

TRABAJO FIN DE GRADO



UCAM

UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA POLITÉCNICA SUPERIOR

Ingeniería en Tecnologías de Telecomunicación

Diseño y evaluación de un dispositivo conectado para la
estimación del porcentaje de cubierta vegetal mediante
procesamiento de imágenes

Autor:

Álvaro Martínez López

Director:

Dr. Juan Miguel Navarro Ruiz

Murcia, julio de 2026

TRABAJO FIN DE GRADO



UCAM

UNIVERSIDAD CATÓLICA
DE MURCIA

ESCUELA POLITÉCNICA SUPERIOR

Ingeniería en Tecnologías de Telecomunicación

Diseño y evaluación de un dispositivo conectado para la
estimación del porcentaje de cubierta vegetal mediante
procesamiento de imágenes

Autor:

Álvaro Martínez López

Director:

Dr. Juan Miguel Navarro Ruiz

Murcia, julio de 2026

Agradecimientos

A mis padres, por su apoyo incondicional, por animarme en todo momento y por acompañarme durante todos estos años; y a la UCAM y a todos los profesores que han formado parte de esta etapa, por la formación recibida y por los conocimientos y experiencias que me han permitido llegar hasta aquí.

Listado de Abreviatura

BLE, Bluetooth Low Energy, tecnología Bluetooth de bajo consumo.

CIVE, Color Index of Vegetation Extraction, índice de color para la extracción de vegetación.

CMOS, Complementary Metal-Oxide-Semiconductor, tecnología de semiconductor complementario de óxido metálico.

ESP32, familia de microcontroladores de Espressif Systems con conectividad inalámbrica integrada.

ExG, Excess Green, índice de Exceso de Verde.

ExGR, Excess Green minus Excess Red, índice de Exceso de Verde menos Exceso de Rojo.

HSV, Hue, Saturation, Value, espacio de color basado en tono, saturación y valor.

IA, inteligencia artificial.

IDE, Integrated Development Environment, entorno de desarrollo integrado.

IoT, Internet of Things, Internet de las Cosas.

IP, Internet Protocol, protocolo de Internet.

JPEG, Joint Photographic Experts Group, formato de compresión de imágenes con pérdidas.

NDRE, Normalized Difference Red Edge, índice de diferencia normalizada del borde rojo.

NDVI, Normalized Difference Vegetation Index, índice de vegetación de diferencia normalizada.

PSRAM, Pseudo-Static Random-Access Memory, memoria de acceso aleatorio pseudoestática.

RGB, Red, Green, Blue, espacio de color basado en rojo, verde y azul.

RGB565, formato RGB de 16 bits, con 5 bits para rojo, 6 para verde y 5 para azul.

RGB888, formato RGB de 24 bits, con 8 bits para cada canal de color.

ROI, Region of Interest, región de interés.

RTC, Real-Time Clock, reloj en tiempo real.

TinyML, Tiny Machine Learning, aprendizaje automático aplicado a microcontroladores y dispositivos con recursos limitados.

VGA, Video Graphics Array, resolución de imagen de 640 × 480 píxeles.

Wi-Fi, tecnología de comunicación inalámbrica basada en la familia de estándares IEEE 802.11.

YUV, modelo de color que separa la luminancia, Y, de las componentes de crominancia, U y V.

ÍNDICE

RESUMEN	21
1. INTRODUCCIÓN	25
2. JUSTIFICACIÓN.....	29
3. MARCO TEÓRICO	33
3.1. Área de estudio y contexto tecnológico.....	33
3.1.1. Agricultura de precisión	33
3.1.2. Monitorización agrícola mediante sensores e imágenes	34
3.1.3. Visión artificial embebida aplicada a la agricultura.....	36
3.2. Cubierta vegetal y métodos de estimación	37
3.2.1. Concepto de cubierta vegetal	37
3.2.2. Importancia de la cubierta vegetal	38
3.2.3. Métodos visuales y procedimientos manuales.....	39
3.2.4. Métodos basados en imágenes	39
3.2.5. Ventajas y limitaciones de los métodos de estimación	40
3.3. Fundamentos del procesamiento digital de imágenes	41
3.3.1. Imagen digital y píxel	42
3.3.2. Resolución espacial.....	42
3.3.3. Representación digital del color.....	43
3.3.4. Espacio de color RGB.....	44
3.3.5. Formatos JPEG y RGB888.....	44
3.3.6. Región de interés.....	45
3.3.7. Factores que afectan a la adquisición de imágenes	46
3.4. Segmentación de vegetación en imágenes RGB.....	47
3.4.1. Segmentación binaria de vegetación y fondo	47
3.4.2. Umbralización fija y adaptativa	48
3.4.3. Detección mediante dominancia verde	49

3.4.4.	Índices de vegetación visibles y Exceso de Verde.....	50
3.4.5.	Segmentación mediante el espacio de color HSV	51
3.4.6.	Calibración y limitaciones de la segmentación por color.....	52
3.5.	Microcontroladores y sistemas embebidos de visión	54
3.5.1.	Sistemas embebidos aplicados al procesamiento de imágenes .	55
3.5.2.	Familia ESP32	55
3.5.3.	Cámaras integradas y sensores de imagen.....	56
3.5.4.	Restricciones del procesamiento en microcontroladores	57
3.6.	Procesamiento local y procesamiento remoto	58
3.7.	Internet de las Cosas y monitorización remota	60
3.8.	Estado del arte de la estimación de cubierta vegetal	62
3.8.1.	Canopeo	63
3.8.2.	VitiCanopy	63
3.8.3.	Teledetección mediante satélites	64
3.8.4.	Monitorización mediante drones y sensores multiespectrales	65
3.8.5.	Otras herramientas y trabajos previos	66
3.8.6.	Métodos empleados en investigaciones anteriores	66
3.8.7.	Comparación de las soluciones existentes	67
3.8.8.	Ventajas, limitaciones y retos pendientes	68
3.9.	Síntesis del estado del arte y posicionamiento del proyecto.....	69
4.	OBJETIVOS.....	73
4.1.	Objetivo general	73
4.2.	Objetivos específicos	73
5.	METODOLOGÍA	77
5.1.	Enfoque metodológico y fases del trabajo	77
5.2.	Arquitectura general del sistema.....	78
5.3.	Hardware utilizado y montaje del prototipo	80

5.4.	Software y entorno de desarrollo	82
5.5.	Configuración de la cámara, memoria y conectividad.....	83
5.6.	Captura, preprocesado de imagen y región de interés	85
5.7.	Algoritmos de estimación de cubierta vegetal.....	88
5.7.1.	Algoritmo basado en Exceso de Verde normalizado	89
5.7.2.	Algoritmo basado en HSV.....	89
5.7.3.	Comparación inicial entre algoritmos	90
5.8.	Interfaz web local	91
5.9.	Integración con Thinger.io.....	93
5.10.	Modo automático y monitorización periódica.....	94
5.11.	Diseño de las pruebas experimentales y criterios de validación	95
5.12.	Repositorio GitHub y documentación del código	98
6.	RESULTADOS	101
6.1.	Validación funcional del sistema	101
6.2.	Resultados obtenidos desde la interfaz web local	104
6.3.	Resultados de la integración con Thinger.io	107
6.4.	Comparación entre ExG, HSV y Canopeo	111
6.5.	Prueba de repetibilidad	119
6.6.	Funcionamiento del modo automático y Deep Sleep	120
6.7.	Limitaciones observadas.....	123
7.	CONCLUSIONES	127
7.1.	Líneas futuras	128
8.	REFLEXIÓN Y VALORACIÓN PERSONAL	131
9.	REFERENCIAS	133
10.	ANEXOS	137
10.1.	Anexo 1. Repositorio GitHub y código fuente	137

ÍNDICE DE ELEMENTOS GRÁFICOS

FIGURA

Fig. 2.1. Uso de Canopeo para la estimación de la cubierta vegetal mediante un dispositivo móvil.	29
Fig. 3.1. Escalas de adquisición de imágenes utilizadas en la monitorización agrícola.	35
Fig. 3.2. Representación conceptual de diferentes porcentajes de cubierta vegetal.....	37
Fig. 3.3. Representación conceptual de los modelos de color RGB y HSV.	51
Fig. 3.4. Arquitectura conceptual de un sistema IoT con procesamiento local de imágenes.....	60
Fig. 3.5. Interfaz de Thinger.io con el estado del dispositivo conectado.	62
Fig. 5.1. Esquema general de la arquitectura del sistema.....	79
Fig. 5.2. Montaje físico del prototipo y disposición utilizada para la captura de las escenas vegetales.....	81
Fig. 5.3. Flujo de captura, procesamiento y generación de resultados del sistema.	86
Fig. 5.4. Página principal del repositorio GitHub del proyecto.	98
Fig. 6.1. Resultado de la interfaz web local para una escena analizada mediante ExG.	106
Fig. 6.2. Comandos disponibles en la Remote Console de Thinger.io.	108
Fig. 6.3. Dashboard de Thinger.io con los principales valores enviados por el prototipo.	109
Fig. 6.4. Resultado obtenido para la primera planta mediante Exceso de Verde.	112
Fig. 6.5. Resultado obtenido para la primera planta mediante HSV.....	113
Fig. 6.6. Resultado obtenido para la primera planta mediante Canopeo.....	114
Fig. 6.7. Resultado obtenido para la segunda planta mediante Exceso de Verde.	115
Fig. 6.8. Resultado obtenido para la segunda planta mediante HSV.	116
Fig. 6.9. Resultado obtenido para la segunda planta mediante Canopeo.	117

TABLA

Tabla 3.1. Comparativa de soluciones y enfoques para la estimación de cubierta vegetal.....	68
Tabla 5.1. Principales parámetros de configuración de la cámara, la memoria y la conectividad.....	85
Tabla 6.1. Resultados de la validación funcional del sistema.	103
Tabla 6.2. Registro de mediciones almacenadas en el bucket_vegetacion.	110
Tabla 6.3. Comparación de los porcentajes obtenidos mediante ExG, HSV y Canopeo.....	118
Tabla 6.4. Resultados de la prueba de repetibilidad.	120
Tabla 6.5. Comparación funcional de los modos automáticos comprobados.	123

RESUMEN

El seguimiento periódico de la cubierta vegetal es relevante en agricultura y monitorización ambiental, aunque los métodos manuales dificultan automatizar las mediciones. Este TFG tuvo como objetivo diseñar, implementar y validar un sistema embebido capaz de estimar el porcentaje de cubierta vegetal mediante el procesamiento local de imágenes RGB. El prototipo utiliza una M5Stack TimerCamera-F con microcontrolador ESP32 y sensor OV3660. Captura imágenes en resolución VGA, las convierte de JPEG a RGB888 y analiza una región de interés mediante Exceso de Verde normalizado (ExG) y HSV (Hue, Saturation, Value; tono, saturación y valor). El sistema calcula el porcentaje de cubierta vegetal y genera una máscara binaria y una imagen de superposición. También incorpora una interfaz web local para configurar parámetros y consultar los resultados, y se integra con la plataforma IoT Thinger.io para enviar, almacenar y visualizar las mediciones y controlar los modos automáticos, incluido Deep Sleep (modo bajo consumo). Las pruebas confirmaron el funcionamiento conjunto de la captura, el procesamiento, la visualización y la comunicación de datos. En dos escenas comparadas con Canopeo, ExG produjo los valores más próximos a esta referencia, mientras que HSV ofreció porcentajes superiores. La prueba de repetibilidad, realizada con cinco capturas, mostró rangos de 0,2 puntos porcentuales para ExG y 0,1 para HSV, con desviaciones estándar de 0,084 y 0,045 puntos porcentuales, respectivamente. También se verificaron los ciclos automáticos y la conservación de la configuración entre despertares. La autonomía teórica, calculada a partir de las hipótesis de consumo adoptadas, fue de 1,44 horas en funcionamiento continuo y de 32,6 días en Deep Sleep, con una captura y un envío cada 24 horas. En conclusión, el prototipo demuestra la viabilidad de estimar autónoma y periódicamente la cubierta vegetal, aunque los resultados dependen de la iluminación, el encuadre, la región de interés y los umbrales seleccionados.

Palabras claves: Cubierta vegetal, procesamiento de imágenes, sistemas embebidos, ESP32, Exceso de Verde, HSV, Internet de las Cosas.

ABSTRACT

Periodic monitoring of vegetation cover is relevant in agriculture and environmental monitoring, although manual methods make it difficult to automate measurements. The objective of this Final Degree Project was to design, implement, and validate an embedded system capable of estimating the percentage of vegetation cover through local processing of RGB images. The prototype uses an M5Stack TimerCamera-F equipped with an ESP32 microcontroller and an OV3660 sensor. It captures images at VGA resolution, converts them from JPEG to RGB888, and analyses a region of interest using Normalised Excess Green (ExG) and HSV (Hue, Saturation, Value). The system calculates the percentage of vegetation cover and generates a binary mask and an overlay image. It also incorporates a local web interface for configuring parameters and viewing the results, and it is integrated with the Thingier.io IoT platform to send, store, and display measurements and control the automatic modes, including Deep Sleep (Low-power mode). The tests confirmed the combined operation of image capture, processing, visualisation, and data communication. In two scenes compared with Canopeo, ExG produced the values closest to this reference, whereas HSV yielded higher percentages. The repeatability test, performed with five captures, showed ranges of 0.2 percentage points for ExG and 0.1 for HSV, with standard deviations of 0.084 and 0.045 percentage points, respectively. The automatic cycles and the preservation of the configuration between wake-up events were also verified. The theoretical battery life, calculated from the adopted power consumption assumptions, was 1.44 hours in continuous operation and 32.6 days in Deep Sleep, with one capture and one transmission every 24 hours. In conclusion, the prototype demonstrates the feasibility of estimating vegetation cover autonomously and periodically, although the results depend on lighting conditions, framing, the region of interest, and the selected thresholds.

Keywords: Vegetation cover, image processing, embedded systems, ESP32, Excess Green, HSV, Internet of Things.

1. INTRODUCCIÓN

El seguimiento del desarrollo de la vegetación es una tarea de interés en ámbitos como la agricultura, la investigación agronómica y la gestión de espacios naturales. Una de las variables empleadas para describir dicho desarrollo es el porcentaje de cubierta vegetal, entendido como la proporción de una superficie determinada que se encuentra ocupada por vegetación. Su estimación permite analizar la evolución de los cultivos, valorar el establecimiento de las plantas y estudiar el grado de ocupación del terreno [1].

Tradicionalmente, la cubierta vegetal se ha estimado mediante observaciones visuales o procedimientos manuales. Sin embargo, estos métodos pueden depender de la experiencia y del criterio de la persona que realiza la evaluación, lo que puede dificultar la comparación y repetición de las mediciones. Como alternativa, el procesamiento digital de imágenes permite analizar una escena de forma sistemática y diferenciar la vegetación del suelo a partir de las características de color de sus píxeles.

Las imágenes obtenidas mediante cámaras convencionales contienen información de los canales rojo, verde y azul, conocidos como canales RGB, del inglés Red, Green and Blue. Esta información puede utilizarse para identificar la vegetación mediante diferentes criterios de clasificación. Además, el empleo de cámaras RGB presenta un coste y una complejidad inferiores a los de otros sistemas de adquisición de imágenes, por lo que constituye una opción accesible para el desarrollo de dispositivos de seguimiento vegetal [2].

El objetivo de este Trabajo Fin de Grado es diseñar, implementar y validar un sistema basado en la M5Stack TimerCamera-F, un módulo que integra un microcontrolador ESP32 y un sensor de imagen OV3660 [3], capaz de estimar el porcentaje de cubierta vegetal mediante el procesamiento local de imágenes RGB. El prototipo integra la captura de imágenes, la configuración de los parámetros de análisis, la clasificación de los píxeles y el cálculo del porcentaje de cubierta vegetal. Asimismo, dispone de una interfaz web local para configurar el dispositivo y consultar los resultados, que también se envían a la plataforma IoT, del inglés Internet of Things, Thingier.io para su configuración, almacenamiento y visualización remota.

El presente documento se estructura de la siguiente manera: la Justificación expone las razones que motivan el desarrollo de este sistema; el Marco Teórico desarrolla los fundamentos técnicos y el estado del arte en los que se apoya el trabajo; los Objetivos concretan las metas planteadas; la Metodología describe el proceso de diseño e implementación; los Resultados recogen las pruebas realizadas y su análisis; y las Conclusiones sintetizan los logros obtenidos y las posibles líneas de trabajo futuro.

2. JUSTIFICACIÓN

La estimación periódica de la cubierta vegetal mediante métodos manuales resulta adecuada para mediciones puntuales, pero presenta dificultades cuando se pretende realizar un seguimiento continuado desde una ubicación fija. En ese contexto, herramientas como Canopeo, una aplicación que permite estimar la cubierta vegetal a partir de imágenes capturadas con un dispositivo móvil, ha demostrado la utilidad de las imágenes RGB para obtener estimaciones rápidas y no destructivas [1]; sin embargo, su uso habitual requiere la intervención directa de una persona que capture y procese cada imagen de forma individual. Esta dependencia limita la posibilidad de automatizar el registro y reduce la eficiencia del seguimiento en aplicaciones que exigen mediciones frecuentes o desatendidas.

La Fig. 2.1 ilustra el uso de Canopeo en campo mediante un dispositivo móvil.



Fig. 2.1. Uso de Canopeo para la estimación de la cubierta vegetal mediante un dispositivo móvil.

Fuente: Web de Canopeo.

Desde el punto de vista tecnológico, los microcontroladores de bajo coste con capacidad de procesamiento local, conectividad inalámbrica e integración de cámara ofrecen una vía para desarrollar soluciones compactas y autónomas. La M5Stack TimerCamera-F reúne estas características al integrar en un único dispositivo un microcontrolador ESP32, conectividad Wi-Fi, batería y un sensor

de imagen OV3660, lo que permite integrar la adquisición de imágenes, su análisis y la comunicación de resultados sin depender de un ordenador externo [3]. Frente a sistemas de adquisición más complejos, como los sensores multiespectrales, el uso de una cámara RGB reduce el coste y la complejidad del sistema, con la contrapartida de trabajar únicamente en el espectro visible [2].

Desde el punto de vista académico, este proyecto permite aplicar de forma conjunta conocimientos adquiridos a lo largo del Grado en áreas como los sistemas embebidos, el procesamiento digital de imágenes, las comunicaciones inalámbricas y el Internet de las Cosas. La integración de estas disciplinas en un único prototipo funcional responde al carácter interdisciplinar que se espera de un Trabajo Fin de Grado en Ingeniería en Tecnologías de Telecomunicación.

El sistema está concebido para aplicaciones experimentales, educativas y de monitorización a pequeña escala. No pretende sustituir a equipos agrónomos profesionales, sino evaluar hasta qué punto un dispositivo basado en componentes accesibles puede proporcionar estimaciones útiles, repetibles y automatizadas del porcentaje de cubierta vegetal. La comparación con otros métodos de segmentación basados en imágenes RGB y con la herramienta Canopeo permitirá analizar de forma comparativa las posibilidades y limitaciones de la solución desarrollada.

La realización de este TFG se justifica por la necesidad de disponer de un sistema autónomo, compacto y de bajo coste que integre captura, procesamiento y comunicación de resultados para la monitorización periódica de la cubierta vegetal, así como por el interés académico de abordar un problema técnico real aplicando de manera integrada las competencias propias de la titulación.

3. MARCO TEÓRICO

En este capítulo se presentan los fundamentos técnicos y el estado del arte que sustentan el desarrollo del proyecto. Se abordan la agricultura de precisión, la estimación de la cubierta vegetal, el procesamiento de imágenes RGB, los sistemas embebidos, el Internet de las Cosas y las principales soluciones existentes en este ámbito.

