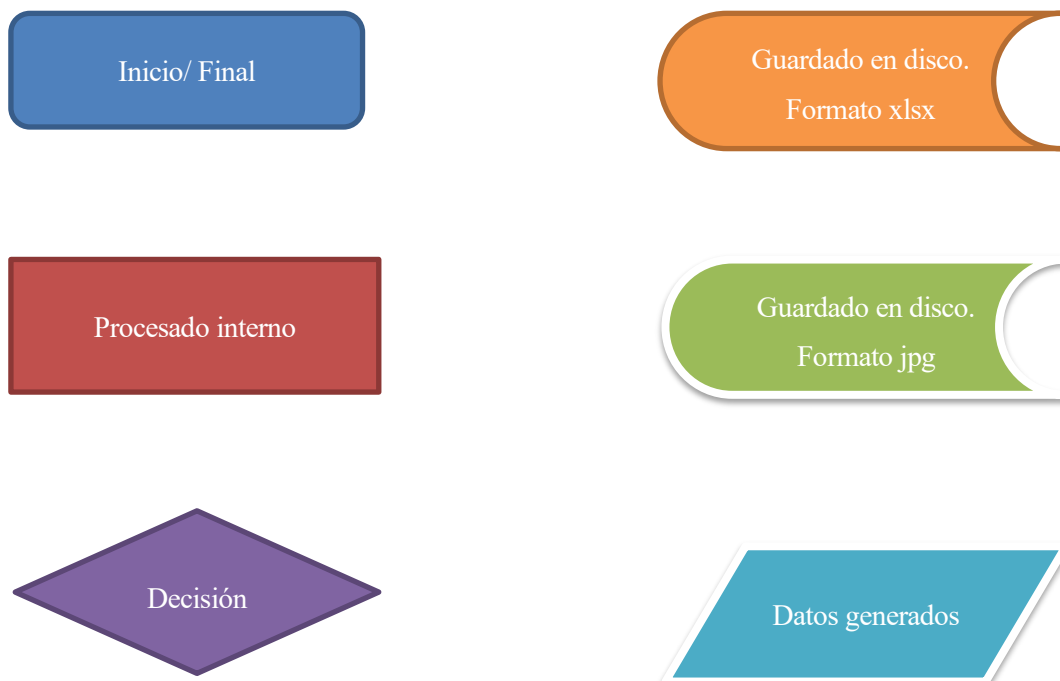


Figura 3-1. Simbología utilizada

- Diagrama de bloque general:

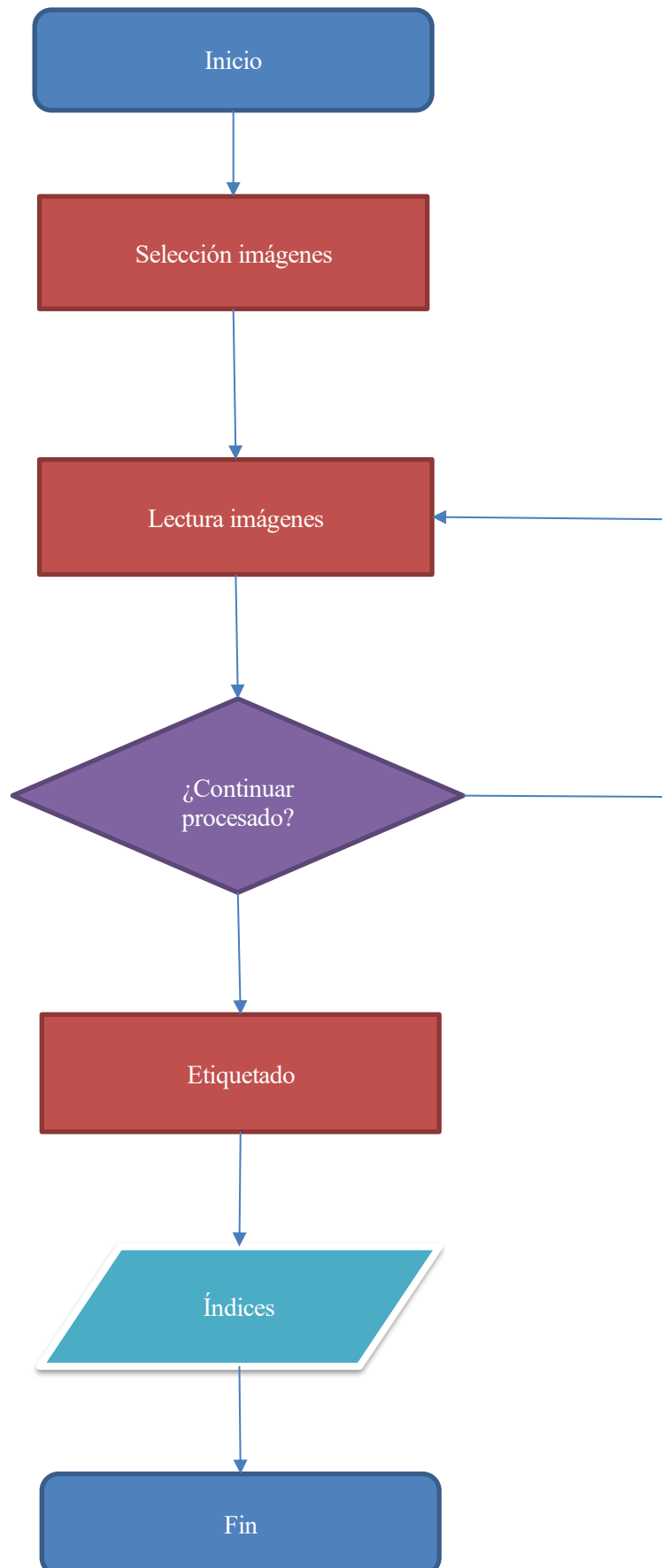


Figura 3-2. Diagrama ventana principal

- Definición Región de Interés:

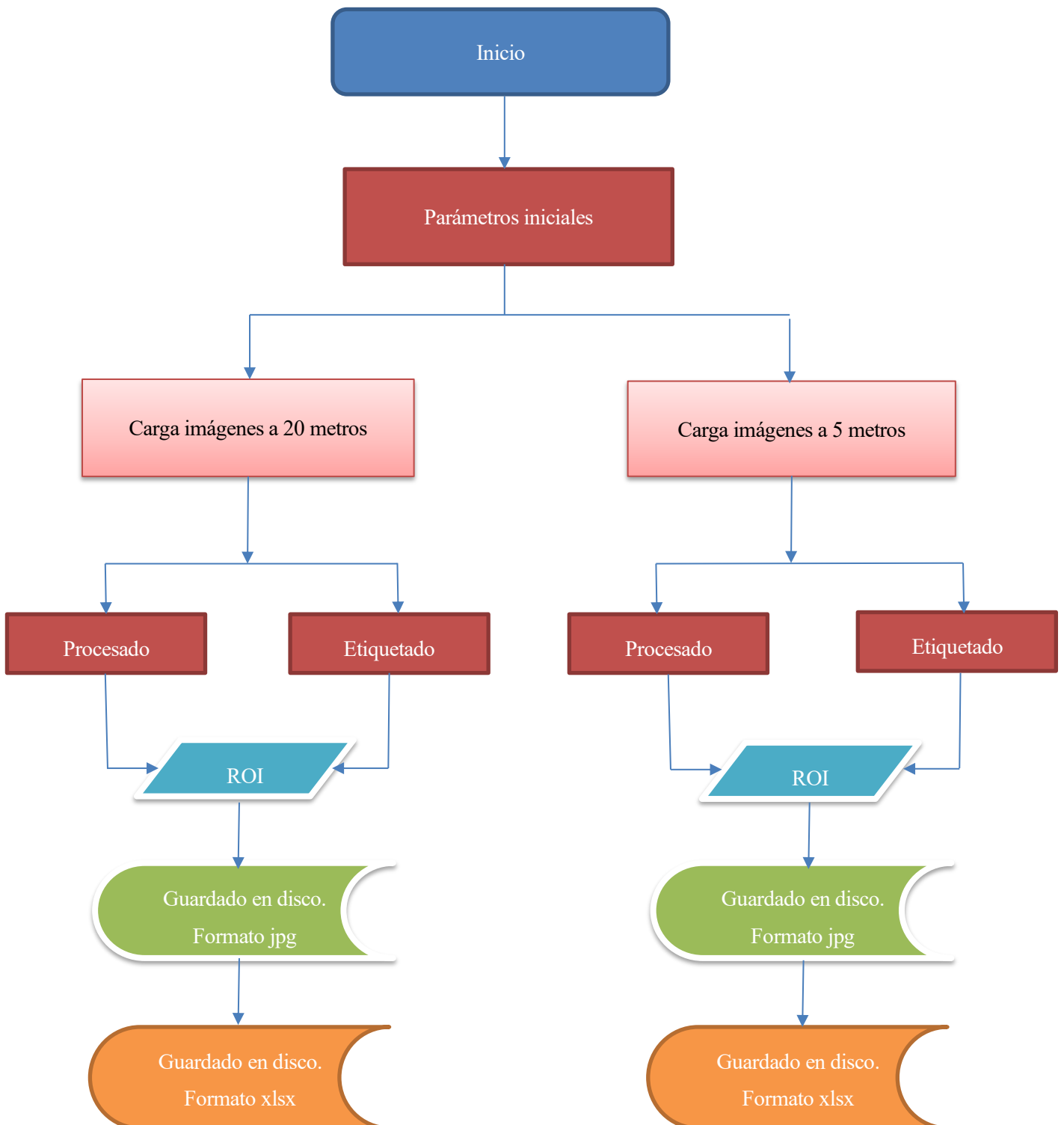


Figura 3-3. Identificación ROI

- Procesado de las imágenes:
 - Dividir imagen en canales R, G y B.

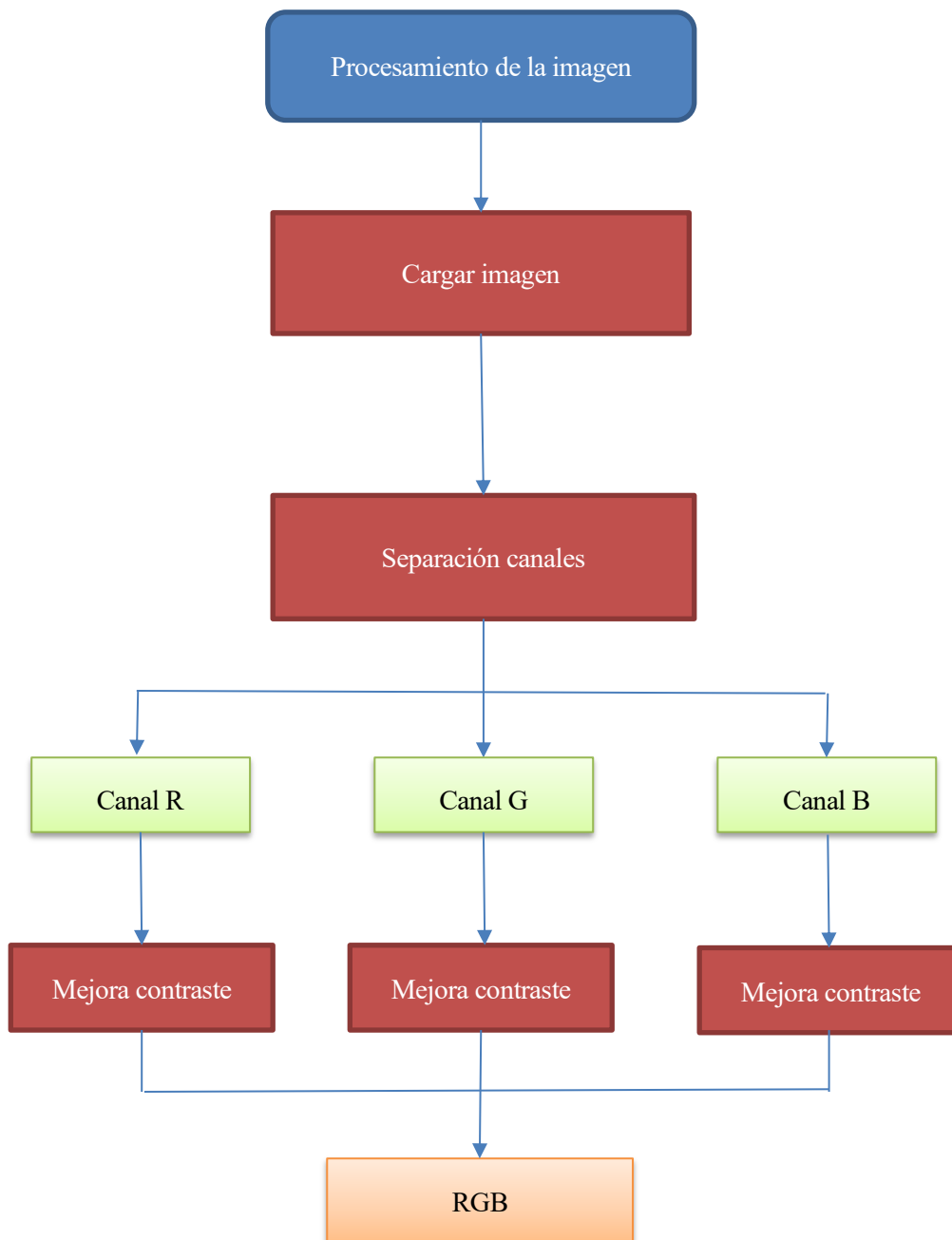


Figura 3-4. Dividir imágenes

- Identificar regiones

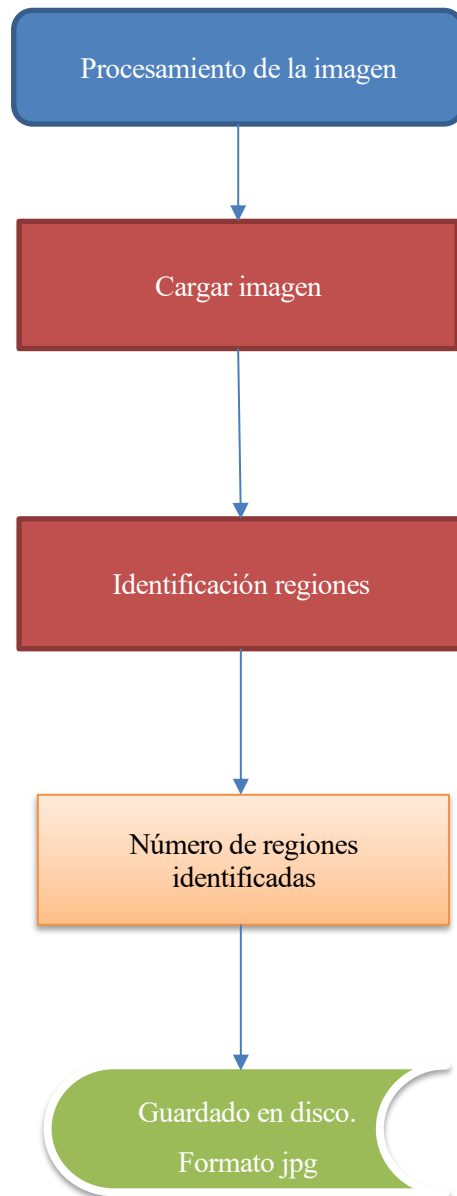


Figura 3-5. Identificar regiones

- Examinar regiones

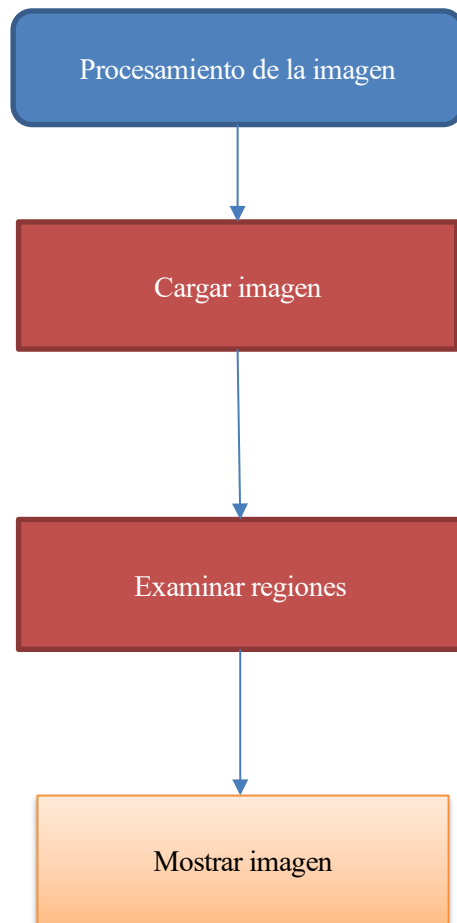


Figura 3-6. Examinar regiones

- Cálculo de áreas

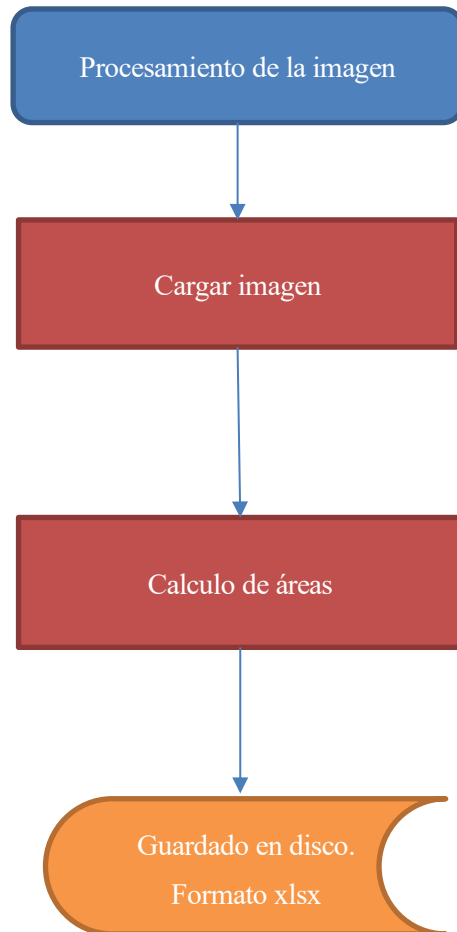


Figura 3-7. Cálculo de áreas

- Cálculo de índices

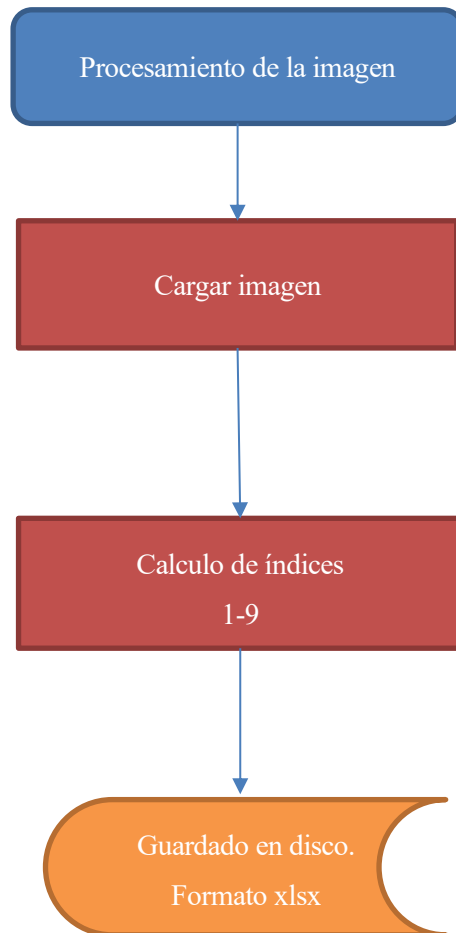


Figura 3-8. Cálculo de índices

4 GUÍA DE USUARIO Y EJEMPLOS DE USO

En esta guía se describe el manejo del programa desarrollado en Matlab®, denominado AgropDrone.

Se ha programado de forma sencilla para que sea para el usuario intuitivo y no tenga dificultad en la ejecución. Se muestran imágenes del programa para mostrar la idea de cómo es su funcionamiento.

4.1 Requisitos del Sistema

Para el uso del programa AgropDrone es recomendable disponer de un ordenador con capacidad para ejecutar **Matlab®** descrito en la sección 2.6 con:

Sistema Operativo Windows® 10.

Software de Hojas de cálculo Microsoft Excel® 2016 o posterior.

Funciones que componen el programa AgropDrone.

Este proyecto ha sido desarrollado en un ordenador personal cumpliendo dichas características y desarrollándose sin problemas.

4.2 Inicio del programa

Para iniciar el programa es necesario abrir el programa Matlab® e ir a la carpeta que contiene las funciones de AgropDrone. Una vez situados en dicha carpeta, abrimos el archivo 'AgropDrone.m' y definimos los 'flags' tomando la decisión de mostrar todas las imágenes que forman el procesado, así como realizar un dilatado en la extracción de la ROI de mayor tamaño. Una vez definido, tecleamos en la línea de comandos de Matlab® el nombre del programa, teniendo en cuenta las mayúsculas y minúsculas:

```
> AgropDrone
```

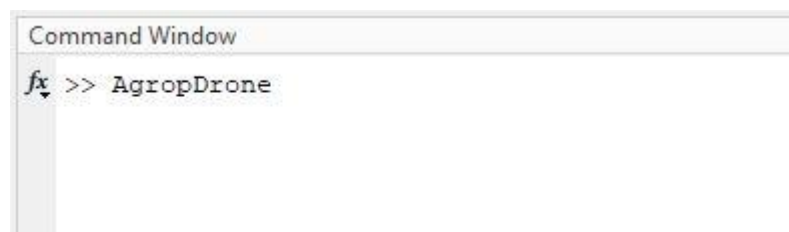
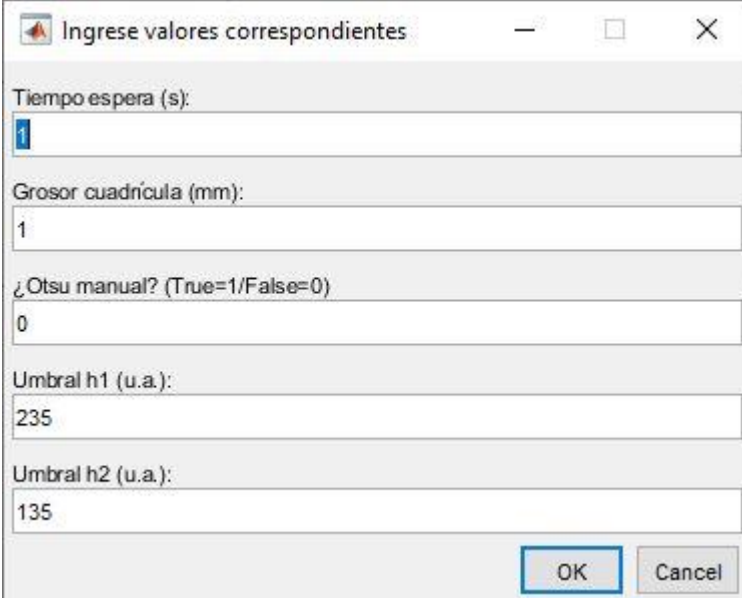


Figura 4-1. Inicio del programa

Tras ejecutarse el archivo 'AgropDrone.m', se abrirá una ventana emergente para introducir los parámetros iniciales previo al procesado de las imágenes.



Ingrese valores correspondientes

Tiempo espera (s):
1

Grosor cuadrícula (mm):
1

¿Otsu manual? (True=1/False=0)
0

Umbral h1 (u.a.):
235

Umbral h2 (u.a.):
135

OK Cancel

Figura 4-2. Introducción de parámetros iniciales

Posteriormente, nos pedirá que seleccionemos las carpetas que contienen las imágenes a las dos alturas, $h_1=20$ metros y $h_2=5$ metros y comenzará el procesado de las imágenes.

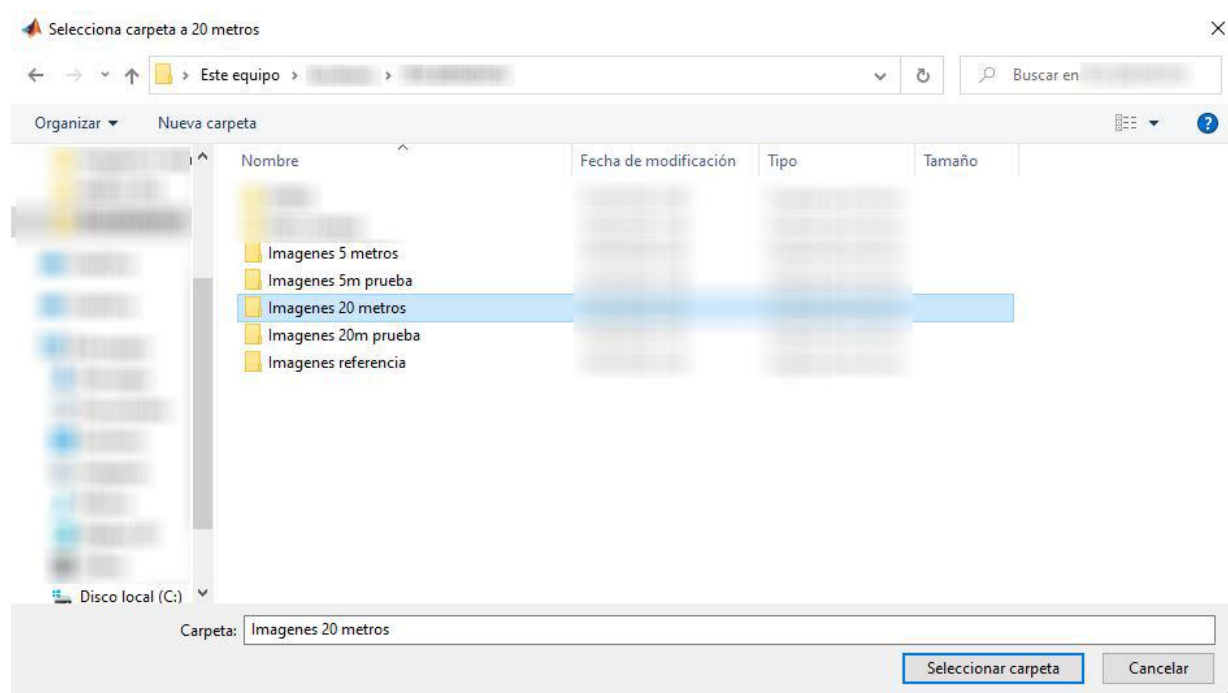


Figura 4-3. Selección carpeta a 20 metros

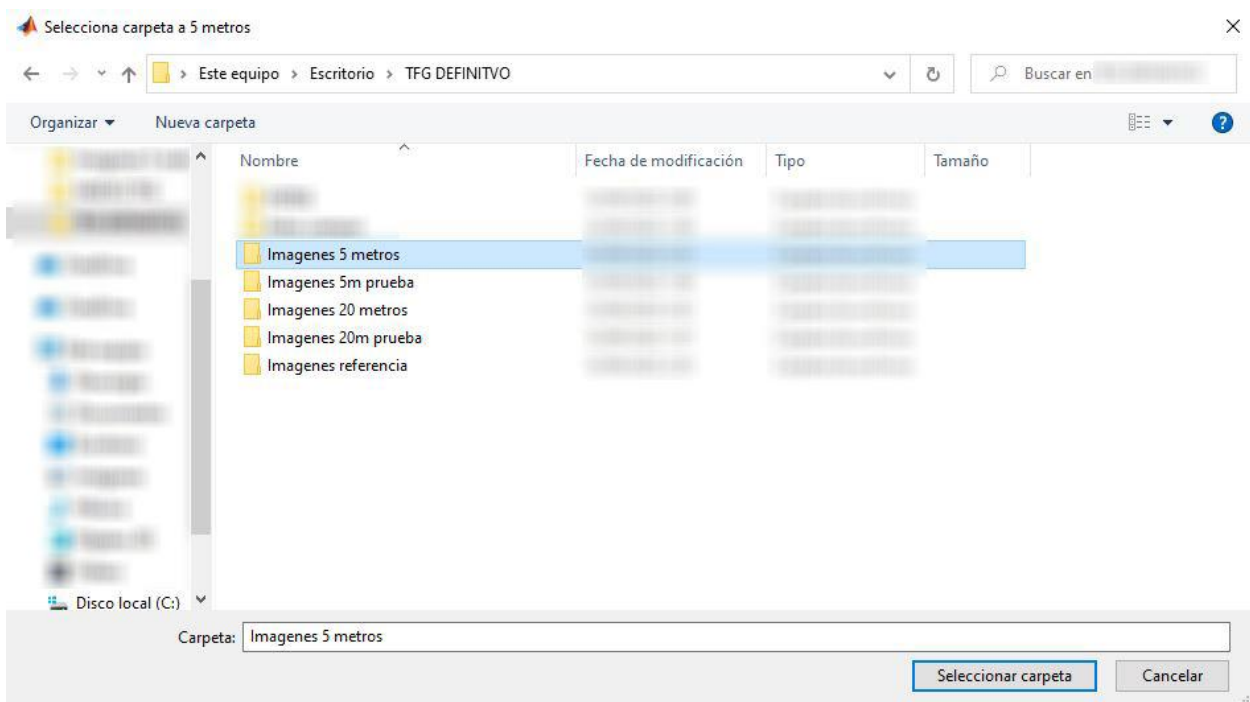


Figura 4-4. Selección carpeta a 5 metros

4.3 Procesado de las imágenes

Una vez que han sido cargadas en memoria ambas carpetas con las fotografías tomadas con el dron a las dos alturas estipuladas en la definición del programa, el programa nos pide que seleccionemos de qué forma calculamos la escala entre los métodos disponibles u omitirla como viene descrito en la sección 2.5.

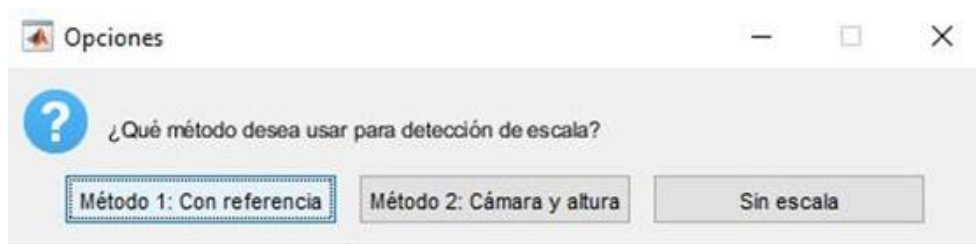


Figura 4-5. Selección cálculo de la escala

- *Lado del píxel ($h1 = 20 m$)* $\approx 0,9$ cm
- *Lado del píxel ($h2 = 5 m$)* $\approx 0,2$ cm

Puede aparecer el **efecto del volumen parcial** como ocurre en imágenes médicas.

A continuación, se crean las subcarpetas donde se almacenará las imágenes en su procesado intermedio, etiquetadas, índices, y las que guardaran los archivos Excel con las propiedades de las etiquetas y de los índices calculados como viene mostrado en la sección 2.6.