3.1. Área de estudio y contexto tecnológico

El presente Trabajo Fin de Grado se sitúa en la intersección de varias áreas de la Ingeniería de Telecomunicación: los sistemas embebidos, el procesamiento digital de imágenes, las comunicaciones inalámbricas y el Internet de las Cosas. Estas tecnologías se aplican al ámbito de la monitorización agrícola con el propósito de obtener información cuantitativa sobre el desarrollo de la vegetación.

La combinación de sensores, capacidad de procesamiento y conectividad permite desarrollar dispositivos capaces de adquirir datos del entorno, analizarlos y transmitir los resultados. Dentro de este contexto, las imágenes digitales constituyen una fuente de información útil para estudiar características visibles de las plantas y de la superficie que ocupan.

3.1.1. Agricultura de precisión

La agricultura de precisión es un enfoque de gestión agrícola basado en la adquisición y el análisis de datos. Su finalidad es conocer con mayor detalle el estado de los cultivos y las condiciones del entorno, de manera que las actuaciones puedan adaptarse a las necesidades observadas. Para ello se emplean tecnologías como sensores, sistemas de posicionamiento, cámaras, vehículos aéreos no tripulados, plataformas digitales y redes de comunicación [4].

Este enfoque tiene en cuenta que las condiciones de una parcela pueden variar tanto espacialmente como a lo largo del tiempo. Factores como la humedad del suelo, el desarrollo de las plantas o la presencia de determinadas alteraciones no tienen por qué ser uniformes en toda la superficie. La toma

periódica de datos permite identificar estas diferencias y realizar un seguimiento más objetivo de su evolución.

La agricultura de precisión no implica necesariamente la automatización completa de las decisiones. Las tecnologías utilizadas también pueden actuar como herramientas de apoyo, proporcionando mediciones que complementen la observación directa. En este sentido, pueden desarrollarse desde sistemas complejos de control hasta dispositivos destinados a registrar una variable concreta [4].

Entre las variables que pueden emplearse para caracterizar el estado de un cultivo se encuentran la temperatura, la humedad, la radiación, las propiedades del suelo y otras variables obtenidas mediante imágenes, como la cubierta vegetal, el vigor o determinados cambios visibles en las plantas [4]–[6]. En este trabajo se ha seleccionado el porcentaje de cubierta vegetal, cuyo seguimiento permite cuantificar qué proporción de una superficie está ocupada por vegetación y observar su evolución en diferentes momentos.

3.1.2. Monitorización agrícola mediante sensores e imágenes

La monitorización agrícola consiste en adquirir información sobre el cultivo y su entorno. Para ello pueden utilizarse sensores de temperatura, humedad, radiación o propiedades del suelo, entre otros. Las técnicas basadas en imágenes complementan estas mediciones, ya que permiten analizar características visibles de una superficie sin establecer contacto directo con ella.

Las imágenes pueden obtenerse desde satélites, aeronaves, drones o cámaras situadas a nivel del suelo. Cada alternativa presenta diferencias en cuanto a superficie observada, resolución, frecuencia de captura y coste. Las imágenes de satélite y las capturas aéreas permiten analizar áreas extensas, mientras que las cámaras próximas al terreno proporcionan un mayor nivel de detalle sobre una zona limitada [5].

La Fig. 3.1 representa estas tres escalas de adquisición y muestra que, a medida que disminuye el área observada, aumenta el nivel de detalle espacial.

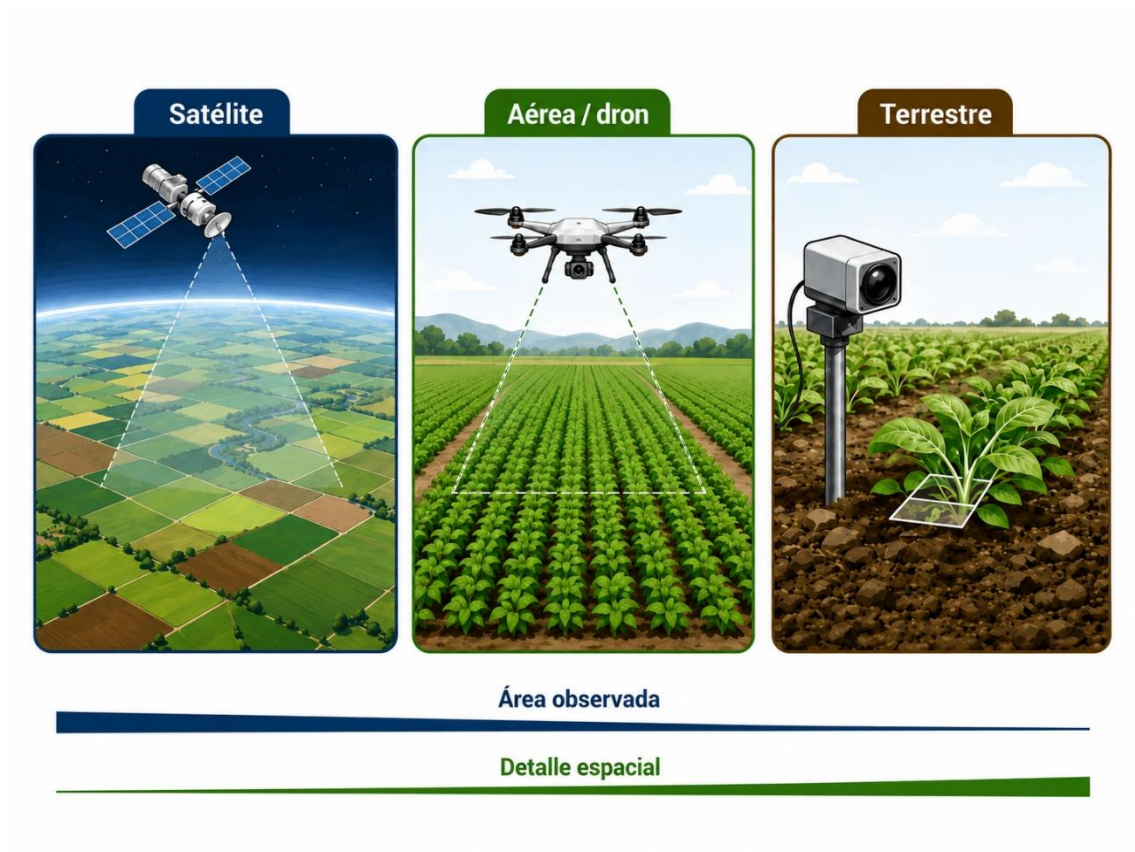


Fig. 3.1. Escalas de adquisición de imágenes utilizadas en la monitorización agrícola.

Fuente: elaboración propia mediante IA generativa.

Las cámaras digitales se utilizan en aplicaciones como el seguimiento del crecimiento, la identificación de plantas, la detección de malas hierbas, la clasificación de productos o la observación de alteraciones visibles. Mediante técnicas de visión artificial, la información capturada puede transformarse en datos cuantitativos que posteriormente se analizan mediante distintos procedimientos [6].

El análisis de imágenes también permite conservar una representación visual de la escena estudiada. Esto facilita la revisión posterior de las mediciones y la identificación de factores que pueden afectar al resultado, como las sombras, los reflejos, el desenfoque o las variaciones de iluminación. Por este motivo, las condiciones de captura deben tenerse en cuenta al diseñar y validar un sistema de monitorización.

3.1.3. *Visión artificial embebida aplicada a la agricultura*

La visión artificial embebida consiste en integrar la adquisición y el procesamiento de imágenes en un dispositivo de tamaño reducido. En lugar de depender de un ordenador convencional para realizar todo el análisis, parte o la totalidad del procesamiento se ejecuta en el propio dispositivo que efectúa la captura.

Este planteamiento permite obtener una medida cuantitativa a partir de la imagen cerca del lugar donde se generan los datos. Cuando únicamente es necesario comunicar el resultado del análisis, el volumen de información transmitido puede ser menor que el requerido para enviar de forma continua las imágenes completas. Este enfoque se enmarca en el concepto de edge computing, un paradigma en el que la capacidad de cómputo se sitúa próxima a la fuente de los datos, de manera que el procesamiento y la toma de decisiones pueden realizarse localmente antes de transmitir los resultados a otros sistemas o a la nube [7].

Los dispositivos embebidos disponen de una capacidad de memoria y procesamiento inferior a la de un ordenador. Por ello, es necesario adaptar la resolución de las imágenes, el formato utilizado y la complejidad de los algoritmos a los recursos disponibles. El diseño debe buscar un equilibrio entre la calidad de la información obtenida y el tiempo y la memoria necesarios para procesarla.

La incorporación de conectividad inalámbrica permite complementar el análisis local con servicios de almacenamiento y visualización remota. De este modo, la visión artificial embebida, las comunicaciones inalámbricas y las plataformas IoT pueden integrarse en sistemas compactos de monitorización. El aprendizaje automático y, en particular, TinyML, entendido como la ejecución de modelos de aprendizaje automático en microcontroladores y otros dispositivos con recursos limitados de memoria, procesamiento y energía, han ganado interés en los sistemas embebidos [8]. Este enfoque permite realizar tareas de inferencia directamente en el dispositivo, reduciendo la dependencia de servidores externos. En el ámbito de la monitorización agrícola, estas técnicas pueden utilizarse para apoyar la interpretación automática de imágenes o

señales capturadas por sensores [6]. No obstante, en este Trabajo Fin de Grado se ha optado por un enfoque basado en la segmentación y clasificación de píxeles mediante criterios de color, por lo que TinyML se menciona únicamente como una tecnología relacionada y una posible línea futura de desarrollo.

3.2. Cubierta vegetal y métodos de estimación

La cubierta vegetal representa la proporción de una superficie ocupada por vegetación. Cuando el análisis se limita a las partes verdes y visibles de las plantas, se emplea el concepto de cubierta vegetal verde. Su valor se expresa habitualmente como un porcentaje comprendido entre el 0 %, cuando no se identifica vegetación, y el 100 %, cuando toda la superficie analizada aparece cubierta por ella.

Esta variable puede estimarse mediante observación visual, procedimientos manuales, imágenes digitales o técnicas de teledetección. La elección del método depende de la superficie estudiada, la frecuencia de las mediciones, los recursos disponibles y el nivel de detalle requerido.

3.2.1. Concepto de cubierta vegetal

La cubierta vegetal representa la fracción del terreno cubierta por las partes visibles de las plantas al observar una superficie desde una posición determinada. En una imagen digital, puede interpretarse como la proporción de píxeles clasificados como vegetación respecto al número total de píxeles analizados. Esta interpretación se ilustra en la Fig. 3.2.



Fig. 3.2. Representación conceptual de diferentes porcentajes de cubierta vegetal.

Fuente: elaboración propia mediante IA generativa.

No debe confundirse con otras variables relacionadas con las plantas. La biomasa expresa la cantidad de materia vegetal presente, mientras que el índice de área foliar relaciona la superficie de las hojas con la superficie de terreno ocupada. Por tanto, dos zonas pueden presentar porcentajes de cubierta similares y contener cantidades diferentes de biomasa.

En la literatura científica, y concretamente en trabajos como el de Canopeo [1], se utiliza el término fractional green canopy cover, que puede traducirse como fracción de cubierta vegetal verde. Esta variable describe la proporción de superficie ocupada por vegetación verde visible y puede relacionarse con procesos como la interceptación de la radiación y la evapotranspiración.

Cuando la estimación se realiza mediante fotografías, el resultado depende de la superficie incluida en la imagen y de la posición de la cámara. Por este motivo, para comparar capturas realizadas en diferentes momentos conviene mantener unas condiciones similares de distancia, orientación y encuadre.

3.2.2. Importancia de la cubierta vegetal

El porcentaje de cubierta vegetal proporciona información sobre el grado de ocupación del terreno y la evolución visible de las plantas. Su seguimiento permite comparar diferentes momentos del ciclo de crecimiento, zonas de una misma parcela o tratamientos aplicados sobre el cultivo.

En los cultivos de cobertura, la presencia de vegetación contribuye a reducir la superficie de suelo expuesta y puede favorecer su protección frente a la acción directa de la lluvia. Por esta razón, la cubierta vegetal se utiliza como una variable para analizar el establecimiento y desarrollo de estos cultivos [9].

También puede emplearse como dato de apoyo en estudios agronómicos e hidrológicos, ya que la cantidad de suelo cubierta influye en la radiación que alcanza la superficie y en la distribución de la evapotranspiración [1]. Sin embargo, no permite conocer por sí sola el estado completo de un cultivo y debe complementarse con otras variables cuando se requiere una evaluación más amplia.

3.2.3. *Métodos visuales y procedimientos manuales*

Uno de los métodos más sencillos consiste en observar directamente una superficie y asignarle un porcentaje aproximado de cubierta vegetal. Este procedimiento es rápido y no requiere equipos específicos, aunque su resultado puede depender de la experiencia y del criterio del observador.

Esta subjetividad dificulta la comparación de pequeñas diferencias y la repetición de las mediciones. Dos personas pueden asignar porcentajes distintos a una misma escena e incluso un mismo observador puede modificar su criterio en momentos diferentes.

Para reducir esta variabilidad pueden utilizarse cuadrículas, transectos o puntos de muestreo. Estos procedimientos dividen la superficie o la imagen en posiciones definidas y registran cuántas coinciden con vegetación. Aunque aplican un criterio más estructurado, requieren más tiempo cuando deben analizarse numerosas muestras.

Los métodos visuales y manuales continúan siendo útiles para obtener una estimación inicial o como procedimiento de comprobación. No obstante, presentan limitaciones cuando se pretende realizar un seguimiento periódico y comparar mediciones de forma sistemática [9].

3.2.4. *Métodos basados en imágenes*

El análisis digital de imágenes permite estimar la cubierta vegetal mediante la clasificación de los píxeles que componen una fotografía. El procedimiento general consiste en capturar una imagen, identificar los píxeles asociados a vegetación y calcular la relación entre estos y el número total de píxeles analizados.

Las imágenes pueden obtenerse desde satélites, vehículos aéreos no tripulados o cámaras situadas cerca del terreno. Las primeras alternativas permiten estudiar superficies amplias, mientras que las cámaras próximas proporcionan un mayor nivel de detalle sobre zonas reducidas [5].

Las técnicas de teledetección emplean habitualmente sensores multiespectrales para registrar información en diferentes regiones del espectro

electromagnético. A partir de estos datos pueden calcularse índices relacionados con la presencia y el estado de la vegetación, aunque requieren sensores y procedimientos de tratamiento más complejos [2], [10].

Las cámaras digitales convencionales registran la información visible mediante los canales rojo, verde y azul. A partir de estos valores pueden aplicarse relaciones entre canales, índices visibles o umbrales de clasificación para diferenciar la vegetación del fondo. Esta alternativa presenta un coste y una complejidad menores, aunque puede verse afectada por las condiciones de iluminación y por los colores presentes en la escena [11].

Canopeo constituye un ejemplo de herramienta basada en este principio. La aplicación utiliza relaciones entre los canales RGB y el índice de Exceso de Verde para estimar la fracción de cubierta vegetal verde [1]. Sus características se estudiarán posteriormente en el apartado dedicado al Estado del Arte.

De forma general, el porcentaje de cubierta vegetal obtenido mediante una imagen puede calcularse mediante la ecuación (3.1):

$$C_v = \left(\frac{N_v}{N_t} \right) \cdot 100 \quad (3.1)$$

En la ecuación **¡Error! No se encuentra el origen de la referencia.**) C_v es el porcentaje de cubierta vegetal, N_v representa el número de píxeles clasificados como vegetación y N_t es el número total de píxeles incluidos en el análisis.

La expresión anterior no establece cómo debe clasificarse cada píxel. Las diferencias entre los métodos se encuentran en los criterios, índices o umbrales empleados para separar la vegetación del resto de la imagen.

3.2.5. *Ventajas y limitaciones de los métodos de estimación*

Los distintos métodos de estimación presentan ventajas y limitaciones. La evaluación visual es rápida y no requiere equipamiento, pero puede estar condicionada por la subjetividad del observador. Los procedimientos basados en cuadrículas o puntos de muestreo aplican criterios más definidos, aunque aumentan el tiempo necesario para realizar el análisis.

Los métodos digitales permiten utilizar el mismo procedimiento de clasificación en todas las imágenes, conservar una representación visual de cada medición y automatizar el cálculo. Sin embargo, el resultado depende de la calidad de la imagen y del criterio empleado para identificar la vegetación.

Las variaciones de iluminación, las sombras, los reflejos, el suelo húmedo o la presencia de elementos con colores similares a las plantas pueden provocar clasificaciones incorrectas. Por este motivo, deben controlarse las condiciones de captura en la medida de lo posible y comprobarse visualmente los resultados de la segmentación.

Los sensores multiespectrales proporcionan información que no está disponible en una cámara convencional, pero presentan generalmente un coste y una complejidad mayores. Las cámaras RGB constituyen una alternativa accesible para sistemas de monitorización de menor escala, siempre que se tengan en cuenta las limitaciones asociadas al uso exclusivo del espectro visible.

No existe, por tanto, un único procedimiento adecuado para todas las situaciones. La elección depende de la superficie observada, la frecuencia de captura, los recursos disponibles y el nivel de exactitud requerido. En los siguientes apartados se desarrollarán los fundamentos de las imágenes digitales y los principales métodos de segmentación de vegetación basados en información RGB.

3.3. Fundamentos del procesamiento digital de imágenes

El procesamiento digital de imágenes reúne las técnicas utilizadas para representar, transformar y analizar imágenes mediante sistemas informáticos. Una cámara convierte la información visual de una escena en datos numéricos que pueden almacenarse y procesarse. A partir de estos datos es posible realizar operaciones como cambiar el formato de la imagen, seleccionar una zona concreta o extraer información de los objetos representados [12].

En este trabajo, el procesamiento de imágenes permite analizar la información de color registrada por una cámara para diferenciar posteriormente la vegetación del resto de la escena. Para comprender este proceso es necesario

introducir los conceptos de píxel, resolución espacial, representación del color, formatos de imagen y región de interés.

3.3.1. Imagen digital y píxel

Una imagen digital puede representarse como una matriz formada por filas y columnas. Cada posición de esta matriz corresponde a un píxel, término procedente de la expresión inglesa *picture element*. El píxel constituye la unidad básica de una imagen y almacena información asociada a la intensidad o al color registrado en una posición determinada [12].

En una imagen en escala de grises, cada píxel contiene un único valor relacionado con su nivel de luminosidad. En una imagen en color se almacenan varios componentes por píxel, cuyo significado depende del espacio de color utilizado.

La posición de cada píxel se identifica mediante dos coordenadas espaciales. Por tanto, una imagen digital puede expresarse de forma general mediante una función bidimensional definida en la ecuación **¡Error! No se encuentra el origen de la referencia.!**):

$$f(x, y) \quad (3.2)$$

donde x e y representan la posición horizontal y vertical del píxel, respectivamente, y $f(x, y)$ contiene la información registrada.

El análisis digital consiste en aplicar operaciones sobre estos valores. Dichas operaciones pueden realizarse sobre toda la imagen o sobre una zona seleccionada, dependiendo del objetivo del procesamiento.

3.3.2. Resolución espacial

La resolución espacial describe la capacidad de una imagen para representar detalles. Habitualmente se expresa mediante su anchura y altura, $W \times H$, donde W representa el número de columnas y H el número de filas. El número total de píxeles se obtiene mediante la ecuación (3.3):

$$N = W \cdot H \quad (3.3)$$

Una resolución mayor puede permitir representar más detalles, ya que la escena se divide en un número superior de elementos. Sin embargo, también incrementa la cantidad de datos que deben almacenarse y procesarse.

La resolución no determina por sí sola la calidad de una imagen. Una fotografía con un número elevado de píxeles puede presentar desenfoque, ruido o una iluminación inadecuada. Del mismo modo, una resolución menor puede resultar suficiente cuando los elementos que se desean analizar aparecen correctamente definidos.