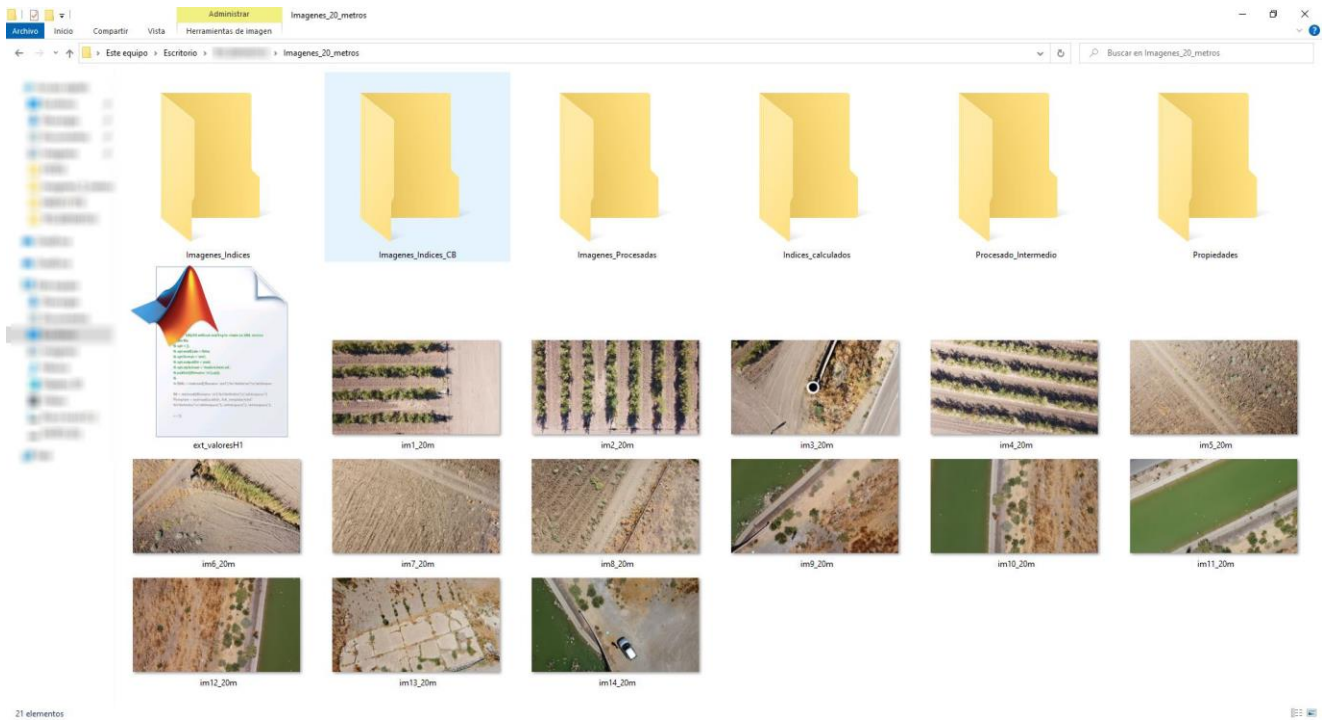


Figura 4-6. Subcarpetas creadas en la carpeta de las imágenes a h1

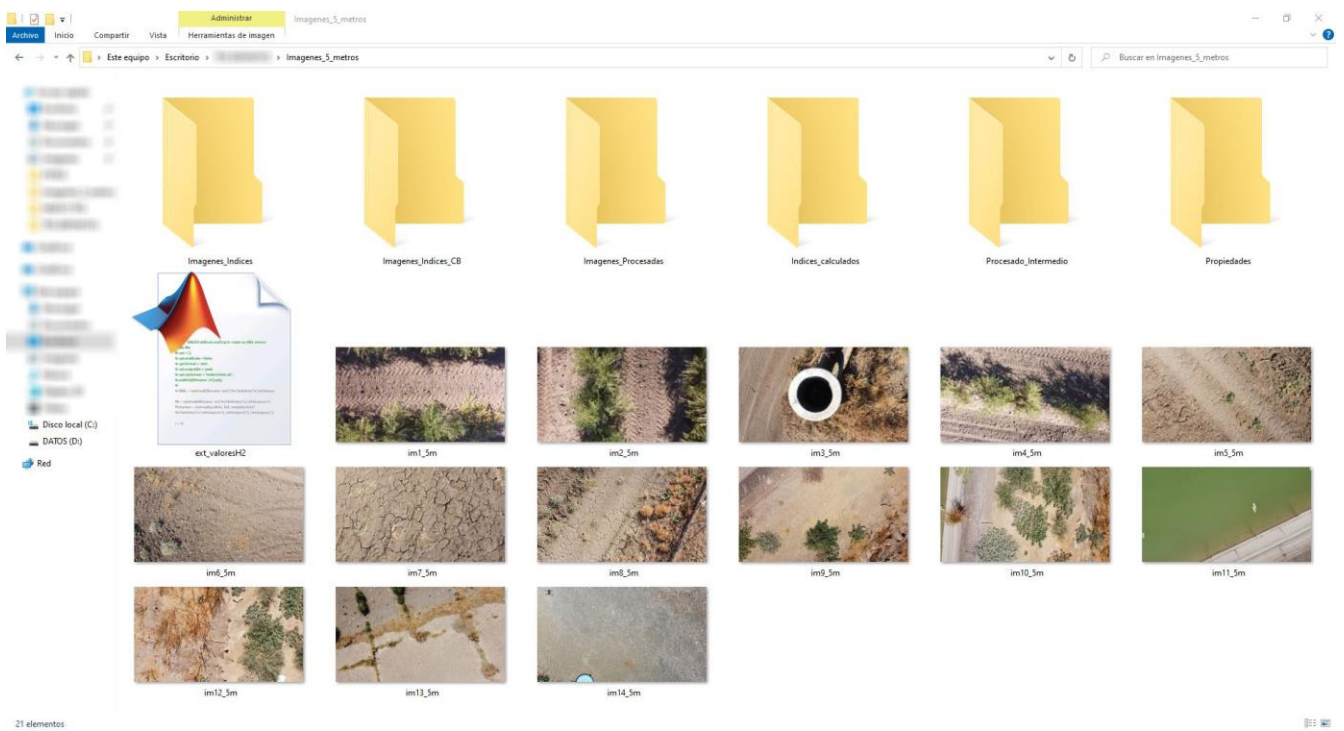


Figura 4-7. Subcarpetas creadas en la carpeta de las imágenes a h2

Finalmente, se recorre en primer lugar la carpeta de las imágenes a 20 metros y tras ser todas procesadas, comienza a recorrer la correspondiente a 5 metros.

En cada iteración en la lectura de las imágenes de cada carpeta (a 20 metros y a 5 metros) nos muestra la imagen actual en pantalla y nos da la opción de continuar el procesado de la fotografía o pasar a la siguiente.

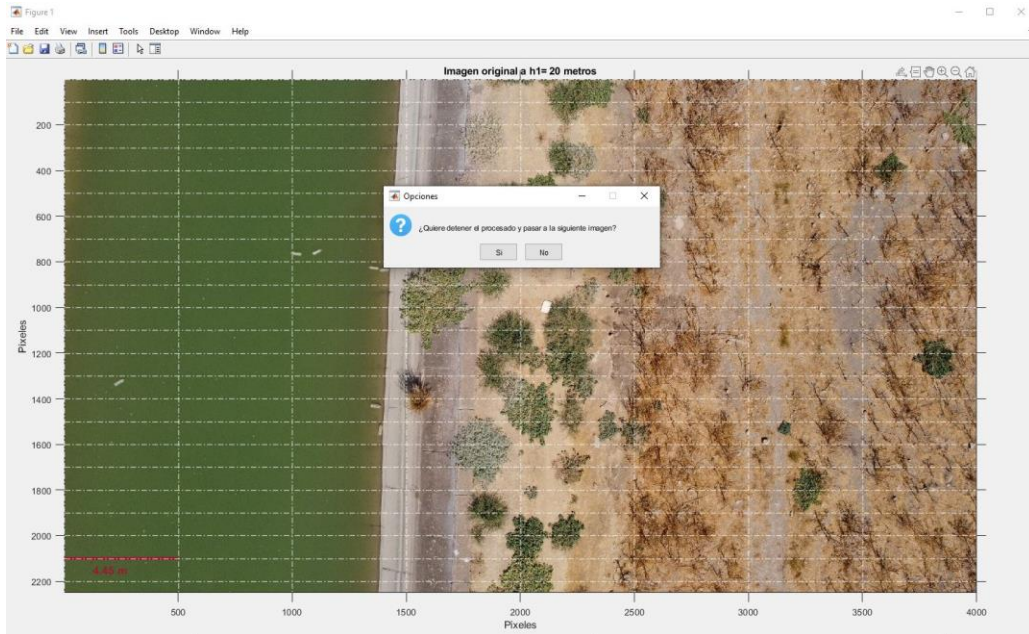


Figura 4-8. Imagen actual y opción de continuar procesado

Si decidimos que continúe el procesado, continuará la estructura del programa hasta el cálculo de los índices como está descrito en la sección 3.1.

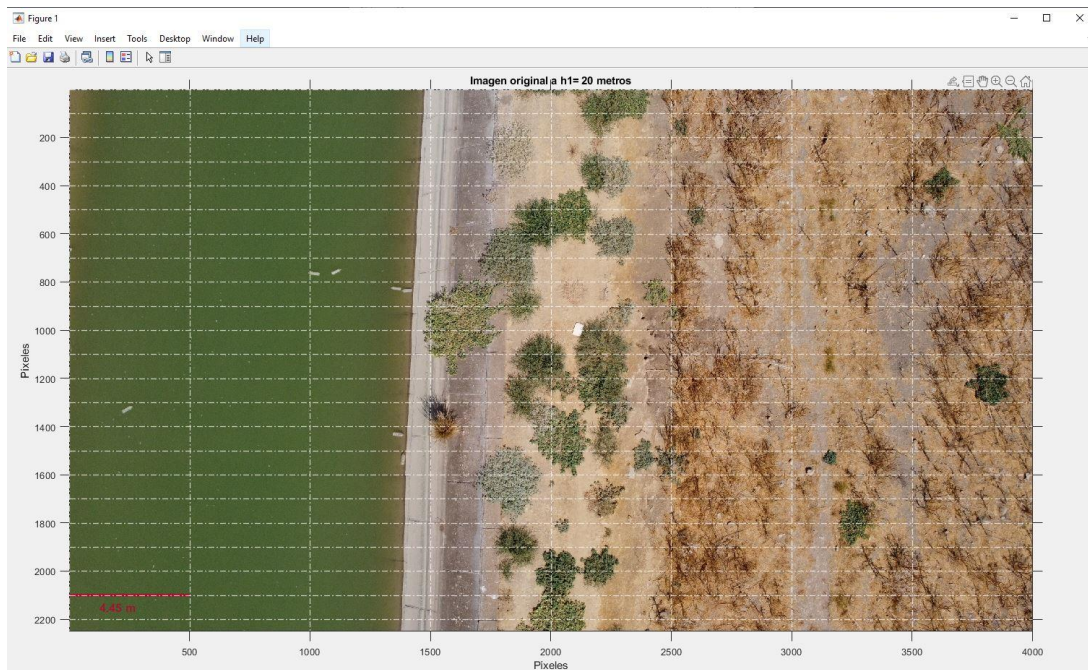


Figura 4-9. Imagen original con escala

4.3.1 Separación de los canales R, G y B

Separo las imágenes en sus tres canales R, G y B para su mejora por cada canal y posterior unión en tres dimensiones, RGB, y continuar con el procesado.

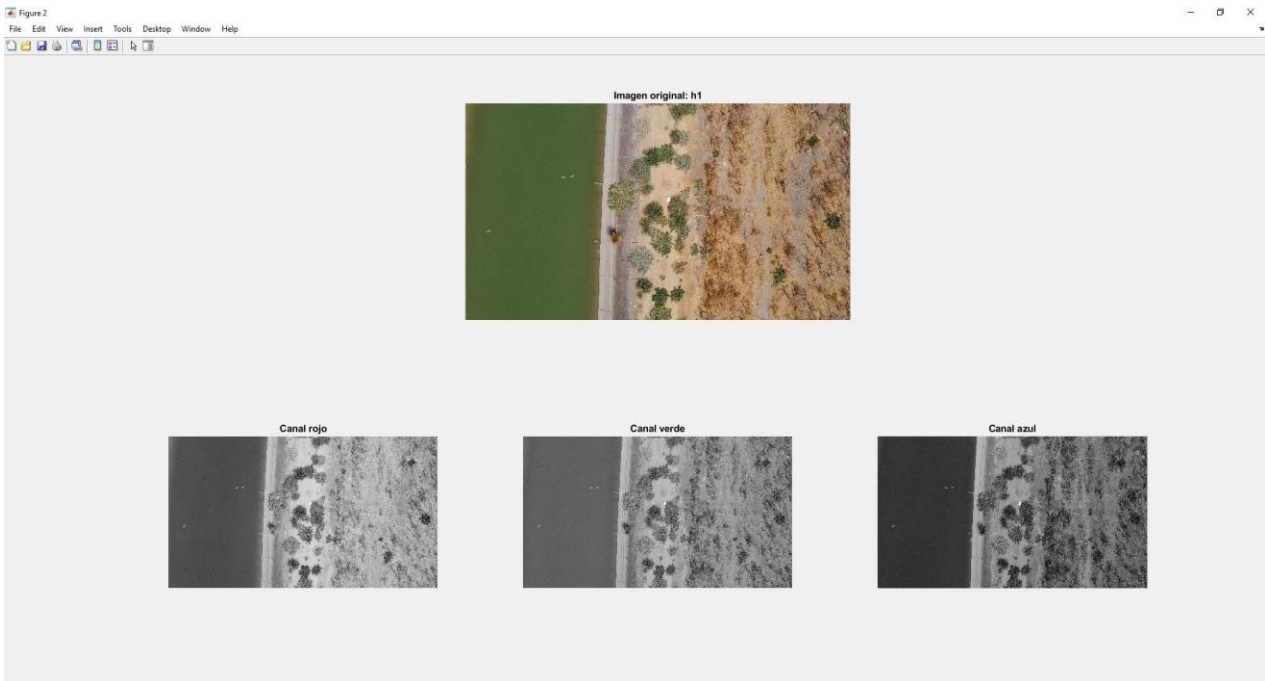


Figura 4-10. Imagen separada por canales

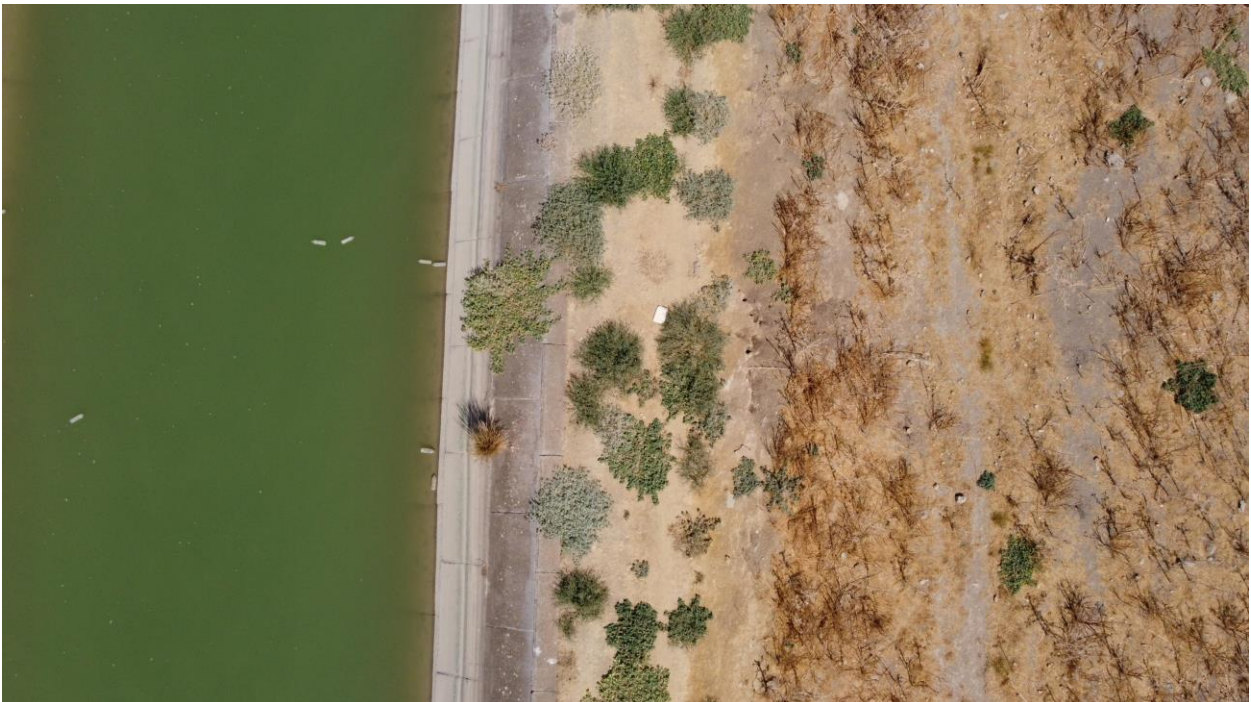


Figura 4-11. Imagen con nitidez mejorada

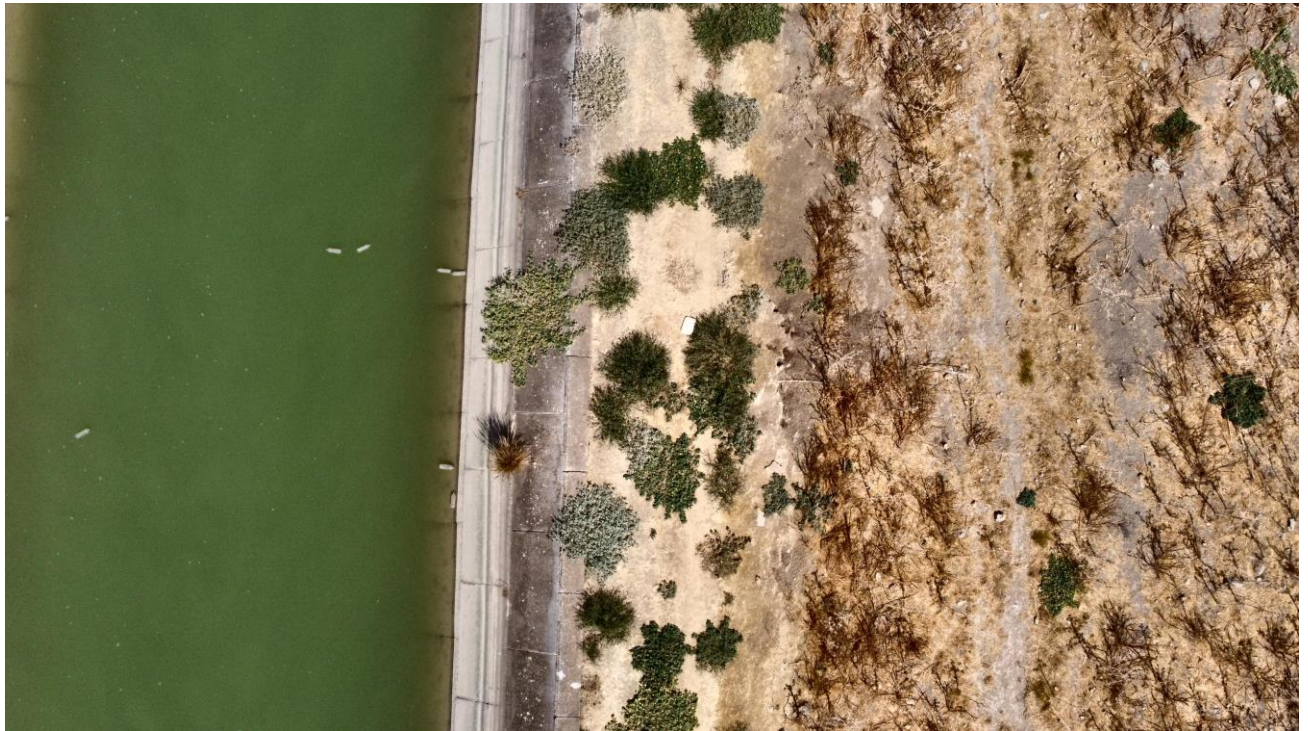


Figura 4-12. Imagen con mejora de contraste

4.3.2 Operaciones Morfológicas

Posterior a la mejora de la nitidez y contraste de las imágenes, es pasada a escala de grises para realizar la binarización (calculando el Umbral Otsu), el relleno de huecos y la aplicación de las Operaciones Morfológicas oportunas, quedando abierta al usuario la opción de aplicar otras a las ejecutadas de inicio. En este caso se aplica una erosión y posterior dibujo del contorno.

Además, a elección de usuario está la aplicación de la Transformada de Hough.



Figura 4-13. Imagen binarizada



Figura 4-14. Imagen con relleno de huecos



Figura 4-15. Imagen erosionada

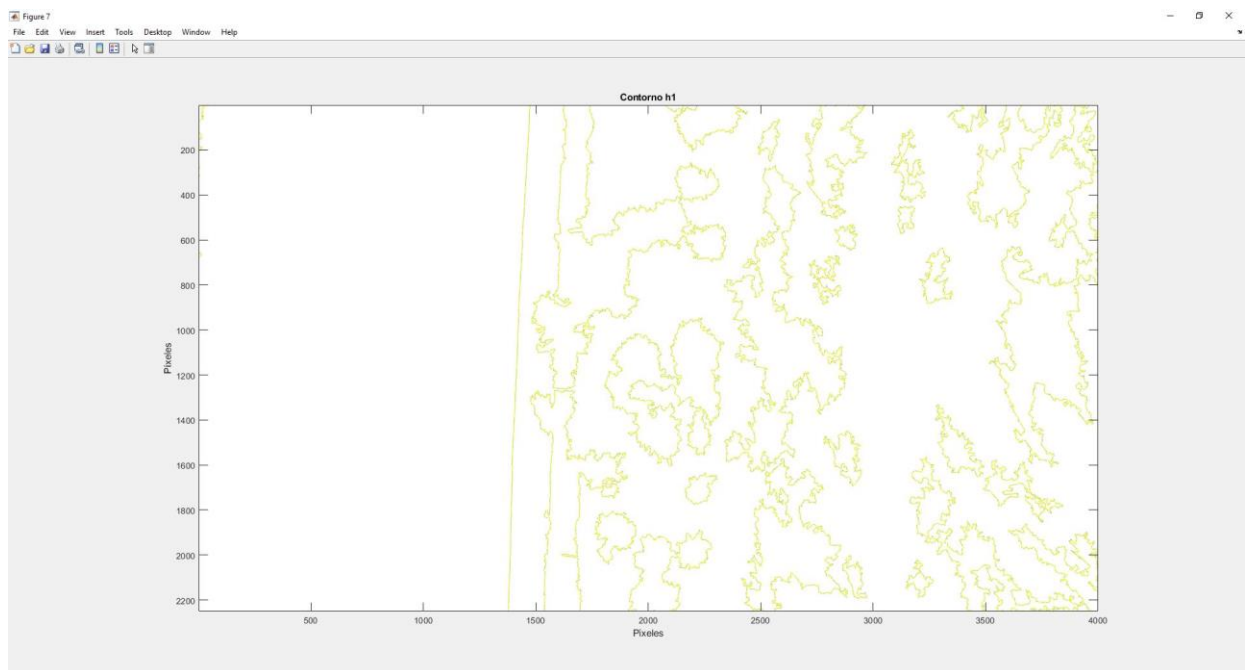


Figura 4-16. Contorno de la imagen

4.3.3 Ampliación sincronizada de las imágenes

Todas las imágenes mostradas por grupos, tienen por el zoom sincronizado para poder hacer una comparación más precisa a la hora de realizar una ampliación en una de las imágenes que la componen.



Figura 4-17. Ampliación sincronizada

4.4 Etiquetado de las imágenes

Etiquetado de los componentes conectados de la imagen binaria y la extracción de sus propiedades. El programa nos pregunta sobre qué etiqueta queremos saber sus características y después de ello el cálculo de los índices del terreno en cuestión.

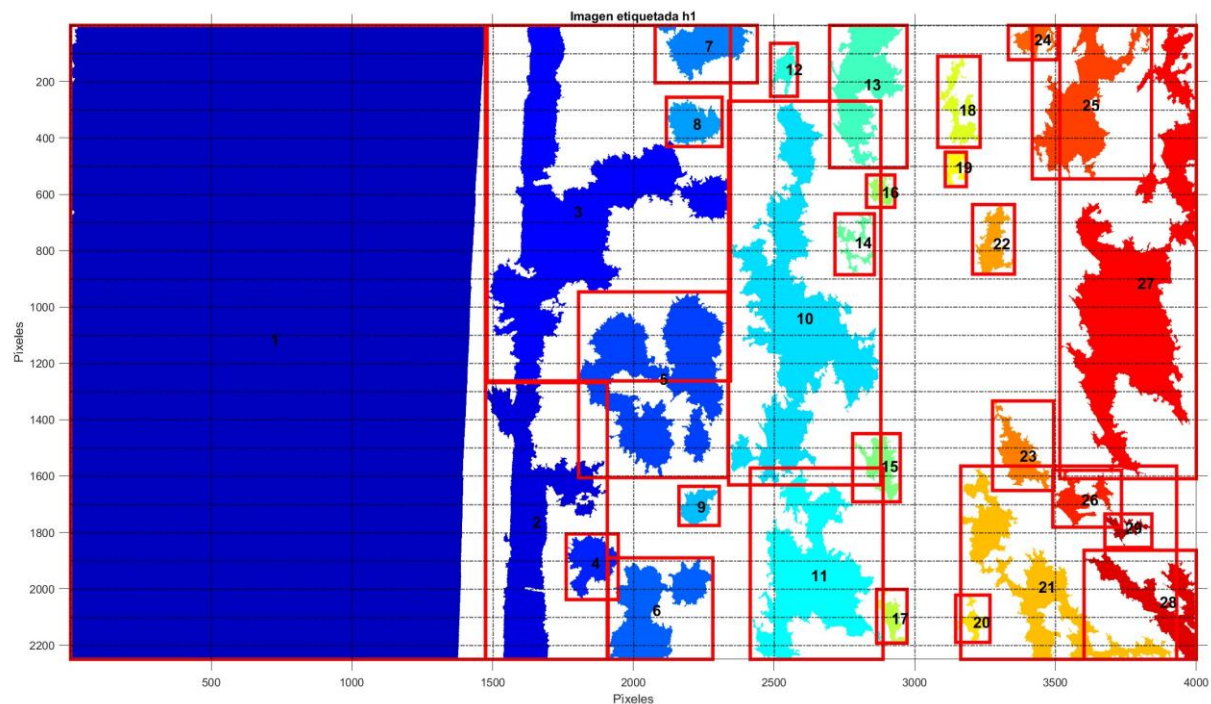


Figura 4-18. Imagen etiquetada



Figura 4-19. Imagen original etiquetada

4.5 Cálculo de los índices

El cálculo de estos índices es una adaptación para la cámara VIS RGB del dron usado de las correspondientes para cámaras VIS NIR como menciona en la Tabla 1-1.

El programa nos pregunta sobre que etiqueta queremos calcular los índices.

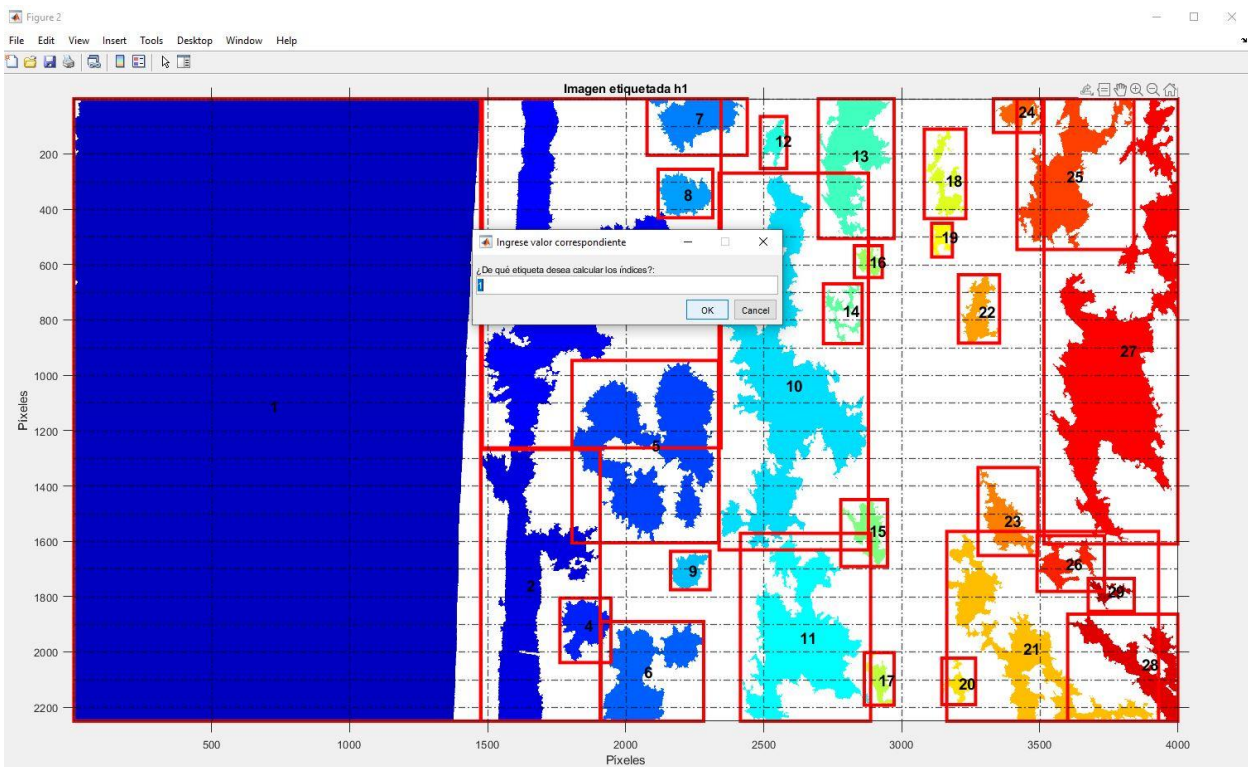


Figura 4-20. Elección etiqueta

En este caso, elegimos la etiqueta número 1 que contiene agua y nos da dos opciones para calcular los índices: 'Por zonas' y 'Por índices'.

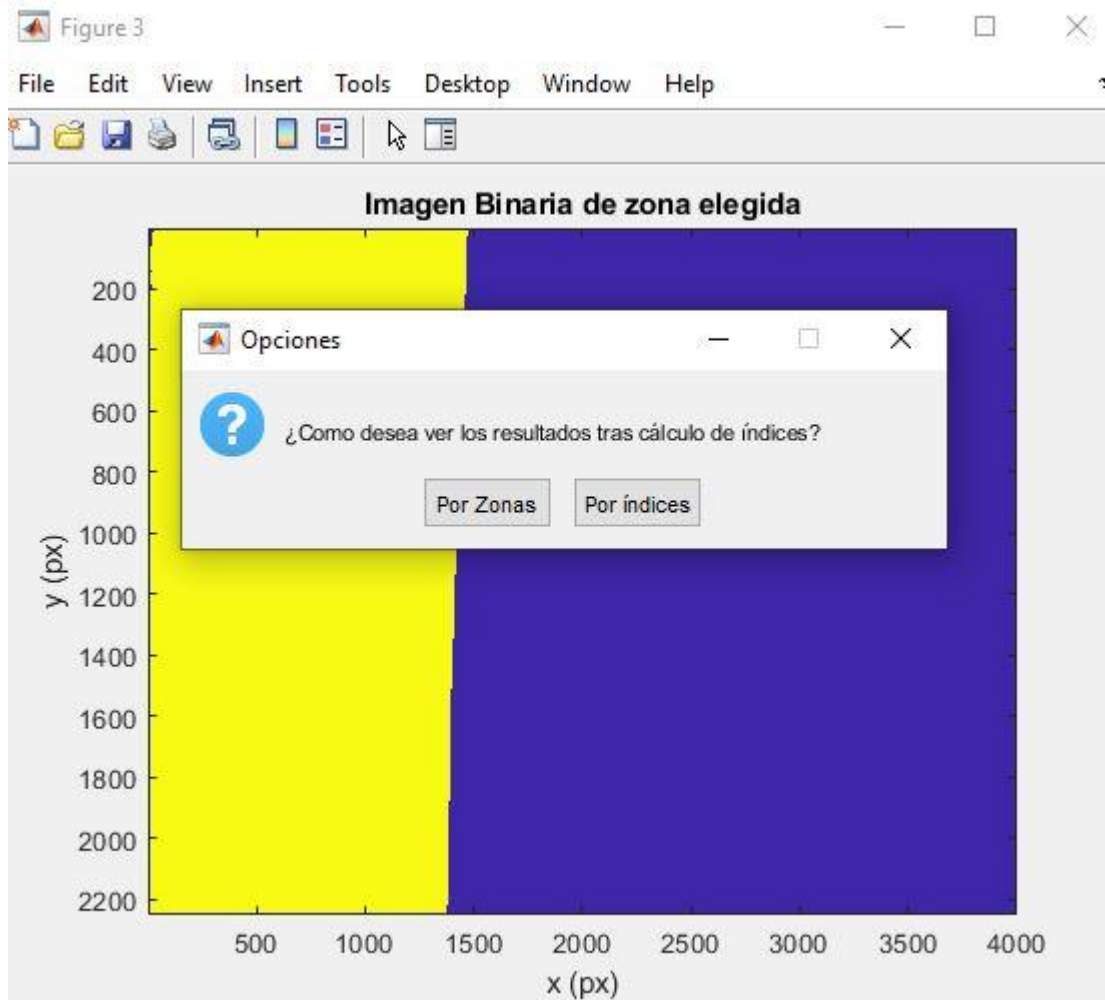


Figura 4-21. Elección del cálculo de los índices

4.5.1 Por zonas

Si la elección del usuario es "Por zonas", calcula todos los índices de la etiqueta elegida y la muestra junto a la original.

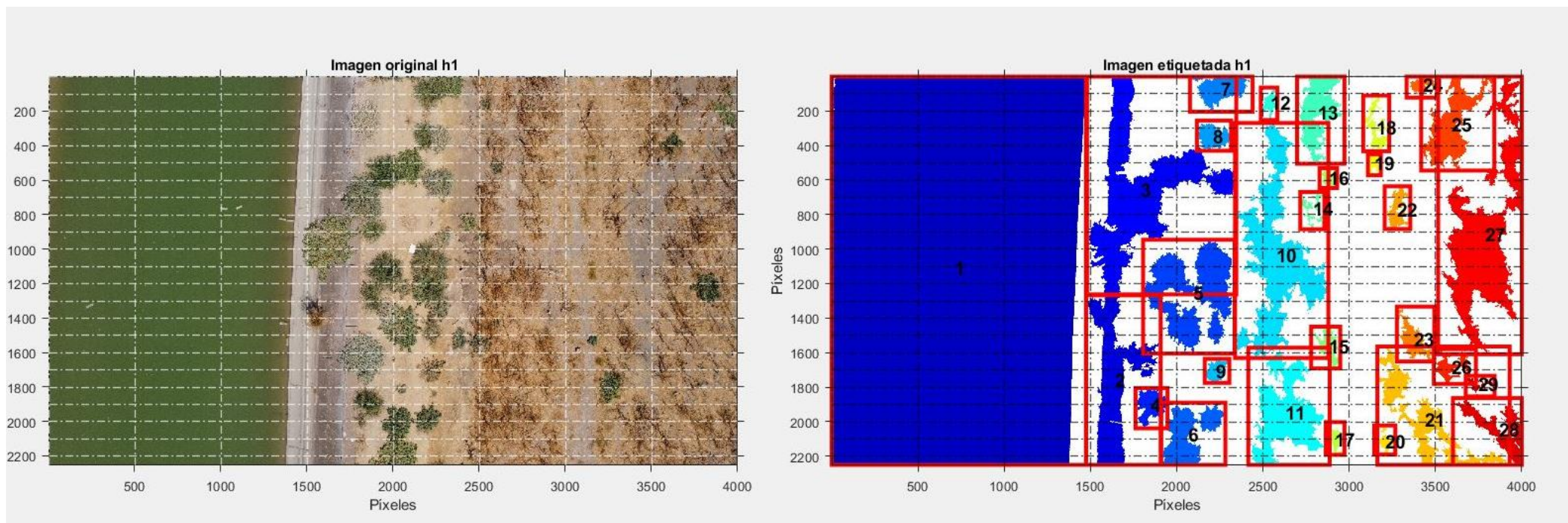


Figura 4-22. Original y etiquetada

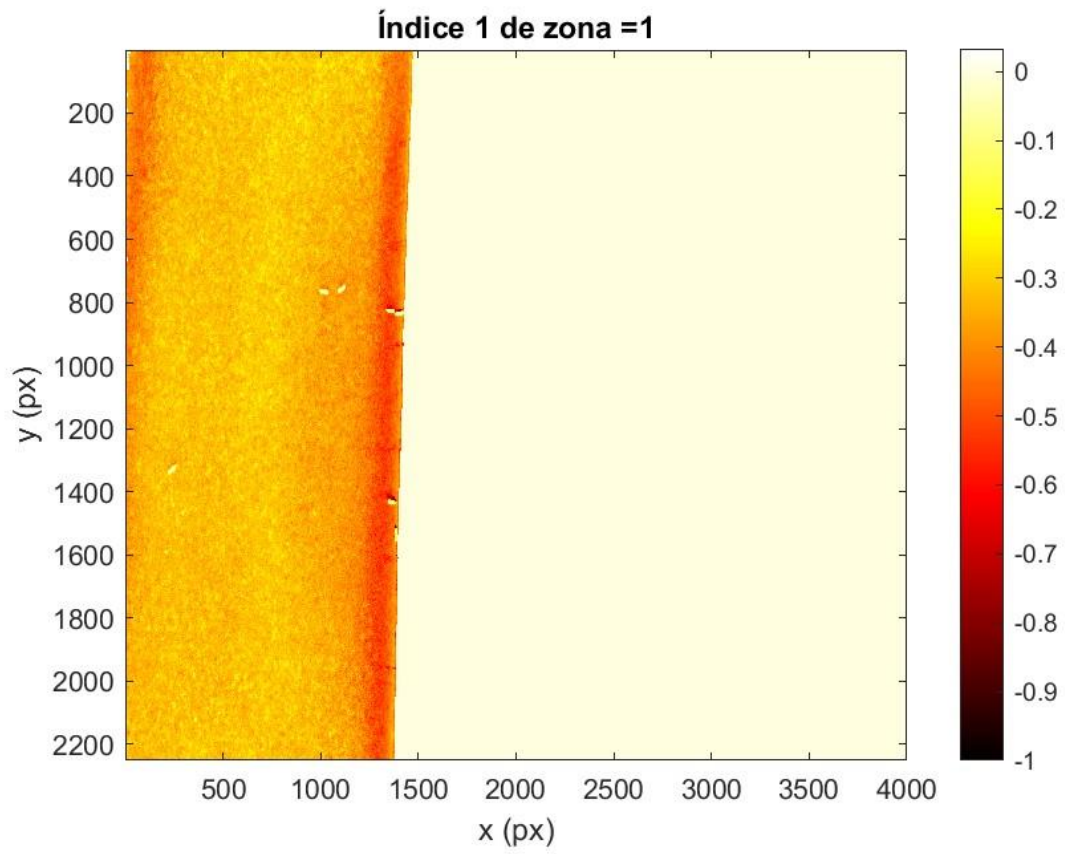


Figura 4-23. Índice 1 de la etiqueta elegida

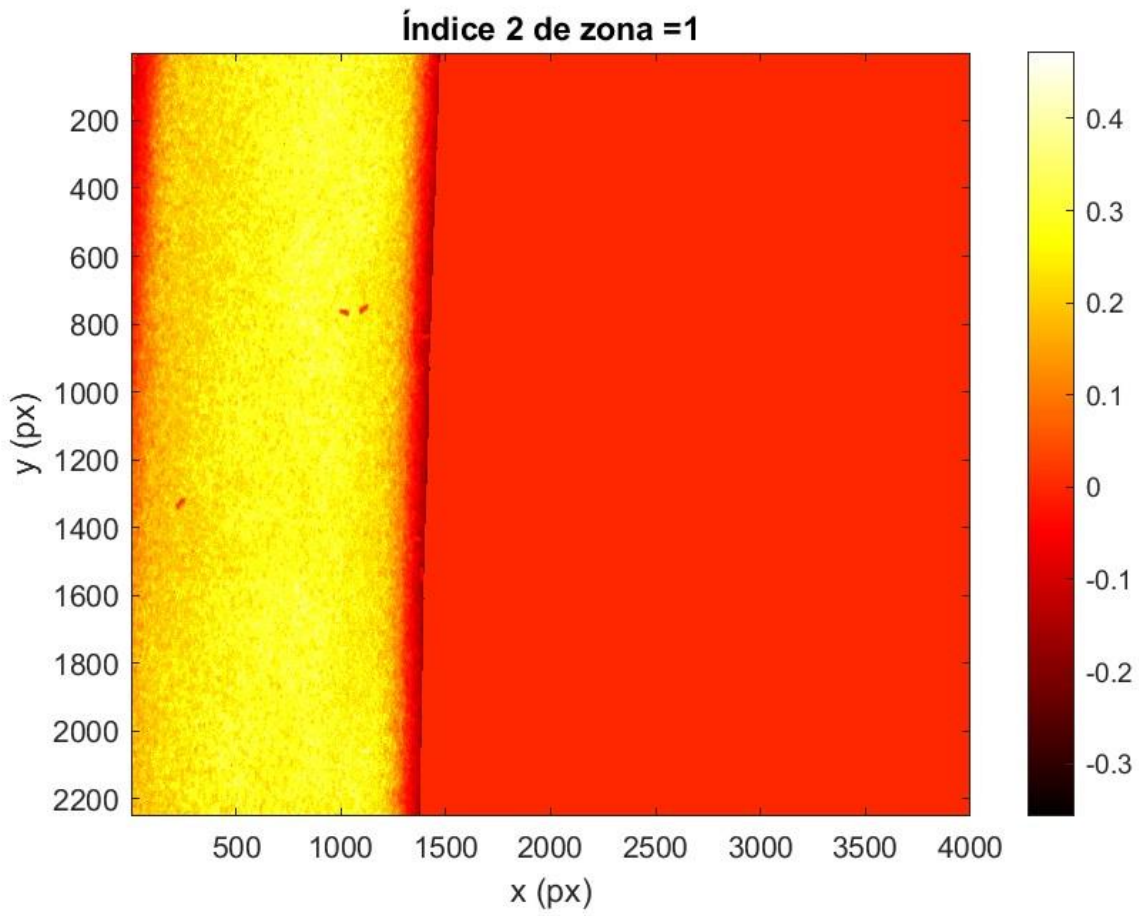


Figura 4-24. Índice 2 de la etiqueta elegida

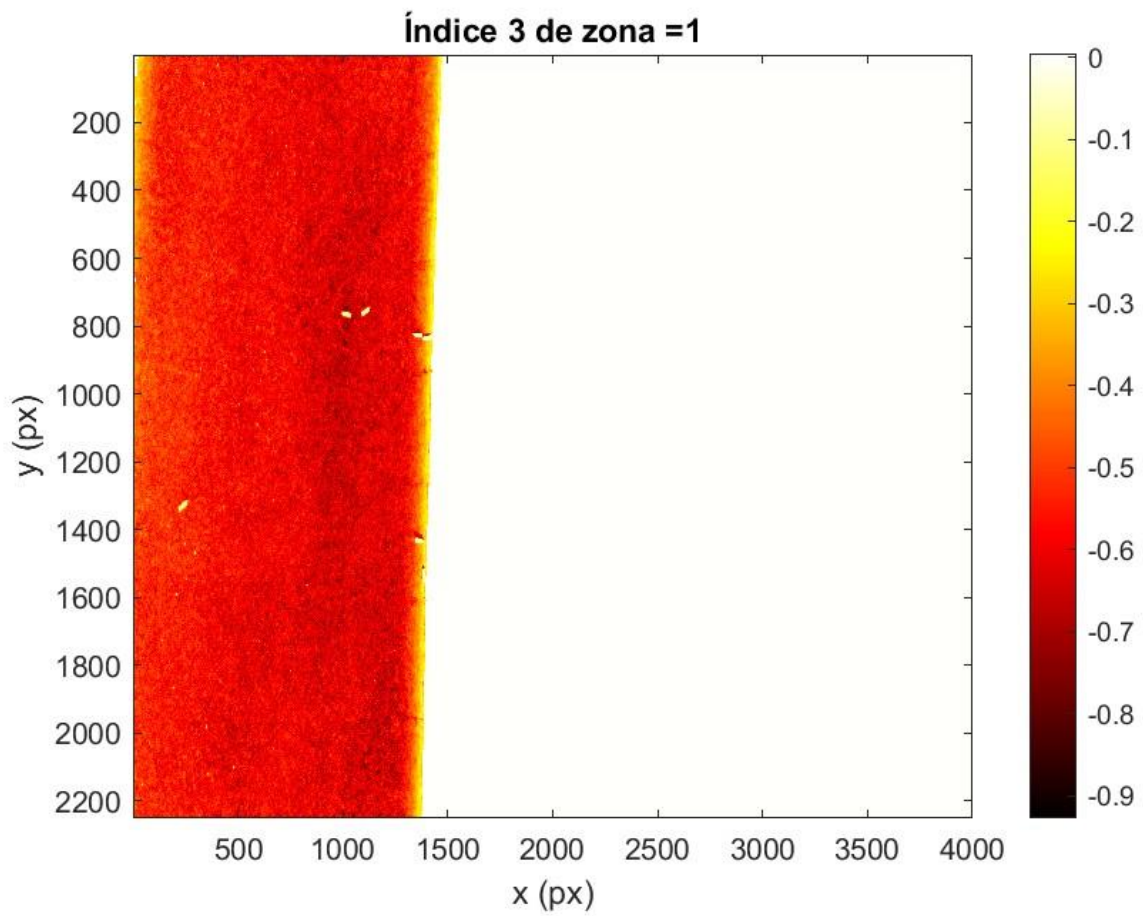


Figura 4-25. Índice 3 de la etiqueta elegida

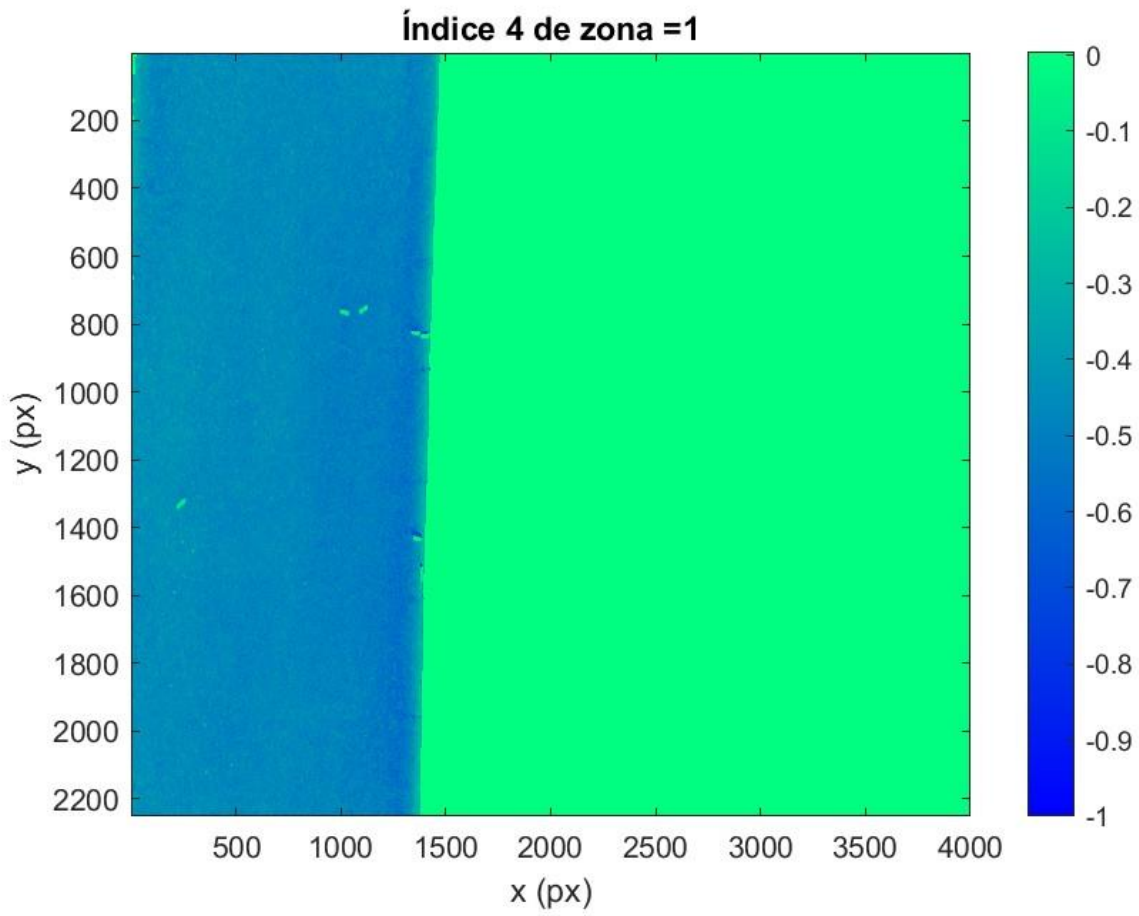


Figura 4-26. Índice 4 de la etiqueta elegida

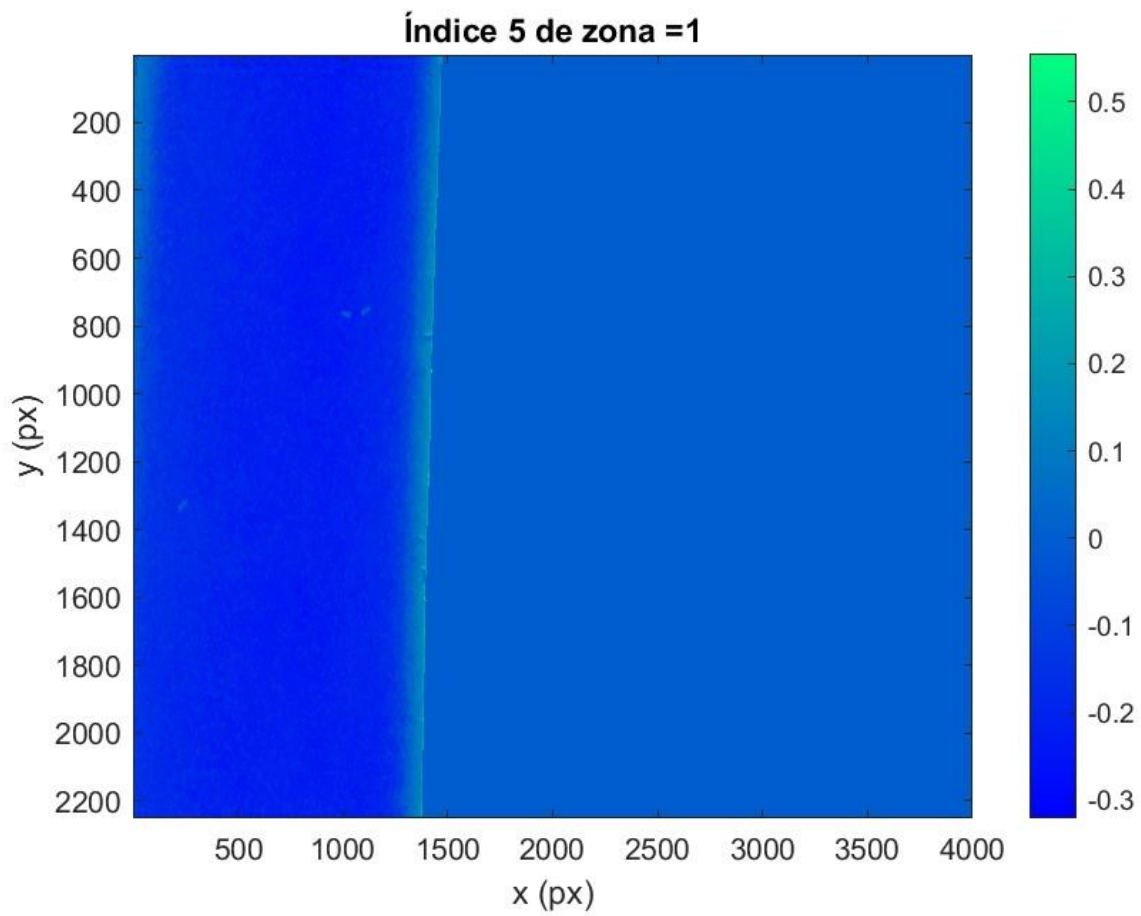


Figura 4-27. Índice 5 de la etiqueta elegida

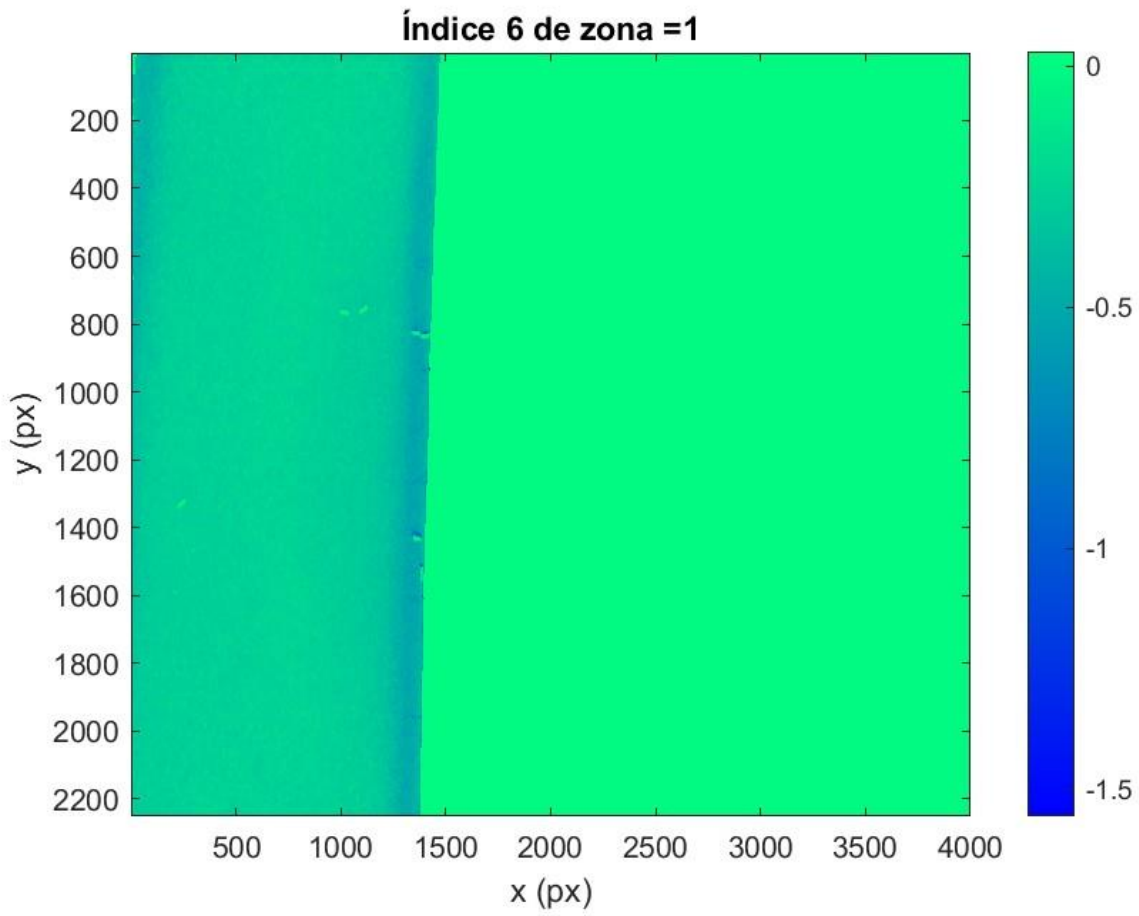


Figura 4-28. Índice 6 de la etiqueta elegida

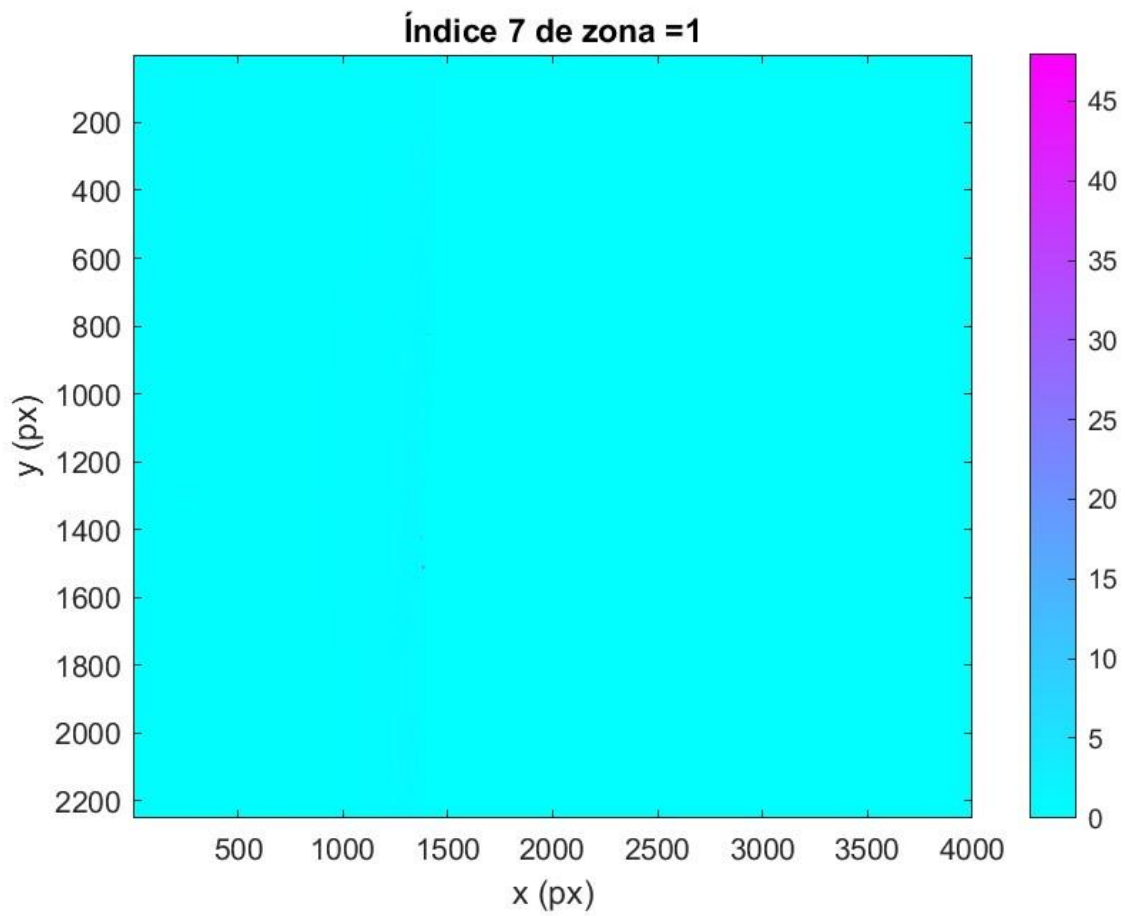


Figura 4-29. Índice 7 de la etiqueta elegida

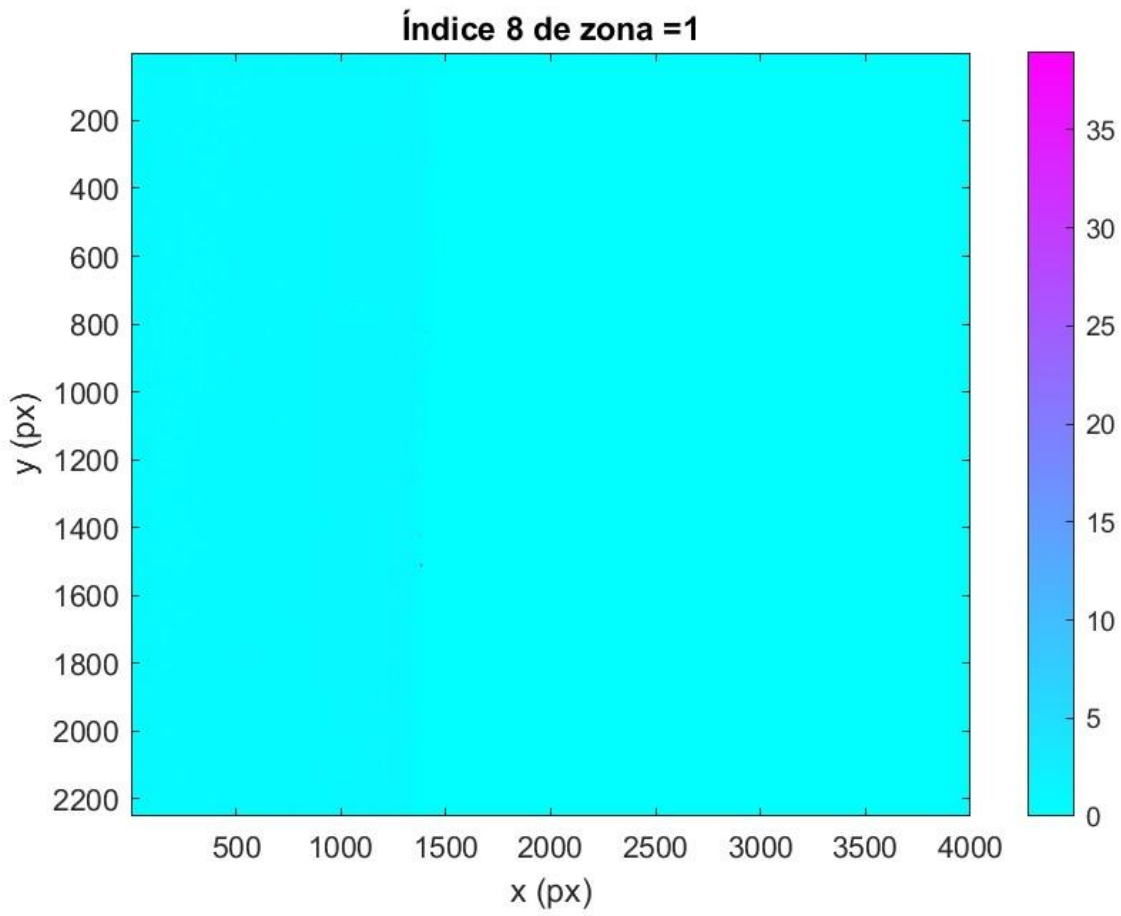


Figura 4-30. Índice 8 de la etiqueta elegida

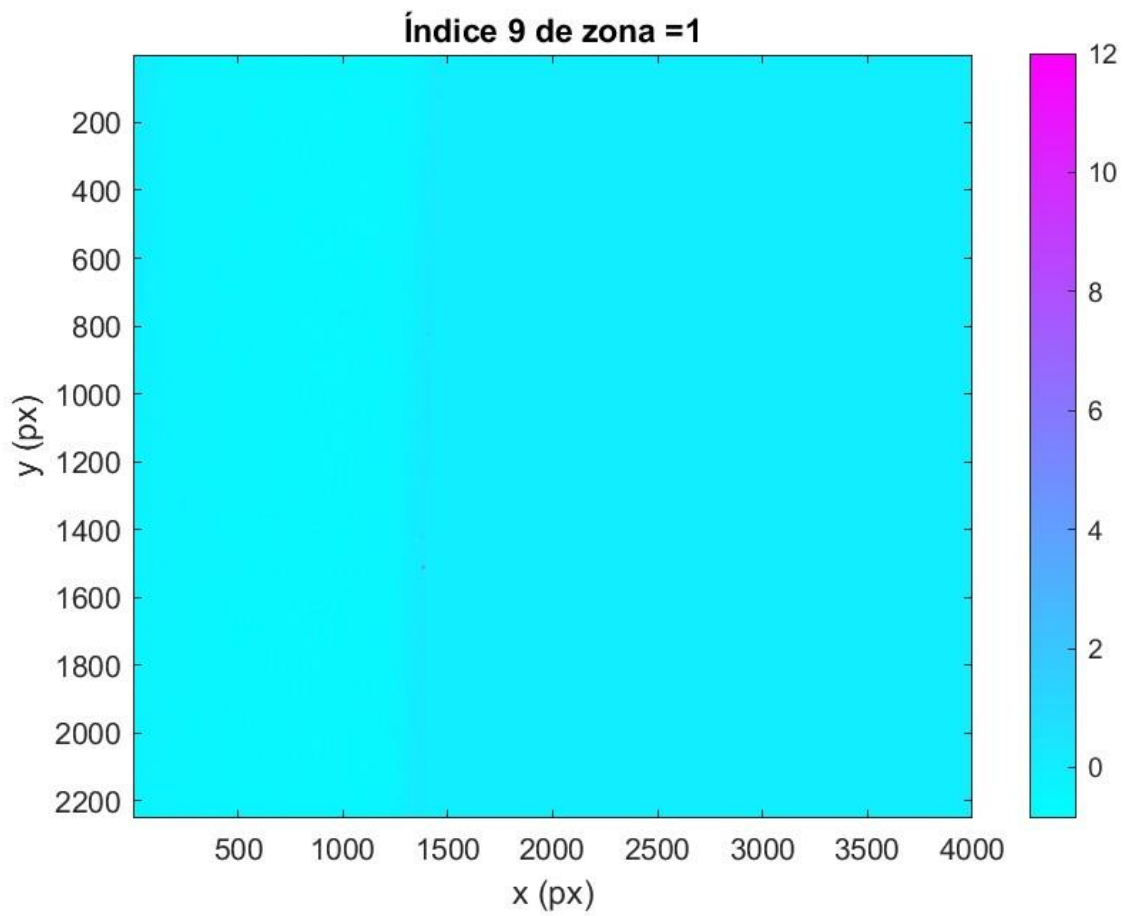


Figura 4-31. Índice 9 de la etiqueta elegida

Mostrando todos los índices juntos:

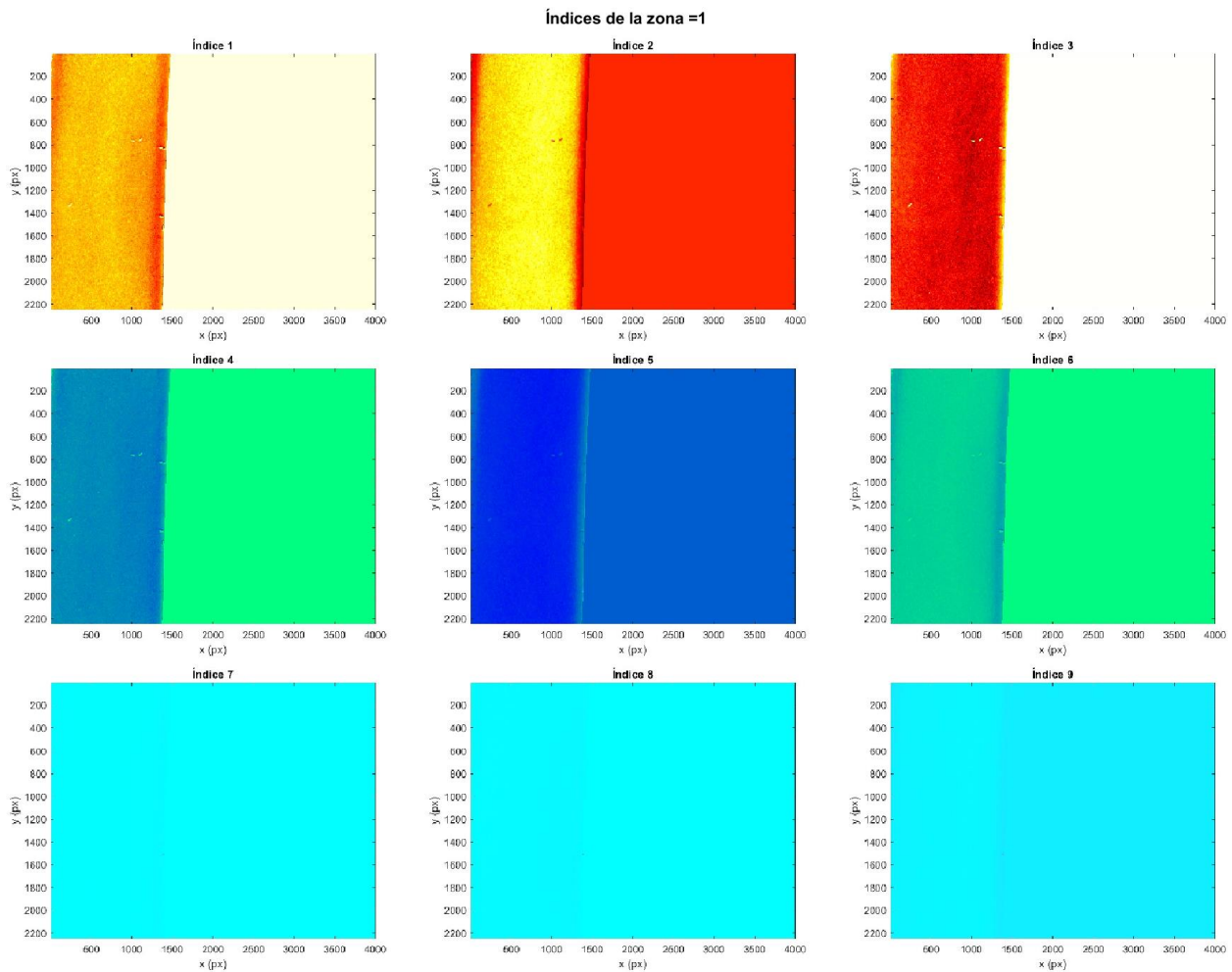


Figura 4-32. Índices de la etiqueta elegida

4.5.2 Por índices

Si es elegida esta opción, nos muestra la referencia de los canales R, G y B con los índices calculados.