En sistemas embebidos, la resolución influye directamente en el uso de memoria y en el tiempo de procesamiento. Por ello, debe seleccionarse buscando un equilibrio entre el nivel de detalle requerido y los recursos disponibles.

3.3.3. Representación digital del color

El color registrado por una cámara se representa mediante valores numéricos organizados según un determinado espacio de color. Estos espacios permiten describir un color a partir de diferentes componentes.

Existen modelos basados directamente en las intensidades de luz y otros que separan propiedades como el tono, la saturación o la luminosidad. La elección del espacio de color depende de la aplicación y del tipo de información que se desea analizar [12].

La cantidad de valores disponibles para cada componente viene determinada por la profundidad de bits. Cuando se emplean 8 bits, pueden representarse $2^8 = 256$ niveles, comprendidos normalmente entre 0 y 255.

Los valores registrados por la cámara pueden variar según la sensibilidad del sensor, el balance de blancos, la exposición y las condiciones de iluminación. Por esta razón, el color obtenido debe interpretarse como una medida dependiente de las condiciones de captura.

3.3.4. *Espacio de color RGB*

El espacio RGB representa los colores mediante tres componentes: rojo, verde y azul, denominadas R, G y B. Se trata de un modelo aditivo, ya que los colores se forman combinando distintas intensidades de luz de estos tres canales.

Cuando cada componente utiliza 8 bits, sus valores se encuentran entre 0 y 255. La combinación (0,0,0) representa el negro, mientras que (255,255,255) corresponde al blanco. Los demás colores se obtienen mediante distintas combinaciones de los tres canales.

Un píxel puede presentar una tonalidad predominantemente verde cuando el valor del canal G es superior a los valores de R y B. Sin embargo, esta condición no permite asegurar por sí sola que el píxel pertenezca a vegetación, ya que otros objetos o fondos también pueden presentar una componente verde elevada.

El espacio RGB resulta adecuado para analizar relaciones entre canales, ya que se corresponde con la información proporcionada por muchas cámaras digitales. Su principal limitación es su sensibilidad a los cambios de iluminación, que pueden modificar simultáneamente los valores de los tres componentes.

3.3.5. *Formatos JPEG y RGB888*

Las imágenes pueden almacenarse mediante diferentes formatos según el uso previsto. En este trabajo resultan relevantes JPEG y RGB888, ya que responden a necesidades distintas.

JPEG es un formato de compresión con pérdidas diseñado para reducir el tamaño de las imágenes fotográficas. Durante la compresión se elimina parte de la información visual, lo que permite disminuir el espacio necesario para almacenar o transmitir la imagen [13].

El nivel de compresión determina el equilibrio entre tamaño y calidad. Una compresión elevada reduce el volumen de datos, pero puede introducir alteraciones en bordes, texturas y zonas con cambios de color.

RGB888 es una representación directa en la que cada píxel utiliza 8 bits para el canal rojo, 8 bits para el verde y 8 bits para el azul. Por tanto, cada píxel ocupa 24 bits, equivalentes a 3 bytes. La memoria necesaria para almacenar una imagen RGB888 puede estimarse mediante la ecuación (3.4):

$$M = 3 \cdot W \cdot H \quad (3.4)$$

En la ecuación (3.4), M representa el número de bytes necesarios, W la anchura de la imagen y H la altura.

La principal ventaja de RGB888 es que permite acceder directamente a los componentes de cada píxel, facilitando la aplicación de operaciones matemáticas y criterios de clasificación. Como inconveniente, requiere más memoria que una imagen comprimida.

En sistemas con cámara es habitual utilizar JPEG para la captura o el almacenamiento y convertir posteriormente la imagen a RGB cuando se necesita analizar individualmente sus canales. La librería de cámara de Espressif utilizada en placas ESP32 admite distintos formatos y funciones de conversión para este tipo de aplicaciones [14], [15]. La forma concreta en que se utiliza esta conversión se describirá en la Metodología.

3.3.6. *Región de interés*

Una región de interés, denominada habitualmente ROI por las siglas de *Region of Interest*, es una zona seleccionada dentro de una imagen sobre la que se realiza el análisis.

El uso de una ROI permite excluir bordes, objetos ajenos al estudio o zonas que no resultan representativas. También reduce el número de píxeles procesados, lo que puede disminuir el tiempo de ejecución.

La región puede definirse mediante coordenadas o porcentajes respecto al tamaño total de la imagen. Su selección debe realizarse de acuerdo con la escena analizada. Una ROI demasiado pequeña puede dejar fuera información relevante, mientras que una región excesivamente amplia puede incluir elementos que alteren el resultado.

El porcentaje calculado sobre una ROI representa únicamente la superficie incluida en ella. Por ello, cuando se comparan diferentes imágenes es necesario utilizar regiones equivalentes o conocer cómo se ha seleccionado cada zona.

3.3.7. Factores que afectan a la adquisición de imágenes

Las condiciones de captura pueden modificar los valores registrados por la cámara y afectar al análisis posterior. Entre los factores más importantes se encuentran la iluminación, las sombras, los reflejos, el enfoque, la distancia y el fondo.

La iluminación influye en la intensidad y en el color registrado. Una iluminación insuficiente puede dificultar la diferenciación entre elementos, mientras que una iluminación excesiva puede provocar zonas saturadas. Además, un mismo objeto puede presentar valores RGB diferentes si cambia la fuente o la dirección de la luz.

Las sombras reducen localmente la intensidad de la imagen, mientras que los reflejos generan zonas muy claras o con colores distintos a los originales. Estos efectos pueden alterar las relaciones entre los canales y dificultar la clasificación.

El enfoque determina la definición de los bordes y de los pequeños elementos. Una imagen desenfocada mezcla visualmente zonas próximas y reduce la precisión con la que pueden identificarse sus límites. La distancia de captura también modifica el tamaño aparente de los objetos y la superficie incluida en la imagen.

El fondo tiene una influencia especial en los métodos basados en color. Los suelos, recipientes u objetos con tonalidades parecidas a la vegetación pueden producir clasificaciones incorrectas. Del mismo modo, la humedad puede modificar el color aparente del suelo y generar reflejos.

En las aplicaciones de análisis vegetal mediante imágenes RGB conviene mantener las condiciones de adquisición lo más constantes posible. No obstante,

en entornos reales no siempre pueden evitarse las variaciones, por lo que los métodos utilizados deben evaluarse bajo diferentes condiciones de captura [16].

Los conceptos desarrollados en este apartado permiten comprender cómo se representa, almacena y prepara una imagen para su análisis. El siguiente apartado se centra en los métodos empleados para segmentar la imagen y diferenciar la vegetación del resto de la escena.

3.4. Segmentación de vegetación en imágenes RGB

La segmentación consiste en dividir una imagen en regiones o grupos de píxeles que presentan características comunes. Su finalidad es separar los elementos de interés del resto de la escena para facilitar su análisis posterior [12].

En la estimación de la cubierta vegetal, este proceso permite distinguir los píxeles correspondientes a vegetación de aquellos asociados al suelo, las sombras u otros elementos del fondo. Una vez realizada la clasificación, puede calcularse la proporción de la imagen ocupada por vegetación.

Existen técnicas de segmentación basadas en la intensidad, el color, la textura, los bordes o el aprendizaje automático. En este trabajo resultan especialmente relevantes los métodos basados en color, ya que utilizan directamente la información de los canales rojo, verde y azul proporcionado por una cámara convencional.

3.4.1. Segmentación binaria de vegetación y fondo

Cuando únicamente se distinguen dos clases, el resultado se denomina segmentación binaria. En este caso, cada píxel se clasifica como vegetación o como fondo mediante una regla de decisión.

El resultado puede representarse mediante una máscara binaria definida en la ecuación (3.5):

$$M(x, y) = \begin{cases} 1, & \text{si el píxel pertenece a la vegetación} \\ 0, & \text{si el píxel pertenece al fondo} \end{cases} \quad (3.5)$$

La máscara permite visualizar qué zonas han sido reconocidas como vegetación y contar el número de píxeles asignados a cada clase. El porcentaje de cubierta vegetal depende directamente de esta clasificación.

Cuando un elemento del fondo se identifica como vegetación se produce un falso positivo. Por el contrario, si una parte de la planta se clasifica como fondo, aparece un falso negativo. Ambos errores modifican el resultado final.

La segmentación binaria basada en color presenta una complejidad computacional reducida, ya que utiliza operaciones sencillas sobre cada píxel. Esta característica facilita su ejecución en dispositivos con recursos limitados, aunque su comportamiento puede verse afectado por las variaciones de iluminación y por la similitud de color entre la vegetación y el fondo.

3.4.2. Umbralización fija y adaptativa

La umbralización clasifica los píxeles comparando una característica determinada con un valor de referencia o umbral. De forma general, puede expresarse mediante la ecuación (3.6):

$$M(x, y) = \begin{cases} 1, & S(x, y) > T \\ 0, & S(x, y) \leq T \end{cases} \quad (3.6)$$

donde $S(x, y)$ representa la característica analizada y T es el umbral utilizado para separar ambas clases.

La característica puede ser el valor de un canal, la diferencia entre varios canales o el resultado de un índice de color. El umbral determina a partir de qué valor el píxel se considera vegetación.

En la umbralización fija se aplica el mismo valor a todas las imágenes. Se trata de un procedimiento rápido, sencillo y fácil de reproducir. Sin embargo, puede ofrecer resultados distintos cuando cambian las condiciones de captura, ya que la iluminación modifica los valores RGB de la vegetación y del fondo.

Los métodos adaptativos ajustan el umbral utilizando información obtenida de la imagen o de una referencia previa. Esta adaptación puede mejorar

la separación cuando las condiciones no son constantes, aunque depende de que la información empleada siga siendo representativa.

Existen también métodos automáticos como el de Otsu, que selecciona el umbral a partir de la distribución de los valores de la imagen [17]. Su comportamiento es adecuado cuando las clases están claramente diferenciadas, pero puede empeorar cuando sus valores se solapan.

3.4.3. *Detección mediante dominancia verde*

Un criterio básico para identificar vegetación en una imagen RGB consiste en comprobar la condición de dominancia verde definida en la ecuación (3.7):

$$G > R \text{ y } G > B \quad (3.7)$$

Esta condición identifica píxeles en los que el verde tiene una mayor intensidad relativa. Su principal ventaja es su sencillez, ya que únicamente requiere comparar los tres canales de cada píxel.

La condición puede complementarse con una diferencia mínima entre el canal verde y los otros componentes. De este modo, no basta con que (G) sea ligeramente superior, sino que debe existir un margen suficiente para considerar que el píxel presenta una dominancia verde significativa.

Un margen pequeño puede incluir tonos grises u oscuros, mientras que uno demasiado elevado puede excluir vegetación afectada por sombras o con tonalidades menos intensas. Por ello, la elección del criterio influye directamente en el resultado.

Además, el color verde no es exclusivo de las plantas. Fondos artificiales, recipientes u otros objetos pueden cumplir las mismas condiciones. Del mismo modo, algunas partes de la vegetación pueden presentar colores amarillentos o marrones y no ser detectadas.

La dominancia verde constituye, por tanto, un método sencillo y de bajo coste computacional, pero puede necesitar umbrales adicionales o referencias del fondo para mejorar su comportamiento.

3.4.4. Índices de vegetación visibles y Exceso de Verde

Los índices de vegetación visibles combinan los canales RGB para aumentar la diferencia entre la vegetación y el fondo. El resultado es un único valor por píxel sobre el que puede aplicarse posteriormente un umbral.

En algunos índices, los canales se normalizan mediante las relaciones recogidas en la ecuación (3.8):

$$r = \frac{R}{R + G + B}, \quad g = \frac{G}{R + G + B}, \quad b = \frac{B}{R + G + B} \quad (3.8)$$

Las componentes normalizadas representan la contribución relativa de cada canal respecto a la intensidad total del píxel y reducen parcialmente la influencia de los cambios globales de luminosidad.

Uno de los índices más utilizados es el Exceso de Verde, conocido como ExG, definido en la ecuación (3.9):

$$ExG = 2g - r - b \quad (3.9)$$

Este índice asigna un peso mayor al canal verde y resta las contribuciones roja y azul. Los píxeles con una mayor presencia relativa de verde presentan, por tanto, valores superiores.

ExG fue propuesto para facilitar la separación entre vegetación, suelo y residuos agrícolas en diferentes condiciones [18]. No produce directamente una clasificación binaria, sino un valor que debe compararse con un umbral.

El resultado final depende tanto del índice como del procedimiento utilizado para seleccionar dicho umbral. Este puede establecerse de forma fija, adaptarse a la imagen o calcularse mediante un método automático.

También se han propuesto otros índices visibles, como ExGR o CIVE, que modifican la ponderación de los canales para mejorar la separación frente a determinados fondos [19], [20]. Sin embargo, no existe un índice que proporcione siempre el mejor resultado. Su eficacia depende del tipo de vegetación, la cámara, el fondo y las condiciones de iluminación.

3.4.5. Segmentación mediante el espacio de color HSV

El espacio de color HSV representa los colores mediante tres componentes: el tono (*Hue*), la saturación (*Saturation*) y el valor o brillo (*Value*). El tono identifica el tipo de color, la saturación indica su grado de pureza o vivacidad y el valor representa el nivel de brillo del píxel [12].

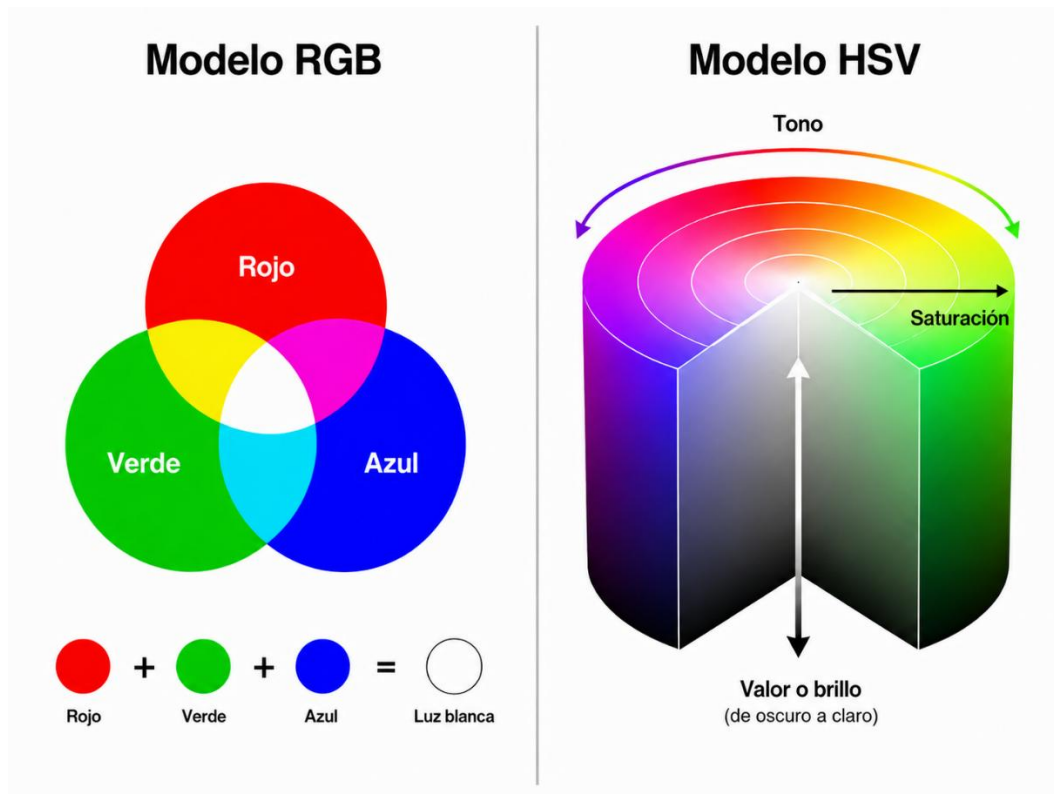


Fig. 3.3. Representación conceptual de los modelos de color RGB y HSV.

Fuente: elaboración propia mediante IA generativa.

A diferencia del espacio RGB, HSV permite separar parcialmente la información asociada al color de la relacionada con la intensidad luminosa. Esta característica resulta especialmente útil para la segmentación de vegetación, ya que puede definirse un intervalo de tonos correspondiente al verde para aislar la zona de interés. Junto a este intervalo, pueden establecerse valores mínimos de saturación y brillo, de forma que las zonas grises, oscuras o poco definidas queden excluidas de la clasificación como vegetación.

La complejidad computacional de este método es moderada, lo que permite aplicarlo en sistemas embebidos con recursos limitados. No obstante, su resultado sigue dependiendo en buena medida de las condiciones en las que se

captura la imagen. Las sombras, los reflejos, los cambios de iluminación, las hojas secas o la propia variedad de tonalidades verdes presentes en la vegetación pueden dar lugar a falsos positivos o falsos negativos en la clasificación.

En este Trabajo Fin de Grado, HSV se emplea como uno de los métodos seleccionables para estimar el porcentaje de cubierta vegetal. Los intervalos y umbrales concretos utilizados se describen en el capítulo de Metodología.

3.4.6. Calibración y limitaciones de la segmentación por color

La calibración consiste en obtener información previa sobre la escena para utilizarla durante la clasificación. En los métodos basados en color puede emplearse para caracterizar el fondo, la vegetación o ambos.

Una referencia del fondo permite conocer los valores que aparecen normalmente cuando no existe vegetación. Los píxeles de una nueva imagen pueden compararse con dicha referencia y clasificarse como vegetación cuando presentan una diferencia suficiente.

También pueden utilizarse imágenes con vegetación para establecer valores representativos de las plantas. La comparación entre ambas referencias permite adaptar el criterio de separación a una escena determinada.

Las referencias pueden obtenerse sobre toda la imagen o sobre zonas concretas. Esta segunda opción resulta útil cuando la iluminación o el fondo no son uniformes en toda la superficie.

Sin embargo, una calibración puede perder validez si cambian de forma importante la iluminación, la posición de la cámara o las características del fondo. Por este motivo, debe considerarse una forma de adaptación al entorno y no una solución que elimine todos los errores. En una primera fase de este Trabajo Fin de Grado se implementó un procedimiento de calibración mediante imágenes de referencia con y sin vegetación. No obstante, tras las pruebas preliminares, este procedimiento se descartó de la versión definitiva del sistema. Su funcionamiento y las razones que motivaron esta decisión se describen en el capítulo de Metodología.

La principal limitación de los métodos RGB es su dependencia de las condiciones de captura. Los cambios de iluminación modifican los valores de los canales, mientras que las sombras, los reflejos y las zonas saturadas pueden alterar el color aparente de los objetos.

También pueden producirse errores cuando el fondo contiene elementos con tonalidades parecidas a las plantas. En sentido contrario, las hojas secas, los tallos o determinadas etapas de desarrollo pueden no mostrar una dominancia verde suficiente.

La normalización de los canales y los índices visibles reducen parcialmente la influencia de la intensidad, pero no corrigen por completo las variaciones provocadas por sombras, reflejos o cambios en la fuente de luz [11], [20].

Otra limitación es que estos métodos suelen analizar los píxeles de forma individual. No consideran necesariamente la forma, la textura ni la relación espacial entre regiones próximas, por lo que pueden aparecer píxeles aislados o bordes irregulares.

A pesar de estas limitaciones, la segmentación basada en color continúa siendo una alternativa adecuada cuando se busca un equilibrio entre sencillez, coste computacional y capacidad de diferenciación. Su aplicación resulta especialmente interesante en sistemas embebidos, siempre que el método se evalúe con imágenes representativas y se conozcan las condiciones que pueden afectar al resultado.