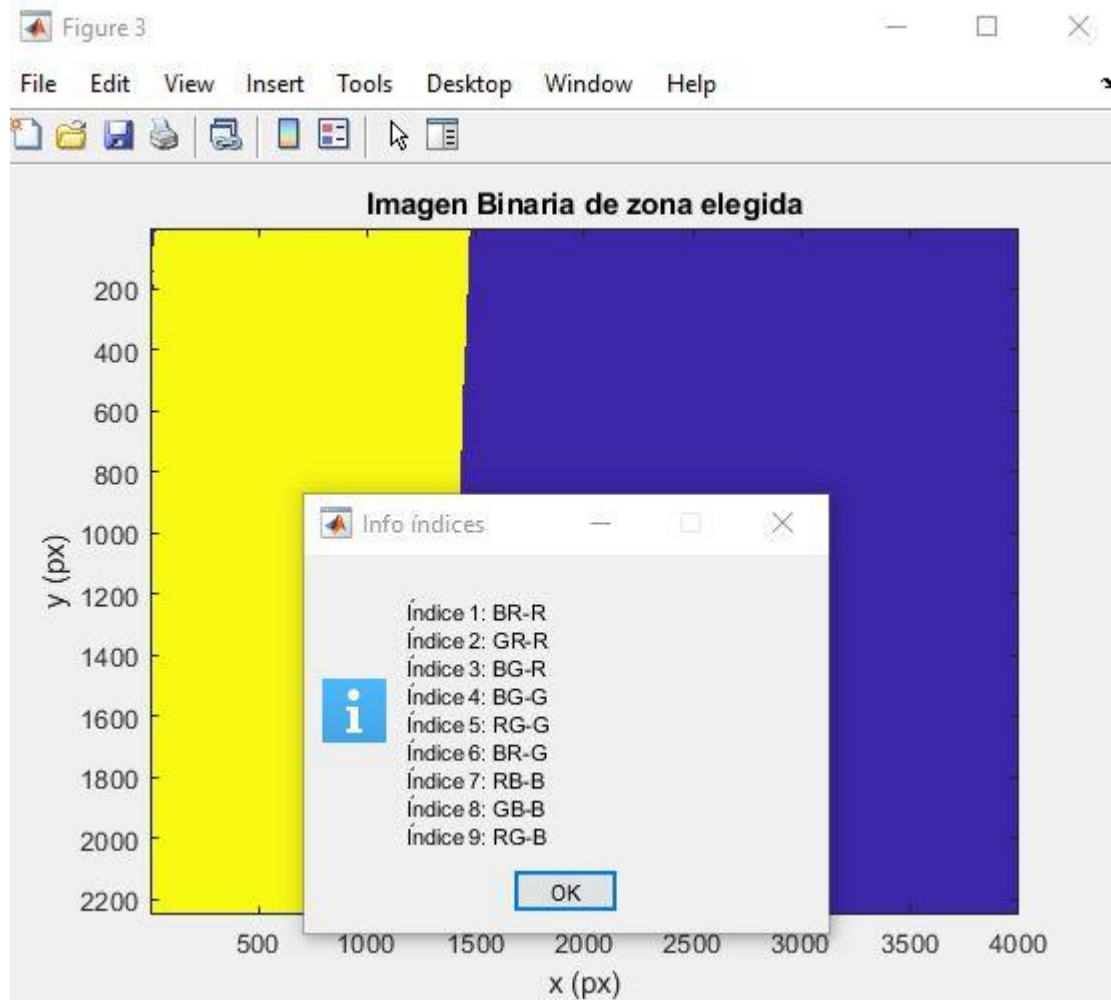


Figura 4-33. Por índices

Índice elegido frente a la imagen original y la etiquetada.



Figura 4-34. Índice elegido de la etiqueta seleccionada

Tras finalizar el cálculo y muestra de los índices, el programa nos da la opción de mantener las ventanas del procesado abiertas o cerrarlas.

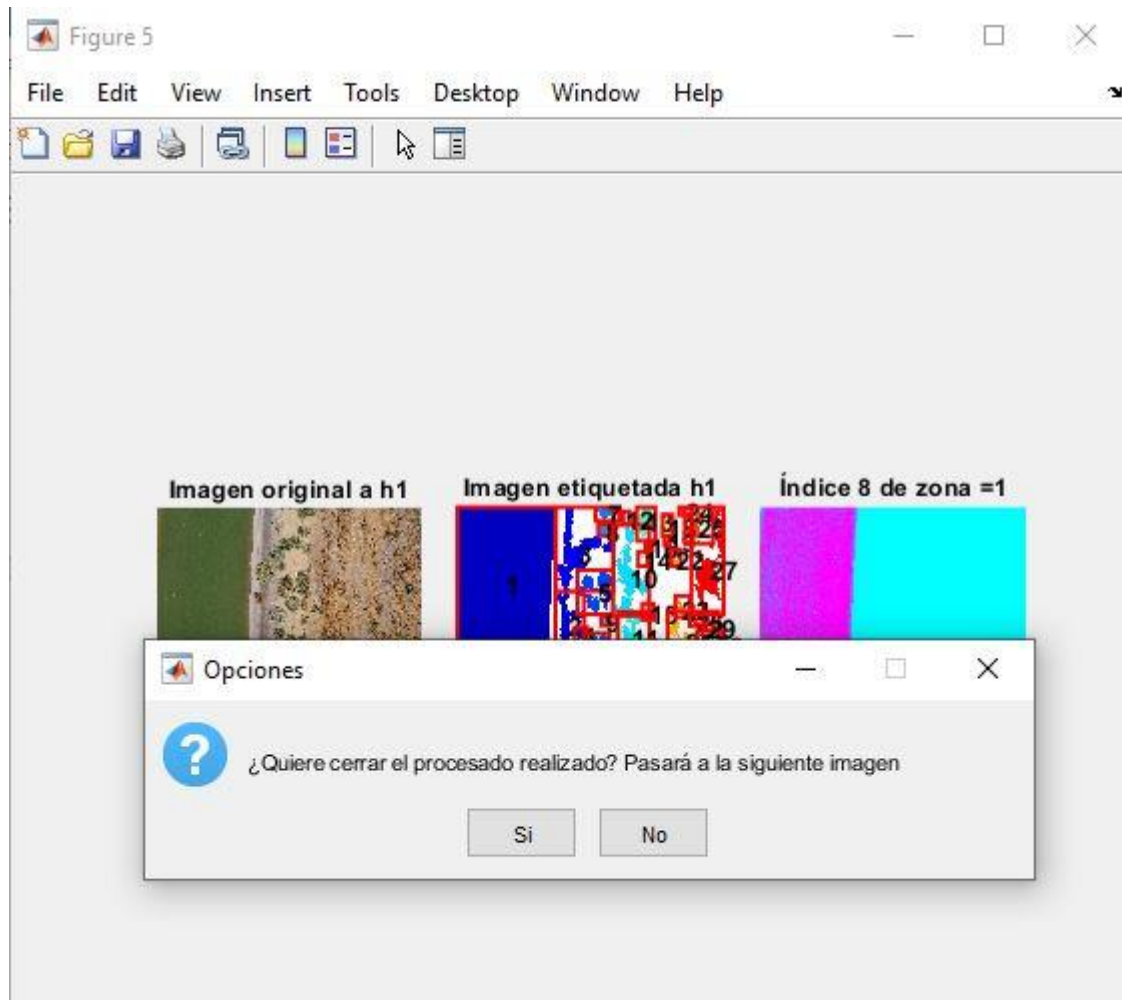


Figura 4-35. Opción de cierre de figuras

4.6 Procesado con escala con el Método “Con referencia”.

Nos pide el programa que seleccionemos una hoja de papel (tamaño A4, 29.7 cm x 21,0 cm) situada como referencia dentro del campo de visión registrado por la cámara y pulsemos “Intro” para calcular la escala.

Como se indicó en el apartado 2.5, el valor de la escala también puede determinarse mediante los datos de ángulo de visión registrado por la cámara a la altura de vuelo considerada.

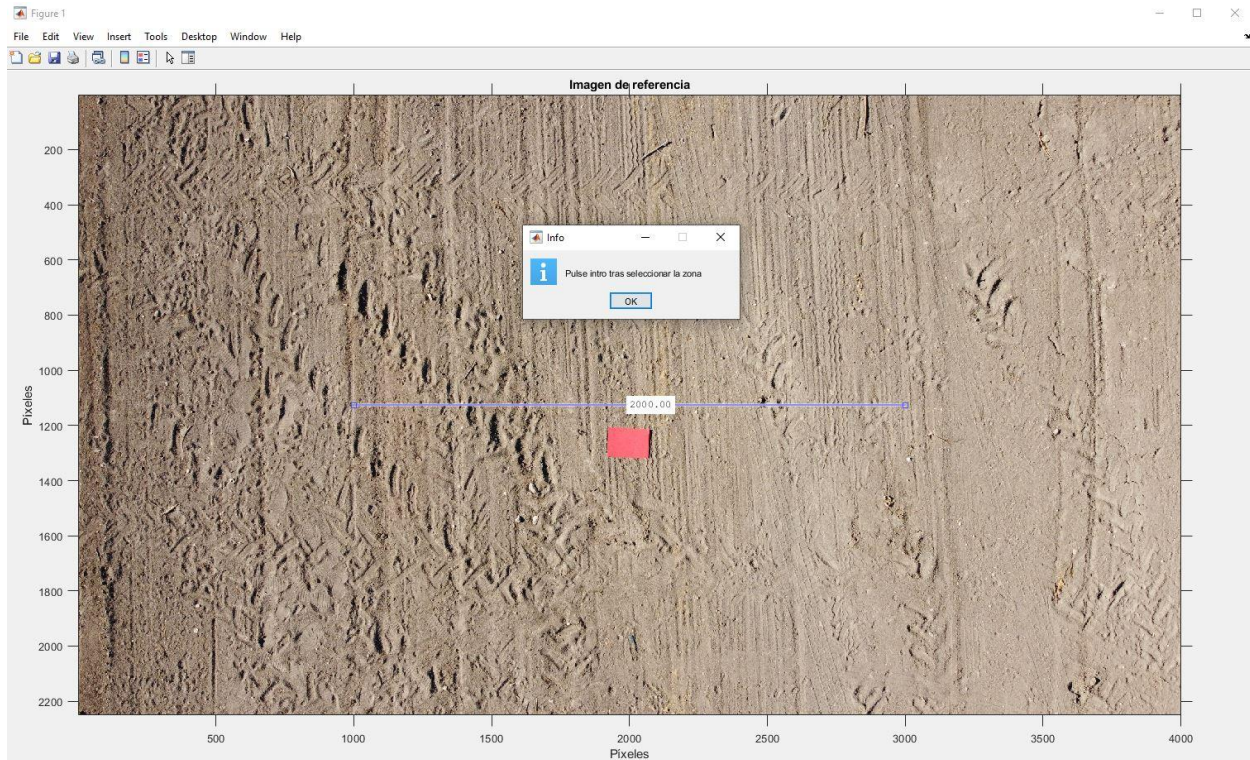


Figura 4-36. Imagen de referencia para escala

4.7 Guardado de los índices y las propiedades en Excel

El programa guarda de manera automática tanto las propiedades de las etiquetas de las ROI de la imagen como los valores de los índices calculados.

4.8 Segmentación por color

Queda abierto para el usuario poder segmentar las imágenes por color para extraer zonas de interés tras la filtración del ruido.

4.9 Programas anexos para ver resultados finales

Se han elaborado dos programas adicionales, correspondientes a las dos alturas, para ver los resultados finales del procesado (ext_valoresH1.m y ext_valoresH2.m).

4.10 Ejemplos de uso

AgropDron ha sido probado con un total de 124 fotografías tomadas con dron a diferentes alturas. Cada una de las imágenes han sido procesadas, etiquetadas y calculados sus índices.

Se ha probado con fotografías tomadas en diferentes zonas de Cádiz y Sevilla, previa consulta de la aplicación ENAIRE® para el permiso del vuelo, con características de terreno diferentes para así probar su funcionamiento; igualmente ha sido comprobado con imágenes de diferentes tamaños y resoluciones.

En el análisis de las fotografías mencionadas, se ha observado que tras la realización de los cálculos de los nueve índices de cada una de las etiquetas reconocidas como ROI, se llega a la conclusión que los índices referidos al canal B (canal azul), desarrollados en la sección 2.6, no aportan nada al estudio de los terrenos comprendidos en tres escenarios como se puede consultar en la sección 2.9, a diferencia de los demás índices que si se puede extraer valores y llegar a conclusiones sobre el estado y la composición del terreno.

Las imágenes que han compuesto este trabajo, además de otras a modo de prueba en escenarios diferentes a los estipulados en este proyecto, están disponibles en una carpeta adjunta. También se muestra las diferentes imágenes generadas en el camino que forma el procesamiento de cada una de ellas y, además los archivos Excel® que incluyen las propiedades y los índices calculados.

Refiriéndolos a la comparación de un mismo escenario a las dos alturas propuestas (20 metros y 5 metros) se sacan diferentes conclusiones que iré detallando individualmente por zona e índice.

- **Escenario 1: Terreno con agua**

Tras analizarlas se observa que a la altura de 20 metros tiene la capacidad de reconocer más regiones de interés (número de etiquetas) y de mejor definición que a una altura de 5 metros; por tanto, obtiene mejores resultados en los cálculos de los índices de la etiqueta como podemos ver en las imágenes.

También añadir que obtendrá mayores propiedades y áreas a una altura de 20 metros.

Los índices referidos al canal B (canal azul) no aportan nada en este escenario.

h1= 20 metros y etiqueta 1

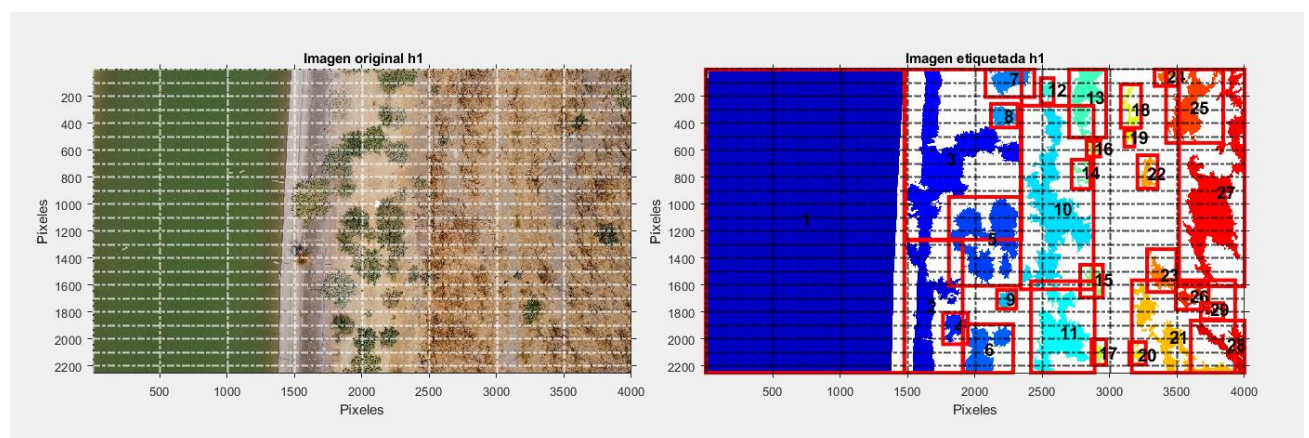


Figura 4-37. Escenario 1 a 20 metros

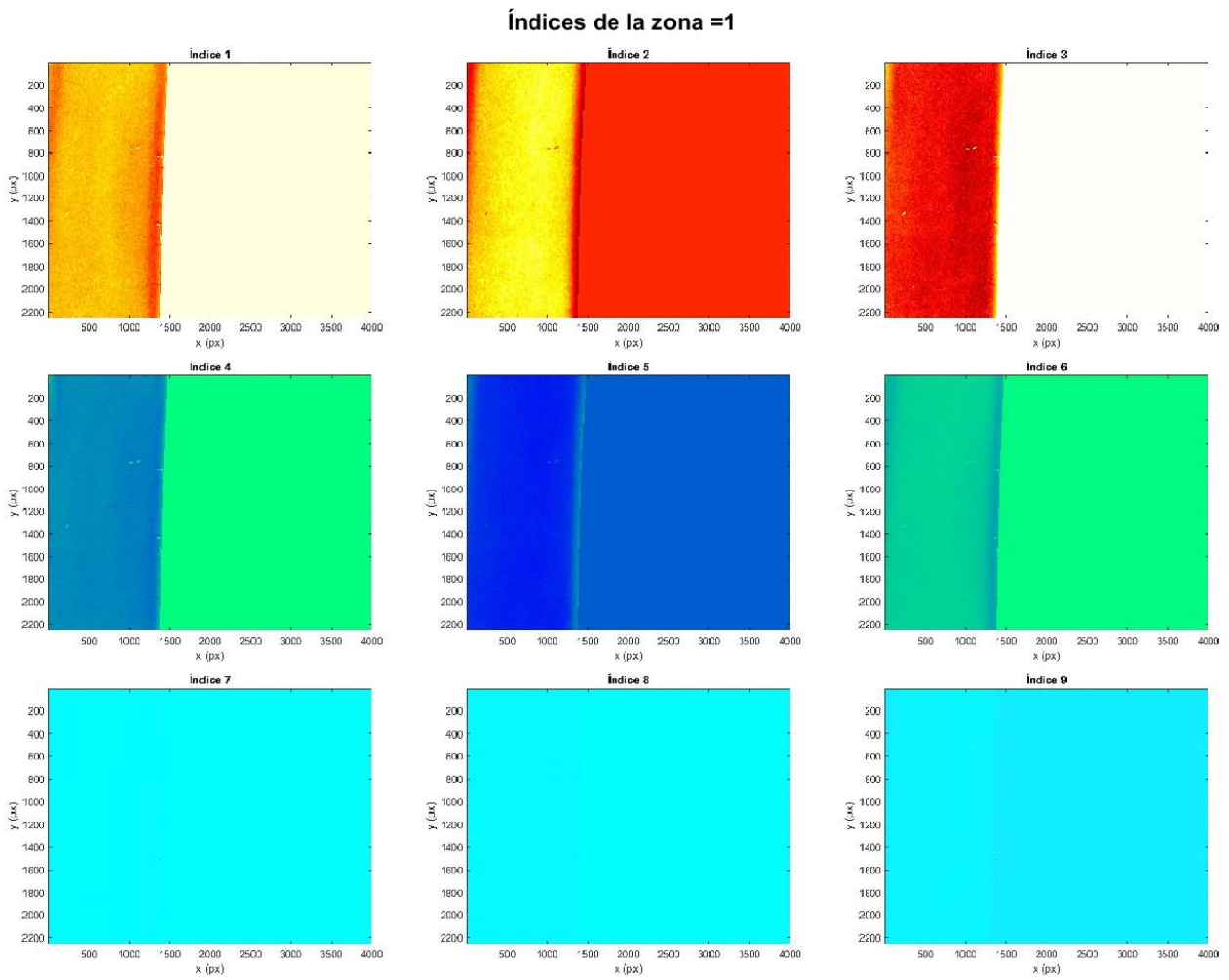


Figura 4-38. Índices de la etiqueta

h2= 5 metros y etiqueta 1

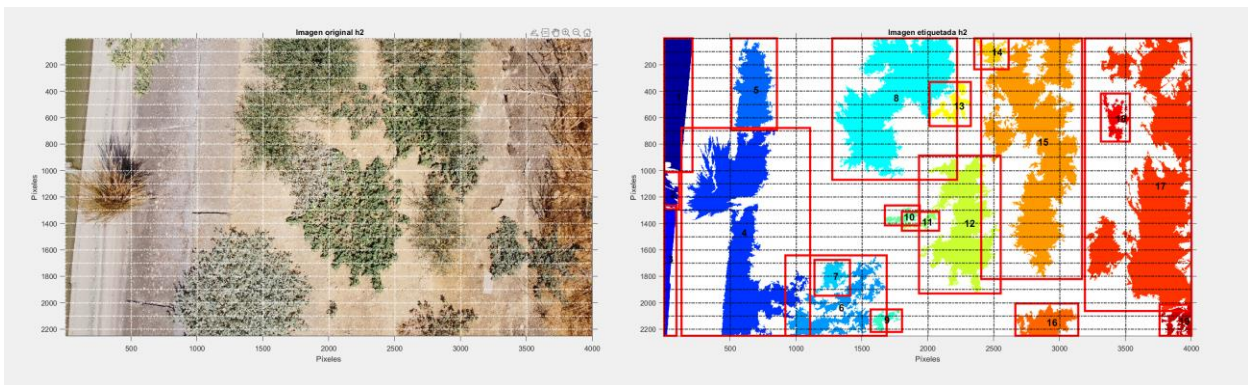


Figura 4-39. Escenario 1 a 5 metros

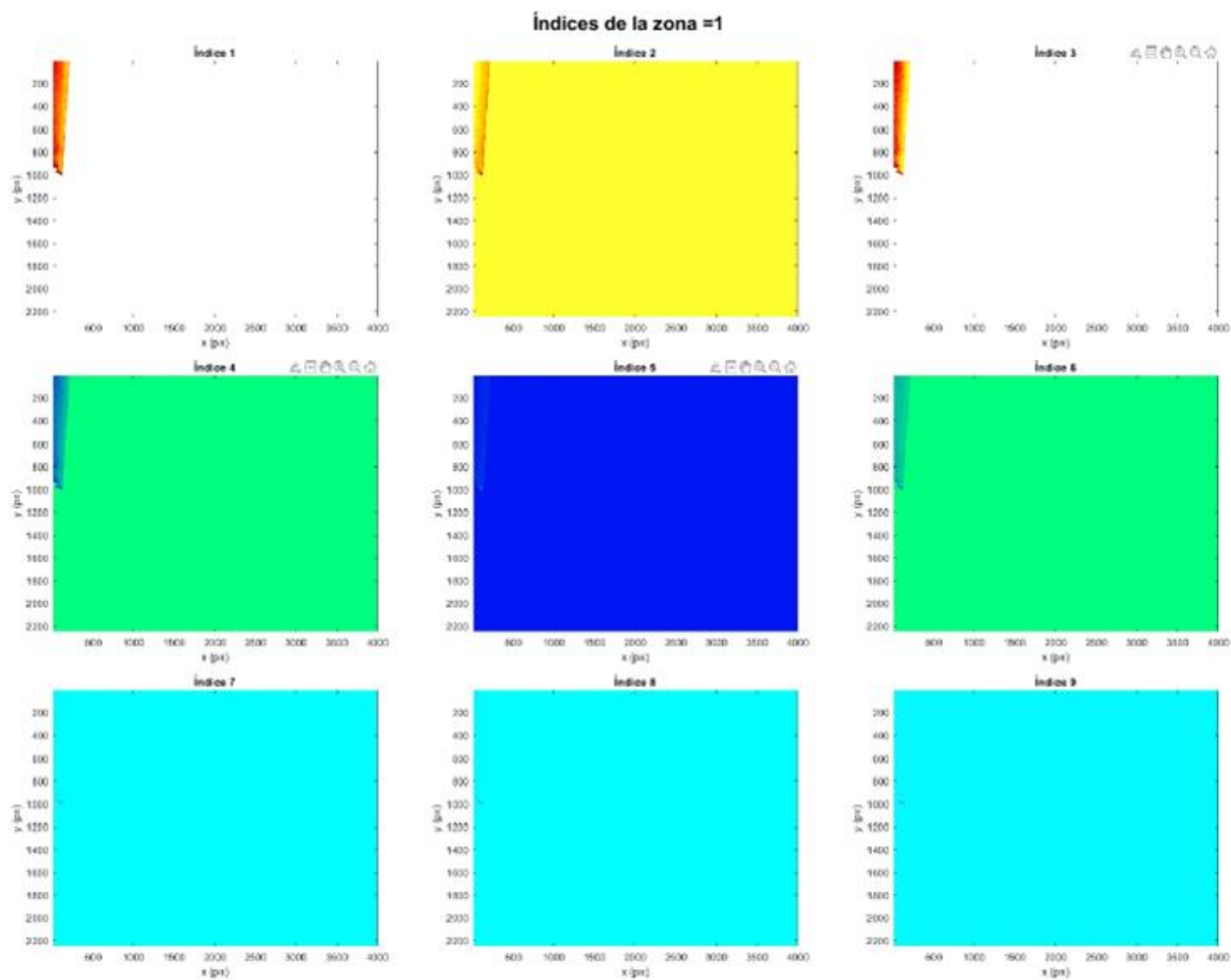


Figura 4-40. Índices de la etiqueta

- **Escenario 2: Terreno sin cultivo**

En este caso ocurre como en el escenario anterior aunque difiere en el cálculo de los índices a ambas alturas, que en este caso sí aportaría algo en el estudio del terreno.

h1= 20 metros y etiqueta 3

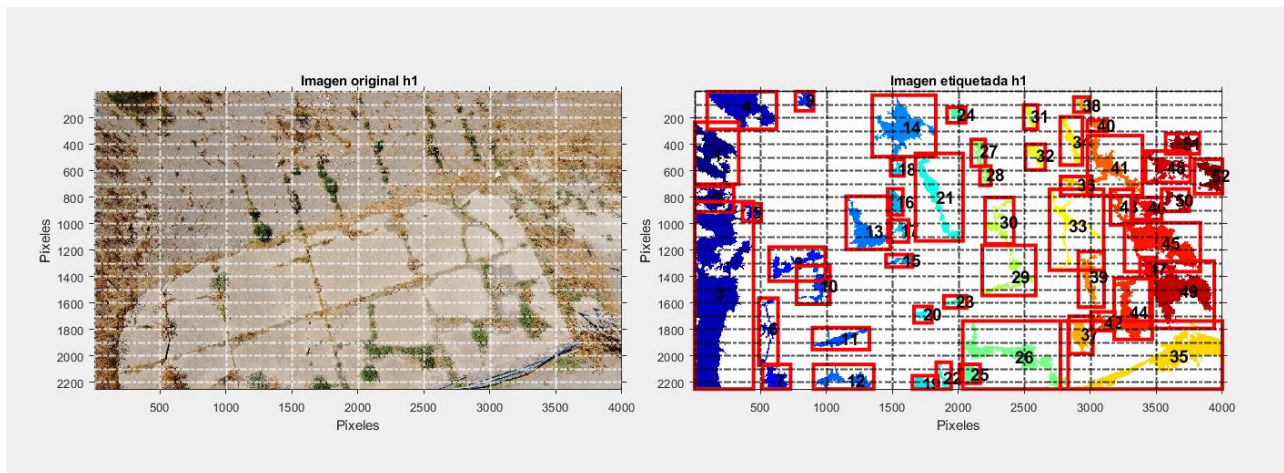


Figura 4-41. Escenario 2 a 20 metros

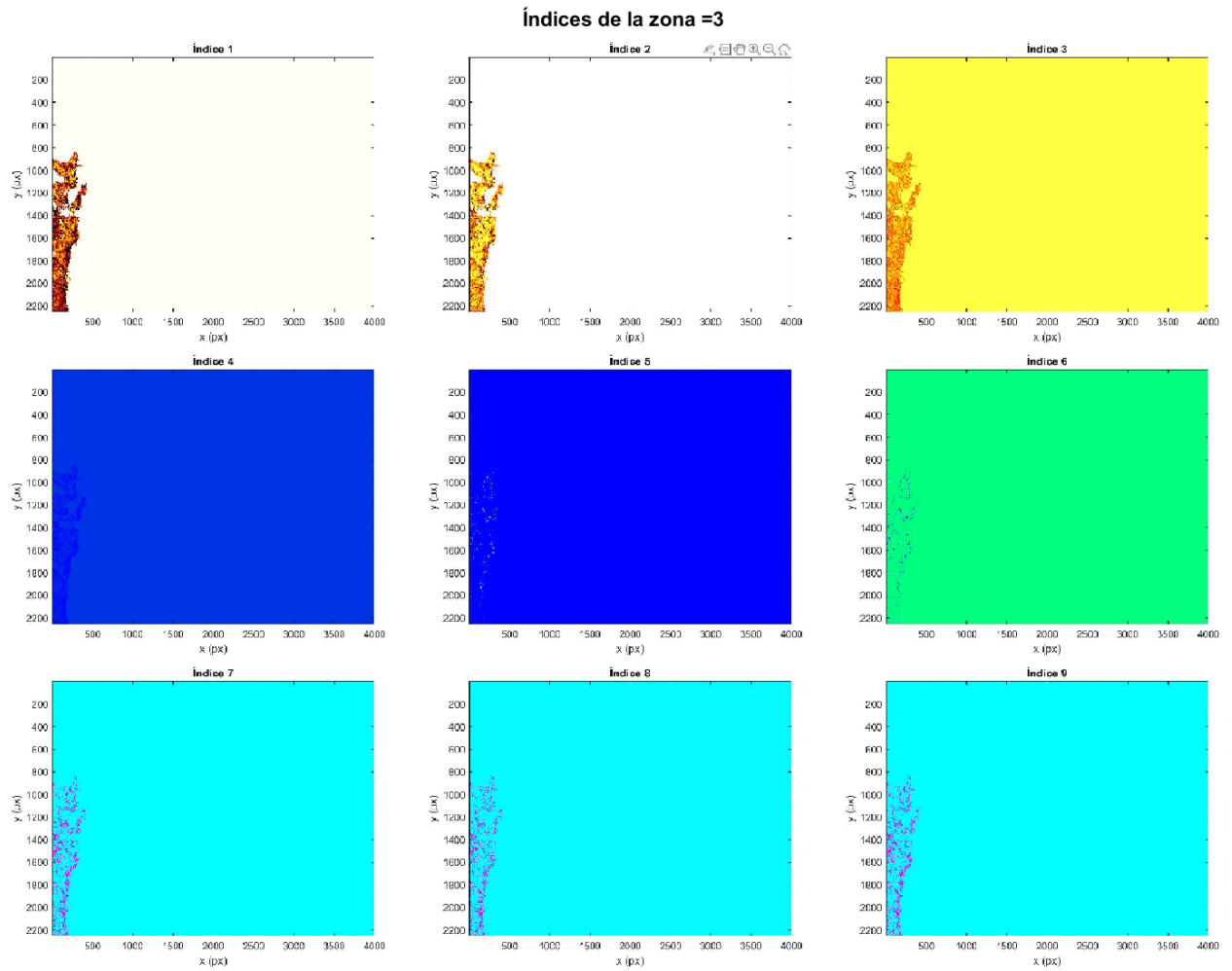


Figura 4-42. Índices de la etiqueta

h2= 5 metros y etiqueta 9

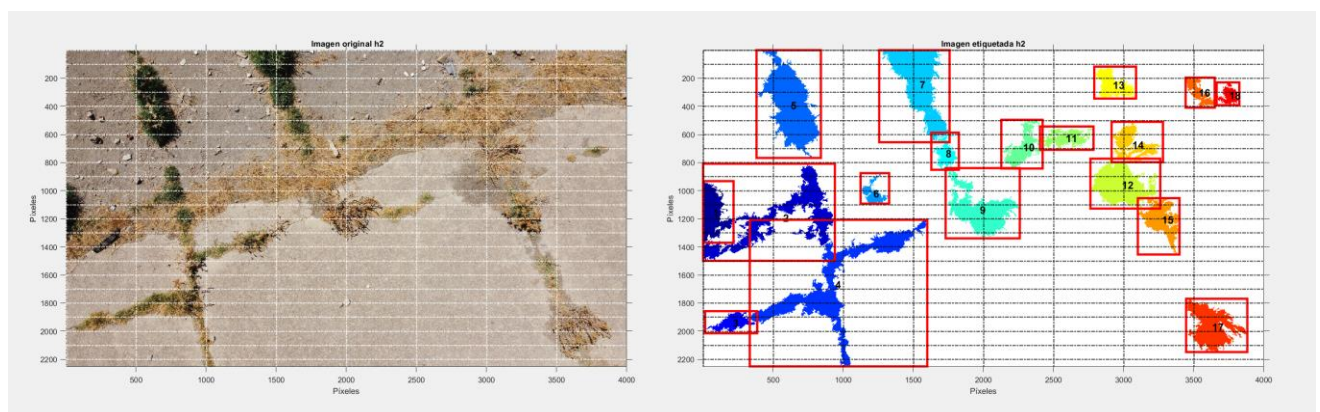


Figura 4-43. Escenario 2 a 5 metros

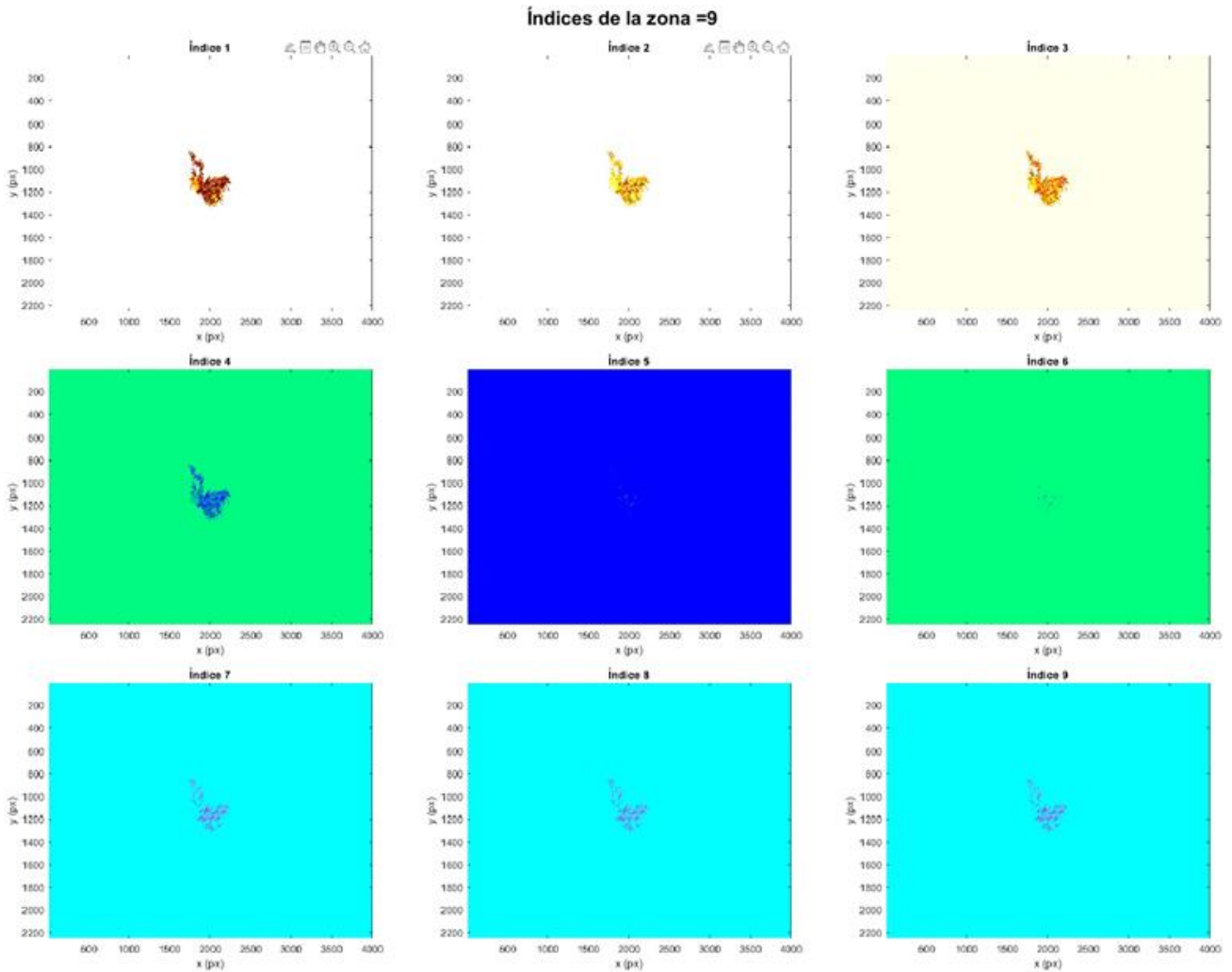


Figura 4-44. Índices de la etiqueta

- **Escenario 3: Terreno cultivado**

En este caso, reconoce mayor número de regiones de interés (número de etiquetas) a 5 metros que a 20 metros como ocurría en los dos escenarios anteriores, pudiendo encontrar más propiedades y áreas a baja altura.