En síntesis, la vegetación puede segmentarse mediante comparaciones entre canales, umbralización, índices visibles como ExG y espacios de color como HSV. La calibración mediante referencias de color constituye otra posible estrategia para adaptar el criterio de clasificación a una escena determinada y fue implementada durante una etapa inicial del desarrollo. Sin embargo, tras su evaluación preliminar, se decidió no mantenerla en la versión definitiva, que utiliza los algoritmos basados en Exceso de Verde normalizado y HSV. La implementación inicial de los tres procedimientos y el proceso seguido para

seleccionar los dos algoritmos definitivos se explican en el capítulo de Metodología.

3.5. Microcontroladores y sistemas embebidos de visión

Los sistemas embebidos son dispositivos electrónicos diseñados para realizar una función específica dentro de un sistema mayor. A diferencia de un ordenador de propósito general, suelen estar optimizados para ejecutar un conjunto limitado de tareas, con restricciones de memoria, capacidad de procesamiento, consumo energético y tamaño físico [21].

Dentro de estos sistemas, los microcontroladores integran en un único circuito elementos como unidad de procesamiento, memoria, periféricos de entrada y salida e interfaces de comunicación. Esta integración permite desarrollar dispositivos compactos capaces de adquirir datos mediante sensores, procesarlos y comunicarse con otros equipos.

Los sistemas embebidos pueden implementarse sobre plataformas con diferentes niveles de capacidad. Los microcontroladores se emplean habitualmente en tareas de adquisición, control y procesamiento de baja o moderada complejidad. Cuando se requiere una mayor capacidad de cálculo, pueden utilizarse ordenadores de placa única, procesadores especializados o dispositivos lógicos programables, que permiten trabajar con algoritmos de visión más exigentes, aunque normalmente presentan un mayor consumo, coste y complejidad. La elección de la plataforma depende de los requisitos de procesamiento, memoria, conectividad, consumo energético y tiempo de respuesta de la aplicación [21].

En aplicaciones de monitorización, los microcontroladores se utilizan para leer sensores, ejecutar algoritmos sencillos, gestionar comunicaciones y controlar actuadores. Cuando se incorpora una cámara, el volumen de información aumenta de forma considerable, ya que una imagen contiene una gran cantidad de píxeles. Por este motivo, el procesamiento de imágenes en microcontroladores requiere seleccionar cuidadosamente la resolución, el formato y el algoritmo empleado.

3.5.1. *Sistemas embebidos aplicados al procesamiento de imágenes*

La visión embebida consiste en integrar la captura y el análisis de imágenes en un dispositivo de recursos limitados. Este enfoque permite obtener información visual sin depender necesariamente de un ordenador externo para realizar todo el procesamiento.

Este tipo de sistemas se emplea en aplicaciones como detección de objetos, lectura de códigos, seguimiento de procesos, inspección visual o monitorización ambiental. En el ámbito agrícola, puede utilizarse para observar el crecimiento de las plantas, detectar cambios visibles o estimar variables relacionadas con la superficie vegetal [6].

El principal reto de la visión embebida es que las imágenes requieren más memoria y más operaciones que otros tipos de datos procedentes de sensores simples. Por ejemplo, una medida de temperatura puede almacenarse en pocos bytes, mientras que una imagen RGB necesita almacenar información de color para cada píxel.

Por esta razón, en dispositivos embebidos suelen emplearse métodos de procesamiento relativamente sencillos, como operaciones por píxel, comparaciones entre canales, umbrales o índices de color. Estos métodos presentan menor complejidad que los modelos de visión artificial más avanzados y pueden ejecutarse con menos recursos computacionales.

3.5.2. *Familia ESP32*

La familia ESP32 está formada por microcontroladores de bajo consumo orientados a aplicaciones conectadas. Una de sus características principales es la integración de conectividad inalámbrica, lo que facilita su uso en sistemas IoT, redes de sensores y dispositivos de monitorización remota.

El microcontrolador ESP32 integrado en la M5Stack TimerCamera-F pertenece a esta familia y combina capacidad de procesamiento, conectividad Wi-Fi y diferentes periféricos para la conexión con sensores y otros dispositivos externos [3]. Estas características permiten utilizarlo como núcleo de sistemas capaces de adquirir datos, procesarlos y enviarlos a otros equipos o plataformas.

En proyectos de visión embebida, el interés de este tipo de microcontroladores se encuentra en la posibilidad de integrar varias funciones en un mismo dispositivo. El sistema puede gestionar la cámara, procesar datos básicos, ofrecer una interfaz local y comunicarse mediante Wi-Fi.

No obstante, aunque el ESP32 integrado en la M5Stack TimerCamera-F permite capturar imágenes, procesarlas localmente y transmitir resultados, sigue siendo un dispositivo con limitaciones frente a un ordenador. La memoria disponible, la velocidad de acceso a los datos y la carga producida por la comunicación inalámbrica condicionan el tipo de procesamiento que puede realizarse.

Por ello, la selección del microcontrolador debe relacionarse con el objetivo de la aplicación. Para tareas de análisis ligero, como la segmentación por color o el cálculo de porcentajes a partir de una imagen, puede ser una plataforma adecuada. Para algoritmos más complejos o modelos de visión artificial de mayor tamaño, puede ser necesario optimizar el procesamiento o recurrir a hardware más potente.

3.5.3. Cámaras integradas y sensores de imagen

Un sensor de imagen convierte la luz recibida en señales eléctricas que posteriormente se transforman en una imagen digital. En los módulos de cámara utilizados con microcontroladores, el sensor suele incorporar funciones internas relacionadas con la exposición, el balance de blancos, el formato de salida y la compresión.

La información proporcionada por la cámara puede entregarse en distintos formatos. Algunos formatos, como RGB565 o RGB888, permiten acceder directamente a la información de color de los píxeles. Otros, como JPEG, reducen el tamaño de los datos mediante compresión. La elección del formato afecta tanto a la memoria necesaria como a la facilidad para aplicar el procesamiento posterior.

La documentación de Espressif indica que el controlador de cámara permite trabajar con formatos como JPEG, RGB565, RGB888 y YUV, dependiendo del sensor utilizado y de su configuración [14], [15]. En aplicaciones

donde se necesita analizar el color de cada píxel, es habitual convertir la imagen a un formato RGB antes de aplicar los criterios de clasificación.

La integración de una cámara en un sistema embebido requiere tener en cuenta la resolución de captura, los búferes de memoria, la velocidad de transferencia, la disponibilidad de memoria externa y la compatibilidad entre el sensor y la placa utilizada. Estos factores condicionan la fluidez de la captura y la viabilidad del procesamiento.

En este TFG, los detalles concretos del módulo de cámara, la resolución empleada y el flujo de conversión de imagen se describirán en la Metodología, ya que forman parte de la implementación del prototipo. En el Marco Teórico únicamente se recogen los fundamentos necesarios para comprender las decisiones de diseño.

3.5.4. Restricciones del procesamiento en microcontroladores

El procesamiento de imágenes en microcontroladores está condicionado principalmente por la memoria disponible, el tiempo de ejecución y el consumo de recursos. Cada imagen ocupa una cantidad de memoria proporcional a su resolución y al número de bytes utilizados por píxel. Por tanto, aumentar la resolución puede mejorar el nivel de detalle, pero también incrementa el volumen de datos que debe manejarse.

El tiempo de procesamiento depende del número de píxeles analizados y de la complejidad de las operaciones aplicadas a cada uno. Los algoritmos que recorren la imagen píxel a píxel pueden ser adecuados si las operaciones son simples, pero pueden volverse lentos al aumentar la resolución o al incorporar cálculos más complejos.

Estas restricciones justifican el uso de estrategias como reducir la resolución, analizar solo una región de interés, capturar en un formato comprimido o emplear algoritmos sencillos basados en comparaciones e índices de color. Todas estas decisiones buscan mantener un equilibrio entre la calidad de la información obtenida, la velocidad de procesamiento y la viabilidad de ejecución en el dispositivo.

La documentación del controlador de cámara de Espressif advierte de la carga que supone trabajar con determinados formatos sin compresión y recomienda tener en cuenta el uso de PSRAM en aplicaciones de cámara [15]. Esto resulta especialmente relevante cuando el sistema debe combinar captura de imágenes, procesamiento y conectividad inalámbrica.

En consecuencia, los sistemas embebidos de visión no deben evaluarse únicamente por la precisión del algoritmo, sino también por su capacidad para ejecutarse de forma estable en el hardware disponible. En el caso de la estimación de cubierta vegetal, los métodos RGB de baja complejidad permiten obtener una aproximación del porcentaje de vegetación sin recurrir a equipos de mayor coste o a procesamiento externo continuo.

El apartado siguiente analiza la diferencia entre realizar el procesamiento en el propio dispositivo y trasladarlo a un servidor o plataforma externa, cuestión directamente relacionada con el diseño de sistemas IoT de monitorización.

3.6. Procesamiento local y procesamiento remoto

En los sistemas de monitorización basados en sensores o imágenes, el tratamiento de la información puede realizarse en distintos puntos de la arquitectura. De forma general, se distingue entre procesamiento local, cuando los datos se analizan en el propio dispositivo que los adquiere, y procesamiento remoto, cuando se envían a un ordenador externo, servidor o plataforma en la nube para su análisis.

El procesamiento local se apoya en el concepto de edge computing, cuyas ventajas principales son la reducción del volumen de datos enviados, la disminución de la dependencia de una conexión permanente y la obtención de una respuesta más inmediata [7].

En el caso de las imágenes, esta diferencia resulta especialmente relevante. Una imagen completa contiene mucha más información que una medida numérica procedente de un sensor simple. Por ello, cuando el objetivo consiste en obtener un valor concreto, como un porcentaje de cubierta vegetal,

puede resultar más eficiente calcularlo en el propio dispositivo y enviar únicamente el resultado.

No obstante, el procesamiento local está limitado por la memoria y la capacidad de cálculo del hardware. Esto obliga a emplear algoritmos ajustados a los recursos disponibles y a controlar aspectos como la resolución, el formato de imagen y el número de píxeles analizados. Por este motivo, los métodos basados en comparaciones entre canales, umbrales o índices de color resultan adecuados para sistemas embebidos.

El procesamiento remoto, por el contrario, permite utilizar equipos con mayor capacidad para ejecutar algoritmos complejos, analizar imágenes de mayor resolución o almacenar grandes volúmenes de datos. Su principal inconveniente es que requiere transmitir más información y depende en mayor medida de la disponibilidad de red.

En muchas aplicaciones se combinan ambos enfoques. El dispositivo realiza la adquisición y un procesamiento básico, mientras que la plataforma remota almacena, representa o analiza los resultados. Esta arquitectura híbrida permite aprovechar la autonomía del procesamiento local y las posibilidades de seguimiento que ofrecen las plataformas IoT.

La Fig. 3.4 resume esta arquitectura híbrida, en la que la adquisición y el procesamiento principal se realizan en el dispositivo, mientras que la plataforma remota permite almacenar y visualizar los resultados.

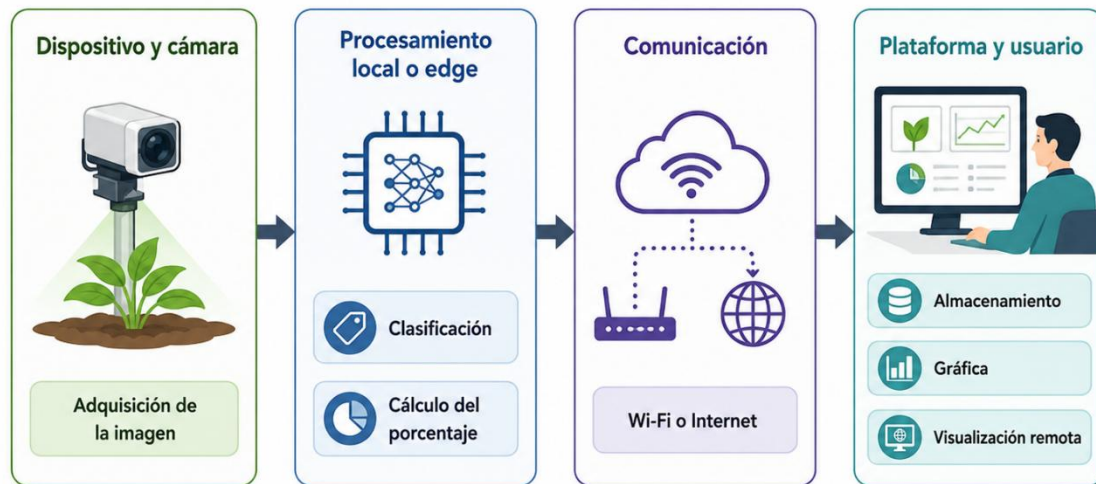


Fig. 3.4. Arquitectura conceptual de un sistema IoT con procesamiento local de imágenes.

Fuente: elaboración propia mediante IA generativa.

En este TFG, el procesamiento local permite obtener la estimación del porcentaje de cubierta vegetal directamente a partir de la imagen capturada. El envío posterior de los resultados a una plataforma IoT complementa esta función con almacenamiento y visualización remota.

3.7. Internet de las Cosas y monitorización remota

El Internet de las Cosas, conocido como IoT, hace referencia a la conexión de objetos físicos a redes de comunicación para que puedan recoger, intercambiar y utilizar datos. Estos dispositivos pueden incorporar sensores, capacidad de procesamiento y conectividad, permitiendo observar variables del entorno y transmitir información a otros equipos o plataformas [22].

En un sistema IoT, el valor principal no se encuentra solo en la medida obtenida, sino en la posibilidad de integrarla dentro de una arquitectura de monitorización. De este modo, los datos pueden almacenarse, representarse y consultarse a lo largo del tiempo, lo que facilita el seguimiento de la variable estudiada.

Como se muestra en la Fig. 3.4, una arquitectura IoT básica integra el dispositivo de adquisición, el procesamiento local, la comunicación y la plataforma de aplicación. En primer lugar, el dispositivo de adquisición, formado por sensores o cámaras que recogen información del entorno. En segundo lugar,

la comunicación, encargada de transmitir los datos mediante tecnologías como Wi-Fi, Bluetooth, redes móviles o Ethernet. Finalmente, la plataforma o aplicación permite almacenar, visualizar o analizar la información recibida [22], [23].

En aplicaciones agrícolas, los dispositivos IoT pueden emplearse para medir variables como temperatura, humedad, radiación, estado del suelo o información obtenida mediante imágenes. La conexión a red permite consultar estos datos sin acceder físicamente al lugar donde se encuentra instalado el sistema. Esto resulta útil cuando se pretende realizar un seguimiento periódico o comparar mediciones tomadas en distintos momentos.

Las plataformas IoT actúan como punto de unión entre los dispositivos conectados y el usuario. Normalmente permiten registrar equipos, recibir datos, almacenarlos y representarlos mediante paneles de visualización. Su uso evita tener que desarrollar desde cero toda la infraestructura de almacenamiento y consulta, lo que resulta especialmente útil en prototipos y trabajos académicos.

Entre las funciones habituales de estas plataformas se encuentran la gestión de dispositivos, la creación de recursos, el almacenamiento de series temporales y la configuración de paneles de visualización. Estos elementos permiten comprobar si los datos se reciben correctamente y observar la evolución de una variable mediante indicadores o gráficas.

Thingier.io es una plataforma IoT orientada a la conexión, gestión y monitorización de dispositivos. Su documentación indica que permite conectar dispositivos, almacenar datos y crear interfaces de visualización [24]. Además, dispone de data buckets, que permiten guardar mediciones como series temporales para su posterior consulta o representación [25]. La plataforma también permite crear dashboards para visualizar los datos mediante indicadores y gráficas [26]. La interfaz de la plataforma y el estado de un dispositivo conectado se muestran en la Fig. 3.5.

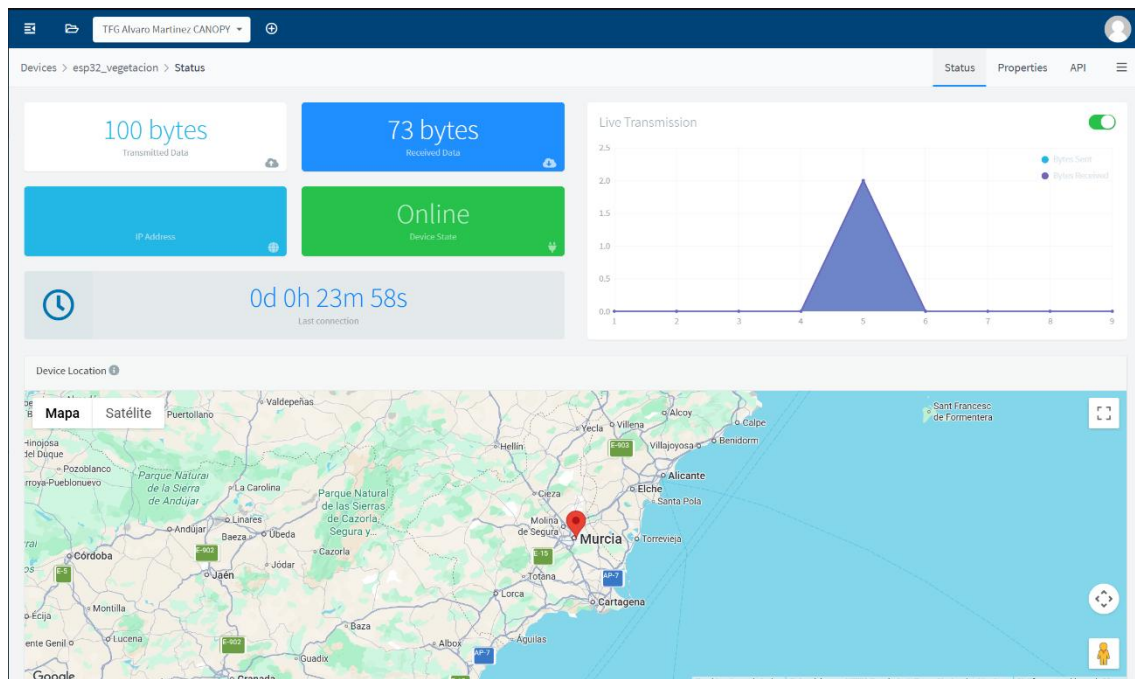


Fig. 3.5. Interfaz de Thingy.io con el estado del dispositivo conectado.

Fuente: elaboración propia a partir de Thingy.io.

En el contexto de este TFG, una plataforma como Thingy.io permite complementar el procesamiento realizado en el dispositivo con almacenamiento y visualización remota. De esta forma, el porcentaje de cubierta vegetal calculado a partir de la imagen puede enviarse a la plataforma y consultarse posteriormente, sin depender únicamente de la visualización local del prototipo.

3.8. Estado del arte de la estimación de cubierta vegetal

El estado del arte permite conocer las soluciones y métodos utilizados previamente para estimar la cubierta vegetal mediante imágenes. En este ámbito se han empleado diferentes enfoques, desde herramientas de análisis manual hasta aplicaciones móviles, software de procesamiento de imagen, teledetección y sistemas conectados.

De forma general, la tendencia ha sido sustituir progresivamente la estimación visual por procedimientos digitales capaces de proporcionar resultados más objetivos y repetibles. Las cámaras RGB han tenido un papel importante en esta evolución, ya que permiten obtener imágenes de forma sencilla y aplicar criterios de clasificación basados en el color de los píxeles.

3.8.1. *Canopeo*

Canopeo es una de las herramientas más conocidas para la estimación de la fracción de cubierta vegetal verde. Fue desarrollada con el objetivo de medir de forma rápida la cubierta vegetal a partir de imágenes y vídeos capturados con dispositivos convencionales [1].

Su funcionamiento se basa en el análisis de imágenes RGB. La herramienta clasifica los píxeles como vegetación o fondo utilizando relaciones entre los canales rojo, verde y azul, junto con el índice de Exceso de Verde. A partir de esta segmentación calcula el porcentaje de píxeles identificados como vegetación respecto al total de la imagen analizada [1].

Una de sus principales ventajas es la facilidad de uso. Permite obtener una estimación rápida sin necesidad de equipamiento especializado, lo que la convierte en una referencia útil para trabajos relacionados con cubierta vegetal. Además, al estar basada en imágenes RGB, resulta adecuada para comparar resultados con otros métodos de segmentación visible.

No obstante, Canopeo depende de la calidad de la imagen capturada y de las condiciones de iluminación. Sombras, reflejos, fondos con colores similares o diferencias en el encuadre pueden afectar al resultado. Además, su uso habitual está orientado al análisis individual de imágenes capturadas o seleccionadas por el usuario, por lo que no constituye por sí solo un sistema fijo de monitorización autónoma y periódica.

3.8.2. *VitiCanopy*

VitiCanopy es una aplicación desarrollada para estimar parámetros relacionados con la arquitectura del dosel vegetal en viñedo. A partir de imágenes tomadas con teléfonos inteligentes o tabletas, la aplicación permite obtener información sobre vigor, porosidad y características estructurales de la cubierta vegetal [27].

Aunque su ámbito de aplicación es más específico que el de Canopeo, resulta relevante porque muestra cómo una cámara convencional puede utilizarse para obtener información agronómica mediante procesamiento digital

de imágenes. En este caso, el análisis se orienta principalmente al estudio del dosel de la vid y a la caracterización de su estructura.

La principal aportación de VitiCanopy es demostrar la utilidad de aplicaciones móviles para realizar mediciones de campo de forma accesible. Sin embargo, al igual que otras soluciones basadas en teléfonos inteligentes, requiere la intervención del usuario para la captura y gestión de las imágenes. Por tanto, no está planteada como un sistema embebido autónomo que capture, procese y envíe resultados de manera periódica.

3.8.3. *Teledetección mediante satélites*

La teledetección mediante satélites permite observar de forma periódica superficies agrícolas extensas sin necesidad de instalar equipos directamente sobre el terreno. Los sensores multiespectrales registran información en varias bandas del espectro visible y del infrarrojo cercano, a partir de las cuales pueden calcularse índices de vegetación como NDVI y analizarse la distribución general del cultivo, así como su evolución a lo largo del tiempo.

Una de las fuentes utilizadas es Sentinel-2, perteneciente al programa europeo Copernicus. Sus sensores registran trece bandas espectrales con resoluciones espaciales de 10, 20 y 60 m, y ofrecen una frecuencia de revisita aproximada de cinco días. Los datos están disponibles sin coste para usuarios particulares, científicos y comerciales a través del Copernicus Data Space Ecosystem [28]. Sin embargo, el usuario no puede elegir el instante exacto de la captura y la presencia de nubes puede limitar la disponibilidad de imágenes realmente útiles. Además, en las bandas de mayor resolución, cada píxel de 10×10 m representa una superficie de 100 m^2 , lo que restringe el nivel de detalle cuando se quiere analizar una zona pequeña.

Cuando se requiere una mayor resolución espacial, pueden emplearse imágenes comerciales con tamaños de píxel inferiores a un metro. En este caso, el nivel de detalle mejora de forma notable, pero también lo hacen el coste y las condiciones de contratación. Como referencia, SkyWatch publica precios desde 2,50 dólares por km^2 para imágenes ópticas archivadas de resolución media y desde 10 dólares por km^2 para imágenes de alta resolución. Cuando se solicita

una nueva captura de alta resolución, el precio puede partir de 12 dólares por km² [29]. En la captura programada de imágenes ópticas de alta resolución, SkyWatch indica una superficie mínima de 25 km²; con un precio inicial de 12 dólares por km², el coste mínimo sería de aproximadamente 300 dólares [30]. No obstante, antes de optar por esta alternativa conviene valorar la resolución disponible, la frecuencia de revisita, la posible presencia de nubosidad, la superficie mínima exigida y el precio final del servicio.

3.8.4. *Monitorización mediante drones y sensores multiespectrales*

Los drones permiten obtener imágenes de parcelas agrícolas con un nivel de detalle superior al ofrecido habitualmente por los satélites. En función de la altura de vuelo y del sensor empleado, pueden alcanzarse resoluciones de pocos centímetros por píxel, lo que facilita el análisis de zonas concretas y la identificación de variaciones dentro de una misma parcela [31]. Además, el usuario puede seleccionar el momento de la captura y utilizar cámaras RGB, multiespectrales o térmicas según el tipo de información que se quiera obtener.

Las imágenes obtenidas pueden combinarse para generar ortomosaicos, mapas de la parcela e índices de vegetación como NDVI o NDRE [32]. Sin embargo, este procedimiento requiere planificar el vuelo, calibrar el sensor cuando sea necesario y procesar posteriormente un número elevado de imágenes. También implica disponer del dron, el sistema de captura, las baterías y los programas de planificación y tratamiento de datos. Su utilización está condicionada, además, por la meteorología, la autonomía del equipo y la normativa aérea. En la categoría abierta europea, las operaciones deben realizarse generalmente por debajo de 120 m y respetar las restricciones aplicables a la zona de vuelo, la proximidad a personas y los requisitos de registro y formación correspondientes [33]. Por tanto, los drones resultan especialmente útiles para inspecciones puntuales y para generar mapas detallados de parcelas completas, aunque su uso periódico requiere organizar nuevas campañas de vuelo y procesar las imágenes obtenidas.

3.8.5. *Otras herramientas y trabajos previos*

Además de Canopeo y VitiCanopy, existen trabajos que utilizan software general de análisis de imágenes para estimar la cubierta vegetal. Herramientas como ImageJ permiten segmentar imágenes, aplicar umbrales y calcular áreas, aunque suelen requerir una mayor intervención manual y conocimientos previos de procesamiento de imagen [34].

Estas herramientas ofrecen flexibilidad para probar diferentes criterios de segmentación y revisar de forma detallada los resultados. Sin embargo, normalmente requieren que el usuario cargue y procese las imágenes de manera individual, por lo que su utilización puede resultar menos práctica cuando se pretende realizar un seguimiento periódico y automatizado.

3.8.6. *Métodos empleados en investigaciones anteriores*

Los métodos más habituales para estimar cubierta vegetal mediante imágenes RGB se basan en la segmentación de la imagen en dos clases: vegetación y fondo. Para ello se emplean comparaciones entre canales, umbrales fijos, umbrales automáticos o índices de vegetación visibles.

Los métodos basados en dominancia verde aprovechan que, en muchas situaciones, la vegetación presenta un valor del canal verde superior al rojo y al azul. Esta aproximación es sencilla y tiene bajo coste computacional, aunque puede fallar cuando existen sombras, vegetación poco verde o fondos con colores similares.

Numerosos trabajos han estudiado el uso de índices visibles calculados a partir de los canales RGB. Entre ellos se encuentran el Exceso de Verde, ExG, y otros índices como ExGR o CIVE, empleados para aumentar la diferencia entre la vegetación y el fondo en imágenes agrícolas. Estos métodos han mostrado utilidad en distintas aplicaciones, especialmente cuando se acompañan de un proceso adecuado de umbralización [18]–[20]. Además, resultan especialmente interesantes en soluciones de bajo coste, ya que no requieren cámaras o sensores multiespectrales.

Otra alternativa empleada en estudios previos es la segmentación en el espacio HSV, que permite identificar tonos verdes mediante límites de tono, saturación y brillo. Su coste computacional moderado facilita su uso en sistemas embebidos, aunque sigue siendo sensible a la iluminación y a las sombras [35].

También se han aplicado métodos automáticos de selección de umbral, como Otsu, y técnicas más avanzadas basadas en aprendizaje automático. Estas últimas pueden mejorar la clasificación en escenarios complejos, pero requieren conjuntos de datos representativos, entrenamiento previo y una validación específica. Por este motivo, en sistemas embebidos de bajo coste siguen siendo habituales los métodos basados en reglas simples e índices de color.

3.8.7. Comparación de las soluciones existentes

A partir de las alternativas descritas anteriormente, la Tabla 3.1 resume sus principales características, campos de aplicación, ventajas y limitaciones.

Solución	Escala de aplicación	Modalidad de seguimiento	Ventaja principal	Limitación principal
Canopeo	Zona pequeña	Puntual, con intervención del usuario	Estimación rápida con imágenes RGB	Requiere la intervención del usuario
VitiCanopy	Zona concreta de viñedo	Puntual, con intervención del usuario	Análisis específico del dosel vegetal	Aplicación especializada e intervención del usuario
ImageJ	Variable	Manual	Gran flexibilidad de segmentación	Requiere conocimientos y procesamiento posterior

Solución	Escala de aplicación	Modalidad de seguimiento	Ventaja principal	Limitación principal
Teledetección y satélites	Superficies extensas	Periódica	Seguimiento de grandes áreas	Resolución, nubosidad y momento de captura
Drones	Parcela completa	Programable	Resolución espacial muy alta	Vuelo, coste y procesamiento de imágenes
Cámara RGB fija y conectada	Zona concreta	Automática	Captura y seguimiento periódico	Dependencia de la iluminación y campo de visión limitado

Tabla 3.1. Comparativa de soluciones y enfoques para la estimación de cubierta vegetal.

Fuente: elaboración propia a partir de la revisión bibliográfica.

La comparación muestra que no existe una única solución válida para todos los escenarios. La observación visual, las aplicaciones móviles y los programas de análisis de imágenes resultan útiles para mediciones puntuales, aunque exigen la intervención del usuario. Los satélites permiten hacer un seguimiento periódico de superficies extensas, pero presentan limitaciones asociadas a la resolución, la nubosidad y el momento de captura. Los drones ofrecen un mayor nivel de detalle y permiten planificar los vuelos, aunque requieren equipamiento específico, procesamiento posterior y una mayor planificación. Por su parte, las cámaras RGB fijas y conectadas están orientadas al seguimiento automático de una zona concreta, si bien sus resultados pueden verse afectados por la iluminación y por el campo de visión establecido.

3.8.8. Ventajas, limitaciones y retos pendientes

Los trabajos revisados muestran que la cubierta vegetal puede estimarse mediante distintas tecnologías, desde la observación visual y el análisis de imágenes RGB hasta la teledetección con satélites, drones y sensores multiespectrales. Entre estas alternativas, el análisis de imágenes RGB se presenta como una opción viable, accesible y ampliamente utilizada, ya que

permite obtener una medida cuantitativa mediante cámaras convencionales en aplicaciones en las que no es necesario recurrir a información multiespectral.

Sin embargo, también existen limitaciones comunes. La iluminación, las sombras, el fondo, la resolución de la imagen y el criterio de segmentación influyen directamente en el porcentaje obtenido. Por ello, no basta con aplicar un algoritmo de clasificación, sino que es necesario comprobar su comportamiento en condiciones reales de captura.

Otro aspecto relevante es el grado de automatización. Muchas herramientas permiten analizar imágenes de forma sencilla, pero requieren que el usuario capture, seleccione o procese manualmente cada fotografía. Esto puede ser suficiente para mediciones puntuales, pero resulta menos adecuado cuando se desea realizar un seguimiento periódico desde una posición fija.

En este contexto, los sistemas embebidos con cámara y conectividad IoT representan una línea de interés. Estos sistemas pueden integrar captura, procesamiento local y envío de resultados en un mismo prototipo. No obstante, deben equilibrar la precisión del análisis con las restricciones de memoria, tiempo de ejecución y comunicación propias del hardware utilizado.

A partir de esta revisión, se identifica como oportunidad el desarrollo de una solución que combine la sencillez de los métodos RGB con la autonomía de un dispositivo embebido conectado. Esta idea sirve de base para el planteamiento del presente TFG, cuya aportación específica se sintetiza en el apartado siguiente.

3.9. Síntesis del estado del arte y posicionamiento del proyecto

El análisis realizado en este marco teórico permite situar el presente Trabajo Fin de Grado dentro de un contexto que combina agricultura de precisión, procesamiento digital de imágenes, sistemas embebidos e Internet de las Cosas. La estimación del porcentaje de cubierta vegetal es una variable útil para observar el desarrollo visible de la vegetación y comparar su evolución a lo largo del tiempo.

Los métodos tradicionales de estimación visual son sencillos y rápidos, pero pueden verse afectados por la subjetividad del observador. Frente a ellos, los métodos basados en imágenes RGB permiten aplicar criterios de clasificación más objetivos y repetibles. Sin embargo, su resultado depende de factores como la iluminación, las sombras, el fondo de la escena, la resolución de la imagen y el criterio de segmentación empleado.

Las soluciones revisadas abarcan desde aplicaciones como Canopeo y VitiCanopy y programas generales de análisis de imágenes hasta sistemas de teledetección mediante satélites y drones. Las primeras permiten obtener información cuantitativa a partir de cámaras convencionales, aunque suelen requerir la intervención del usuario para capturar o procesar las imágenes. Por su parte, los satélites y los drones permiten estudiar superficies más amplias, pero presentan otras exigencias relacionadas con la resolución, la planificación, el coste o el procesamiento posterior.

A partir de esta revisión, se identifica como línea de interés el desarrollo de sistemas compactos capaces de integrar en un mismo dispositivo la captura de imágenes, el procesamiento local y la comunicación de resultados. Esta aproximación resulta especialmente adecuada cuando se desea realizar una monitorización periódica de una zona concreta sin depender continuamente de un ordenador externo.

El presente TFG se posiciona dentro de esta línea mediante el diseño de un prototipo basado en la M5Stack TimerCamera-F, un módulo que integra un microcontrolador ESP32 y un sensor de imagen OV3660 [3]. El sistema se plantea como una solución de bajo coste y complejidad moderada, orientada a estimar el porcentaje de cubierta vegetal mediante técnicas de segmentación RGB ejecutadas en el propio dispositivo.

La aportación prevista no consiste en sustituir a herramientas profesionales ni a sistemas de teledetección avanzados, sino en estudiar la viabilidad de una alternativa embebida, accesible y conectada. Para ello, el prototipo integra la captura de imágenes, la clasificación de píxeles, el cálculo del porcentaje de cubierta vegetal y la comunicación de las mediciones a una plataforma IoT.

En consecuencia, el marco teórico permite justificar tanto la elección de imágenes RGB como el uso de un microcontrolador conectado. Los apartados siguientes desarrollarán los objetivos del trabajo y la metodología seguida para implementar, probar y evaluar el sistema propuesto.

4. OBJETIVOS

El presente apartado recoge el objetivo general y los objetivos específicos del Trabajo Fin de Grado. Su finalidad es definir de manera clara las metas del proyecto y orientar el diseño, la implementación y la validación del sistema desarrollado.

4.1. Objetivo general

Diseñar, implementar y validar un sistema basado en la M5Stack TimerCamera-F, un módulo que integra un microcontrolador ESP32 y un sensor de imagen OV3660, capaz de capturar imágenes en formato JPEG, convertirlas al formato RGB888, procesarlas localmente para estimar el porcentaje de cubierta vegetal y enviar los resultados obtenidos a una plataforma IoT para su almacenamiento y visualización remota.

4.2. Objetivos específicos

- Analizar los fundamentos del procesamiento digital de imágenes RGB, la segmentación de vegetación y el uso de sistemas embebidos conectados aplicados a la estimación de cubierta vegetal.
- Seleccionar y configurar el hardware necesario para la captura de imágenes, el procesamiento local y la comunicación inalámbrica del sistema.
- Desarrollar el software del dispositivo para la adquisición de imágenes, su tratamiento local y el cálculo del porcentaje de cubierta vegetal.
- Implementar una interfaz web local que permita capturar imágenes, visualizar resultados, configurar parámetros básicos, seleccionar la región de interés, elegir el algoritmo de detección y ajustar los umbrales de segmentación.
- Integrar el sistema con una plataforma IoT para posibilitar el envío, almacenamiento y consulta remota de las mediciones obtenidas.

- Evaluar el funcionamiento del prototipo mediante pruebas experimentales en distintas condiciones de captura y comparar los resultados obtenidos con otros métodos o herramientas de referencia.

5. METODOLOGÍA

Este capítulo describe la metodología seguida para diseñar, implementar y validar el prototipo. Se presentan la arquitectura del sistema, la configuración del hardware y del software, el procesamiento de las imágenes, la integración de las interfaces y el diseño de las pruebas experimentales.

5.1. Enfoque metodológico y fases del trabajo

La metodología seguida en este Trabajo Fin de Grado adopta un enfoque de desarrollo secuencial por fases, combinado con un proceso iterativo de implementación y validación. El trabajo se orientó al desarrollo de un prototipo funcional capaz de estimar el porcentaje de cubierta vegetal mediante el procesamiento local de imágenes RGB. Al tratarse de un proyecto de carácter práctico, no se limitó al estudio teórico del problema, sino que incluyó la selección de componentes, la programación del dispositivo, la integración de interfaces de usuario y la realización de pruebas experimentales para comprobar el funcionamiento del sistema.

El desarrollo se ha organizado en cinco fases principales. En la primera fase se analizaron los fundamentos necesarios para abordar el problema, incluyendo la estimación de cubierta vegetal, los métodos de segmentación basados en imágenes RGB y el uso de sistemas embebidos conectados. Esta revisión permitió orientar el proyecto hacia una solución de bajo coste, con capacidad de procesamiento local, y sirvió como base para definir la arquitectura general del sistema, que se presenta en el apartado 5.2.

A continuación, en la segunda fase, se seleccionó y configuró el hardware necesario, así como el entorno de desarrollo. En esta etapa se definieron la placa utilizada, el sensor de cámara, la memoria disponible, la conectividad inalámbrica y las herramientas de programación. Estos aspectos se desarrollan en los apartados 5.3 a 5.5.

La tercera fase se centró en el desarrollo del software del prototipo. En esta etapa se implementó la adquisición de imágenes, la conversión al formato necesario para el análisis, la selección de una región de interés, el ajuste de los

parámetros de segmentación y los algoritmos encargados de clasificar los píxeles para estimar el porcentaje de cubierta vegetal. Este desarrollo se explica con más detalle en los apartados 5.6 y 5.7.

La cuarta fase se dedicó a la visualización y comunicación de los resultados. Para ello se implementó una interfaz web local, accesible desde un navegador, y se integró la comunicación con Thinger.io para enviar, almacenar y consultar los resultados de forma remota. Además, esta integración permite controlar determinados parámetros del sistema a distancia. En esta fase también se incorporó un modo de funcionamiento automático orientado a realizar mediciones periódicas. Estos elementos se describen en los apartados 5.8 a 5.10.

Por último, la quinta fase consistió en el diseño de las pruebas experimentales. Estas pruebas se diseñaron para comprobar el comportamiento del sistema en distintas condiciones de captura y analizar la repetibilidad de las mediciones. El planteamiento de estas pruebas se recoge en el apartado 5.11, mientras que la documentación y publicación del código se describen en el apartado 5.12. Los resultados obtenidos se presentan en el capítulo 6.

5.2. Arquitectura general del sistema

El sistema desarrollado se ha organizado en tres bloques funcionales: adquisición de imagen, procesamiento local y visualización y comunicación de resultados. La Fig. 5.1 muestra el flujo general del sistema, desde la captura de la escena vegetal hasta la consulta local mediante la interfaz web y la monitorización y el control remoto a través de Thinger.io.