El cálculo de los índices a 20 metros no aportan al análisis, siendo diferente a la otra altura propuesta y aportando datos al estudio del terreno, aunque los índices referidos al canal G (canal verde) no aporta a 5 metros.

h1= 20 metros y etiqueta 1

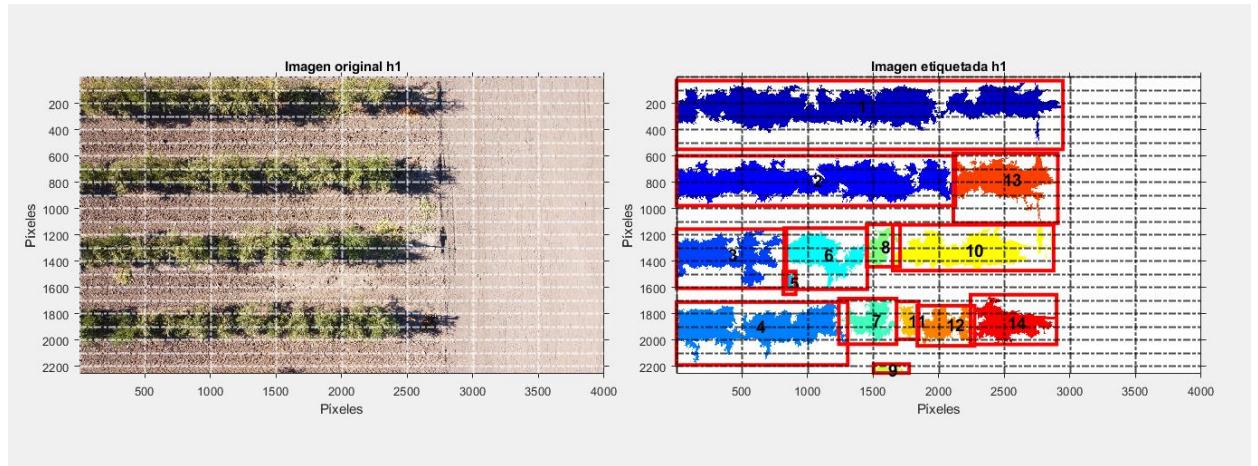


Figura 4-45. Escenario 3 a 20 metros

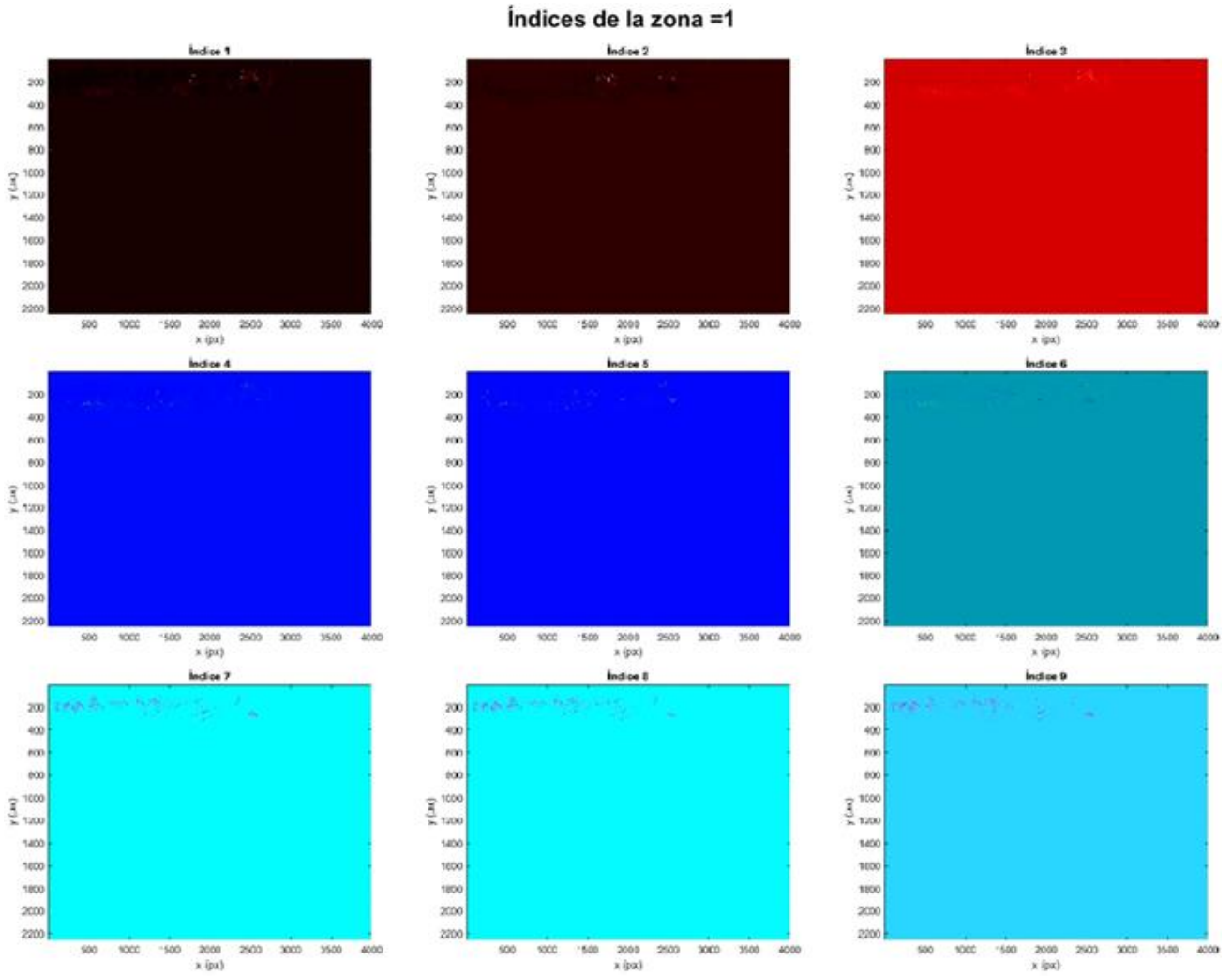


Figura 4-46. Índices de la etiqueta

h2= 5 metros y etiqueta 1

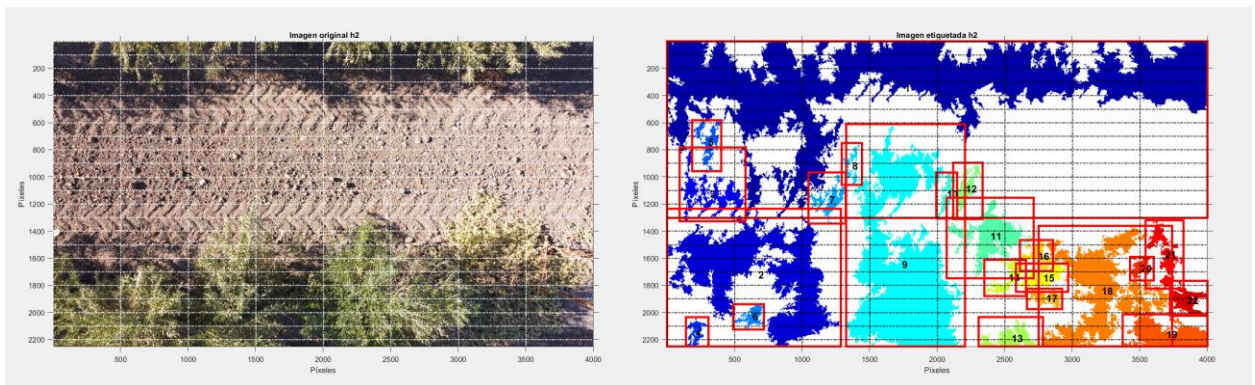


Figura 4-47. Escenario 3 a 5 metros

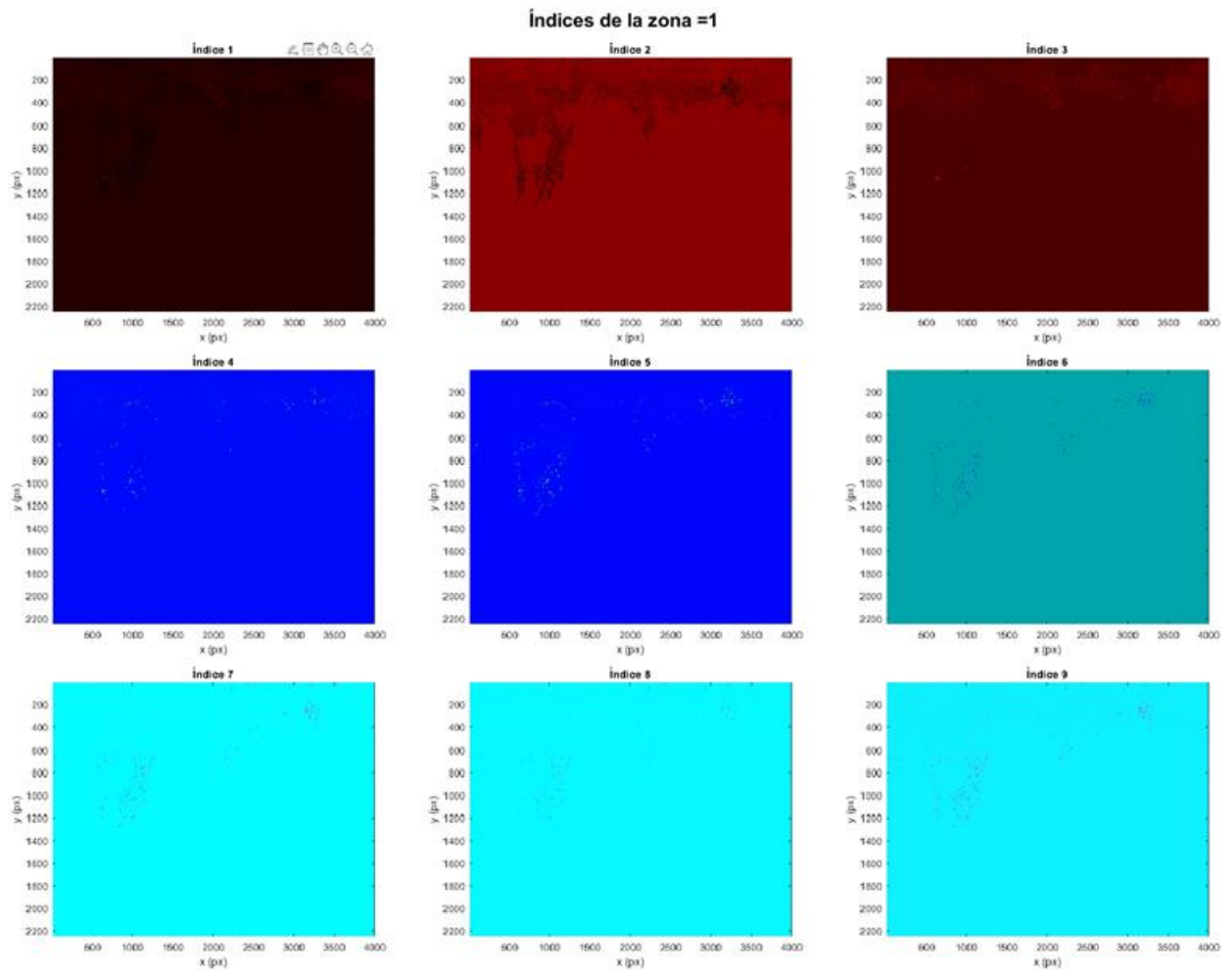


Figura 4-48. Índices de la etiqueta

5 CONCLUSIONES

En este proyecto se ha desarrollado un programa para el procesado de imágenes en formato JPG en un ordenador personal. Es una herramienta válida para el estudio del terreno agrícola que aporta como ventaja fundamental el uso del dron descrito en la sección 2.3 lo que evita el desplazamiento del usuario por la zona de estudio.

Este trabajo servirá para seleccionar la región de interés de las imágenes obtenidas, así como identificar objetos en ella, calcular áreas y centroides o separar zonas, y finalmente extraer sus propiedades e índices calculados, según el interés del usuario.

Las dificultades encontradas en la realización del trabajo se refieren, particularmente, a la elección de funciones de Matlab® que disminuyeran el costo computacional a la hora de procesar las imágenes de alta definición y por supuesto, elegir el camino correcto en el desarrollo para que no perdiera calidad ni características cada fotografía para obtener unos resultados más exactos y el análisis realizado no tuviera apenas margen de error.

Se ha realizado pruebas en áreas no pertenecientes a las definidas en este proyecto además con diferentes dimensiones, para comprobar el correcto funcionamiento y obteniendo tiempos de procesado de las imágenes muy satisfactorios (en orden de dos minutos por imagen) en un ordenador personal de gama media.

La finalidad del software empleado es su correcto funcionamiento al procesar imágenes RGB y extraer sus índices, aunque también podríamos usar drones con cámaras hiperespectrales, para obtener otros índices y características.

El programa cumple con el planteamiento inicial de servir como herramienta de análisis de imágenes RGB con dron 4K para usuarios de diferentes ámbitos que deseen estudiar sus terrenos y obtener valores que le aporten beneficio en sus cuidados, ahorro de tiempo y coste. Ofrece una interfaz sencilla con varias utilidades para manejar tanto las imágenes originales como las procesadas.

Finalmente, considero importante destacar el valor que tiene el entorno de investigación en el aprendizaje durante el TFG, habiendo estado en contacto con agricultores, lo que me ha permitido conocer sus preocupaciones e inquietudes ante la digitalización de la agricultura. Esta digitalización está en pleno crecimiento y cada vez serán más demandados este tipo de programas para el análisis del medio natural y la optimización de los procesos productivos y de conservación del mismo.

6 LÍNEAS DE DESARROLLO

Se plantean las siguientes:

1. Realizar pruebas sobre otros escenarios que permita corroborar los resultados obtenidos en el procesado de los terrenos.
2. Llevar a cabo pruebas a diferentes alturas de las desarrolladas en este proyecto.
3. Profundizar en el análisis del terreno con extracción de otras características o índices.
4. Mejorar los tiempos de ejecución.
5. Realizar análisis de viabilidad de la aplicabilidad de los algoritmos desarrollados.
6. Obtener otros índices diferentes que nos aporte información del terreno.
7. Convertir el programa AgropDrone en un formato ejecutable, por ejemplo, en App Designer de Matlab®, que permita ampliar la posibilidad de uso del mismo y una mejora visual.

BIBLIOGRAFÍA Y REFERENCIAS

- [1] Q. Xie *et al.*, «Vegetation Indices Combining the Red and Red-Edge Spectral Information for Leaf Area Index Retrieval», *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 11, n.º 5, pp. 1482-1492, may 2018, doi: 10.1109/JSTARS.2018.2813281.
- [2] «Mavic Mini - Especificaciones - DJI». <https://www.dji.com/es/mavic-mini/specs> (accedido sep. 12, 2022).
- [3] «Satellite VS Drone Imagery: Knowing the Difference and Effectiveness of SuperVision Earth's Complementary Synergy | by Suvama Satish | supervisionearth | Medium». <https://medium.com/supervisionearth/satellite-vs-drone-imagery-knowing-the-difference-and-effectiveness-of-supervision-earths-90e98b78777c> (accedido sep. 12, 2022).
- [4] «plantilla-espectro-luz-visible_53562-9303.jpg (1380×849)». https://img.freepik.com/vector-premium/plantilla-espectro-luz-visible_53562-9303.jpg?w=1380 (accedido sep. 12, 2022).
- [5] «Mavic Mini - DJI». https://www.dji.com/es/mavic-mini?site=brandsite&from=insite_search (accedido sep. 12, 2022).
- [6] D. G. Manolakis, *Hyperspectral imaging remote sensing : physics, sensors, and algorithms*. Cambridge, United Kingdom: Cambridge University Press, 2016.
- [7] E. Gómez González, *Guía para conocer las cámaras de fotografía digital : [desde los consejos prácticos al análisis técnico avanzado, incluyendo de forma sencilla sus fundamentos de óptica]*. Sevilla: Victoria, 2012.
- [8] «MATLAB - Wikipedia, la enciclopedia libre». <https://es.wikipedia.org/wiki/MATLAB> (accedido sep. 12, 2022).
- [9] «MATLAB - El lenguaje del cálculo técnico - MATLAB & Simulink». <https://es.mathworks.com/products/matlab.html> (accedido sep. 12, 2022).
- [10] «JPG vs JPEG: El formato de archivo de imagen más común». <https://kinsta.com/es/blog/jpg-vs-jpeg/> (accedido sep. 12, 2022).
- [11] «Qué es una imagen JPG o JPEG: Significado y Definición». <https://aulacm.com/que-es/pg-jpeg-significado-definicion/> (accedido sep. 12, 2022).
- [12] «Tangent of argument in degrees - MATLAB tand - MathWorks España». <https://es.mathworks.com/help/matlab/ref/tand.html> (accedido sep. 12, 2022).
- [13] «Synchronize limits of multiple axes - MATLAB linkaxes - MathWorks España». https://es.mathworks.com/help/matlab/ref/linkaxes.html?searchHighlight=linkaxes&s_tid=srchtitle_linkaxes_1 (accedido sep. 12, 2022).
- [14] M. Merino Arias, *Fotografía y drones : guía completa para convertirte en un experto*. Boadilla del Monte (Madrid): JdeJ Editores, 2018.
- [15] «Mostrar varios cuadros de imagen como un montaje rectangular - MATLAB montage - MathWorks España». <https://es.mathworks.com/help/images/ref/montage.html> (accedido sep. 12, 2022).

ANEXOS

Anexo A

Ficheros de datos

Propiedades. Áreas

- *HI = 20 metros*

Imagen 1: Terreno con agua

Area	Centroid_1	Centroid_2	BoundingBox_1	BoundingBox_2	BoundingBox_3	BoundingBox_4	Area_metros2	Num_etiq
3188993	710,3997833	1113,370415	0,5	0,5	1474	2250	402,599798	1
148774	1640,556374	1760,097013	1475,5	1268,5	432	982	18,78222447	2
298501	1790,05076	660,5104841	1480,5	0,5	865	1262	37,68476202	3
23704	1850,797418	1905,638753	1761,5	1805,5	185	233	2,992551446	4
168968	2093,045032	1254,195309	1805,5	946,5	531	659	21,33165004	5
66480	2067,653761	2074,300737	1910,5	1890,5	372	360	8,392879687	6
35876	2253,712092	72,2833928	2077,5	0,5	363	204	4,529226108	7
20517	2211,439489	347,8998879	2117,5	255,5	198	175	2,590203257	8
10783	2227,758138	1706,405082	2162,5	1636,5	143	139	1,361318015	9
241491	2579,147761	1038,588759	2338,5	269,5	541	1362	30,48743845	10
133671	2630,201151	1951,280958	2415,5	1571,5	472	679	16,87552077	11
7063	2540,731984	154,6590684	2487,5	64,5	96	188	0,891680343	12
66415	2818,78544	207,6372958	2697,5	0,5	275	505	8,384673652	13
7895	2787,269411	768,488537	2716,5	669,5	140	216	0,996717586	14
16264	2881,366454	1563,126045	2779,5	1449,5	169	242	2,053276102	15
5703	2881,66053	590,1271261	2828,5	531,5	101	115	0,71998485	16
7532	2916,913436	2102,099442	2864,5	2002,5	108	190	0,950890039	17
14939	3157,619988	297,7485106	3081,5	110,5	151	323	1,885999243	18
5121	3144,438196	501,9627026	3108,5	450,5	75	122	0,646509279	19
6250	3207,7	2116,03056	3145,5	2022,5	122	168	0,789041788	20
95889	3439,344648	1990,370387	3163,5	1564,5	767	686	12,10566848	21
15312	3279,08046	772,4713297	3205,5	636,5	149	247	1,933089256	22
22389	3372,119344	1526,946983	3276,5	1333,5	217	318	2,826537053	23
9490	3423,981876	49,06174921	3332,5	0,5	176	122	1,19808105	24
80140	3595,304979	281,1385575	3417,5	0,5	424	545	10,11740942	25
15987	3592,123288	1682,177582	3490,5	1579,5	244	202	2,018305769	26
259436	3790,744411	913,9715614	3515,5	0,5	485	1609	32,75293524	27
44062	3870,129	2045,490377	3601,5	1863,5	399	387	5,56268148	28
5112	3747,384781	1782,483764	3675,5	1734,5	167	119	0,645373059	29

Imagen 2: Terreno mixto

Area	Centroid_1	Centroid_2	BoundingBox_1	BoundingBox_2	BoundingBox_3	BoundingBox_4	Area_metros2	Num_etiq
128697	249,813158	116,4085721	0,5	0,5	769	363	16,24756975	1
164589	433,1100803	353,0822108	0,5	0,5	1042	729	20,77881581	2
5600016	1841,204622	1022,223963	0,5	0,5	4000	2250	706,9834617	3
7930	641,9557377	473,7262295	559,5	418,5	166	120	1,00113622	4
6347	710,289113	362,8359855	651,5	310,5	123	103	0,801287716	5
6770	935,910192	349,1373708	864,5	265,5	132	144	0,854690064	6
16433	1311,57421	206,2724396	1166,5	117,5	240	187	2,074611791	7
976916	2977,861395	1710,043191	1319,5	958,5	2681	1292	123,3324075	8
17841	1546,723782	47,1591839	1421,5	0,5	314	168	2,252367125	9
39257	3191,118476	1902,585501	3069,5	1703,5	215	353	4,956066153	10
11094	3302,524337	2097,440959	3218,5	1992,5	160	173	1,400580735	11
21297	3515,315115	2004,693384	3419,5	1918,5	221	179	2,688675672	12
28597	3624,594363	2195,714306	3429,5	2130,5	404	120	3,61027648	13
8210	3799,859562	1029,490743	3550,5	919,5	450	237	1,036485292	14
7065	3960,953715	2072,209342	3873,5	1992,5	127	123	0,891932837	15

Imagen 3: Terreno mixto

Area	Centroid_1	Centroid_2	BoundingBox_1	BoundingBox_2	BoundingBox_3	BoundingBox_4	Area_metros2	Num_etiq
615338	1380,172622	221,9098609	0,5	26,5	2946	522	77,68438328	1
444429	1045,07977	790,7180022	0,5	597,5	2127	380	56,10768842	2
170523	390,7877119	1353,193047	0,5	1155,5	838	450	21,52796364	3
281672	608,0087549	1900,9444	0,5	1710,5	1305	479	35,56015655	4
6384	860,8074875	1569,913064	814,5	1478,5	95	173	0,805958844	5
116902	1123,588519	1353,313502	820,5	1147,5	634	466	14,75849009	6
65206	1487,843297	1851,807456	1238,5	1684,5	440	346	8,232041409	7
36293	1558,351004	1288,401565	1449,5	1120,5	255	321	4,581870976	8
8829	1615,094801	2227,990033	1502,5	2182,5	271	68	1,114631991	9
197020	2203,622932	1319,756517	1648,5	1122,5	1227	349	24,87312208	10
20232	1762,747726	1858,781089	1677,5	1706,5	169	287	2,554222952	11
67454	2055,535357	1886,756145	1835,5	1738,5	437	305	8,515843959	12
163942	2492,909023	781,1221164	2110,5	579,5	797	531	20,6971342	13
107591	2527,888959	1867,130634	2241,5	1654,5	656	378	13,5830072	14

Imagen 4: Terreno sin siembra

Area	Centroid_1	Centroid_2	BoundingBox_1	BoundingBox_2	BoundingBox_3	BoundingBox_4	Area_metros2	Num_etiq
16047	376,8729358	1456,939802	222,5	1314,5	293	278	2,025880571	1
13776	435,0852207	996,5807927	304,5	807,5	250	408	1,739174347	2
11658	444,4945102	1900,060388	342,5	1742,5	241	291	1,471783866	3
21124	717,311494	1975,226756	581,5	1723,5	324	476	2,666834996	4
7798	678,1981277	996,8470121	586,5	873,5	201	239	0,984471658	5
17284	867,0632955	1988,830479	720,5	1722,5	296	497	2,182047721	6
6195	856,8521388	1547,123002	788,5	1438,5	151	177	0,78209822	7
9320	966,4201717	1939,602468	864,5	1777,5	200	333	1,176619114	8
6269	1067,392567	1587,5591	1013,5	1487,5	125	196	0,791440475	9
14446	1295,257303	2000,261941	1176,5	1800,5	240	382	1,823759626	10
6222	1335,716008	2134,188846	1269,5	2025,5	153	225	0,78550688	11
5686	1432,605346	1727,175871	1361,5	1651,5	141	160	0,717838657	12
7304	1461,400192	2130,699754	1392,5	1989,5	159	261	0,922105795	13
10115	1742,349184	1273,395156	1692,5	1147,5	92	270	1,276985229	14
7179	1880,641315	1670,837164	1793,5	1566,5	160	200	0,906324959	15
16067	1944,804817	1971,130578	1822,5	1760,5	228	426	2,028405504	16
5159	1934,892809	1498,644311	1856,5	1415,5	138	164	0,651306653	17
8116	1996,291153	2043,112124	1914,5	1885,5	141	282	1,024618104	18
19472	2175,237212	1581,622073	2076,5	1455,5	264	299	2,45827547	19
33361	2212,73448	1983,043374	2123,5	1818,5	204	405	4,211715692	20
8968	2277,191012	2127,85426	2195,5	1980,5	164	270	1,13218028	21
5911	2260,970056	1329,369142	2204,5	1251,5	120	153	0,746244161	22
20614	2586,661686	2077,822499	2500,5	1951,5	178	299	2,602449186	23
5027	2633,063855	48,4463895	2567,5	0,5	122	121	0,634642091	24
7694	2722,698726	964,7978945	2629,5	891,5	152	153	0,971342002	25
8004	2798,32009	734,5752124	2717,5	679,5	168	124	1,010478475	26
9731	2838,860754	100,4402425	2738,5	0,5	186	300	1,228506502	27
55270	2961,960467	1860,74545	2825,5	1585,5	299	593	6,977654337	28
12039	3050,003987	146,3713764	2894,5	67,5	312	160	1,519883853	29
15613	3065,695254	880,0315122	2911,5	787,5	305	199	1,971089509	30
7405	3076,33788	1840,161648	3005,5	1722,5	138	208	0,93485671	31
13162	3211,294636	2141,742136	3131,5	1994,5	145	256	1,661658881	32
35026	3290,578028	313,5023126	3162,5	119,5	286	407	4,421916425	33
8730	3266,121077	1145,16575	3197,5	1051,5	136	174	1,102133569	34
7194	3319,346539	1449,831109	3269,5	1361,5	113	182	0,908218659	35
8666	3322,333141	2089,150935	3278,5	1975,5	93	235	1,094053781	36
12782	3470,818808	1264,826631	3344,5	1165,5	229	207	1,613685141	37
5846	3461,657544	1684,533185	3408,5	1613,5	103	143	0,738038126	38
14906	3525,684489	1496,143164	3439,5	1361,5	143	271	1,881833102	39
93580	3600,141451	2030,909649	3454,5	1700,5	320	550	11,81416488	40
78450	3863,130159	491,9891523	3614,5	107,5	386	734	9,904052519	41
7921	3696,779068	1140,543366	3619,5	1075,5	161	171	1	42
34421	3731,639551	1445,629267	3628,5	1266,5	204	413	4,34553718	43
12613	3862,754301	1187,004361	3768,5	1097,5	188	169	1,592349451	44
5336	3874,917729	260,2861694	3812,5	190,5	128	130	0,673652317	45
5693	3933,990866	1664,351133	3889,5	1602,5	111	147	0,718722384	46

- $HI = 5$ metros

Imagen 1: Terreno con agua

Area	Centroid_1	Centroid_2	BoundingBox_1	BoundingBox_2	BoundingBox_3	BoundingBox_4	Area_metros2	Num_etiq
166768	88,94537921	444,0190384	0,5	0,5	214	1012	3,445619835	1
18577	47,25854551	1175,766162	0,5	1013,5	112	280	0,383822314	2
51083	30,60429106	1667,777558	0,5	1265,5	93	985	1,055433884	3
508994	585,7042794	1469,070325	131,5	678,5	976	1572	10,51640496	4
132886	675,2401231	388,9562934	510,5	0,5	345	695	2,745578512	5
134633	1323,790438	2035,031033	919,5	1642,5	769	608	2,781673554	6
28027	1282,341813	1797,305991	1140,5	1676,5	269	271	0,579070248	7
478626	1740,529637	448,8055141	1275,5	0,5	947	1070	9,888966942	8
14146	1668,918493	2130,046232	1566,5	2051,5	238	170	0,292272727	9
14492	1817,371722	1349,294714	1674,5	1264,5	265	151	0,299421488	10
11148	1954,541084	1389,713132	1803,5	1312,5	283	143	0,230330579	11
282049	2276,258717	1398,356091	1933,5	887,5	620	1043	5,827458678	12
31361	2196,814037	509,6145531	2013,5	330,5	313	334	0,647954545	13
23792	2485,473899	102,4817165	2352,5	0,5	261	234	0,491570248	14
607891	2833,754444	782,9441347	2405,5	0,5	765	1821	12,5597314	15
65005	2900,778879	2148,68807	2664,5	2006,5	477	244	1,343078512	16
761325	3724,445111	1116,200232	3192,5	0,5	808	2063	15,72985537	17
30388	3424,988318	605,0004278	3313,5	417,5	220	364	0,62785124	18
31983	3904,084045	2136,790482	3758,5	2015,5	242	235	0,660805785	19

Imagen 2: Terreno mixto

Area	Centroid_1	Centroid_2	BoundingBox_1	BoundingBox_2	BoundingBox_3	BoundingBox_4	Area_metros2	Num_etiq
56582	151,3334629	542,3116185	0,5	247,5	331	574	1,169049587	1
51131	359,563924	2079,946295	50,5	1942,5	559	308	1,05642562	2
1712954	1267,458194	931,9492059	70,5	0,5	2184	2250	35,391611157	3
17657	530,8563176	553,5239282	364,5	464,5	349	188	0,36481405	4
21625	553,6908671	378,5724393	419,5	181,5	318	311	0,446797521	5
42071	704,7369209	101,8028095	552,5	0,5	303	248	0,869235537	6
77814	843,8767188	1822,95425	576,5	1585,5	517	445	1,607727273	7
13638	1143,874835	66,29835753	1060,5	0,5	181	192	0,28177686	8
12589	1167,636587	1955,01263	1079,5	1861,5	192	193	0,260103306	9
30633	2051,992753	1594,510038	1784,5	1437,5	457	284	0,632913223	10
17595	2131,132253	1112,652856	2037,5	1025,5	189	186	0,363533058	11
12869	2206,456601	780,849561	2101,5	680,5	197	188	0,26588843	12
50007	2446,746655	98,87985682	2292,5	0,5	358	241	1,033202479	13
188840	2988,082599	1999,793725	2545,5	1655,5	848	595	3,901652893	14
848301	3009,041088	955,4061212	2577,5	0,5	953	1762	17,52688017	15
22108	3314,821241	1830,035779	3140,5	1753,5	328	175	0,45677686	16
23590	3425,971429	971,5821958	3282,5	853,5	260	223	0,487396694	17
129828	3739,895454	546,0456142	3484,5	214,5	516	623	2,682396694	18
28209	3702,688645	113,4004041	3513,5	5,5	434	230	0,582830579	19
204930	3850,094671	1482,219421	3572,5	923,5	428	1147	4,234090909	20
12250	3940,438041	1205,960245	3869,5	1121,5	131	183	0,253099174	21

Imagen 3: Terreno mixto

Area	Centroid_1	Centroid_2	BoundingBox_1	BoundingBox_2	BoundingBox_3	BoundingBox_4	Area_metros2	Num_etiq
1687241	1760,98287	342,8603857	0,5	0,5	4000	1303	34,86035124	1
497805	677,2328281	1721,815609	0,5	1235,5	1287	1015	10,28522727	2
66618	317,9873758	1126,814735	93,5	783,5	488	544	1,376404959	3
13300	213,2524812	2146,778872	140,5	2035,5	167	215	0,274793388	4
24359	305,7120571	740,9677737	186,5	582,5	215	376	0,503285124	5
16583	631,5300609	2029,365495	489,5	1939,5	228	187	0,342623967	6
27282	1202,513525	1166,512279	1047,5	967,5	275	377	0,563677686	7
11069	1371,767549	915,8531033	1293,5	750,5	150	309	0,228698347	8
682674	1740,61433	1645,607514	1326,5	611,5	880	1639	14,10483471	9
15551	2072,322166	1126,351103	1993,5	968,5	155	339	0,321301653	10
104054	2393,824082	1436,985335	2070,5	1154,5	644	594	2,149876033	11
27584	2212,643126	1087,897078	2118,5	896,5	218	415	0,569917355	12
37282	2551,614076	2189,18757	2307,5	2034,5	476	216	0,770289256	13
24180	2527,750496	1742,116377	2349,5	1611,5	308	268	0,499586777	14
30960	2784,292894	1744,314083	2583,5	1639,5	386	209	0,639669421	15
20597	2746,566733	1582,827693	2612,5	1466,5	245	228	0,425557851	16
17874	2806,994629	1893,778058	2664,5	1825,5	261	150	0,369297521	17
303268	3219,025631	1840,608043	2751,5	1361,5	988	889	6,265867769	18
89080	3695,533812	2162,163303	3373,5	2014,5	627	236	1,840495868	19
14234	3508,253267	1673,699312	3428,5	1590,5	175	174	0,294090909	20
40354	3685,189201	1572,313575	3542,5	1320,5	281	504	0,833760331	21
30030	3850,478621	1909,662837	3723,5	1767,5	277	257	0,620454545	22

Imagen 4: Terreno sin siembra

Area	Centroid_1	Centroid_2	BoundingBox_1	BoundingBox_2	BoundingBox_3	BoundingBox_4	Area_metros2	Num_etiq
13394	170,2045692	467,9156339	0,5	363,5	284	175	0,276735537	1
19233	226,9823221	1451,438101	81,5	1312,5	278	278	0,397376033	2
74523	627,3036378	1237,90407	356,5	957,5	468	492	1,539731405	3
20010	556,7395302	1873,550525	411,5	1757,5	308	189	0,413429752	4
22494	590,742109	1604,923224	414,5	1442,5	307	342	0,464752066	5
12251	743,6973308	2187,791282	645,5	2098,5	176	152	0,253119835	6
42516	974,2798946	1545,426545	703,5	1306,5	469	519	0,878429752	7
17951	897,8914267	2128,767812	795,5	2023,5	253	227	0,37088843	8
21923	1079,20572	1699,686767	849,5	1567,5	428	259	0,452954545	9
35050	1103,765649	1909,144708	933,5	1776,5	333	276	0,724173554	10
22344	1473,153106	1862,507698	1265,5	1722,5	362	301	0,461652893	11
43213	2048,476847	1210,397288	1803,5	1044,5	422	305	0,892830579	12
31201	1974,943592	2171,10833	1838,5	2001,5	277	249	0,64464876	13
110283	2572,741855	1790,388546	2207,5	1518,5	660	581	2,27857438	14
17323	2368,215205	174,680194	2208,5	86,5	336	233	0,357913223	15
58056	2568,834281	463,1641346	2332,5	241,5	400	454	1,199504132	16
14957	2813,227185	1167,551915	2645,5	1101,5	295	157	0,309028926	17
69493	3145,062222	273,0246931	2782,5	123,5	622	322	1,435805785	18
16101	2934,198497	790,5874169	2795,5	666,5	255	255	0,332665289	19
16759	2911,783519	2142,223641	2819,5	2019,5	206	231	0,346260331	20
16246	3118,833744	2192,191247	2965,5	2115,5	334	135	0,335661157	21
150896	3452,329267	777,8071718	3039,5	402,5	747	801	3,11768595	22
18727	3165,90794	865,7703316	3051,5	747,5	254	256	0,386921488	23
15761	3150,67096	1397,695641	3064,5	1216,5	194	329	0,325640496	24
14479	3316,375993	1678,575178	3175,5	1584,5	255	185	0,299152893	25
11976	3365,478457	513,4607548	3212,5	436,5	328	159	0,247438017	26
16662	3334,188813	406,7803985	3219,5	299,5	222	212	0,344256198	27
43545	3419,919233	1376,181881	3267,5	1152,5	311	431	0,899690083	28
16912	3491,134934	1821,752306	3273,5	1703,5	426	234	0,349421488	29
12081	3551,382915	109,0898104	3440,5	0,5	218	252	0,249607438	30
21229	3631,896557	1331,795092	3500,5	1226,5	238	213	0,438615702	31
12733	3648,74303	887,7712244	3534,5	777,5	200	223	0,263078512	32
88716	3863,48316	286,3508161	3560,5	0,5	440	663	1,832975207	33
32025	3808,810679	2039,875098	3685,5	1844,5	242	406	0,661673554	34
95145	3874,805024	1142,329592	3695,5	656,5	305	959	1,965805785	35
18198	3937,166777	2149,350148	3857,5	2033,5	143	217	0,375991736	36
18125	3951,857986	1888,072772	3864,5	1767,5	136	273	0,374483471	37
18568	3961,875	1588,350118	3887,5	1392,5	113	372	0,383636364	38

Anexo B

Cálculo de índices

- $HI = 20$ metros

Imagen 1: Terreno con agua

nums_indices	indices_cal
Indice 1	-0,276315789
Indice 2	-0,171052632
Indice 3	-0,105263158
Indice 4	-0,126984127
Indice 5	0,206349206
Indice 6	-0,333333333
Indice 7	0,381818182
Indice 8	0,145454545
Indice 9	0,236363636

Imagen 2: Terreno mixto

nums_indices	indices_cal
Indice 1	-0,448598131
Indice 2	-0,056074766
Indice 3	-0,392523364
Indice 4	-0,415841584
Indice 5	0,059405941
Indice 6	-0,475247525
Indice 7	0,813559322
Indice 8	0,711864407
Indice 9	0,101694915

Imagen 3: Terreno mixto

nums_indices	indices_cal
Indice 1	-0,886363636
Indice 2	-0,454545455
Indice 3	-0,431818182
Indice 4	-0,791666667
Indice 5	0,833333333
Indice 6	-1,625
Indice 7	7,8
Indice 8	3,8
Indice 9	4

Imagen 4: Terreno sin siembra

nums_indices	indices_cal
Indice 1	-0,72
Indice 2	-0,4
Indice 3	-0,32
Indice 4	-0,533333333
Indice 5	0,666666667
Indice 6	-1,2
Indice 7	2,571428571
Indice 8	1,142857143
Indice 9	1,428571429

- $HI = 5$ metros

Imagen 1: Terreno con agua

nums_indices	indices_cal
Indice 1	-0,300884956
Indice 2	-0,221238938
Indice 3	-0,079646018
Indice 4	-0,102272727
Indice 5	0,284090909
Indice 6	-0,386363636
Indice 7	0,430379747
Indice 8	0,113924051
Indice 9	0,316455696

Imagen 2: Terreno mixto

nums_indices	indices_cal
Indice 1	-0,625
Indice 2	-0,318181818
Indice 3	-0,306818182
Indice 4	-0,45
Indice 5	0,466666667
Indice 6	-0,916666667
Indice 7	1,666666667
Indice 8	0,818181818
Indice 9	0,848484848

Imagen 3: Terreno mixto

nums_indices	indices_cal
Indice 1	-0,962962963
Indice 2	-0,074074074
Indice 3	-0,888888889
Indice 4	-0,96
Indice 5	0,08
Indice 6	-1,04
Indice 7	26
Indice 8	24
Indice 9	2

Imagen 4: Terreno sin siembra

nums_indices	indices_cal
Indice 1	-0,422222222
Indice 2	0,244444444
Indice 3	-0,666666667
Indice 4	-0,535714286
Indice 5	-0,196428571
Indice 6	-0,339285714
Indice 7	0,730769231
Indice 8	1,153846154
Indice 9	-0,423076923

Anexo C

Códigos del programa

AgropDrone.m

```

clear all
close all
clc

%% Ma Auxiliadora Peña Moreno

% TFG: Desarrollo de análisis de imagen de dron en el rango visible 4K

% Programa AgropDrone

%% Parámetros iniciales de la cámara/imagen

% Parámetros de la cámara del dron DJI Mavic Mini

aFOV= 83; % 83 grados
factor = round(2*tand(aFOV/2),2); % Factor de
multiplicación campo de visión [12]

% Tamaño folio de referencia para escala
ancho_ref=297; %mm
alto_ref=210; %mm

%% Parámetros iniciales del procesado:

% Carpeta origen

carp_0=cd;

% Realizar el procesado sin mostrar imágenes

flagFiguras = 0; % 0 no muestro, 1
muestro

% Aplicar dilatación a la imagen

flagDilatar = 1; % 0 no aplico, 1
aplico

% Mostrar colormap en la figura

flagColorMap = 1; % 0 no aplico, 1
aplico

%% Parámetros variables

prompt={'Tiempo espera (s):', 'Grosor cuadrícula (mm):', '¿Otsu manual?
(True=1/False=0)', 'Umbral h1 (u.a.):', 'Umbral h2 (u.a.):'};
dlg_title='Ingrese valores correspondientes';
width = 60;
height = 1;
num_lines=[height, width];
def_out= {'1', '1', '0', '235', '135'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
t= str2num(inputH{1});

```

```

grosor= str2num(inputH{2});
otsu= str2num(inputH{3});
umbral1= str2num(inputH{4});
umbral2= str2num(inputH{5});

%% Selección carpetas donde están las imágenes

carp_1 = uigetdir('C:\', 'Selecciona carpeta a 20 metros');
carp_2 = uigetdir('C:\', 'Selecciona carpeta a 5 metros');

mkdir Imagenes_20_metros Procesado_Intermedio %creamos
subcarpetas
mkdir Imagenes_5_metros Procesado_Intermedio

mkdir Imagenes_20_metros Imagenes_Procesadas %creamos
subcarpetas
mkdir Imagenes_5_metros Imagenes_Procesadas

mkdir Imagenes_20_metros Propiedades %creamos
subcarpetas
mkdir Imagenes_5_metros Propiedades

mkdir Imagenes_20_metros Indices_calculados %creamos
subcarpetas
mkdir Imagenes_5_metros Indices_calculados

mkdir Imagenes_20_metros Imagenes_Indices %creamos
subcarpetas
mkdir Imagenes_5_metros Imagenes_Indices

mkdir Imagenes_20_metros Imagenes_Indices_CB %creamos
subcarpetas
mkdir Imagenes_5_metros Imagenes_Indices_CB

% mkdir Imagenes_20_metros Propiedades_pixel %creamos
subcarpetas
% mkdir Imagenes_5_metros Propiedades_pixel % si guardamos
separado

Cim_20m = fullfile(carp_1, '\*.jpg'); %guardar en formato jpg
Cim_5m = fullfile(carp_2, '\*.jpg');

excel_20m = fullfile(carp_1, '\*.xlsx'); %guardar en formato excel
excel_5m = fullfile(carp_2, '\*.xlsx');

Imagenes_20_metros=dir(Cim_20m);
Imagenes_5_metros=dir(Cim_5m);

im_20m_excel=dir(excel_20m);
im_5m_excel=dir(excel_5m);

outputDir_20=[carp_1 '\Procesado_Intermedio\']; %Carpetas de salida
outputDir_5=[carp_2 '\Procesado_Intermedio\'];

outputDir_20P=[carp_1 '\Imagenes_procesadas\']; %Carpetas de salida
outputDir_5P=[carp_2 '\Imagenes_procesadas\'];

outputDir_20I=[carp_1 '\Imagenes_Indices']; %Carpetas de salida
outputDir_5I=[carp_2 '\Imagenes_Indices'];

```



```

outputDir_20Ic=[carp_1 '\\Imágenes_Indíces_CB'];           %Carpetas de salida
outputDir_5Ic=[carp_2 '\\Imágenes_Indíces_CB'];

% outputExc_20P=[carp_1 '\\Propiedades_pixel\\'];         %Carpetas de salida
% outputExc_5P=[carp_2 '\\Propiedades_pixel\\'];         %guardado separado

outputExc_20=[carp_1 '\\Propiedades\\'];                 %Carpetas de salida
outputExc_5=[carp_2 '\\Propiedades\\'];

outputExcInd_20=[carp_1 '\\Indíces_calculados\\'];       %Carpetas de salida
outputExcInd_5=[carp_2 '\\Indíces_calculados\\'];

% Decidir si se calcula la escala y de que forma
Q_escal=questdlg('¿Qué método desea usar para detección de escala?', 'Opciones',
'Método 1: Cámara y altura', 'Método 2: Con referencia', 'Sin escala', 'Sin escala');

switch Q_escal
    case 'Método 1: Con referencia'

        im_ref = uigetfile('*.jpg', 'Selecciona imagen de referencia');
        im_ref = imread(im_ref);           % Leo imágenes

        % Extraigo las dimensiones de la imagen

        dim_imR=size(im_ref);
        im_alto = dim_imR(1);              % Pixeles de alto
        im_ancho = dim_imR(2);            % Pixeles de ancho

        div= im_ancho*0.125;               % División imagen
        figuras (pixeles)

        figure; imshow(im_ref), caxis([0 1920]), title('Imagen de referencia'); %Imagen
referencia con ejes
        xlabel('Píxeles')
        ylabel('Píxeles')
        axis on

        help = msgbox("Pulse intro tras seleccionar la zona","Info","help");
        calc=imdistline;
        pause();
        pixeles=getDistance(calc);
        close(gcf);

        escalaH1=(ancho_ref*div)/pixeles;   %mm
        convH1=round((escalaH1/1000), 3);   %metros

        flagEscala = 1;

    case 'Sin escala'

        flagEscala = 2;

    case 'Método 2: Cámara y altura'

        flagEscala = 3;

```

```

end

%% Imagen 1 en RGB

h1= 20;      % Altura, h1= 20 metros

for i = 1:length(Imagenes_20_metros)                % Recorre el
número de imagenes en el directorio
    imagenes_20=Imagenes_20_metros(i).name;        % Obtengo nombre
imagenes

    nombre_im1=[carp_1 '\'];                        % Recorre el directorio
    f = imread(strcat(nombre_im1, imagenes_20));    % Leo imagenes

    % Extraigo las dimensiones de la imagen

    dim_imH1=size(f);
    im_alto = dim_imH1(1);                          % Pixeles de alto
    im_ancho = dim_imH1(2);                          % Pixeles de
ancho

    div= im_ancho*0.125;                             % División imagen
figuras (pixeles)

    if(flagEscala == 3)

        FOV_h1 = round((h1*factor),2);              % Campo
visión en metros
        escalaH1 = round((FOV_h1/im_ancho)*100,2);  % Escala,
pixeles por centímetro

        convH1 = round((escalaH1*div)/100, 2);      % Conversión
a metros
    end

figure; imshow(f), caxis([0 1920]), title('Imagen original a h1= 20 metros'); %Imagen
original con ejes
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(f);

for row = 1 : 100 : rows
    yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
end

for col = 1 : 500 : columns
    xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
end

line([500,0],[2100,2100], 'Color', [0.6350 0.0780 0.1840], 'LineWidth', 3);
text (125,2150,[num2str(convH1), ' m'], 'Color', [0.6350 0.0780 0.1840],
'FontSize', 12, 'FontWeight', 'bold');

pause(t);

```

```

% Decidir si se continua con el procesado
seleccion=questdlg('¿Quiere detener el procesado y pasar a la siguiente imagen?',
'Opciones', 'Si', 'No', 'No');
switch seleccion
    case 'Si'
        close(gcf);

    case 'No'

end

if(seleccion=='Si')           %Si el usuario cierra, no procesa esa imagen y pasa a la
siguiente.
    continue
    i=i+1;
end

[M, N, canales] = size (f);           % M filas, N columnas

% Tres matrices bidimensionales, correspondientes a planos R,G y B.

Im1_R = f(:,:,1);           %Plano rojo
Im1_G = f(:,:,2);           %Plano verde
Im1_B = f(:,:,3);           %Plano azul

if(flagFiguras == 1)           % Mostrará figura

F1= figure;
ax1_F1 =subplot(2,3,[1,2,3]); imshow(f); title('Imagen original: h1')
ax2_F1 =subplot(2,3,4); imshow(Im1_R); title('Canal rojo');
ax3_F1 =subplot(2,3,5); imshow(Im1_G); title('Canal verde');
ax4_F1 =subplot(2,3,6); imshow(Im1_B); title('Canal azul');

    ax_v_F1 = [ax1_F1 ax2_F1 ax3_F1 ax4_F1];

    linkaxes(ax_v_F1);           % Sincronización imagenes de la figura [13]

pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.
end

%% Mejora de las imagenes
%% Filtrado de ruido de la imagen h1

rfiltro1=medfilt2(Im1_R);           %Filtro mediana h1
gfiltro1=medfilt2(Im1_G);
bfiltro1=medfilt2(Im1_B);

iRGB_h1 = cat(3, rfiltro1, gfiltro1, bfiltro1);           % Unimos para formar la
imagen RGB h1

% Si se desea mostrar la imagen filtrada:
% F2= figure;
% ax1_F2 = subplot (2,2, [1,2]); imshow(f); title('Imagen original h1')
% ax2_F2 = subplot (2,2,[3,4]); imshow(iRGB_h1); title('Imagen filtrada h1')
% ax_v_F2 = [ax1_F2 ax2_F2];
%
% linkaxes(ax_v_F2);

```

```

% Guardar imagenes filtradas h1
imgsave=sprintf('filtrada_h1_%g.jpg',i);
imwrite(iRGB_h1, [outputDir_20 imgsave]);

% Pasamos la imagen filtrada RGB h1 a L*a*b para tratamiento

lab_h1=rgb2lab(iRGB_h1);

L_1=lab_h1(:,:,1)/100;
L_1= adapthisteq(L_1,'NumTiles',[8 8], 'ClipLimit',0.005);
lab_h1(:,:,1)=L_1*100;

% Vuelta a RGB
fcontrast= lab2rgb(lab_h1);

if(flagFiguras == 1) % Mostrará figura
figure;
imshowpair(iRGB_h1, fcontrast, 'montage')
title('Original - Mejora de contraste h1') [14]

pause(t); % Dejo un tiempo prudencial de decisión de cierre de figura.
end

% Guardar imagenes mejoradas h1
imgsave=sprintf('Mcontraste_h1_%g.jpg',i);
imwrite(fcontrast, [outputDir_20 imgsave]);

% [A,B] = size (fcontrast); % A filas, B columnas
fcontrast8=uint8(round(fcontrast*255));
% Tres matrices bidimensionales, correspondientes a planos R,G y B.
Im1c_R = fcontrast8(:,:,1); %Plano rojo
Im1c_G = fcontrast8(:,:,2); %Plano verde
Im1c_B = fcontrast8(:,:,3); %Plano azul

iRGBc_h1 = cat(3,Im1c_R, Im1c_G, Im1c_B); % Unimos para formar la imagen RGB h1

% Imagen h1 en tonos de gris
f_gris = rgb2gray(iRGBc_h1); %Imagen escala de grises

% max(max(f_grises)); % Aumento automático del contraste y valor
del umbral
% f_gris=adapthisteq(f_grises); % Si lo hago con la imagen BW
%f_gris=imadjust(f_grises); % Mejores resultados con adapthisteq
%figure, imshow(f_gris), title('Imagen h1 en tonos de gris') %Comprobación

%% Binarización de la imagen h1(+ Otsu)

modoManual = otsu;

if modoManual
% Umbral elegido a mano, analizando el histograma y obtención de la plantilla
binaria:
%prompt2= 'Introduce el valor del umbral: '; % Introduciendo el valor por ventana
comandos
%umbral1= input(prompt2);

```

```

    umbralNorm = umbral1/255;
else
    % Umbral elegido automáticamente mediante función de Matlab que usa el método de
    Otsu:
    umbralNorm = graythresh (f_gris);          % Calcula un umbral global a partir de
    im. escala grises
    umbral1 = umbralNorm * 255;
end

PlantillaF = im2bw (f_gris,umbralNorm);

% Si se desea imprimir el valor del umbral en la imagen:
% line ([umbral1,umbral1],[0,400],'Color','r','LineWidth',2);    % Imprime en la
imagen el valor del umbral
% text (115,500,['U=',num2str(umbral1)]);

fBinarizada = 255*uint8(PlantillaF);
a = f_gris .* uint8(PlantillaF);

if(flagFiguras == 1)                                     % Mostrará figura
figure; imshow([f_gris,fBinarizada,a]), title('Imagen escala de grises - Imagen
binarizada - Plantilla aplicada: h1');

pause(t);        % Dejo un tiempo prudencial de decisión de cierre de figura.
end

umbral_1 = CalculaUmbralOtsu (imhist(f_gris));

[umbral_1,tabVarIntraclase] = CalculaUmbralOtsu (imhist(f_gris));
fBinarizada = 255*uint8(f_gris<umbral1); %0: fondo, 255: objeto

if(flagFiguras == 1)                                     % Mostrará figura
figure; imshow(fBinarizada), title('Imagen binarizada h1');
%axis on          % Si deseo ejes
end
% Guardar imagenes binarizadas h1
imgsave=sprintf('binarizada_h1_%g.jpg',i);
imwrite(fBinarizada, [outputDir_20 imgsave]);

%% Operaciones Morfológicas h1

% Relleno de regiones y huecos de la imagen (aparentemente no aporta nada)

rellenoI1 = imfill(fBinarizada,'holes');

%figure, imshow(rellenoI1), title('Huecos rellenos h1');    %Si deseo sola

% Guardar imagenes tras relleno de huecos
imgsave=sprintf('relleno_h1_%g.jpg',i);
imwrite(rellenoI1, [outputDir_20 imgsave]);

%Creamos las diferentes estructuras
se1 = strel('square',3);          % Cuadrado 3x3
se2 = strel('line',5,45);        % Línea de longitud 5 y ángulo de 45°
se3 = strel('disk',2);          % Disco de radio 2
se4 = strel('ball',15,5);        % Bola de radio 15 y alto 5
se5 = strel('diamond',1);        % Diamante de tamaño 1
se6 = strel('rectangle',[7,4]);  % Rectangulo de 7x4

```

```

% Erosión (Puede aportar)

erodeI1 = imerode(rellenoI1,se3);
erodeI1=bwareaopen(erodeI1,5000);           %Valor tras ver valores area
% figure, imshow(erodeI1), title('Erosión h1'); %Si deseo imagen sola

if(flagFiguras == 1)                       % Mostrará figura
F3= figure;
ax1_F3 = subplot (2,2, [1,2]); imshow(rellenoI1); title('Huecos rellenos h1')
ax2_F3 =subplot (2,2,[3,4]); imshow(erodeI1); title('Erosión h1')

ax_v_F3 = [ax1_F3 ax2_F3];

linkaxes(ax_v_F3);

pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.
end

% Guardar imagenes erosionadas
imgsave=sprintf('erosion_h1_%g.jpg',i);
imwrite(erodeI1, [outputDir_20 imgsave]);

% Dibujo el contorno de la imagen h1

if(flagFiguras == 1)                       % Mostrará figura
figure, imcontour(erodeI1,3), title('Contorno h1');
xlabel('Píxeles')
ylabel('Píxeles')

pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.
end
% Apertura: Erosión+Dilatación

% OpenI1 = imopen(f_gris,se5);
% figure, imshow(OpenI1), title('Apertura h1');

% Cierre: Dilatación+Erosión (No aporta casi nada)

% closeI1 = imclose(f_gris,se5);
% figure, imshow(closeI1), title('Cierre h1');

% Dilatación (No aporta casi nada)

% dilateI1 = imdilate(f_gris,se3);
% figure, imshow(dilateI1), title('Dilatación h1')

%% Transformada de Hough h1 (A usar si se cree conveniente)

%% Preprocesado para acentuar bordes más importantes.
% Bordes_1 = edge(f_gris,'canny');
% Bordes_1 = bwareaopen(Bordes_1, 40);
%% T. de Hough y búsqueda de máximos.
% [H,T,R] = hough(Bordes_1);
% peaks = houghpeaks(H,110);
%% Buscamos las líneas y las mostramos con valores por defecto
% lineas_1 = houghlines(Bordes_1,T,R,peaks);

```

```

% figure, imshow(f_gris), hold on
% max_len = 0;
% for n = 1:length(lineas_1)
% xy = [lineas_1(n).point1; lineas_1(n).point2];
% plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
% % Dibujamos inicios y finales de las líneas
% plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
% plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','green');
% end
% % Cambiamos los valores de FillGap y MinLength para mostrar líneas más largas
% lineas_1 = houghlines(Bordes_1,T,R,peaks,'FillGap',120,'MinLength',15);
% figure, imshow(f_gris), hold on
% max_len = 0;
% for n = 1:length(lineas_1)
% xy = [lineas_1(n).point1; lineas_1(n).point2];
% plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
% % Dibujamos los principios y finales de las líneas.
% plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
% plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','green');
% end

%% Etiquetado h1
figure, imshow(f);
etiq=bwlabel(erodeI1);
imshow(label2rgb(etiq)), caxis([0 1920]), title('Imagen etiquetada h1');
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(f);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

CC2=bwconncomp(etiq);           %Busca y cuenta los componentes conectados
disp(CC2)

propiedades_h1=regionprops(etiq, 'basic');   %Todas las propiedades 'all' sino solo
las básicas

% Identificamos las regiones encontrados

for u=1:length(propiedades_h1)
    box1=propiedades_h1(u).BoundingBox;
    point1=propiedades_h1(u).Centroid;
    rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
        'LineWidth',3)
    text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 14,
'FontWeight', 'bold');
end

pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.

% Guardar imagenes etiquetadas h1

```

```

imgsave=fullfile(outputDir_20P,['etiquetada_h1_' num2str(i) '.jpg']);
saveas(gcf, imgsave);

% Muestro el área de cada región interés en metros cuadrados

% for u=1:length(propiedades_h1)
%
%     Area1_m=round((propiedades_h1(u).Area)/(escalaH1^2),3)           % metros
cuadrados
%     T = table(Area1_m);
% end
p1 = struct2table(propiedades_h1);
p1.Area_metros2 = (p1.Area/escalaH1^2)/(10^4);                       % metros cuadrados

netiq=length(propiedades_h1);
num_etiq= (1:1:netiq)';
p1.Num_etiq =num_etiq;

% Unir tablas propiedades y area en metros cuadrados
propiedades_1= join(struct2table(propiedades_h1), p1);

% Etiquetas sobre imagen original

if(flagFiguras == 1)                                               % Mostrará figura
figure;
imshow(f), caxis([0 1920]); title(' Imagen original etiquetada')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(f);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

% Mostrar las regiones
for u=1:length(propiedades_h1)
    box1=propiedades_h1(u).BoundingBox;
    point1=propiedades_h1(u).Centroid;
    rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
        'LineWidth',3)
    text(point1(1), point1(2), sprintf('%d', u), 'Color', 'w', 'FontSize', 14,
'FontWeight', 'bold');
end

pause(t);                % Dejo un tiempo prudencial de decisión de cierre de figura.
end

%% Cálculo de índices de la etiqueta seleccionada

% Elegir etiqueta deseada

prompt={'¿De qué etiqueta desea calcular los índices?'};

```



```

dlg_title='Ingrese valor correspondiente';
width = 60;
height = 1;
num_lines=[height, width];
def_out= {'1'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
number= str2num(inputH{1});

[r, c] = find(etiq==number);           % [filas, columnas], buscar
coordenadas de la etiqueta
rc = [r c];
%hold on; plot(1197,154,'y*')         %comprobar
iRGBc_h1 = double(iRGBc_h1);

ind_1 = zeros(im_alto,im_ancho);
ind_2 = zeros(im_alto,im_ancho);
ind_3 = zeros(im_alto,im_ancho);
ind_4 = zeros(im_alto,im_ancho);
ind_5 = zeros(im_alto,im_ancho);
ind_6 = zeros(im_alto,im_ancho);
ind_7 = zeros(im_alto,im_ancho);
ind_8 = zeros(im_alto,im_ancho);
ind_9 = zeros(im_alto,im_ancho);

% Comprobamos que la imagen binaria corresponde con la zona elegida

im_bin = [];

im_bin = zeros(im_alto,im_ancho);

% ponemos a 1 los puntos de la zona

N_fil = length(r);

for ind_q = 1:N_fil
    im_bin(r(ind_q),c(ind_q)) = 1;
end

figure;
imagesc(im_bin)
title('Imagen Binaria de zona elegida')
xlabel('x (px)')
ylabel('y (px)')

pause(t);

% Cálculo de índices

for ind = 1:N_fil           % Pongo a 1 los puntos de la zona

    R = iRGBc_h1(r(ind),c(ind),1);
    G = iRGBc_h1(r(ind),c(ind),2);
    B = iRGBc_h1(r(ind),c(ind),3);

```

```
% Cálculo índice 1
ind1_calc = (B - R)./ R;
ind_1(r(ind),c(ind)) = ind1_calc;
% Cálculo índice 2
ind2_calc = (G - R)./ R;
ind_2(r(ind),c(ind)) = ind2_calc;
% Cálculo índice 3
ind3_calc = (B - G)./ R;
ind_3(r(ind),c(ind)) = ind3_calc;
% Cálculo índice 4
ind4_calc = (B - G)./ G;
ind_4(r(ind),c(ind)) = ind4_calc;
% Cálculo índice 5
ind5_calc = (R - G)./ G;
ind_5(r(ind),c(ind)) = ind5_calc;
% Cálculo índice 6
ind6_calc = (B - R)./ G;
ind_6(r(ind),c(ind)) = ind6_calc;
% Cálculo índice 7
ind7_calc = (R - B)./ B;
ind_7(r(ind),c(ind)) = ind7_calc;
% Cálculo índice 8
ind8_calc = (G - B)./ B;
ind_8(r(ind),c(ind)) = ind8_calc;
% Cálculo índice 9
ind9_calc = (R - G)./ B;
ind_9(r(ind),c(ind)) = ind9_calc;
```

```
end
```

```
% Guardo los valores de los índices en una tabla
```

```

nums_indices={'Indice 1'; 'Indice 2'; 'Indice 3'; 'Indice 4'; 'Indice 5'; 'Indice 6';
'Indice 7'; 'Indice 8'; 'Indice 9'};
indices_cal=[ind1_calc; ind2_calc; ind3_calc; ind4_calc; ind5_calc; ind6_calc;
ind7_calc; ind8_calc; ind9_calc];
Indices_Value= table(nums_indices, indices_cal);

% Guardo en excel los valores de los índices

excsave=sprintf('indices_img20_%g_etiq%g.xlsx',i, number);
writetable(Indices_Value, [outputExcInd_20 excsave]);

% Decidir si se muestran los resultados por zonas o por índices

Q_indice=questdlg('¿Como desea ver los resultados tras cálculo de índices?',
'Opciones', 'Por Zonas', 'Por índices', 'Por índices');

switch Q_indice

    case 'Por Zonas'

        flagIndice = 1;

    case 'Por índices'

        flagIndice = 2;

end

    if(flagIndice == 1)

% Índice 1

        % Mostrando el colorbar
        figure;
        imagesc(ind_1)
        tit_IND1 = strcat('Índice 1 de zona = ', num2str(number));
        title(tit_IND1)
        xlabel('x (px)')
        ylabel('y (px)')
        colorbar
        colormap hot

        pause(t);

% Guardar imagenes con índice 1 con colorbar
        imgsave=fullfile(outputDir_20Ic,['Imh1_' num2str(i) '_Ind1etiq_' num2str(number)
'.jpg']);
        saveas(gcf, imgsave);

        % Sin colorbar
        figure;
        imagesc(ind_1)
        title('Índice 1')
        xlabel('x (px)')
        ylabel('y (px)')
        colormap hot

```

```

    pause(0);

% Guardar imagenes con índice 1 sin colorbar
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind1letiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 2

% Mostrando el colorbar

    figure;
    imagesc(ind_2)
    tit_IND2 = strcat('Índice 2 de zona = ', num2str(number));
    title(tit_IND2)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap hot

    pause(t);

% Guardar imagenes con índice 2 con colorbar
imgsave=fullfile(outputDir_20Ic,['Imh1_' num2str(i) '_Ind2etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
    figure;
    imagesc(ind_2)
    title('Índice 2')
    xlabel('x (px)')
    ylabel('y (px)')
    colormap hot

    pause(0);

% Guardar imagenes con índice 2
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind2etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 3

% Mostrando el colorbar
    figure;
    imagesc(ind_3)
    tit_IND3 = strcat('Índice 3 de zona = ', num2str(number));
    title(tit_IND3)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap hot

    pause(t);

```

```
% Guardar imagenes con índice 3 con colorbar
imgsave=fullfile(outputDir_20Ic,['Imh1_' num2str(i) '_Ind3etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar

figure;
imagesc(ind_3)
title('Índice 3')
xlabel('x (px)')
ylabel('y (px)')
colormap hot

pause(0);

% Guardar imagenes con índice 3 sin colorbar
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind3etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 4

% Mostrando el colorbar

figure;
imagesc(ind_4)
tit_IND4 = strcat('Índice 4 de zona = ', num2str(number));
title(tit_IND4)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 4 con colorbar
imgsave=fullfile(outputDir_20Ic,['Imh1_' num2str(i) '_Ind4etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_4)
title('Índice 4')
xlabel('x (px)')
ylabel('y (px)')
colormap winter

pause(0);

% Guardar imagenes con índice 4 sin colorbar
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind4etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);
```

```
% Índice 5

% Mostrando el colorbar
figure;
imagesc(ind_5)
tit_IND5 = strcat('Índice 5 de zona = ', num2str(number));
title(tit_IND5)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 5 con colorbar
imgsave=fullfile(outputDir_20Ic,['Imh1_' num2str(i) '_Ind5etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_5)
title('Índice 5')
xlabel('x (px)')
ylabel('y (px)')
colormap winter

pause(0);

% Guardar imagenes con índice 5 sin colorbar
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind5etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 6

% Mostrando el colorbar
figure;
imagesc(ind_6)
tit_IND6 = strcat('Índice 6 de zona = ', num2str(number));
title(tit_IND6)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 6 con colorbar
imgsave=fullfile(outputDir_20Ic,['Imh1_' num2str(i) '_Ind6etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_6)
title('Índice 6')
xlabel('x (px)')
```

```
ylabel('y (px)')
colormap winter

pause(0);

% Guardar imagenes con índice 6 sin colorbar
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind6etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 7

% Mostrando el colorbar

figure;
imagesc(ind_7)
tit_IND7 = strcat('Índice 7 de zona = ', num2str(number));
title(tit_IND7)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap cool

pause(t);

% Guardar imagenes con índice 7 con colorbar
imgsave=fullfile(outputDir_20Ic,['Imh1_' num2str(i) '_Ind7etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_7)
title('Índice 7')
xlabel('x (px)')
ylabel('y (px)')
colormap cool

pause(0);

% Guardar imagenes con índice 7 sin colorbar
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind7etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 8

% Mostrando el colorbar

figure;
imagesc(ind_8)
tit_IND8 = strcat('Índice 8 de zona = ', num2str(number));
title(tit_IND8)
xlabel('x (px)')
ylabel('y (px)')
colorbar
```

```

colormap cool

pause(t);

% Guardar imagenes con índice 8 con colorbar
imgsave=fullfile(outputDir_20Ic,['Imh1_' num2str(i) '_Ind8etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_8)
title('Índice 8')
xlabel('x (px)')
ylabel('y (px)')
colormap cool

pause(0);

% Guardar imagenes con índice 8 sin colorbar
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind8etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 9

% Mostrando el colorbar

figure;
imagesc(ind_9)
tit_IND9 = strcat('Índice 9 de zona = ', num2str(number));
title(tit_IND9)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap cool

pause(t);

% Guardar imagenes con índice 9 con colorbar
imgsave=fullfile(outputDir_20Ic,['Imh1_' num2str(i) '_Ind9etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_9)
title('Índice 9')
xlabel('x (px)')
ylabel('y (px)')
colormap cool

pause(0);

% Guardar imagenes con índice 9 sin colorbar
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind9etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