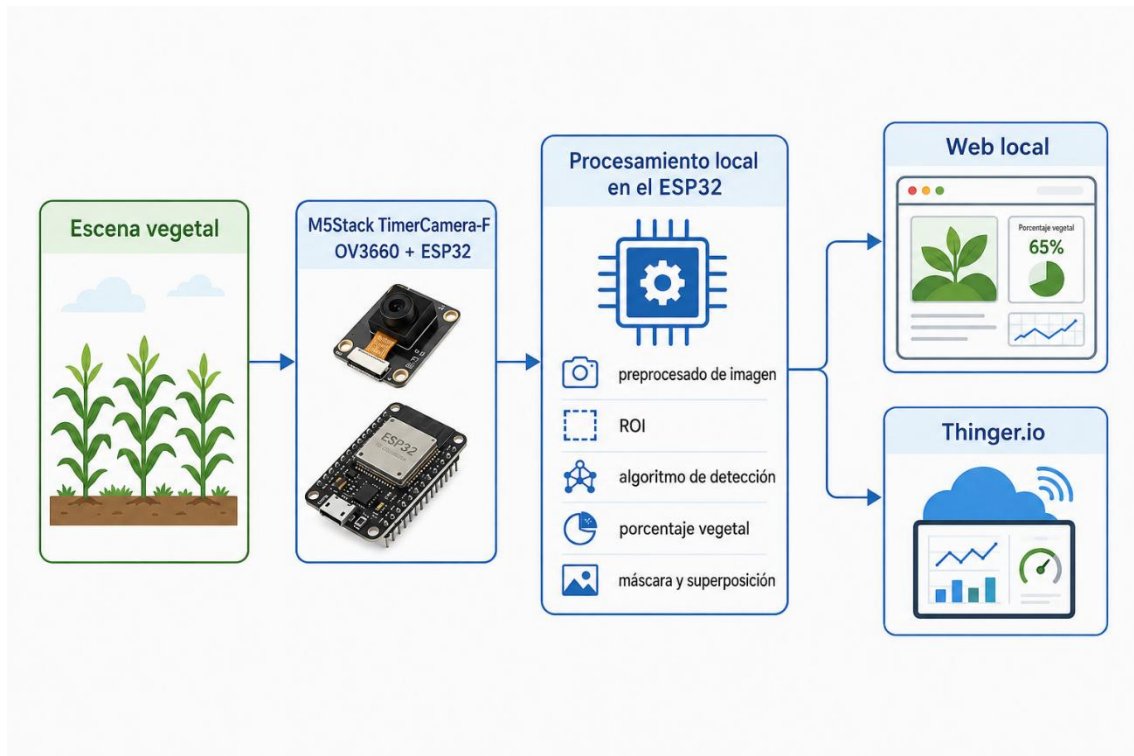


Fig. 5.1. Esquema general de la arquitectura del sistema.

Fuente: elaboración propia.

El primer bloque corresponde a la adquisición de imagen. La escena que se desea analizar, formada por una zona con vegetación y fondo, se captura mediante la cámara integrada en el dispositivo. Esta imagen constituye la entrada del sistema y sirve de base para estimar qué proporción de la región analizada corresponde a cubierta vegetal.

El segundo bloque corresponde al procesamiento local, que se ejecuta en el microcontrolador. A partir de la imagen capturada, el sistema la prepara para su análisis, selecciona la región de interés y aplica el procedimiento de clasificación de píxeles para diferenciar la vegetación del fondo. El resultado de este proceso es el porcentaje de cubierta vegetal, calculado como la relación entre los píxeles identificados como vegetación y el total de píxeles evaluados en la región seleccionada.

El tercer bloque corresponde a la visualización y comunicación de resultados. El dispositivo dispone de una interfaz web local accesible desde un navegador, desde la que el usuario puede interactuar con el sistema, configurar parámetros y consultar los resultados obtenidos. Además, los resultados

numéricos y las variables de estado se envían a la plataforma Thinger.io, lo que permite almacenar las mediciones, consultarlas de forma remota y disponer de funciones de control sobre determinados parámetros del sistema. De este modo, el sistema combina el procesamiento local de la imagen con la monitorización y el control remoto a través de una plataforma IoT, en línea con el planteamiento descrito en el apartado 3.6.

5.3. Hardware utilizado y montaje del prototipo

El prototipo desarrollado se ha basado en una M5Stack TimerCamera-F, seleccionada por integrar en un único módulo un microcontrolador ESP32, un sensor de imagen OV3660, conectividad Wi-Fi, memoria PSRAM y una batería. Esta integración permite capturar imágenes, procesarlas localmente y comunicar los resultados sin depender de un ordenador externo.

El sensor de imagen integrado en la placa es el OV3660, un sensor CMOS en color con una resolución de hasta 3 Mpx. La cámara incorpora una óptica con un campo de visión de 120 grados, que permite abarcar una zona amplia de la escena. La integración de estos elementos en una placa compacta resulta adecuada para un sistema embebido, donde los recursos disponibles son más limitados que en un ordenador convencional.

La placa dispone de conectividad Wi-Fi, empleada tanto para acceder a la interfaz web local como para enviar datos a la plataforma Thinger.io. Además, incorpora memoria PSRAM, utilizada para almacenar temporalmente los fotogramas y los datos generados durante el análisis. En cuanto a la alimentación, la TimerCamera-F cuenta con una batería integrada de 270 mAh, lo que permite realizar pruebas sin mantener el dispositivo conectado de forma permanente a una fuente externa.

El montaje utilizado durante las pruebas consistió en colocar la cámara orientada hacia la escena vegetal y fijarla sobre un trípode, con el fin de mantener una posición estable entre capturas. Esta disposición reduce la variabilidad producida por cambios de encuadre, distancia u orientación, algo importante cuando se comparan mediciones realizadas en distintas sesiones.



Fig. 5.2. Montaje físico del prototipo y disposición utilizada para la captura de las escenas vegetales.

Fuente: elaboración propia.

En la Fig. 5.2. se muestra el prototipo utilizado durante las pruebas experimentales. En ella puede observarse la M5Stack TimerCamera-F montada sobre el trípode y orientada hacia la escena analizada. Este montaje constituye la base física sobre la que se desarrollan los apartados posteriores de adquisición de imagen, procesamiento local, comunicación y validación del sistema.

5.4. Software y entorno de desarrollo

El desarrollo del software del prototipo se ha realizado mediante Arduino IDE, un entorno de programación ampliamente utilizado en placas basadas en ESP32. Este entorno permite escribir, compilar y cargar el código directamente en el microcontrolador, además de facilitar el uso de librerías específicas para la cámara, la conexión Wi-Fi, el servidor web local y la comunicación con plataformas IoT.

El lenguaje empleado ha sido C/C++ dentro del entorno Arduino, ya que permite trabajar directamente con los recursos del microcontrolador y con las librerías disponibles para el hardware seleccionado. En Arduino IDE se seleccionó el perfil de placa M5TimerCAM / M5Stack-Timer-CAM, correspondiente al módulo utilizado.

Para el desarrollo se emplearon distintas librerías relacionadas con la gestión de la cámara, la conversión de imágenes, la conectividad Wi-Fi, el servidor web local y la integración con Thinger.io. Entre las principales se encuentran `esp_camera.h`, `img_converters.h`, `WiFi.h`, `WebServer.h`, `ThingerESP32.h` y `ThingerConsole.h`. También se utilizaron `esp_sleep.h` y `esp_timer.h` para el funcionamiento con Deep Sleep y la gestión temporal, y `Preferences.h` para almacenar determinados parámetros de configuración de forma persistente.

Además del entorno Arduino IDE, se utilizó la plataforma Thinger.io para la comunicación remota con el dispositivo y el almacenamiento de las mediciones. Su integración se describe con más detalle en el apartado 5.9.

La interfaz web local se desarrolló como parte del propio código del dispositivo y se accede a ella mediante un navegador desde un equipo conectado a la misma red Wi-Fi. Esta interfaz permite interactuar con el sistema de forma directa, realizar capturas, modificar parámetros de funcionamiento y consultar los resultados obtenidos sin necesidad de instalar una aplicación específica.

Durante el desarrollo se trabajó mediante versiones sucesivas del código, incorporando nuevas funcionalidades de forma progresiva. La versión empleada

para la redacción de esta memoria corresponde al archivo `codigo_base_funcional_v9.ino`. Esta forma de trabajo permitió comprobar cada mejora de manera independiente y mantener un control claro de la evolución del prototipo.

5.5. Configuración de la cámara, memoria y conectividad

La configuración inicial del sistema se centró en preparar los elementos necesarios para capturar imágenes, almacenar temporalmente los datos y establecer la comunicación inalámbrica del dispositivo. Esta fase es importante porque el funcionamiento de la cámara, la disponibilidad de memoria y la conexión Wi-Fi condicionan el resto del sistema.

En primer lugar, se configuró el sensor de imagen OV3660 integrado en la M5Stack TimerCamera-F. Para el desarrollo del proyecto se seleccionó una resolución VGA de 640×480 píxeles, al considerarse una opción equilibrada entre calidad de imagen y consumo de memoria. Una resolución superior permitiría obtener más detalle de la escena, pero también aumentaría el tamaño de los fotogramas y la duración del procesamiento de la imagen, algo que puede ser problemático en un sistema embebido. Por este motivo, la resolución VGA se consideró adecuada para las pruebas de estimación de cubierta vegetal.

La imagen capturada por la cámara se obtiene inicialmente en formato JPEG, lo que reduce el tamaño del fotograma y facilita su visualización desde la interfaz web. Sin embargo, para analizar los valores de color de cada píxel y aplicar los algoritmos de segmentación es necesario trabajar con los valores RGB de la imagen. Por ello, el sistema realiza una conversión al formato RGB888, proceso que se describe con más detalle en el apartado 5.6.

Otro aspecto importante fue el uso de la memoria PSRAM incorporada en la placa. El tratamiento de imágenes requiere almacenar temporalmente los fotogramas y los datos derivados del análisis, lo que puede superar la capacidad de la memoria interna del microcontrolador. Los 8 MB de PSRAM disponibles en la TimerCamera-F permiten trabajar con imágenes en resolución VGA y realizar el análisis local de forma adecuada para los objetivos del proyecto. El programa comprueba la disponibilidad de esta memoria durante la inicialización y, cuando

está disponible, configura dos búferes de cámara alojados en la PSRAM. Si no se detecta, utiliza un único búfer en la memoria interna del ESP32.

Además de la cámara y la memoria, se configuró la conectividad Wi-Fi del dispositivo en modo estación, de manera que el ESP32 se conecta a una red inalámbrica existente. Esta conexión permite integrar el sistema dentro de una red local y acceder a la interfaz web mediante la dirección IP asignada al dispositivo, siempre que el equipo del usuario esté conectado a la misma red. La conectividad Wi-Fi también es la base de la comunicación con Thinger.io. A través de esta conexión, el dispositivo puede enviar los resultados calculados y otros parámetros de funcionamiento, lo que permite su almacenamiento y visualización remota. La configuración de esta integración se desarrolla en el apartado 5.9.

En la Tabla 5.1 se recogen los principales parámetros de configuración del sistema, que sirven como referencia para los apartados posteriores.

Parámetro	Configuración utilizada
Sensor de imagen	OV3660
Resolución máxima del sensor	3 Mpx
Resolución de captura	VGA, 640 × 480 píxeles
Formato inicial de captura	JPEG
Formato de análisis	RGB888
Memoria PSRAM	8 MB integrados
Conectividad	Wi-Fi 802.11 b/g/n, 2,4 GHz

Parámetro	Configuración utilizada
Acceso local	Web local vía IP
Plataforma IoT	Thingier.io
Tipo de procesamiento	Local en el ESP32

Tabla 5.1. Principales parámetros de configuración de la cámara, la memoria y la conectividad.

Fuente: elaboración propia.

Las configuraciones descritas en este apartado establecen la base técnica sobre la que se apoyan las fases siguientes: la adquisición y el preprocesado de imagen, la estimación de cubierta vegetal, la interfaz web local y el envío de datos a la plataforma IoT.

5.6. Captura, preprocesado de imagen y región de interés

Una vez configuradas la cámara, la memoria y la conectividad del dispositivo, el siguiente paso consiste en capturar la imagen y prepararla para su análisis. Esta fase es común a los algoritmos de estimación utilizados posteriormente, por lo que se explica antes de describir los métodos concretos de clasificación de píxeles.

La Fig. 5.3 resume el flujo general seguido desde la captura de la imagen hasta la generación de los resultados.

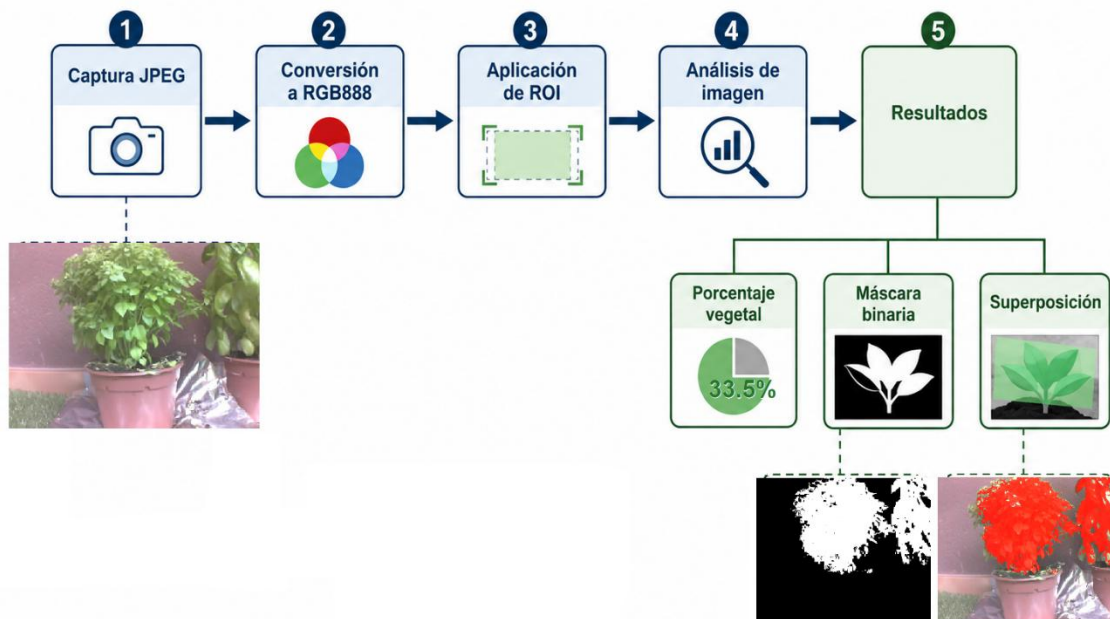


Fig. 5.3. Flujo de captura, procesamiento y generación de resultados del sistema.

Fuente: elaboración propia

El proceso comienza con la captura de una imagen mediante el sensor OV3660. Para evitar utilizar un fotograma que ya estuviera disponible en el búfer de la cámara, el programa obtiene y descarta una primera captura. A continuación, espera 150 ms y solicita una nueva imagen, que es la utilizada para el análisis. La imagen se obtiene inicialmente en formato JPEG, ya que este formato reduce el tamaño del fotograma y facilita su almacenamiento temporal y su visualización desde la interfaz web. Sin embargo, para analizar la imagen píxel a píxel es necesario trabajar con los valores de color de cada punto de la escena. Por ello, el sistema convierte la imagen al formato RGB888, en el que cada píxel queda representado por sus tres componentes de color: rojo, verde y azul.

Tras esta conversión, el sistema dispone de la información necesaria para recorrer la imagen y aplicar los algoritmos de segmentación. No obstante, en lugar de analizar siempre el fotograma completo, se define una región de interés o ROI. Esta región delimita la zona concreta de la imagen que será evaluada,

permitiendo centrar el análisis en la parte de la escena que contiene la vegetación y reduciendo la influencia de zonas no relevantes.

En la configuración predeterminada del sistema, la ROI se sitúa en una zona central de la imagen mediante un recorte del 20 % en cada uno de sus cuatro lados. Esta selección permite excluir zonas periféricas del encuadre que puedan incluir bordes, sombras o partes del montaje. La región analizada puede modificarse desde la interfaz web local y desde Thinger.io, permitiendo utilizar la imagen completa o adaptar la ROI a las características de la escena.

Durante el recorrido de la ROI, el sistema contabiliza el número total de píxeles evaluados y el número de píxeles clasificados como vegetación. La decisión de si un píxel pertenece o no a vegetación depende del algoritmo seleccionado, basado en Exceso de Verde normalizado o en el espacio de color HSV, que se describen en el apartado 5.7. A partir de estos valores, el sistema calcula el porcentaje de cubierta vegetal como la relación entre los píxeles identificados como vegetación y el total de píxeles analizados en la región seleccionada. Después de realizar el cálculo, los resultados inferiores al 0,5 % se fijan en 0 %.

Además del cálculo numérico, el sistema genera una máscara binaria para representar visualmente el resultado de la segmentación. En esta máscara, los píxeles clasificados como vegetación se representan en blanco, mientras que los píxeles considerados como no vegetación se representan en negro. Esta salida permite comprobar de forma visual qué zonas han sido identificadas como cubierta vegetal por el algoritmo.

De forma complementaria, se genera una imagen de superposición sobre la captura original. En esta representación, los píxeles detectados como vegetación se marcan sobre la imagen original en color rojo. Esto facilita la interpretación del resultado y permite comprobar si la clasificación realizada por el sistema coincide razonablemente con la percepción visual de la escena. Esta salida resulta especialmente útil durante las pruebas, ya que ayuda a detectar errores de segmentación y a valorar si los parámetros del algoritmo necesitan ajuste.

El resultado de todo este proceso no se limita a un dato numérico. Junto al porcentaje de cubierta vegetal, el sistema ofrece la máscara binaria y la superposición visual como apoyo para interpretar y validar lo que el algoritmo ha detectado en cada captura.

5.7. Algoritmos de estimación de cubierta vegetal

Una vez capturada y preprocesada la imagen, el sistema aplica un algoritmo de segmentación para clasificar los píxeles de la región de interés como vegetación o no vegetación. Esta clasificación se realiza de forma local en el microcontrolador y es la base para calcular el porcentaje de cubierta vegetal.

El sistema se planteó inicialmente con tres procedimientos de detección: un método de calibración mediante imágenes de referencia, un algoritmo basado en Exceso de Verde normalizado y otro basado en el espacio de color HSV. Durante las pruebas preliminares realizadas en la fase de desarrollo, el método de calibración ofreció resultados claramente insatisfactorios, con clasificaciones inestables y poco coherentes ante cambios de iluminación y de fondo. Por este motivo, se descartó antes de la validación experimental definitiva. Los métodos seleccionables por el usuario quedaron limitados a Exceso de Verde normalizado y HSV. Canopeo se utilizó posteriormente como herramienta externa de comparación durante la validación experimental.

La selección del algoritmo activo se gestiona mediante la variable `detectionMode`, que permite alternar entre los modos `DETECTION_EXG_NORMALIZED` y `DETECTION_HSV_GREEN`. En el código, esta selección se aplica en la función `isVegetationByActiveAlgorithm()`, encargada de decidir qué método se utiliza para clasificar cada píxel durante el análisis. Esta configuración puede modificarse tanto desde la interfaz web local como desde el panel de control de Thinger.io, donde también se ajustan los parámetros principales de cada algoritmo.

El procedimiento común a ambos métodos consiste en recorrer los píxeles incluidos dentro de la ROI, extraer sus valores RGB y comprobar si cumplen los criterios definidos por el algoritmo seleccionado. A partir del número de píxeles clasificados como vegetación y del número total de píxeles evaluados en la

región analizada, se calcula el porcentaje de cubierta vegetal mediante la ecuación **¡Error! No se encuentra el origen de la referencia.!**:

$$P_{cv} = (N_{veg} / N_{ROI}) \times 100 \quad (5.1)$$

Donde, P_{cv} , es el porcentaje de cubierta vegetal, N_{veg} es el número de píxeles clasificados como vegetación y N_{ROI} es el número total de píxeles analizados en la región de interés.

En el código, este cálculo se realiza dentro de la función `performAnalysis()`, donde se actualizan las variables `lastVegPercent`, `lastVegetationPx` y `lastRoiPx`.