```



```

%% Muestro la imagen Original y la etiquetada

F4= figure;
ax1_F4 = subplot (1,4, [1,2]); imshow(f), caxis([0 1920]); title(' Imagen original
h1')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(f);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

ax2_F4 =subplot (1,4,[3,4]); imshow(label2rgb(etiq)), caxis([0 1920]); title('Imagen
etiquetada h1')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(f);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

for u=1:length(propiedades_h1)
    box1=propiedades_h1(u).BoundingBox;
    point1=propiedades_h1(u).Centroid;
    rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
        'LineWidth',3)
    text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 14,
'FontWeight', 'bold');
end

ax_v_F4 = [ax1_F4 ax2_F4];

    linkaxes(ax_v_F4);
pause(t);

%% Montaje con todos los índices

carp_1I = outputDir_20I;

Cim_20mI = fullfile(carp_1I, '*.jpg'); %guardar en formato jpg

im_20mI=dir(Cim_20mI);

```

```

cd(carp_1I)

imagenes_20I= string({im_20mI.name});

figure;
montage(imagenes_20I, 'Size',[3 3]); [15]
tit_20m = strcat('Índices de la zona = ', num2str(number));
title(tit_20m)

pause(t);

cd(carp_0);

    end

    if(flagIndice == 2)

        select_indice = msgbox(["Índice 1: BR-R"; "Índice 2: GR-R"; "Índice 3: BG-R"; ...
            "Índice 4: BG-G"; "Índice 5: RG-G"; "Índice 6: BR-G"; "Índice 7: RB-B"; ...
            "Índice 8: GB-B"; "Índice 9: RG-B"], "Info índices","help");
        pause(t);

prompt={'Número del índice que desea ver (1-9):'};
dlg_title='Ingrese valor correspondiente';
width = 60;
height = 1;
num_lines=[height, width];
def_out= {'1'};

inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
number_ind= str2num(inputH{1});

% Índice 1

if(number_ind==1)
    figure;
    imagesc(ind_1)
    tit_IND1 = strcat('Índice 1 de zona = ', num2str(number));
    title(tit_IND1)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap hot

    pause(t);

% Guardar imagenes con índice 1
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind1eti_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F5 = figure;

ax1_F5 = subplot (1,6, [1,2]); imshow(f); title('Imagen original a h1')

```

```

    ax2_F5 = subplot (1,6, [3,4]); imshow(label2rgb(etiq)), caxis([0 1920]);
title('Imagen etiquetada h1')

    for u=1:length(propiedades_h1)
    box1=propiedades_h1(u).BoundingBox;
    point1=propiedades_h1(u).Centroid;
    rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
        'LineWidth',1.5)
    text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
    end

    ax3_F5 = subplot (1,6, [5,6]); imshow(ind_1); colormap hot; title(tit_IND1)

    ax_v_F5 = [ax1_F5 ax2_F5 ax3_F5];

    linkaxes(ax_v_F5);

    pause(t);
    end

% Índice 2

if(number_ind==2)
    figure;
    imagesc(ind_2)
    tit_IND2 = strcat('Índice 2 de zona = ', num2str(number));
    title(tit_IND2)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap hot

    pause(t);

    % Guardar imagenes con índice 2
    imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind2etiq_' num2str(number)
'.jpg']);
    saveas(gcf, imgsave);

    F5 = figure;

    ax1_F5 = subplot (1,6, [1,2]); imshow(f); title('Imagen original a h1')
    ax2_F5 = subplot (1,6, [3,4]); imshow(label2rgb(etiq)), caxis([0 1920]);
title('Imagen etiquetada h1')

    for u=1:length(propiedades_h1)
    box1=propiedades_h1(u).BoundingBox;
    point1=propiedades_h1(u).Centroid;
    rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
        'LineWidth',1.5)
    text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
    end

    ax3_F5 = subplot (1,6, [5,6]); imshow(ind_2); colormap hot; title(tit_IND2)

```

```

ax_v_F5 = [ax1_F5 ax2_F5 ax3_F5];

linkaxes(ax_v_F5);

pause(t);
end

% Índice 3

if(number_ind==3)
    figure;
    imagesc(ind_3)
    tit_IND3 = strcat('Índice 3 de zona = ', num2str(number));
    title(tit_IND3)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap hot

    pause(t);

    % Guardar imagenes con índice 3
    imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind3eti' num2str(number)
    '.jpg']);
    saveas(gcf, imgsave);

    F5 = figure;

    ax1_F5 = subplot (1,6, [1,2]); imshow(f); title('Imagen original a h1')
    ax2_F5 = subplot (1,6, [3,4]); imshow(label2rgb(etiq)), caxis([0 1920]);
title('Imagen etiquetada h1')

    for u=1:length(propiedades_h1)
        box1=propiedades_h1(u).BoundingBox;
        point1=propiedades_h1(u).Centroid;
        rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
        'LineWidth',1.5)
        text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 10,
        'FontWeight', 'bold');
    end

    ax3_F5 = subplot (1,6, [5,6]); imshow(ind_3); colormap hot; title(tit_IND3)

    ax_v_F5 = [ax1_F5 ax2_F5 ax3_F5];

    linkaxes(ax_v_F5);

    pause(t);
end

% Índice 4
if(number_ind==4)
    figure;
    imagesc(ind_4)
    tit_IND4 = strcat('Índice 4 de zona = ', num2str(number));
    title(tit_IND4)
    xlabel('x (px)')
    ylabel('y (px)')

```

```

colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 4
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind4etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F5 = figure;

ax1_F5 = subplot (1,6, [1,2]); imshow(f); title('Imagen original a h1')

ax2_F5 = subplot (1,6, [3,4]); imshow(label2rgb(etiq)), caxis([0 1920]);
title('Imagen etiquetada h1')
for u=1:length(propiedades_h1)
box1=propiedades_h1(u).BoundingBox;
point1=propiedades_h1(u).Centroid;
rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
'LineWidth',1.5)
text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
end

ax3_F5 = subplot (1,6, [5,6]); imshow(ind_4); colormap winter; title(tit_IND4)

ax_v_F5 = [ax1_F5 ax2_F5 ax3_F5];

linkaxes(ax_v_F5);

pause(t);
end

% Índice 5

if(number_ind==5)

figure;
imagesc(ind_5)
tit_IND5 = strcat('Índice 5 de zona = ', num2str(number));
title(tit_IND5)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 5
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind5etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F5 = figure;

```

```

ax1_F5 = subplot (1,6, [1,2]); imshow(f); title('Imagen original a h1')
ax2_F5 = subplot (1,6, [3,4]); imshow(label2rgb(etiq)), caxis([0 1920]);
title('Imagen etiquetada h1')

for u=1:length(propiedades_h1)
box1=propiedades_h1(u).BoundingBox;
point1=propiedades_h1(u).Centroid;
rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
'LineWidth',1.5)
text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
end

ax3_F5 = subplot (1,6, [5,6]); imshow(ind_5); colormap winter; title(tit_IND5)

ax_v_F5 = [ax1_F5 ax2_F5 ax3_F5];

linkaxes(ax_v_F5);

pause(t);
end

% Índice 6

if(number_ind==6)
figure;
imagesc(ind_6)
tit_IND6 = strcat('Índice 6 de zona = ', num2str(number));
title(tit_IND6)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 6
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind6etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F5 = figure;

ax1_F5 = subplot (1,6, [1,2]); imshow(f); title('Imagen original a h1')
ax2_F5 = subplot (1,6, [3,4]); imshow(label2rgb(etiq)), caxis([0 1920]);
title('Imagen etiquetada h1')

for u=1:length(propiedades_h1)
box1=propiedades_h1(u).BoundingBox;
point1=propiedades_h1(u).Centroid;
rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
'LineWidth',1.5)
text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
end

ax3_F5 = subplot (1,6, [5,6]); imshow(ind_6); colormap winter; title(tit_IND6)

```

```

    ax_v_F5 = [ax1_F5 ax2_F5 ax3_F5];

    linkaxes(ax_v_F5);

    pause(t);

end

% Índice 7

if(number_ind==7)
    figure;
    imagesc(ind_7)
    tit_IND7 = strcat('Índice 7 de zona = ', num2str(number));
    title(tit_IND7)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap cool

    pause(t);

    % Guardar imagenes con índice 7
    imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind7etiq_' num2str(number)
    '.jpg']);
    saveas(gcf, imgsave);

    F5 = figure;

    ax1_F5 = subplot (1,6, [1,2]); imshow(f); title('Imagen original a h1')
    ax2_F5 = subplot (1,6, [3,4]); imshow(label2rgb(etiq)), caxis([0 1920]);
    title('Imagen etiquetada h1')

    for u=1:length(propiedades_h1)
        box1=propiedades_h1(u).BoundingBox;
        point1=propiedades_h1(u).Centroid;
        rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
            'LineWidth',1.5)
        text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 10,
            'FontWeight', 'bold');
    end

    ax3_F5 = subplot (1,6, [5,6]); imshow(ind_7); colormap cool; title(tit_IND7)

    ax_v_F5 = [ax1_F5 ax2_F5 ax3_F5];

    linkaxes(ax_v_F5);

    pause(t);

end

% Índice 8

if(number_ind==8)
    figure;
    imagesc(ind_8)
    tit_IND8 = strcat('Índice 8 de zona = ', num2str(number));

```

```

title(tit_IND8)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap cool

pause(t);

% Guardar imagenes con índice 8
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind8etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F5 = figure;

ax1_F5 = subplot (1,6, [1,2]); imshow(f); title('Imagen original a h1')
ax2_F5 = subplot (1,6, [3,4]); imshow(label2rgb(etiq)), caxis([0 1920]);
title('Imagen etiquetada h1')

for u=1:length(propiedades_h1)
box1=propiedades_h1(u).BoundingBox;
point1=propiedades_h1(u).Centroid;
rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
'LineWidth',1.5)
text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
end

ax3_F5 = subplot (1,6, [5,6]); imshow(ind_8); colormap cool; title(tit_IND8)

ax_v_F5 = [ax1_F5 ax2_F5 ax3_F5];

linkaxes(ax_v_F5);

pause(t);
end

% Índice 9

if(number_ind==9)
figure;
imagesc(ind_9)
tit_IND9 = strcat('Índice 9 de zona = ', num2str(number));
title(tit_IND9)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap cool

pause(t);

% Guardar imagenes con índice 9
imgsave=fullfile(outputDir_20I,['Imh1_' num2str(i) '_Ind9etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F5 = figure;

ax1_F5 = subplot (1,6, [1,2]); imshow(f); title('Imagen original a h1')

```



```

    ax2_F5 = subplot (1,6, [3,4]); imshow(label2rgb(etiq)), caxis([0 1920]);
title('Imagen etiquetada h1')

    for u=1:length(propiedades_h1)
    box1=propiedades_h1(u).BoundingBox;
    point1=propiedades_h1(u).Centroid;
    rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
        'LineWidth',1.5)
    text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
    end

    ax3_F5 = subplot (1,6, [5,6]); imshow(ind_9); colormap cool; title(tit_IND9)

    ax_v_F5 = [ax1_F5 ax2_F5 ax3_F5];

    linkaxes(ax_v_F5);

    pause(t);
end

    end

% % Recortamos regiones de interés
%
% for u=1:length(propiedades_h1)
%     box1=propiedades_h1(u).BoundingBox;
%     figure
%     imshow(fcontrast(box1(2):box1(2)+ box1(4), box1(1):box1(1)+box1(3),:));
% end
% pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.
%

% Guardo en Excel las propiedades h1 en pixeles
%writetable(struct2table(propiedades_h1),'propiedades_h1.xlsx');

% excsave=sprintf('props_img20m_%g.xlsx',i);
% writetable(struct2table(propiedades_h1), [outputExc_20P excsave]);

% Guardo en Excel las propiedades h1 en metros

% excsave=sprintf('propsmetros_img10m_%g.xlsx',i); %Si las guardo
% separadas
% writetable(p1, [outputExc_20 excsave]);

excsave=sprintf('props_img20m_%g.xlsx',i);
writetable(p1, [outputExc_20 excsave]);

% Tomamos el área máxima
[x pam]=max([propiedades_h1.Area]);

% "ROI" contiene solo la imagen más grande
roi_h1=ismember(etiq,pam);
ee=strel('disk',18,0);

if(flagDilatar == 1)                               % Dilatación aplicada
    roi_h1=imdilate(roi_h1, ee);
end

```

```

end

% Obtener los límites del área máxima
limites_h1=propiedades_h1(pam).BoundingBox;

% Mostramos la región de interés (ROI)

if(flagFiguras == 1)                                % Mostrará figura
figure;
imagesc(roi_h1)
colormap gray
title('Objeto de área máxima a h1');
xlabel('Píxeles')
ylabel('Píxeles')
end

%% Resultados finales h1

if(flagFiguras == 1)                                % Mostrará figura
% Si enfrente tan solo dos imágenes
F6= figure;
ax1_F6 = subplot (1,4, [1,2]); imshow(f), caxis([0 1920]); title(' Imagen original
h1')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(f);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

ax2_F6 = subplot (1,4,[3,4]); imshow(label2rgb(etiq)), caxis([0 1920]); title('Imagen
etiquetada h1')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(f);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

    % Muestro las regiones
for u=1:length(propiedades_h1)
    box1=propiedades_h1(u).BoundingBox;

```

```

    point1=propiedades_h1(u).Centroid;
    rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...
        'LineWidth',3)
    text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 14,
'FontWeight', 'bold');
end

ax_v_F6 = [ax1_F6 ax2_F6];

linkaxes(ax_v_F6);

pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.

% Si enfrento cuatro imagenes:
F7= figure;
ax1_F7 = subplot (2,2,1); imshow(f), caxis([0 1920]); title(' Imagen original h1')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(f);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

ax2_F7 = subplot (2,2,2); imshow(fcontrast), caxis([0 1920]); title(' Imagen mejorada
h1')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

ax3_F7 = subplot (2,2,3); imshow(erodeI1); title(' Imagen erosionada h1')

ax4_F7 = subplot (2,2,4); imshow(label2rgb(etiq)), caxis([0 1920]); title('Imagen
etiquetada h1')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(f);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

for u=1:length(propiedades_h1)
    box1=propiedades_h1(u).BoundingBox;
    point1=propiedades_h1(u).Centroid;
    rectangle('Position', [box1(1), box1(2), box1(3), box1(4)], 'EdgeColor','r',...

```

```

        'LineWidth',3)
    text(point1(1), point1(2), sprintf('%d', u), 'Color', 'k', 'FontSize', 14,
'FontWeight', 'bold');
end

ax_v_F7 = [ax1_F7 ax2_F7 ax3_F7 ax4_F7];

linkaxes(ax_v_F7);

pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.
end

%% Segmentación por color tras filtración ruido (elección usuario ejecución)
imR1= double(Im1c_R);
imG1= double(Im1c_G);
imB1= double(Im1c_B);

% CANAL ROJO

f_R= imR1>120 & imG1>10 & imG1<151 & imB1>10 & imB1<120;

Rmascara_1 = 1 - f_R;
iR_roja=double(erodeI1);           %Muestra las partes rojas
iR_verde=double(erodeI1).* Rmascara_1; %Multiplicacion por 0
iR_azul=double(erodeI1).* Rmascara_1; %Multiplicacion por 0

ifinR=cat(3, iR_roja, iR_verde,iR_azul);
% figure, imshow(ifinR), title('Segmentada Rojo h1')
%
% pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.

% CANAL VERDE
f_G= (imG1-imB1-imR1);           % Anulo componentes azules y rojos
im_binariaG= f_G > 0;

Gmascara_1 = 1 - im_binariaG;
iG_verde=double(erodeI1);           %Muestra las partes verdes
iG_roja=double(erodeI1).* Gmascara_1; %Multiplicacion por 0
iG_azul=double(erodeI1).* Gmascara_1; %Multiplicacion por 0

ifinG=cat(3, iG_roja, iG_verde,iG_azul);
%figure, imshow(ifinG), title('Segmentada verde h1')
pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.

% CANAL AZUL
f_B= (imB1-imG1-imR1);           %Anulo componentes verdes y rojos
im_binariaB= f_B > 0;

Bmascara_1 = 1 - im_binariaB;
iB_azul=double(erodeI1);           %Muestra las partes azules
iB_roja=double(erodeI1).* Bmascara_1; %Multiplicacion por 0
iB_verde=double(erodeI1).* Bmascara_1; %Multiplicacion por 0

ifinB=cat(3, iB_roja, iB_verde,iB_azul);
%figure, imshow(ifinB), title('Segmentada azul h1')
% pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.

% TIERRA (Anaranjado)
f_tierra= imR1 >80 & imG1>80 & imG1 <200 & imB1 >100;

```

```

Tmascara_1 = 1 - f_tierra;
iT_azul=double(erodeI1).* Tmascara_1;
iT_roja=double(erodeI1);
iT_verde=double(erodeI1);

ifinT=cat(3, iT_roja, iT_verde,iT_azul);
%figure, imshow(ifinT), title('Segmentada tierra h1')

%pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.

% AMARILLO
f_amar= imR1 >180 & imG1>150 & imG1 <254 & imB1 <130;

Amascara_1 = 1 - f_amar;
iA_azul=double(erodeI1).* Amascara_1;
iA_roja=double(erodeI1);
iA_verde=double(erodeI1);

ifinA=cat(3, iA_roja, iA_verde,iA_azul);
%figure, imshow(ifinA), title('Segmentada amarillo h1')

% pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.

% Decidir si se deja las figuras abiertas de la imagen procesada
seleccion=questdlg('¿Quiere cerrar el procesado realizado? Pasará a la siguiente
imagen', 'Opciones', 'Si', 'No', 'No');
switch seleccion
    case 'Si'
        close all;

    case 'No'

end

if(seleccion=='Si')           %Si el usuario elije 'Si', cierra el procesado y pasa a la
siguiente.
    continue
    i=i+1;
end

end

%% Imagen 2 en RGB

h2= 5;           % Altura, h2= 5 metros

FOV_h2 = round((h2*factor),2);           % Campo visión
en metros
escalaH2 = round((FOV_h2/im_ancho)*1000,2);           % Escala,
pixeles por milímetros
convH2 = round((escalaH2*div)/1000, 2);           % Conversión a
metros

for j = 1:length(Imagenes_5_metros)           %Recorre el número de imagenes en el
directorio
    imagenes_5=Imagenes_5_metros(j).name;           %Obtengo nombres imagenes

```

```

nombre_im2=[carp_2 '\']; %Recorre el directorio
g = imread(strcat(nombre_im2, imagenes_5)); %Leo imagenes

% Extraigo las dimensiones de la imagen

dim_imH2=size(g);
im_alto = dim_imH2(1); % Pixeles de alto
im_ancho = dim_imH2(2); % Pixeles de
ancho

div= im_ancho*0.125; % División imagen
figuras (pixeles)

if(flagEscala == 3)

    FOV_h2 = round((h2*factor),2); % Campo
visión en metros
    escalaH2 = round((FOV_h2/im_ancho)*100,2); % Escala,
pixeles por centímetro

    convH2 = round((escalaH2*div)/100, 2); % Conversión
a metros
end

figure; imshow(g), caxis([0 1920]), title('Imagen original a h2= 5 metros');
%Imagen original con ejes
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(g);

for row = 1 : 100 : rows
    yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
end

for col = 1 : 500 : columns
    xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
end

line([500,0],[2100,2100], 'Color', [0.6350 0.0780 0.1840], 'LineWidth', 3);
text (125,2150,[num2str(convH2), ' m'], 'Color', [0.6350 0.0780 0.1840],
'FontSize', 12, 'FontWeight', 'bold');
% text (0,950,'x', 'Color', [0.6350 0.0780 0.1840], 'FontSize', 18,
'FontWeight', 'bold');
% text (200,950,'x', 'Color', [0.6350 0.0780 0.1840], 'FontSize', 18,
'FontWeight', 'bold');

pause(t); % Dejo un tiempo prudencial de decisión de cierre de figura.

% Decidir si se continua con el procesado
seleccion=questdlg('¿Quiere detener el procesado y pasar a la siguiente imagen?',
'Opciones', 'Si', 'No', 'No');
switch seleccion
case 'Si'

```

```

        close(gcf);

        case 'No'

end

if(seleccion=='Si')           %Si el usuario cierra, no procesa esa imagen y pasa a la
siguiente.
    continue
    j=j+1;
end

[P,Q, canales] = size (g);      % P filas, Q columnas

% Tres matrices bidimensionales, correspondientes a planos R,G y B.
Im2_R = g(:,:,1);             %Plano rojo
Im2_G = g(:,:,2);             %Plano verde
Im2_B = g(:,:,3);             %Plano azul

if(flagFiguras == 1)           % Mostrará figura

F8= figure;
ax1_F8= subplot(2,3,[1,2,3]); imshow(g); title('Imagen original: h2')
ax2_F8= subplot(2,3,4); imshow(Im2_R); title('Canal rojo');
ax3_F8= subplot(2,3,5); imshow(Im2_G); title('Canal verde');
ax4_F8= subplot(2,3,6); imshow(Im2_B); title('Canal azul');

ax_v_F8 = [ax1_F8 ax2_F8 ax3_F8 ax4_F8];

linkaxes(ax_v_F8);

pause(t);
end

%% Mejora de las imágenes h2
%% Filtrado de ruido de la imagen h2

rfiltro2=medfilt2(Im2_R);      %Filtro mediana h2
gfiltro2=medfilt2(Im2_G);
bfiltro2=medfilt2(Im2_B);

iRGB_h2 = cat(3, rfiltro2, gfiltro2, bfiltro2);    % Unimos para formar la imagen RGB
h2

% Si se desea mostrar la imagen filtrada:
% F9= figure;
% ax1_F9= subplot (2,2, [1,2]); imshow(g); title('Imagen original h2')
% ax2_F9= subplot (2,2,[3,4]); imshow(iRGB_h2); title('Imagen filtrada h2')
% ax_v_F9 = [ax1_F9 ax2_F9];
%
% linkaxes(ax_v_F9);

% Guardar imagenes filtradas h2
imgsave=sprintf('filtrada_h2_%g.jpg',j);
imwrite(iRGB_h2, [outputDir_5 imgsave]);

% Pasamos la imagen filtrada RGB h2 a L*a*b para tratamiento

```

```

lab_h2=rgb2lab(iRGB_h2);

L_2=lab_h2(:,:,1)/100;
L_2= adapthisteq(L_2,'NumTiles',[8 8], 'ClipLimit',0.005);
lab_h2(:,:,1)=L_2*100;

% Vuelta a RGB
gcontrast= lab2rgb(lab_h2);

if(flagFiguras == 1) % Mostrará figura
figure;
imshowpair(iRGB_h2, gcontrast, 'montage')
title('Original - Mejora de contraste h2')

pause(t); % Dejo un tiempo prudencial de decisión de cierre de figura.
end

% Guardar imagenes mejoradas h2
imgsave=sprintf('Mcontraste_h2_%g.jpg',j);
imwrite(gcontrast, [outputDir_5 imgsave]);

%[C,D] = size (gcontrast); % C filas, D columnas
gcontrast8=uint8(round(gcontrast*255));
% Tres matrices bidimensionales, correspondientes a planos R,G y B.
Im2c_R = gcontrast8(:,:,1); %Plano rojo
Im2c_G = gcontrast8(:,:,2); %Plano verde
Im2c_B = gcontrast8(:,:,3); %Plano azul

iRGBc_h2= cat(3, Im2c_R, Im2c_G, Im2c_B); % Unimmos para formar la imagen RGB h2

% Imagen h2 en tonos de gris
% sum_h2 = double(Im2c_R) + double(Im2c_G) + double(Im2c_B);
g_gris = rgb2gray(iRGBc_h2); %Imagen escala de grises

%% Binarización de la imagen h2(+ Otsu)

modoManual = otsu;

if modoManual
% Umbral elegido a mano, analizando el histograma y obtención de la plantilla
binaria:
% prompt= 'Introduce el valor del umbral: '; % Introduciendo el valor por ventana
comandos
% umbral2= input(prompt);

umbralNorm = umbral2/255;
else
% Umbral elegido automáticamente mediante función de Matlab que usa el método de
Otsu:
umbralNorm = graythresh (g_gris); % Calcula un umbral global a partir de
im. escala grises
umbral2 = umbralNorm * 255;
end

PlantillaG = im2bw (g_gris,umbralNorm);

```



```

% Si se desea imprimir el valor del umbral en la imagen:
% line ([umbral2,umbral2],[0,400],'Color','r','LineWidth',2);    % Imprime en la
imagen el valor del umbral
% text (115,500,['U=',num2str(umbral2)]);

gBinarizada = 255*uint8(PlantillaG);
b = g_gris .* uint8(PlantillaG);

if(flagFiguras == 1)                                % Mostrará figura
figure; imshow([g_gris,gBinarizada,b]);

pause(t);      % Dejo un tiempo prudencial de decisión de cierre de figura.
end

umbral_2 = CalculaUmbralOtsu (imhist(g_gris));

[umbral_2,tabVarIntraclase] = CalculaUmbralOtsu (imhist(g_gris));
gBinarizada = 255*uint8(g_gris<umbral2); %0: fondo, 255: objeto

if(flagFiguras == 1)                                % Mostrará figura
figure; imshow(gBinarizada), title('Imagen binarizada h2');

% Guardar imagenes binarizadas h2
imgsave=sprintf('binarizada_h2_%g.jpg',j);
imwrite(gBinarizada, [outputDir_5 imgsave]);

pause(t);
end
%% Operaciones Morfológicas h2

% Relleno de regiones y huecos de la imagen (aparentemente no aporta nada)

rellenoI2 = imfill(gBinarizada,'holes');

% figure, imshow(rellenoI2), title('Huecos rellenos h2');    % Si deseo sola

% Guardar imagenes tras relleno de huecos
imgsave=sprintf('relleno_h2_%g.jpg',j);
imwrite(rellenoI2, [outputDir_5 imgsave]);

% Estructuras ya creadas antes
se1 = strel('square',3);           % Cuadrado 3x3
se2 = strel('line',5,45);         % Línea de longitud 5 y ángulo de 45°
se3 = strel('disk',2);           % Disco de radio 2
se4 = strel('ball',15,5);        % Bola de radio 15 y alto 5
se5 = strel('diamond',1);        % Diamante de tamaño 1
se6 = strel('rectangle',[7,4]);   % Rectangulo de 7x4

% Erosión (Puede aportar)

erodeI2 = imerode(rellenoI2,se5);
erodeI2=bwareaopen(erodeI2, 11000);    % Tras observar los valores,
elijo ese.
%figure, imshow(erodeI2), title('Erosión h2'); %Si deseo la img sola

if(flagFiguras == 1)                                % Mostrará figura
F10= figure;
ax1_F10= subplot (2,2, [1,2]); imshow(rellenoI2); title('Huecos rellenos h2')

```

```

ax2_F10=subplot (2,2,[3,4]); imshow(erodeI2); title('Erosión h2')

ax_v_F10 = [ax1_F10 ax2_F10];

    linkaxes(ax_v_F10);                % Sincronización imagenes de la figura

pause(t);          % Dejo un tiempo prudencial de decisión de cierre de figura.
end
% Guardar imagenes erosionadas
imgsave=sprintf('erosion_h2_%g.jpg',j);
imwrite(erodeI2, [outputDir_5 imgsave]);

% Dibujo el contorno de la imagen h2

if(flagFiguras == 1)                    % Mostrará figura
figure, imcontour(erodeI2,3), title('Contorno h2');
xlabel('Píxeles')
ylabel('Píxeles')

pause(t);          % Dejo un tiempo prudencial de decisión de cierre de figura.
end

% Apertura: Erosión+Dilatación

% OpenI2 = imopen(g_gris,se5);
% figure, imshow(OpenI2), title('Apertura h2');

% Cierre: Dilatación+Erosión (No aporta casi nada)

% closeI2 = imclose(g_gris,se5);
% figure, imshow(closeI2), title('Cierre h2');

% Dilatación (No aporta casi nada)

% dilateI2 = imdilate(g_gris,se3);
% figure, imshow(dilateI2), title('Dilatación h2')

%% Transformada de Hough h2 (A usar si se cree conveniente)
%% Preprocesado para acentuar bordes más importantes.
% Bordes_2 = edge(g_gris,'canny');
% Bordes_2 = bwareaopen(Bordes_2, 40);
%% T. de Hough y búsqueda de máximos.
% [H2,T2,R2] = hough(Bordes_2);
% peaks = houghpeaks(H2,110);
%% Buscamos las líneas y las mostramos con valores por defecto
% lineas_2 = houghlines(Bordes_2,T2,R2,peaks);
% figure, imshow(g_gris), hold on
% max_len = 0;
% for m = 1:length(lineas_2)
% xy = [lineas_2(m).point1; lineas_2(m).point2];
% plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
%% Dibujamos inicios y finales de las líneas
% plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
% plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','green');
% end
%% Cambiamos los valores de FillGap y MinLength para mostrar líneas más largas
% lineas_2 = houghlines(Bordes_2,T2,R2,peaks,'FillGap',120,'MinLength',15);
% figure, imshow(g_gris), hold on
% max_len = 0;
% for m = 1:length(lineas_2)

```

```

% xy = [lineas_2(m).point1; lineas_2(m).point2];
% plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','red');
% % Dibujamos los principios y finales de las líneas.
% plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
% plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','green');
% end

%% Etiquetado h2
figure, imshow(g);
etiq_2=bwlabel(erodeI2);
imshow(label2rgb(etiq_2)), caxis([0 1920]), title('Imagen etiquetada h2');
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(g);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

CC=bwconncomp(etiq_2);          %Busca y cuenta los componentes conectados
disp(CC)

propiedades_h2=regionprops(etiq_2, 'basic');      % Todas las propiedades 'all' sino
basicas. (CC en vez de etiq_2 vale)

% Identificamos las regiones encontradas

for w=1:length(propiedades_h2)
    box2=propiedades_h2(w).BoundingBox;
    point2=propiedades_h2(w).Centroid;
    rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
        'LineWidth',3)
    text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 14,
        'FontWeight', 'bold');
end

pause(t);          % Dejo un tiempo prudencial de decisión de cierre de figura.

% Guardar imagenes etiquetadas h2

imgsave=fullfile(outputDir_5P,['etiquetada_h2_' num2str(j) '.jpg']);
saveas(gcf, imgsave);

% Muestro el área de cada región interés en metros cuadrados

p2 = struct2table(propiedades_h2);
p2.Area_metros2 = (p2.Area/escalaH2^2)/(10^6);          % metros cuadrados

netiq2=length(propiedades_h2);
num_etiq2= (1:1:netiq2)';
p2.Num_etiq =num_etiq2;

```

```

% Unir tablas propiedades y area en metros cuadrados
propiedades_2= join(struct2table(propiedades_h2), p2);

% Etiquetas sobre imagen original

if(flagFiguras == 1)                                     % Mostrará figura
figure;
imshow(g), caxis([0 1920]); title(' Imagen original etiquetada h2')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(g);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

    % Mostrar las regiones
for w=1:length(propiedades_h2)
    box2=propiedades_h2(w).BoundingBox;
    point2=propiedades_h2(w).Centroid;
    rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
        'LineWidth',3)
    text(point2(1), point2(2), sprintf('%d', w), 'Color', 'w', 'FontSize', 14,
        'FontWeight', 'bold');
end

pause(t);          % Dejo un tiempo prudencial de decisión de cierre de figura.
end

%% Cálculo de índices de la etiqueta seleccionada

prompt={'¿De qué etiqueta desea ver los índices?'};
dlg_title='Ingrese valor correspondiente';
width = 60;
height = 1;
num_lines=[height, width];
def_out= {'1'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
number= str2num(inputH{1});

[r, c] = find(etiq_2==number);                          % [filas, columnas], buscar
coordenadas de la etiqueta
rc2 = [r c];
%hold on; plot(1197,154,'y*')                            %comprobar
iRGBc_h2 = double(iRGBc_h2);

ind_2 = zeros(im_alto,im_ancho);
ind_3 = zeros(im_alto,im_ancho);
ind_4 = zeros(im_alto,im_ancho);
ind_5 = zeros(im_alto,im_ancho);
ind_6 = zeros(im_alto,im_ancho);

```

```

ind_7 = zeros(im_alto,im_ancho);
ind_8 = zeros(im_alto,im_ancho);
ind_9 = zeros(im_alto,im_ancho);

% Comprobamos que la imagen binaria corresponde con la zona elegida

    im_bin2 = [];

    im_bin2 = zeros(im_alto,im_ancho);

    % ponemos a 1 los puntos de la zona

    N_fil = length(r);

    for ind_q = 1:N_fil

        im_bin2(r(ind_q),c(ind_q)) = 1;

    end

    figure;
    imagesc(im_bin2)
    title('Imagen Binaria de zona elegida')
    xlabel('x (px)')
    ylabel('y (px)')

    pause(t);

% Cálculo de índices

for ind = 1:N_fil                                % Pongo a 1 los puntos de la zona

    R = iRGBc_h2(r(ind),c(ind),1);
    G = iRGBc_h2(r(ind),c(ind),2);
    B = iRGBc_h2(r(ind),c(ind),3);

    % Cálculo índice 1

    ind1_calc = (B - R)./ R;

    ind_1(r(ind),c(ind)) = ind1_calc;

    % Cálculo índice 2

    ind2_calc = (G - R)./ R;

    ind_2(r(ind),c(ind)) = ind2_calc;

    % Cálculo índice 3

    ind3_calc = (B - G)./ R;

    ind_3(r(ind),c(ind)) = ind3_calc;

    % Cálculo índice 4

    ind4_calc = (B - G)./ G;

```

```

ind_4(r(ind),c(ind)) = ind4_calc;

% Cálculo índice 5
ind5_calc = (R - G)./ G;
ind_5(r(ind),c(ind)) = ind5_calc;

% Cálculo índice 6
ind6_calc = (B - R)./ G;
ind_6(r(ind),c(ind)) = ind6_calc;

% Cálculo índice 7
ind7_calc = (R - B)./ B;
ind_7(r(ind),c(ind)) = ind7_calc;

% Cálculo índice 8
ind8_calc = (G - B)./ B;
ind_8(r(ind),c(ind)) = ind8_calc;

% Cálculo índice 9
ind9_calc = (R - G)./ B;
ind_9(r(ind),c(ind)) = ind9_calc;

end

% Guardo los valores de los índices en una tabla
nums_indices={'Indice 1'; 'Indice 2'; 'Indice 3'; 'Indice 4'; 'Indice 5'; 'Indice 6';
'Indice 7'; 'Indice 8'; 'Indice 9'};
indices_cal=[ind1_calc; ind2_calc; ind3_calc; ind4_calc; ind5_calc; ind6_calc;
ind7_calc; ind8_calc; ind9_calc];
Indices_Value= table(nums_indices, indices_cal);

% Guardo en excel los valores de los índices
excsave=sprintf('indices_img5_%g_etiq%g.xlsx',j, number);
writetable(Indices_Value, [outputExcInd_5 excsave]);

% Decidir si se muestran los resultados por zonas o por índices
Q_indice=questdlg('¿Como desea ver los resultados tras cálculo de índices?',
'Opciones', 'Por Zonas', 'Por índices', 'Por índices');

switch Q_indice
    case 'Por Zonas'
        flagIndice = 1;
    case 'Por índices'

```

```
flagIndice = 2;

end

if(flagIndice == 1)

    % Índice 1

    % Mostrando el colorbar
    figure;
    imagesc(ind_1)
    tit_IND1 = strcat('Índice 1 de zona = ', num2str(number));
    title(tit_IND1)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap hot

    pause(t);

% Guardar imagenes con índice 1 con colorbar
imgsave=fullfile(outputDir_5Ic,['Imh2_' num2str(j) '_Ind1letiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_1)
title('Índice 1')
xlabel('x (px)')
ylabel('y (px)')
colormap hot

pause(0);

% Guardar imagenes con índice 1 sin colorbar
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind1letiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 2

% Mostrando el colorbar

figure;
imagesc(ind_2)
tit_IND2 = strcat('Índice 2 de zona = ', num2str(number));
title(tit_IND2)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap hot

pause(t);

% Guardar imagenes con índice 2 con colorbar
```

```

    imgsave=fullfile(outputDir_5Ic,['Imh2_' num2str(j) '_Ind2etiq_' num2str(number)
'.jpg']);
    saveas(gcf, imgsave);

    % Sin colorbar
    figure;
    imagesc(ind_2)
    title('Índice 2')
    xlabel('x (px)')
    ylabel('y (px)')
    colormap hot

    pause(0);

    % Guardar imagenes con índice 2
    imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind2etiq_' num2str(number)
'.jpg']);
    saveas(gcf, imgsave);
    close(gcf);

% Índice 3

    % Mostrando el colorbar
    figure;
    imagesc(ind_3)
    tit_IND3 = strcat('Índice 3 de zona = ', num2str(number));
    title(tit_IND3)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap hot

    pause(t);

    % Guardar imagenes con índice 3 con colorbar
    imgsave=fullfile(outputDir_5Ic,['Imh2_' num2str(j) '_Ind3etiq_' num2str(number)
'.jpg']);
    saveas(gcf, imgsave);

    % Sin colorbar

    figure;
    imagesc(ind_3)
    title('Índice 3')
    xlabel('x (px)')
    ylabel('y (px)')
    colormap hot

    pause(0);

    % Guardar imagenes con índice 3 sin colorbar
    imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind3etiq_' num2str(number)
'.jpg']);
    saveas(gcf, imgsave);
    close(gcf);

% Índice 4

```



```
% Mostrando el colorbar

figure;
imagesc(ind_4)
tit_IND4 = strcat('Índice 4 de zona = ', num2str(number));
title(tit_IND4)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 4 con colorbar
imgsave=fullfile(outputDir_5Ic,['Imh2_' num2str(j) '_Ind4etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_4)
title('Índice 4')
xlabel('x (px)')
ylabel('y (px)')
colormap winter

pause(0);

% Guardar imagenes con índice 4 sin colorbar
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind4etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 5

% Mostrando el colorbar
figure;
imagesc(ind_5)
tit_IND5 = strcat('Índice 5 de zona = ', num2str(number));
title(tit_IND5)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 5 con colorbar
imgsave=fullfile(outputDir_5Ic,['Imh2_' num2str(j) '_Ind5etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_5)
title('Índice 5')
xlabel('x (px)')
```

```
ylabel('y (px)')
colormap winter

pause(0);

% Guardar imagenes con índice 5 sin colorbar
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind5etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 6

% Mostrando el colorbar
figure;
imagesc(ind_6)
tit_IND6 = strcat('Índice 6 de zona = ', num2str(number));
title(tit_IND6)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 6 con colorbar
imgsave=fullfile(outputDir_5Ic,['Imh2_' num2str(j) '_Ind6etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_6)
title('Índice 6')
xlabel('x (px)')
ylabel('y (px)')
colormap winter

pause(0);

% Guardar imagenes con índice 6 sin colorbar
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind6etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 7

% Mostrando el colorbar

figure;
imagesc(ind_7)
tit_IND7 = strcat('Índice 7 de zona = ', num2str(number));
title(tit_IND7)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap cool
```

```
pause(t);

% Guardar imagenes con índice 7 con colorbar
imgsave=fullfile(outputDir_5Ic,['Imh2_' num2str(j) '_Ind7etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_7)
title('Índice 7')
xlabel('x (px)')
ylabel('y (px)')
colormap cool

pause(0);

% Guardar imagenes con índice 7 sin colorbar
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind7etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

% Índice 8

% Mostrando el colorbar

figure;
imagesc(ind_8)
tit_IND8 = strcat('Índice 8 de zona = ', num2str(number));
title(tit_IND8)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap cool

pause(t);

% Guardar imagenes con índice 8 con colorbar
imgsave=fullfile(outputDir_5Ic,['Imh2_' num2str(j) '_Ind8etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_8)
title('Índice 8')
xlabel('x (px)')
ylabel('y (px)')
colormap cool

pause(0);

% Guardar imagenes con índice 8 sin colorbar
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind8etiq_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);
```

```

% Índice 9

% Mostrando el colorbar

figure;
imagesc(ind_9)
tit_IND9 = strcat('Índice 9 de zona = ', num2str(number));
title(tit_IND9)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap cool

pause(t);

% Guardar imagenes con índice 9 con colorbar
imgsave=fullfile(outputDir_5Ic,['Imh2_' num2str(j) '_Ind9eti_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

% Sin colorbar
figure;
imagesc(ind_9)
title('Índice 9')
xlabel('x (px)')
ylabel('y (px)')
colormap cool

pause(0);

% Guardar imagenes con índice 9 sin colorbar
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind9eti_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);
close(gcf);

%% Muestro la imagen Original y la etiquetada

F11= figure;
ax1_F11 = subplot (1,4, [1,2]); imshow(g), caxis([0 1920]); title(' Imagen original
h2')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(g);

for row = 1 : 100 : rows
    yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
end

for col = 1 : 500 : columns
    xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
end

ax2_F11 =subplot (1,4,[3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
title('Imagen etiquetada h2')
xlabel('Píxeles')

```

```

ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(g);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

for w=1:length(propiedades_h2)
    box2=propiedades_h2(w).BoundingBox;
    point2=propiedades_h2(w).Centroid;
    rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
        'LineWidth',3)
    text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 14,
'FontWeight', 'bold');
end

ax_v_F11 = [ax1_F11 ax2_F11];

    linkaxes(ax_v_F11);
pause(t);

    %% Montaje con todos los índices

carp_2I = outputDir_5I;

Cim_5mI = fullfile(carp_2I, '*.jpg'); %guardar en formato jpg

im_5mI=dir(Cim_5mI);

cd(carp_2I)

imagenes_5I= string({im_5mI.name});

figure;
montage(imagenes_5I, 'Size',[3 3]);
tit_5m = strcat('Índices de la zona = ', num2str(number));
title(tit_5m)

pause(t);

cd(carp_0);

    end

if(flagIndice == 2)

    select_indice = msgbox(["Índice 1: BR-R"; "Índice 2: GR-R"; "Índice 3: BG-R"; ...
        "Índice 4: BG-G"; "Índice 5: RG-G"; "Índice 6: BR-G"; "Índice 7: RB-B"; ...
        "Índice 8: GB-B"; "Índice 9: RG-B"], "Info índices","help");
    pause(t);

```

```

prompt={'Número del índice que desea ver (1-9):'};
dlg_title='Ingrese valor correspondiente';
width = 60;
height = 1;
num_lines=[height, width];
def_out= {'1'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
number_ind= str2num(inputH{1});

% Índice 1

if(number_ind==1)
    figure;
    imagesc(ind_1)
    tit_IND1 = strcat('Índice 1 de zona = ', num2str(number));
    title(tit_IND1)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap hot

    pause(t);

% Guardar imagenes con índice 1
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind1eti_q_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F12 = figure;

ax1_F12 = subplot (1,6, [1,2]); imshow(g); title('Imagen original a h2')
ax2_F12 = subplot (1,6, [3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
title('Imagen etiquetada h2')

for w=1:length(propiedades_h2)
    box2=propiedades_h2(w).BoundingBox;
    point2=propiedades_h2(w).Centroid;
    rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
        'LineWidth',1.5)
    text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 10,
        'FontWeight', 'bold');
end

ax3_F12 = subplot (1,6, [5,6]); imshow(ind_1); colormap hot; title(tit_IND1)

ax_v_F12 = [ax1_F12 ax2_F12 ax3_F12];

linkaxes(ax_v_F12);

pause(t);
end

% Índice 2

if(number_ind==2)
    figure;
    imagesc(ind_2)

```

```

tit_IND2 = strcat('Índice 2 de zona = ', num2str(number));
title(tit_IND2)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap hot

pause(t);

% Guardar imagenes con índice 2
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(w) '_Ind2eti_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F12 = figure;

ax1_F12 = subplot (1,6, [1,2]); imshow(g); title('Imagen original a h2')
ax2_F12 = subplot (1,6, [3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
title('Imagen etiquetada h2')

for w=1:length(propiedades_h2)
box2=propiedades_h2(w).BoundingBox;
point2=propiedades_h2(w).Centroid;
rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
'LineWidth',1.5)
text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
end

ax3_F12 = subplot (1,6, [5,6]); imshow(ind_2); colormap hot; title(tit_IND2)

ax_v_F12 = [ax1_F12 ax2_F12 ax3_F12];

linkaxes(ax_v_F12);

pause(t);
end

% Índice 3
if(number_ind==3)
figure;
imagesc(ind_3)
tit_IND3 = strcat('Índice 3 de zona = ', num2str(number));
title(tit_IND3)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap hot

pause(t);

% Guardar imagenes con índice 3
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind3eti_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

```

```

F12 = figure;

ax1_F12 = subplot (1,6, [1,2]); imshow(g); title('Imagen original a h2')
ax2_F12 = subplot (1,6, [3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
title('Imagen etiquetada h2')

for w=1:length(propiedades_h2)
box2=propiedades_h2(w).BoundingBox;
point2=propiedades_h2(w).Centroid;
rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
'LineWidth',1.5)
text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
end

ax3_F12 = subplot (1,6, [5,6]); imshow(ind_3); colormap hot; title(tit_IND3)

ax_v_F12 = [ax1_F12 ax2_F12 ax3_F12];

linkaxes(ax_v_F12);

pause(t);
end

% Índice 4
if(number_ind==4)
figure;
imagesc(ind_4)
tit_IND4 = strcat('Índice 4 de zona = ', num2str(number));
title(tit_IND4)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 4
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind4eti_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F12 = figure;

ax1_F12 = subplot (1,6, [1,2]); imshow(g); title('Imagen original a h2')

ax2_F12 = subplot (1,6, [3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
title('Imagen etiquetada h2')
for w=1:length(propiedades_h2)
box2=propiedades_h2(w).BoundingBox;
point2=propiedades_h2(w).Centroid;
rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
'LineWidth',1.5)
text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
end

```



```

ax3_F12 = subplot (1,6, [5,6]); imshow(ind_4); colormap winter; title(tit_IND4)

ax_v_F12 = [ax1_F12 ax2_F12 ax3_F12];

linkaxes(ax_v_F12);

pause(t);
end

% Índice 5

if(number_ind==5)

    figure;
    imagesc(ind_5)
    tit_IND5 = strcat('Índice 5 de zona = ', num2str(number));
    title(tit_IND5)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap winter

    pause(t);

    % Guardar imagenes con índice 5
    imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind5eti_q_' num2str(number)
'.jpg']);
    saveas(gcf, imgsave);

    F12 = figure;

    ax1_F12 = subplot (1,6, [1,2]); imshow(g); title('Imagen original a h2')
    ax2_F12 = subplot (1,6, [3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
title('Imagen etiquetada h2')

    for w=1:length(propiedades_h2)
    box2=propiedades_h2(w).BoundingBox;
    point2=propiedades_h2(w).Centroid;
    rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
        'LineWidth',1.5)
    text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
    end

    ax3_F12 = subplot (1,6, [5,6]); imshow(ind_5); colormap winter; title(tit_IND5)

    ax_v_F12 = [ax1_F12 ax2_F12 ax3_F12];

    linkaxes(ax_v_F12);

    pause(t);
end

% Índice 6

if(number_ind==6)
    figure;

```

```

imagesc(ind_6)
tit_IND6 = strcat('Índice 6 de zona = ', num2str(number));
title(tit_IND6)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap winter

pause(t);

% Guardar imagenes con índice 6
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind6eti_' num2str(number)
'.jpg']);
saveas(gcf, imgsave);

F12 = figure;

ax1_F12 = subplot (1,6, [1,2]); imshow(g); title('Imagen original a h2')
ax2_F12 = subplot (1,6, [3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
title('Imagen etiquetada h2')

for w=1:length(propiedades_h2)
box2=propiedades_h2(w).BoundingBox;
point2=propiedades_h2(w).Centroid;
rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
'LineWidth',1.5)
text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 10,
'FontWeight', 'bold');
end

ax3_F12 = subplot (1,6, [5,6]); imshow(ind_6); colormap winter; title(tit_IND6)

ax_v_F12 = [ax1_F12 ax2_F12 ax3_F12];

linkaxes(ax_v_F12);

pause(t);

end

% Índice 7

if(number_ind==7)
figure;
imagesc(ind_7)
tit_IND7 = strcat('Índice 7 de zona = ', num2str(number));
title(tit_IND7)
xlabel('x (px)')
ylabel('y (px)')
colorbar
colormap cool

pause(t);

% Guardar imagenes con índice 7
imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind7eti_' num2str(number)
'.jpg']);

```

```

saveas(gcf, imgsave);

F12 = figure;

ax1_F12 = subplot (1,6, [1,2]); imshow(g); title('Imagen original a h2')
ax2_F12 = subplot (1,6, [3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
title('Imagen etiquetada h2')

for w=1:length(propiedades_h2)
    box2=propiedades_h2(w).BoundingBox;
    point2=propiedades_h2(w).Centroid;
    rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
        'LineWidth',1.5)
    text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 10,
        'FontWeight', 'bold');
end

ax3_F12 = subplot (1,6, [5,6]); imshow(ind_7); colormap cool; title(tit_IND7)

ax_v_F12 = [ax1_F12 ax2_F12 ax3_F12];

linkaxes(ax_v_F12);

pause(t);
end

% Índice 8

if(number_ind==8)
    figure;
    imagesc(ind_8)
    tit_IND8 = strcat('Índice 8 de zona = ', num2str(number));
    title(tit_IND8)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap cool

    pause(t);

    % Guardar imagenes con índice 8
    imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind8etiq_' num2str(number)
        '.jpg']);
    saveas(gcf, imgsave);

    F12 = figure;

    ax1_F12 = subplot (1,6, [1,2]); imshow(g); title('Imagen original a h2')
    ax2_F12 = subplot (1,6, [3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
    title('Imagen etiquetada h2')

    for w=1:length(propiedades_h2)
        box2=propiedades_h2(w).BoundingBox;
        point2=propiedades_h2(w).Centroid;
        rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
            'LineWidth',1.5)
        text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 10,
            'FontWeight', 'bold');
        end

```

```

ax3_F12 = subplot (1,6, [5,6]); imshow(ind_8); colormap cool; title(tit_IND8)

ax_v_F12 = [ax1_F12 ax2_F12 ax3_F12];

linkaxes(ax_v_F12);

pause(t);
end

% Índice 9

if(number_ind==9)
    figure;
    imagesc(ind_9)
    tit_IND9 = strcat('Índice 9 de zona = ', num2str(number));
    title(tit_IND9)
    xlabel('x (px)')
    ylabel('y (px)')
    colorbar
    colormap cool

    pause(t);

    % Guardar imagenes con índice 9
    imgsave=fullfile(outputDir_5I,['Imh2_' num2str(j) '_Ind9etiq_' num2str(number)
    '.jpg']);
    saveas(gcf, imgsave);

    F12 = figure;

    ax1_F12 = subplot (1,6, [1,2]); imshow(g); title('Imagen original a h2')
    ax2_F12 = subplot (1,6, [3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
    title('Imagen etiquetada h2')

    for w=1:length(propiedades_h2)
        box2=propiedades_h2(w).BoundingBox;
        point2=propiedades_h2(w).Centroid;
        rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
            'LineWidth',1.5)
        text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 10,
            'FontWeight', 'bold');
    end

    ax3_F12 = subplot (1,6, [5,6]); imshow(ind_9); colormap cool; title(tit_IND9)

    ax_v_F12 = [ax1_F12 ax2_F12 ax3_F12];

    linkaxes(ax_v_F12);

    pause(t);
end

end

```

```

% % Recortamos regiones de interés
%
% for w=1:length(propiedades_h2)
%     box2=propiedades_h2(w).BoundingBox;
%     figure
%     imshow(gcontrast(box2(2):box2(2)+ box2(4), box2(1):box2(1)+box2(3),:));
% end
% pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.
%

% Guardo en Excel las propiedades h2 en pixeles

% excsave2=sprintf('props_img5m_%g.xlsx',j);           % si se hace separado
% writetable(struct2table(propiedades_h2), [outputExc_5 excsave2]);

% Guardo en Excel las propiedades h2 en metros

% excsave=sprintf('propsmetros_img5m_%g.xlsx',j); %Si las guardo separadas
% writetable(p2, [outputExc_5 excsave]);

excsave=sprintf('props_img5m_%g.xlsx',j);
writetable(propiedades_2, [outputExc_5 excsave]);

% Tomamos el área máxima
[z pam2]=max([propiedades_h2.Area]);

% "ROI" contiene solo la imagen más grande
roi_h2=ismember(etiq_2,pam2);
ee2=strel('disk',18,0);

if(flagDilatar == 1)                               % Dilatación aplicada
    roi_h2=imdilate(roi_h2, ee2);
end

% Obtener los límites del área máxima
limites_h2=propiedades_h2(pam2).BoundingBox;

% Mostramos la región de interés (ROI)

if(flagFiguras == 1)                               % Mostrará figura
figure;
imagesc(roi_h2)
colormap gray
title('Objeto de área máxima a h2');
xlabel('Píxeles')
ylabel('Píxeles')
end

%% Resultados finales h2

if(flagFiguras == 1)                               % Mostrará figura

% Si enfrente tan solo dos imágenes

F13= figure;
ax1_F13 = subplot (1,4, [1,2]); imshow(g), caxis([0 1920]); title(' Imagen original
h2')

```

```

xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(g);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

ax2_F13 =subplot (1,4,[3,4]); imshow(label2rgb(etiq_2)), caxis([0 1920]);
title('Imagen etiquetada h2')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(g);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

    for col = 1 : 500 : columns
        xline(col, '-.', 'Color', 'k', 'LineWidth', grosor);
    end

% Muestro las regiones
for w=1:length(propiedades_h2)
    box2=propiedades_h2(w).BoundingBox;
    point2=propiedades_h2(w).Centroid;
    rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
        'LineWidth',3)
    text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 14,
'FontWeight', 'bold');
end

ax_v_F13 = [ax1_F13 ax2_F13];

linkaxes(ax_v_F13);

pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.

% Si enfrente cuatro imagenes:
F14=figure;
ax1_F14 = subplot (2,2,1); imshow(g), caxis([0 1920]); title('Imagen original h2')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(g);

    for row = 1 : 100 : rows
        yline(row, '-.', 'Color', 'w', 'LineWidth', grosor);
    end

```

```

end

for col = 1 : 500 : columns
    xline(col, '-.', 'Color', 'w', 'LineWidth', grosor);
end

ax2_F14 = subplot (2,2,2); imshow(gcontrast), caxis([0 1920]); title('Imagen mejorada
h2')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

ax3_F14 = subplot (2,2,3); imshow(erodeI2); title(' Imagen erosionada h2')

ax4_F14 = subplot (2,2,4); imshow(label2rgb(etiq_2)), caxis([0 1920]); title('Imagen
etiquetada h2')
xlabel('Píxeles')
ylabel('Píxeles')
axis on

% Mostrar cuadrícula que nos servirá de guía
[rows, columns, numberOfColorChannels] = size(g);

for row = 1 : 100 : rows
    yline(row, '-.', 'Color', 'k', 'LineWidth', grosor);
end

for col = 1 : 500 : columns
    xline(col, '-.', 'Color', 'k', 'LineWidth', grosor);
end

% Muestro las regiones
for w=1:length(propiedades_h2)
    box2=propiedades_h2(w).BoundingBox;
    point2=propiedades_h2(w).Centroid;
    rectangle('Position', [box2(1), box2(2), box2(3), box2(4)], 'EdgeColor','r',...
        'LineWidth',3)
    text(point2(1), point2(2), sprintf('%d', w), 'Color', 'k', 'FontSize', 14,
'FontWeight', 'bold');
end

ax_v_F14 = [ax1_F14 ax2_F14 ax3_F14 ax4_F14];

linkaxes(ax_v_F14);

pause(t);          % Dejo un tiempo prudencial de decisión de cierre de figura.
end

%% Segmentación por color tras filtración ruido (elección usuario ejecución)
imR2= double(Im2c_R);
imG2= double(Im2c_G);
imB2= double(Im2c_B);

% CANAL ROJO

g_R= imR2>90 & imG2>8 & imG2<55 & imB2>10 & imB2<60;

Rmascara_2 = 1 - g_R;
iR_red=double(erodeI2);          %Muestra las partes rojas
iR_green=double(erodeI2).* Rmascara_2; %Multiplicacion por 0

```

```

iR_blue=double(erodeI2).* Rmascara_2;           %Multiplicacion por 0

ifinalR=cat(3, iR_red, iR_green,iR_blue);
%figure, imshow(ifinalR), title('Segmentada rojo h2')

% pause(t);           % Dejo un tiempo prudencial de decisi3n de cierre de figura.
%

% CANAL VERDE
g_G= (imG2-imB2-imR2);           %Anulo componentes azules y rojos
im_binG= g_G > 0;

Gmascara_2 = 1 - im_binG;
iG_green=double(erodeI2);           %Muestra las partes verdes
iG_red=double(erodeI2).* Gmascara_2;           %Multiplicacion por 0
iG_blue=double(erodeI2).* Gmascara_2;           %Multiplicacion por 0

ifinalG=cat(3, iG_red, iG_green,iG_blue);
%figure, imshow(ifinalG), title('Segmentada verde h2')

% pause(t);           % Dejo un tiempo prudencial de decisi3n de cierre de figura.
%

% CANAL AZUL
g_B= (imB2-imG2-imR2);           %Anulo componentes verdes y rojos
im_binB= g_B > 0;

Bmascara_2 = 1 - im_binB;
iB_blue=double(erodeI2);           %Muestra las partes azules
iB_red=double(erodeI2).* Bmascara_2;           %Multiplicacion por 0
iB_green=double(erodeI2).* Bmascara_2;           %Multiplicacion por 0

ifinalB=cat(3, iB_red, iB_green,iB_blue);
%figure, imshow(ifinalB), title('Segmentada azul h2')

% pause(t);           % Dejo un tiempo prudencial de decisi3n de cierre de figura.
%

% TIERRA (Anaranjado)
g_tierra= imR2 >80 & imG2>80 & imG2 <200 & imB2 >100;

Tmascara_2 = 1 - g_tierra;
iT_blue=double(erodeI2).* Tmascara_2;
iT_red=double(erodeI2);
iT_green=double(erodeI2);

ifinalT=cat(3, iT_red, iT_green, iT_blue);
%figure, imshow(ifinalT), title('Segmentada tierra h2')

%pause(t);           % Dejo un tiempo prudencial de decisi3n de cierre de figura.

% AMARILLO
g_amar= imR2 >180 & imG2>150 & imG2 <254 & imB2 <130;

Amascara_2 = 1 - g_amar;
iA_blue=double(erodeI2).* Amascara_2;
iA_red=double(erodeI2);
iA_green=double(erodeI2);

```



```
ifinalA=cat(3, iA_red, iA_green,iA_blue);
%figure, imshow(ifinalA), title('Segmentada amarillo h2')

% pause(t);           % Dejo un tiempo prudencial de decisión de cierre de figura.

% Decidir si se deja las figuras abiertas de la imagen procesada
seleccion=questdlg('¿Quiere cerrar el procesado realizado? Pasará a la siguiente
imagen', 'Opciones', 'Si', 'No', 'No');
switch seleccion
    case 'Si'
        close all;

    case 'No'

end

if(seleccion=='Si')    %Si el usuario elije 'Si', cierra el procesado y pasa a la
siguiente.
    continue
    j=j+1;
end
end
```

CalculaUmbralOtsu.m

```
function [umbral,tabVarIntraclase] = CalculaUmbralOtsu (h)
%function umbral = CalculaUmbralOtsu (h)

p = double(h)/sum(h);

tabVarIntraclase = zeros(256,1);

for T = 0:255
    indT = T+1;
    A1 = sum(p(1:indT));
    A2 = sum(p(indT+1:256));

    mu1 = 0;
    mu2 = 0;
    for r=0:T
        mu1 = mu1 + r*p(r+1);
    end
    for r=T+1:255
        mu2 = mu2 + r*p(r+1);
    end
    mu1 = mu1/A1;
    mu2 = mu2/A2;

    var1 = 0;
    var2 = 0;
    for r=0:T
        var1 = var1 + (r-mu1)^2*p(r+1);
    end
    for r=T+1:255
        var2 = var2 + (r-mu2)^2*p(r+1);
    end
    var1 = var1/A1;
    var2 = var2/A2;

    tabVarIntraclase(indT) = A1*var1+A2*var2;
end

[varMin,indMin] = min(tabVarIntraclase);

umbral = indMin-1;
```

ext_valoresH1.m

```

%% Parámetros variables

prompt={'Tiempo pause:', 'Número de filas:', 'Número de columnas:'};
dlg_title='Ingrese valores correspondientes';
width = 55;
height = 1;
num_lines=[height, width];
def_out= {'3', '2', '3'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
t= str2num(inputH{1});
fil= str2num(inputH{2});
col= str2num(inputH{3});

%% Programa para extraer etiqueta deseada a h1= 20 metros

carp_1 = uigetdir('C:\', 'Selecciona carpeta a 20 metros');

Cim_25m = fullfile(carp_1, '*.jpg'); %guardar en formato jpg

Imagenes_20_metros=dir(Cim_25m);

imagenes_20= string({Imagenes_20_metros.name});

figure;
montage(imagenes_20, 'Size',[fil col]), title('Imágenes originales a h1= 25 metros');

offset=0;
offset2=0;

for i = 1:numel(imagenes_20)
if(i <= col)
    text(230+offset,30, ['Imagen ' num2str(i)]
, 'fontsize',12, 'color', 'yellow', 'fontweight', 'bold')
    offset=offset+680;

elseif(i > col)
    text(230+offset2,390, ['Imagen ' num2str(i)]
, 'fontsize',12, 'color', 'yellow', 'fontweight', 'bold')
    offset2=offset2+680;
end
end

end

pause(t);

prompt={'¿Qué imagen desea etiquetar?'};
dlg_title='Ingrese valor correspondiente';
width = 55;
height = 1;
num_lines=[height, width];
def_out= {'1'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
open= str2num(inputH{1});
close(gcf);

open_etiq = fullfile(carp_1, 'Imágenes_Procesadas');

```

```

imE_25m=dir(open_etiq);

imagenesE_25=['etiquetada_h1_' num2str(open) '.jpg']; %Obtengo
nombre imagenes

    nombreE_im1=[open_etiq '\\']; %Recorre el directorio

    f = imread(strcat(nombreE_im1, imagenesE_25)); %Leo imagenes

figure; imshow(f); %Imagen etiquetada elegida

pause(t);

prompt={'Introduzca el número de la etiqueta deseada:'};
dlg_title='Ingrese valor correspondiente';
width = 55;
height = 1;
num_lines=[height, width];
def_out= {'1'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
etiQ= str2num(inputH{1});

% Leo las propiedades de la imagen elegida

open_excel = fullfile(carp_1, 'Propiedades');
Pim_25m=dir(open_excel);

imagenesEx_25=['props_img10m_' num2str(open) '.xlsx']; %Obtengo nombre
excel

    nombreP_im1=[open_excel '\\']; %Recorre el directorio

    f_prop = readtable(strcat(nombreP_im1, imagenesEx_25));
    f_area = readtable(strcat(nombreP_im1, imagenesEx_25), 'Range','H:H ',
'ReadVariableNames',false);
    area=f_area(etiQ,:);
close(gcf);

area_metros= table2array(area);
area_metros=round(area_metros,2)
figure; imshow(f); %Área etiqueta elegida
    text(660,1300, ['Área de la etiqueta nº ', num2str(etiQ), '= ',
num2str(area_metros), ' metros cuadrados']
,'fontsize',12,'color','r','fontweight','bold')

```

ext_valoresH2.m

```

%% Parámetros variables

prompt={'Tiempo pause:', 'Número de filas:', 'Número de columnas:'};
dlg_title='Ingrese valores correspondientes';
width = 55;
height = 1;
num_lines=[height, width];
def_out= {'3', '2', '3'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
t= str2num(inputH{1});
fil= str2num(inputH{2});
col= str2num(inputH{3});

%% Programa para extraer etiqueta deseada a h1= 5 metros

carp_2 = uigetdir('C:\', 'Selecciona carpeta a 5 metros');
Cim_5m = fullfile(carp_2, '\*.jpg');
Imagenes_5_metros=dir(Cim_5m);
imagenes_5= string({Imagenes_5_metros.name});

figure;
montage(imagenes_5, 'Size',[fil col]), title('Imágenes originales a h2= 5 metros');

offset=0;
offset2=0;

for i = 1:numel(imagenes_5)
if(i <= col)
    text(230+offset,30, ['Imagen ' num2str(i)]
, 'fontsize',12, 'color', 'yellow', 'fontweight', 'bold')
    offset=offset+680;

elseif(i > col)
    text(230+offset2,390, ['Imagen ' num2str(i)]
, 'fontsize',12, 'color', 'yellow', 'fontweight', 'bold')
    offset2=offset2+680;
end
end

end

pause(t);

prompt={'¿Qué imagen desea etiquetar?'};
dlg_title='Ingrese valor correspondiente';
width = 55;
height = 1;
num_lines=[height, width];
def_out= {'1'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
open= str2num(inputH{1});
close(gcf);

open_etiq = fullfile(carp_2, 'Imágenes_Procesadas');

```

```

imE_5m=dir(open_etiq);

imagenesE_5=['etiquetada_h2_' num2str(open) '.jpg'];           %Obtengo
nombre imagenes

    nombreE_im2=[open_etiq '\\'];                               %Recorre el directorio

    g = imread(strcat(nombreE_im2, imagenesE_5));               %Leo imagenes

figure; imshow(g); %Imagen etiquetada elegida

pause(t);

prompt={'Introduzca el número de la etiqueta deseada:'};
dlg_title='Ingrese valor correspondiente';
width = 55;
height = 1;
num_lines=[height, width];
def_out= {'1'};
inputH= inputdlg(prompt, dlg_title, num_lines, def_out);
eti= str2num(inputH{1});

% Leo las propiedades de la imagen elegida

open_excel = fullfile(carp_1, 'Propiedades');
Pim_5m=dir(open_excel);

imagenesEx_5=['props_img5m_' num2str(open) '.xlsx'];           %Obtengo nombre
excel

    nombreP_im2=[open_excel '\\'];                               %Recorre el directorio

    g_prop = readtable(strcat(nombreP_im2, imagenesEx_5));
    g_area = readtable(strcat(nombreP_im2, imagenesEx_5), 'Range','H:H ',
'ReadVariableNames',false);
    area= g_area(eti,:);
close(gcf);

area_metros= table2array(area);
area_metros=round(area_metros,2)
figure; imshow(g); %Área etiqueta elegida
    text(660,1300, ['Área de la etiqueta nº ', num2str(eti), '= ',
num2str(area_metros), ' metros cuadrados']
,'fontsize',12,'color','r','fontweight','bold')

```