5.7.1. Algoritmo basado en Exceso de Verde normalizado

El primer método implementado fue el Exceso de Verde normalizado, descrito en el apartado 3.4.4. Para cada píxel, el código calcula las componentes normalizadas r , g y b según la ecuación (3.8), y obtiene el índice $ExG = 2g - r - b$, de acuerdo con la ecuación (3.9). Si la suma $R + G + B$ es igual a cero, el píxel se clasifica directamente como no vegetación para evitar una división entre cero.

Este proceso se encuentra implementado en la función `isVegetationExgNormalized()`. Una vez calculado el valor de ExG , el píxel se clasifica como vegetación si dicho valor es igual o superior al umbral configurado en la variable `exgThreshold`. Como valor predeterminado, este umbral se estableció en 0,12, aunque puede modificarse desde la interfaz web local y desde Thinger.io para adaptar el comportamiento del algoritmo a distintas condiciones de captura.

5.7.2. Algoritmo basado en HSV

El segundo método implementado se basa en el espacio de color HSV, que representa el color mediante tres componentes: tono, saturación y valor o brillo. El tono indica el tipo de color, la saturación representa su intensidad o pureza, y el valor se relaciona con el brillo. Esta representación permite separar de forma más clara el color dominante de un píxel respecto a su luminosidad.

En el código, la conversión de RGB a HSV se realiza mediante la función `rgbToHsv()`. A partir de los valores R, G y B de cada píxel, esta función calcula el tono en grados, dentro del intervalo de 0° a 360° , y la saturación y el valor en una escala de 0 a 255. Después, la función `isVegetationHsvGreen()` comprueba si el tono se encuentra dentro del rango configurado y si la saturación y el valor son iguales o superiores a sus límites mínimos.

Como valores predeterminados, el rango de tono utilizado para detectar vegetación se estableció entre 80° y 150° , la saturación mínima en 50 y el valor o brillo mínimo en 40, ambos en una escala de 0 a 255. Estos valores se almacenan en las variables `hsvHMin`, `hsvHMax`, `hsvSMin` y `hsvVMin`. Tanto la interfaz web local como el panel de control de Thingier.io permiten modificar estos parámetros, lo que facilita ajustar la detección a diferentes tipos de vegetación, condiciones de iluminación o fondos presentes en la escena.

El uso de HSV aporta una ventaja importante frente al análisis directo en RGB, ya que permite definir la vegetación a partir de un rango de tonos asociado al color verde y filtrar píxeles poco saturados o demasiado oscuros. De esta forma, se reduce la posibilidad de que zonas grises, blancas, negras o con poca información cromática sean clasificadas erróneamente como vegetación.

5.7.3. Comparación inicial entre algoritmos

Los dos algoritmos implementados persiguen el mismo objetivo: identificar los píxeles correspondientes a vegetación dentro de la región de interés y calcular el porcentaje de cubierta vegetal. Sin embargo, cada uno realiza esta clasificación con un criterio diferente. El algoritmo basado en Exceso de Verde normalizado trabaja sobre las componentes RGB normalizadas, mientras que el algoritmo basado en HSV utiliza rangos de tono, saturación y brillo.

El algoritmo de Exceso de Verde normalizado destaca por su sencillez y bajo coste computacional, lo que lo hace adecuado para sistemas embebidos. Por su parte, el algoritmo basado en HSV permite definir de forma independiente el rango de tono y los valores mínimos de saturación y valor. Como contrapartida, requiere convertir previamente cada píxel de RGB a HSV y realizar un mayor número de operaciones.

En esta metodología se mantienen ambos algoritmos con el fin de comparar su comportamiento durante las pruebas experimentales. Esta comparación permite justificar su incorporación desde el punto de vista técnico, mientras que la evaluación práctica de los resultados obtenidos con cada método se desarrolla posteriormente en el capítulo de Resultados.

La implementación completa de los métodos ExG y HSV, así como del mecanismo de selección entre ambos, puede consultarse en el código fuente disponible mediante el enlace incluido en el Anexo 1.

5.8. Interfaz web local

Además del procesamiento local de la imagen, el sistema incorpora una interfaz web accesible desde un navegador. Esta interfaz se ejecuta desde el propio microcontrolador ESP32 y permite interactuar con el prototipo sin necesidad de instalar una aplicación adicional. Para acceder a ella, el dispositivo debe estar conectado a una red Wi-Fi y el usuario debe introducir en el navegador la dirección IP asignada al sistema.

La interfaz web local permite controlar las funciones principales del prototipo durante las pruebas. Desde ella se pueden iniciar análisis manuales, modificar parámetros de configuración y consultar los resultados obtenidos. De esta forma, el usuario puede comprobar el comportamiento del sistema de manera directa y ajustar su funcionamiento según las condiciones de captura.

Desde la página principal se accede a la opción Porcentaje manual, desde la que puede iniciarse un nuevo análisis de la escena. Al hacerlo, el sistema captura la imagen, aplica el preprocesado descrito en el apartado 5.6 y utiliza el algoritmo seleccionado para estimar el porcentaje de cubierta vegetal. Una vez finalizado el proceso, la interfaz muestra los resultados principales: el porcentaje calculado, el número de píxeles clasificados como vegetación y el número total de píxeles analizados dentro de la región de interés.

La interfaz también permite consultar la última imagen capturada por la cámara, así como la máscara binaria y la imagen de superposición generadas

durante el análisis. Estas representaciones, descritas en el apartado 5.6, permiten comprobar visualmente la clasificación realizada por el sistema.

Otra función importante es la configuración de la región de interés. El usuario puede modificar la zona del fotograma que será analizada, ajustando sus límites para centrar el procesamiento en el área más relevante de la escena. Esta opción permite adaptar el sistema a diferentes encuadres o distancias de captura sin tener que modificar el código del dispositivo.

La interfaz local también incluye opciones para seleccionar el algoritmo de detección. El usuario puede alternar entre el método basado en Exceso de Verde normalizado y el método basado en HSV. Además, se pueden ajustar los parámetros principales de cada uno, como el umbral del Exceso de Verde o los rangos de tono, saturación y brillo utilizados en HSV. Esta configuración facilita adaptar la segmentación a diferentes tipos de vegetación, condiciones de iluminación y fondos presentes en la imagen.

Junto a las funciones de análisis y configuración, la interfaz muestra información de estado del dispositivo, como el nivel de batería y el tiempo activo desde el último arranque. Estos datos permiten supervisar el funcionamiento general del prototipo durante las sesiones de prueba y valorar su comportamiento en cada ciclo de medida.

La interfaz web local también incorpora opciones relacionadas con el modo automático, permitiendo iniciar o detener la adquisición periódica de medidas. El funcionamiento de este modo se desarrolla con mayor detalle en el apartado 5.10, dedicado a la monitorización periódica del sistema.

La implementación completa de la interfaz web local puede consultarse en el código fuente disponible mediante el enlace incluido en el Anexo 1.

Esta interfaz se complementa con la integración remota mediante Thinger.io, descrita en el apartado 5.9. Así, el sistema dispone de dos vías de interacción: una local, mediante la dirección IP del dispositivo, y otra remota, mediante la plataforma IoT. Esta doble vía de acceso —control directo desde la interfaz local y supervisión remota a través de Thinger.io— permite operar el prototipo tanto durante las pruebas próximas al dispositivo como en escenarios

de monitorización a distancia. Ambas interfaces actúan sobre la configuración común del sistema, por lo que los cambios realizados desde una de ellas afectan al funcionamiento posterior del dispositivo.

Las pantallas correspondientes a la versión final de la interfaz web local y los resultados obtenidos durante su utilización se muestran y describen en el capítulo de Resultados.

5.9. Integración con Thinger.io

La monitorización remota del sistema se implementó mediante la plataforma Thinger.io. La conexión entre el ESP32 y la plataforma se realiza a través de la red Wi-Fi configurada en el dispositivo y mediante las librerías `ThingerESP32.h` y `ThingerConsole.h`.

En el código se definieron distintos recursos para establecer una comunicación bidireccional con la plataforma. Estos recursos permiten enviar los resultados y las principales variables de estado del sistema, como el porcentaje de cubierta vegetal, el número de píxeles clasificados como vegetación, el total de píxeles analizados, el algoritmo seleccionado, el nivel de batería, el tiempo activo y el estado del modo automático. También permiten recibir órdenes y modificar determinados parámetros de funcionamiento de forma remota.

La implementación de los recursos de comunicación con Thinger.io puede consultarse en el código fuente completo disponible mediante el enlace incluido en el Anexo 1.

Para almacenar el histórico de las mediciones se configuró un bucket denominado `bucket_vegetacion`. El envío se realiza mediante el recurso `medicion`, que agrupa los resultados numéricos y las variables de estado asociadas a cada análisis. De esta manera, los datos pueden conservarse en la plataforma para su posterior consulta y representación temporal.

La integración con Thinger.io permite consultar las mediciones y controlar el sistema de forma remota. Además, el dashboard puede mostrar la última imagen capturada mediante la dirección IP local del ESP32, siempre que el equipo desde el que se accede se encuentre conectado a la misma red Wi-Fi.

Cuando la consulta se realiza desde una red diferente, la imagen no está disponible, aunque sí pueden consultarse los datos y utilizarse las funciones de control.

Las pantallas finales de Thinger.io, incluyendo el estado del dispositivo, el almacenamiento de las mediciones y los dashboards de monitorización y control, se muestran y describen en el capítulo de Resultados.

5.10. Modo automático y monitorización periódica

Junto a la captura manual, el sistema incorpora un modo automático orientado a realizar mediciones periódicas sin intervención directa del usuario. Esta funcionalidad permite que el dispositivo capture imágenes, procese la escena y actualice los resultados de forma repetida, permitiendo realizar un seguimiento temporal de la cubierta vegetal. El funcionamiento automático se implementó mediante dos modalidades: un modo normal y un modo basado en Deep Sleep. El intervalo entre mediciones puede configurarse de forma remota, y el funcionamiento automático puede detenerse para volver a la captura manual.

Para las mediciones diarias se configuró un intervalo de 24 horas. Este intervalo programa una nueva captura aproximadamente 24 horas después de la anterior, pero no sincroniza el dispositivo con una hora concreta ni garantiza condiciones equivalentes de iluminación. En el modo automático normal, el dispositivo permanece encendido, conectado a la red Wi-Fi y accesible tanto desde la interfaz web local como desde Thinger.io. Durante este funcionamiento, el sistema realiza capturas periódicas según el intervalo configurado, analiza la imagen, calcula el porcentaje de cubierta vegetal y envía los datos a la plataforma IoT.

El segundo modo automático incorpora Deep Sleep, un modo de bajo consumo del ESP32 en el que el sistema trabaja por ciclos de activación y reposo: el dispositivo despierta, establece la conexión Wi-Fi, se conecta a Thinger.io, recupera la configuración almacenada, captura una imagen, realiza el procesamiento local, envía los datos calculados y, tras permanecer activo durante un tiempo determinado, vuelve a entrar en Deep Sleep hasta la siguiente medición. Durante el periodo de reposo, el ESP32 permanece en un estado de

bajo consumo y no están activas ni la interfaz web local ni la comunicación con Thinger.io. El código incorpora una ventana despierta definida mediante la variable `awake_window_seconds`, que establece el tiempo durante el cual la placa permanece operativa tras despertar. Esta ventana permite al usuario acceder al dispositivo, modificar parámetros o detener el modo automático antes de que vuelva a entrar en reposo.

Una particularidad de este modo es que, al despertar, el ESP32 reinicia la ejecución del programa desde la función `setup()`. Para evitar que se pierda la configuración modificada durante el funcionamiento, el código utiliza memoria RTC mediante `RTC_DATA_ATTR`. Así se conservan entre ciclos parámetros como el modo automático activo, el intervalo de medida, el estado del Deep Sleep, la ventana despierta, el algoritmo seleccionado, los umbrales de detección, la configuración de la ROI y los contadores de funcionamiento. Además, el código utiliza `autoCaptureCount` para registrar el número de capturas automáticas realizadas y `wakeCount` para contabilizar los despertares desde Deep Sleep. Antes de entrar en reposo, estos valores se copian en `rtcAutoCaptureCount` y `rtcWakeCount`, declaradas mediante `RTC_DATA_ATTR`, para conservarlos entre ciclos. El tiempo activo acumulado durante los ciclos anteriores se almacena en `rtcTotalAwakeSeconds`, mientras que la función `getTotalAwakeSeconds()` suma este valor al tiempo transcurrido desde el despertar actual. De esta forma, puede diferenciarse entre el tiempo activo desde el último arranque y el tiempo activo total acumulado.

5.11. Diseño de las pruebas experimentales y criterios de validación

Una vez desarrollado el prototipo, se planteó un conjunto de pruebas experimentales para comprobar su funcionamiento y valorar la coherencia de las mediciones obtenidas. El objetivo no era solo verificar que cada bloque respondiera por separado, sino comprobar que el sistema completo fuera capaz de capturar imágenes, analizarlas, mostrar resultados y enviar datos de forma coordinada.

El diseño experimental se organizó en varios bloques de prueba. En primer lugar, se comprobó el funcionamiento general del sistema: captura de

imagen, conversión al formato de análisis, aplicación de la región de interés, ejecución del algoritmo seleccionado, cálculo del porcentaje de cubierta vegetal y generación de las salidas visuales, es decir, la máscara binaria y la imagen de superposición.

A continuación, se validó la interfaz web local y parte de la interacción básica con el prototipo. En este caso se revisó el acceso mediante la dirección IP asignada al dispositivo y la realización de capturas manuales. También se comprobó la consulta de la última imagen obtenida, la visualización de la máscara y la superposición, así como la modificación de la región de interés y el ajuste de los parámetros de detección. Esta prueba se planteó para comprobar que los cambios realizados desde la interfaz se aplicaban al análisis posterior.

Otro bloque se centró en la integración con Thinger.io. Para ello se verificó el envío de datos desde el dispositivo, el almacenamiento de las mediciones en el bucket ``bucket_vegetacion`` y la actualización de los widgets de monitorización. Además, se comprobaron los controles remotos incluidos en el dashboard, como la selección del algoritmo, el ajuste de umbrales, la configuración de la ROI, la captura manual y la activación o detención del modo automático.

La comparación entre algoritmos se planteó como una parte específica de la validación. Para ello se analizaron las mismas escenas con los dos métodos implementados, Exceso de Verde normalizado y HSV, manteniendo la misma región de interés y modificando únicamente el algoritmo de detección. Como referencia externa, las mismas imágenes se analizaron también mediante Canopeo. Para facilitar la comparación, se utilizó en cada caso la misma zona de la escena y se registró el porcentaje de cubierta vegetal obtenido con cada método. Esta comparación se planteó como un contraste entre herramientas de segmentación y no como una validación frente a un valor real medido de forma independiente.

En cada caso se registraron el porcentaje de cubierta vegetal, el número de píxeles clasificados como vegetación, el número total de píxeles analizados y una valoración visual de la máscara generada. Para esta comparación se utilizaron escenas sencillas pero representativas, con distintas condiciones de

captura: presencia clara de vegetación, fondo uniforme, zonas con sombras, fondos más complejos y diferentes niveles de cubierta vegetal. Con esta selección se evaluó la respuesta de cada algoritmo ante distintos tipos de escena.

También se realizó una prueba de repetibilidad. En este caso se realizaron varias capturas consecutivas de una misma escena sin modificar la posición del prototipo, la distancia de captura, la fuente de iluminación ni la región de interés. El objetivo fue comprobar si el porcentaje estimado se mantenía estable cuando las condiciones externas no cambiaban de forma significativa.

Por último, se comprobó el funcionamiento del modo automático y su variante con Deep Sleep. En la prueba del modo automático normal se verificó que el sistema pudiera realizar capturas periódicas con los intervalos configurados, actualizar el contador de capturas automáticas y enviar los datos a Thinger.io. En la versión con Deep Sleep, además, se comprobó que el dispositivo fuera capaz de despertar, conectarse a la red Wi-Fi, enviar la medición, conservar la configuración mediante memoria RTC (RTC_DATA_ATTR) y volver posteriormente al estado de bajo consumo. La validación de este modo se limitó a comprobar el funcionamiento del ciclo de activación y reposo.

Los criterios de validación empleados fueron funcionales y cualitativos. Desde el punto de vista funcional, se consideró válido que cada parte del sistema realizara correctamente la tarea prevista: captura, análisis local, cálculo del porcentaje, visualización web, envío de datos a Thinger.io, actualización de widgets y funcionamiento del modo automático. Desde el punto de vista cualitativo, se valoró la coherencia visual entre la escena original y la máscara generada, comprobando si las zonas detectadas como vegetación coincidían razonablemente con lo observado en la imagen.

Para ordenar el proceso, cada prueba se registró en una hoja de cálculo. En ella se anotaron las condiciones de captura, el algoritmo utilizado, la configuración de la ROI, los parámetros principales del método, el porcentaje de cubierta vegetal, el número de píxeles clasificados como vegetación, el total de píxeles analizados y las observaciones relevantes. Este registro permitió

comparar los resultados de forma ordenada y sirvió como base para la redacción del capítulo de Resultados.

Con este diseño de pruebas se buscó validar el sistema desde tres perspectivas: el funcionamiento técnico del prototipo, la coherencia de los algoritmos de estimación y el funcionamiento de las herramientas de visualización y monitorización. Los resultados obtenidos a partir de estas pruebas se presentan y analizan en el capítulo 6.

5.12. Repositorio GitHub y documentación del código

Como parte de la documentación final del proyecto, el código desarrollado se ha organizado en un repositorio público en GitHub. Esto permite conservar la versión final del programa en un entorno accesible y ordenado, sin necesidad de incluir todo el código fuente dentro del cuerpo principal de la memoria.

La Fig. 5.4. muestra la página principal del repositorio creado para este Trabajo Fin de Grado. En el directorio principal se incluyen dos archivos: `codigo_base_funcional_v9.ino`, que contiene la implementación completa del sistema, y `README.md`, destinado a documentar el proyecto y facilitar la consulta y reutilización del código.

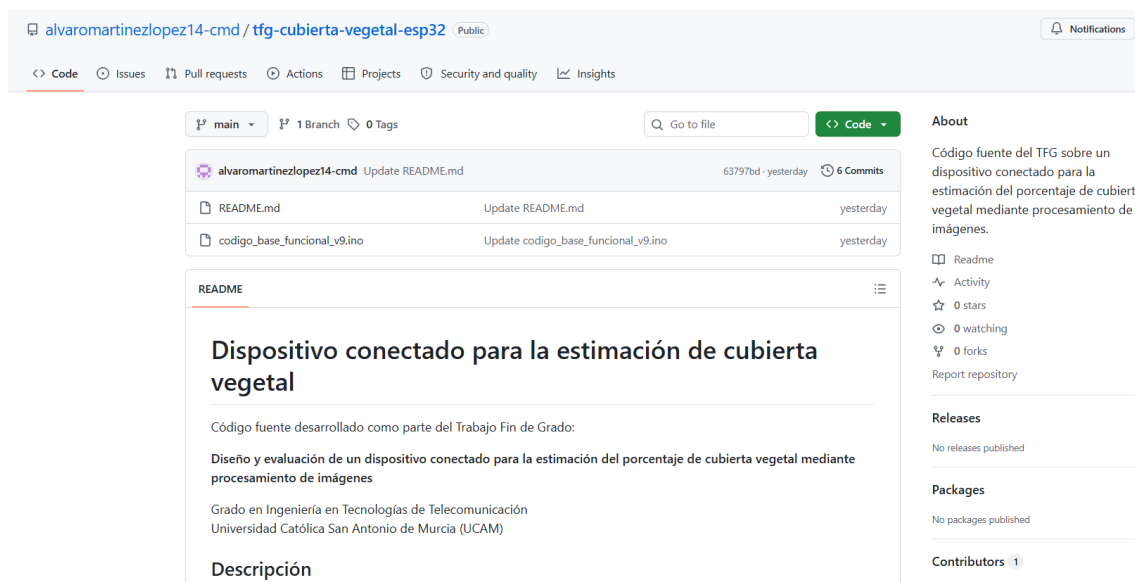


Fig. 5.4. Página principal del repositorio GitHub del proyecto.

Fuente: elaboración propia.

El repositorio presenta una estructura sencilla. El archivo `codigo_base_funcional_v9.ino` reúne el código utilizado en la versión final del prototipo, incluyendo la adquisición y el procesamiento de imágenes, la interfaz web local, la comunicación con Thinger.io y los modos de funcionamiento manual, automático y de bajo consumo. Por su parte, el archivo `README.md` incluye una descripción general del proyecto, el hardware empleado, las librerías necesarias, las indicaciones básicas de configuración y los pasos para cargar el programa en la placa.

Antes de publicar el código, se revisó el archivo para eliminar o sustituir la información sensible. En concreto, se retiró el usuario de Thinger.io, el identificador del dispositivo y las credenciales de acceso a la plataforma. Estos campos se sustituyeron por etiquetas genéricas, de forma que el código pueda reutilizarse o adaptarse sin comprometer datos privados.

De este modo, el repositorio sirve tanto para almacenar la versión final del programa como para complementar la descripción técnica incluida en la memoria. El enlace al repositorio GitHub y la identificación del archivo utilizado se incluyen en el apartado de anexos.

6. RESULTADOS

Este capítulo recopila el comportamiento real del prototipo una vez integradas todas sus funciones, tomando como referencia la versión final del sistema, `codigo_base_funcional_v9.ino`. A diferencia del capítulo anterior, aquí no se vuelve a explicar cómo se construyó cada parte, sino que se comprueba cómo responde el sistema completo durante las pruebas.

Los resultados se organizan en varios bloques. En primer lugar, se presenta el funcionamiento general del prototipo, incluyendo la captura y el procesamiento local de las imágenes, la generación de las salidas visuales, la interfaz web local, la integración con Thingier.io y los modos automáticos de funcionamiento. A continuación, se exponen las pruebas realizadas con escenas vegetales, la comparación entre Exceso de Verde normalizado, HSV y Canopeo y el análisis de la repetibilidad de las mediciones.

Junto al valor numérico de cada análisis, se realizó una revisión visual de la máscara binaria y de la imagen de superposición para comprobar qué zonas de la fotografía habían sido clasificadas como vegetación. Esta revisión permitió identificar aciertos y errores de segmentación que no podían apreciarse únicamente a partir del porcentaje calculado.

6.1. Validación funcional del sistema

Como primera comprobación, se verificó el funcionamiento conjunto del sistema, desde la captura de la imagen hasta la consulta y el envío de los resultados. Esta validación se realizó sobre la versión final del programa.

Durante las pruebas, la M5Stack TimerCamera-F obtuvo imágenes en resolución VGA de 640×480 píxeles. La conversión de JPEG a RGB888 se completó correctamente y permitió ejecutar el análisis de las imágenes mediante los dos algoritmos implementados.

También se comprobó el análisis de la imagen completa y de regiones de interés configuradas por el usuario. Al modificar la ROI, el sistema actualizó el número total de píxeles analizados y recalculó el porcentaje de cubierta vegetal correspondiente a la zona seleccionada.

Los dos algoritmos implementados, Exceso de Verde normalizado y HSV, pudieron seleccionarse sin problemas desde el sistema. En ambos casos se obtuvo el número de píxeles clasificados como vegetación, el número total de píxeles analizados y el porcentaje de cubierta vegetal correspondiente. Además del resultado numérico, se generaron las dos salidas visuales previstas: la máscara binaria, que muestra en blanco los píxeles detectados como vegetación y en negro el resto de la escena, y la imagen de superposición, que marca esas mismas zonas sobre la fotografía original. Esta segunda representación ayuda a comprobar si la detección resulta coherente con lo que se observa a simple vista.

Por último, se comprobó que los resultados podían consultarse desde la interfaz web local y enviarse a Thinger.io mediante el recurso `medicion`. Las mediciones transmitidas quedaron registradas en el bucket `bucket_vegetacion`. También se verificó la actualización de las variables de estado asociadas al modo automático, el tiempo activo, la batería y los ciclos de Deep Sleep. El funcionamiento detallado de estas interfaces y modos se presenta en los apartados posteriores.

Bloque comprobado	Resultado obtenido
Captura de imagen	Captura correcta en resolución VGA 640 × 480
Conversión de formato	Conversión de JPEG a RGB888 para análisis píxel a píxel
Región de interés	Análisis de imagen completa o ROI configurada
Algoritmo ExG	Detección mediante Exceso de Verde normalizado

Bloque comprobado	Resultado obtenido
Algoritmo HSV	Detección mediante tono, saturación y luminosidad
Cálculo del porcentaje	Obtención de píxeles vegetales, píxeles totales y porcentaje
Máscara binaria	Generación de imagen con vegetación en blanco y fondo en negro
Superposición visual	Marcado en rojo de los píxeles detectados sobre la imagen original
Interfaz web local	Consulta de mediciones, imágenes y parámetros desde el navegador
Thingier.io	Envío mediante medición y almacenamiento en bucket_vegetacion
Modo automático	Realización de capturas periódicas y actualización del contador
Deep Sleep	Despertar, restauración de la configuración, captura, envío y retorno al estado de reposo

Tabla 6.1. Resultados de la validación funcional del sistema.

Fuente: elaboración propia.

Con estas comprobaciones queda establecido el punto de partida de las pruebas experimentales: el prototipo captura, procesa, calcula y comunica resultados tal como se había planteado en el diseño. Sobre esta base se desarrollan las pruebas con plantas reales que se describen en los apartados siguientes.

6.2. Resultados obtenidos desde la interfaz web local

La primera forma de consultar los resultados del sistema es a través de la interfaz web local. Desde esta página se puede acceder directamente a la última captura realizada por la cámara y al resultado del análisis, sin necesidad de utilizar herramientas externas. En la parte superior de la interfaz se muestra el porcentaje de cubierta vegetal calculado. Junto a este valor aparecen también los píxeles clasificados como vegetación, el número total de píxeles incluidos en la ROI y el algoritmo utilizado en el análisis, lo que permite comprobar de forma rápida no solo el resultado final, sino también la base sobre la que se ha calculado.

La interfaz web local fue evolucionando durante el desarrollo para facilitar tanto la configuración del sistema como la interpretación de los resultados. Una de las principales mejoras se incorporó en la página de configuración del algoritmo, donde los campos numéricos se complementaron con controles deslizantes y una previsualización de la superposición. De este modo, al modificar el umbral de ExG o los parámetros de HSV, la imagen se vuelve a procesar y la superposición se actualiza de forma prácticamente inmediata sobre la misma captura de referencia. El usuario puede observar cómo aumenta o disminuye el número de píxeles clasificados como vegetación y seleccionar visualmente los parámetros que mejor se ajustan a las condiciones de iluminación y al tipo de escena analizada, sin necesidad de realizar una nueva fotografía con cada modificación.

También se mejoró la configuración de la región de interés mediante una selección visual sobre la imagen capturada. Además de analizar la imagen completa o utilizar una ROI central, la interfaz permite delimitar libremente la zona que se desea procesar y comprobar previamente qué parte de la escena quedará incluida en el cálculo. Esta posibilidad resulta especialmente útil en aplicaciones agrícolas, ya que permite centrar el análisis en una zona concreta del terreno o del cultivo. Por ejemplo, podría seleccionarse únicamente una línea de cultivo, evitando que el césped u otra vegetación situada alrededor alterase el porcentaje de cubierta vegetal obtenido. Con ello, la interfaz deja de ser

únicamente un medio de consulta y se convierte también en una herramienta de configuración y adaptación del análisis a cada escenario.

Debajo de ese bloque de resultados se muestra la imagen original capturada por la M5Stack TimerCamera-F. Esta visualización resulta útil porque permite relacionar el porcentaje obtenido con el encuadre real de la escena y comprobar si la planta queda dentro de la zona de interés. A partir de esa misma captura, la interfaz genera también una superposición sobre la imagen original y una máscara binaria. En la superposición se marcan los píxeles detectados como vegetación, mientras que en la máscara binaria se separa la vegetación del resto de la escena. Estas dos imágenes permiten comprobar de forma visual si el algoritmo está detectando realmente la planta o si incluye zonas del fondo, sombras u otros elementos no deseados.

En la Fig. 6.1 se muestra el resultado completo de un análisis consultado desde la interfaz web local. La captura mostrada corresponde a una escena analizada mediante Exceso de Verde normalizado (umbral ExG = 0,120; recortes de la ROI: izquierda 25 %, derecha 20 %, superior 0 % e inferior 29 %).

45.9%

vegetacion detectada

Píxeles vegetación: 55135**Píxeles ROI total:** 120032**Algoritmo usado:** Exceso de Verde normalizado**Foto analizada:**

Visualización de detección

Los píxeles resaltados son exactamente los clasificados como vegetación en el último análisis.

Superposición sobre la imagen original:**Máscara binaria:****Nueva foto y recalcular**

Fig. 6.1. Resultado de la interfaz web local para una escena analizada mediante ExG.

Fuente: elaboración propia.

La página muestra además los parámetros del algoritmo activo. En el caso de ExG, aparece el umbral empleado para clasificar los píxeles como vegetación; en el caso HSV, se muestran los límites de tono, saturación y luminosidad utilizados. Esta información resulta especialmente útil durante las pruebas, porque un mismo encuadre puede producir resultados distintos si se modifican los umbrales. La interfaz incluye también el botón “Nueva foto y recalcular”, que permite repetir la captura y actualizar el análisis sin reiniciar el dispositivo, algo práctico para ajustar el encuadre, cambiar de algoritmo y comprobar de inmediato cómo varía la detección.

La interfaz web local se convirtió así en una herramienta especialmente útil para revisar el comportamiento del sistema en tiempo real. No solo muestra el porcentaje de cubierta vegetal, sino que permite contrastar la imagen original con la máscara, la superposición y la configuración empleada en cada análisis, lo que facilita detectar rápidamente si el resultado se ajusta a la escena observada.

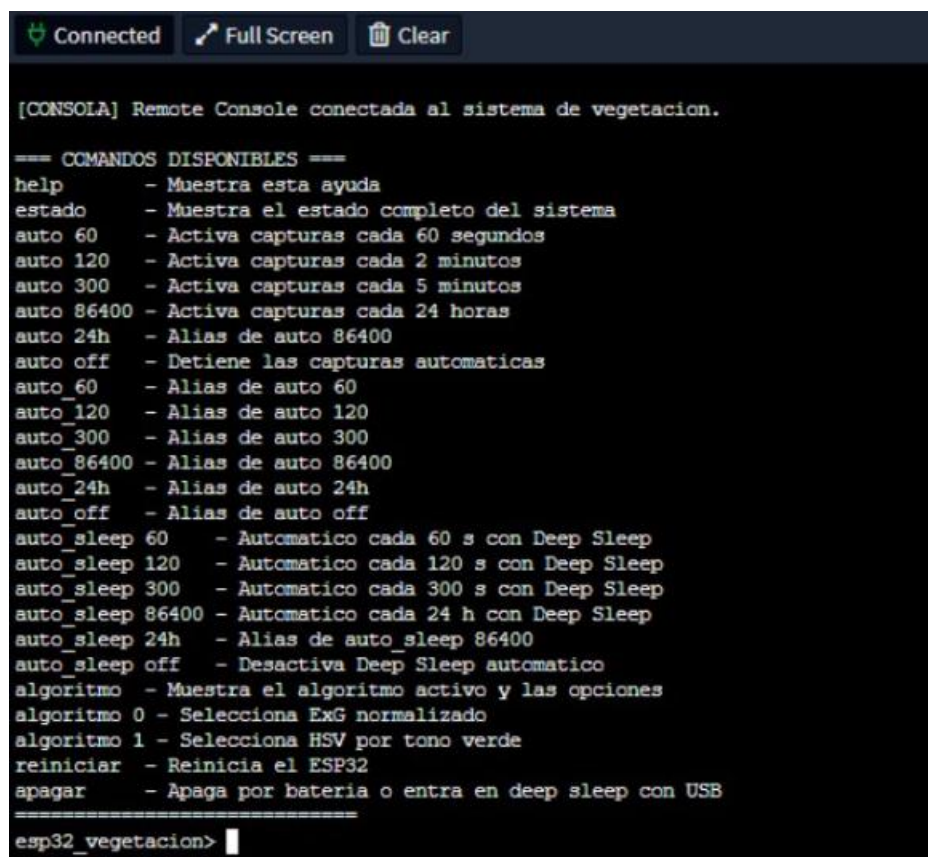
Como funcionalidad adicional de la interfaz web local, se comprobó la configuración y el cambio de la red Wi-Fi mediante Bluetooth Low Energy (BLE). Desde la propia interfaz puede iniciarse el modo de configuración y consultar las indicaciones necesarias para completar el proceso. Mediante la aplicación ESP BLE Provisioning instalada en un teléfono móvil, se seleccionó la red inalámbrica y se introdujeron sus credenciales. Una vez establecida la conexión, estas quedaron almacenadas de forma persistente y el dispositivo se reinició conectado a la nueva red. La prueba confirmó que la conexión Wi-Fi podía configurarse inicialmente o modificarse posteriormente sin cambiar el código ni volver a cargar el programa desde Arduino IDE.

6.3. Resultados de la integración con Thinger.io

Además de la consulta local, el sistema se integró con la plataforma Thinger.io para permitir el seguimiento remoto de las mediciones. Esta parte del proyecto no sustituye a la interfaz web local, ya que no está orientada a revisar la máscara o la superposición, sino a visualizar los principales valores del

sistema desde un panel externo y conservar un registro histórico de las mediciones realizadas.

Durante una primera fase, la interacción remota con el dispositivo se realizó mediante la Remote Console de Thinger.io. Al establecerse la conexión, la propia consola mostraba automáticamente los comandos disponibles y la función asociada a cada uno, pudiendo consultarse de nuevo mediante el comando `help`. Desde esta herramienta fue posible seleccionar el algoritmo, configurar los intervalos de captura, activar o detener el modo automático y el funcionamiento con Deep Sleep, reiniciar el dispositivo y consultar su estado.



```
Connected Full Screen Clear

[CONSOLA] Remote Console conectada al sistema de vegetacion.

=== COMANDOS DISPONIBLES ===
help      - Muestra esta ayuda
estado    - Muestra el estado completo del sistema
auto 60   - Activa capturas cada 60 segundos
auto 120  - Activa capturas cada 2 minutos
auto 300  - Activa capturas cada 5 minutos
auto 86400 - Activa capturas cada 24 horas
auto 24h  - Alias de auto 86400
auto off  - Detiene las capturas automaticas
auto_60   - Alias de auto 60
auto_120  - Alias de auto 120
auto_300  - Alias de auto 300
auto_86400 - Alias de auto 86400
auto_24h  - Alias de auto 24h
auto_off  - Alias de auto off
auto_sleep 60 - Automatico cada 60 s con Deep Sleep
auto_sleep 120 - Automatico cada 120 s con Deep Sleep
auto_sleep 300 - Automatico cada 300 s con Deep Sleep
auto_sleep 86400 - Automatico cada 24 h con Deep Sleep
auto_sleep 24h - Alias de auto_sleep 86400
auto_sleep off - Desactiva Deep Sleep automatico
algoritmo - Muestra el algoritmo activo y las opciones
algoritmo 0 - Selecciona ExG normalizado
algoritmo 1 - Selecciona HSV por tono verde
reiniciar  - Reinicia el ESP32
apagar     - Apaga por bateria o entra en deep sleep con USB
=====
esp32_vegetacion> |
```

Fig. 6.2. Comandos disponibles en la Remote Console de Thinger.io.

Fuente: elaboración propia.

Posteriormente se incorporó un dashboard con botones, selectores, indicadores y gráficas, que permitió realizar estas operaciones mediante una interfaz visual. La Remote Console se mantuvo disponible como herramienta complementaria de control y diagnóstico.

La comunicación se realizó mediante el recurso *medicion*, definido en el dispositivo registrado en *Thinger.io*. A través de este recurso se enviaron el porcentaje de cubierta vegetal, los píxeles clasificados como vegetación, el número total de píxeles analizados, el algoritmo utilizado y distintas variables de estado del sistema. En el dashboard se configuraron widgets para mostrar el porcentaje obtenido, el algoritmo activo, la evolución temporal de las mediciones, el nivel de batería, el tiempo activo, el número de capturas automáticas y el estado de los modos automático y Deep Sleep. Estos indicadores permitieron comprobar desde la plataforma si el sistema estaba realizando mediciones y si los datos se actualizaban correctamente.

En la Fig. 6.3. se muestra el área de monitorización del dashboard durante una de las pruebas.

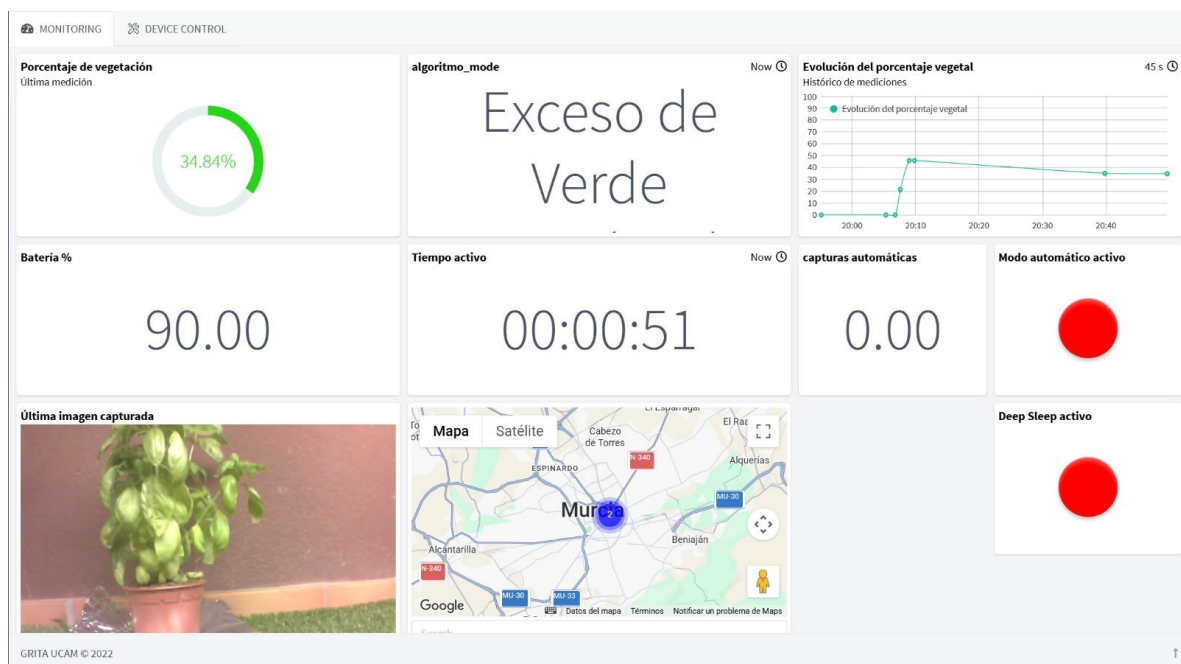


Fig. 6.3. Dashboard de Thinger.io con los principales valores enviados por el prototipo.

Fuente: elaboración propia.

Las mediciones enviadas por el dispositivo se almacenaron en el bucket *bucket_vegetacion*. De esta forma, cada análisis no queda limitado al valor mostrado en el momento de la captura, sino que puede consultarse posteriormente como parte de un histórico. Esto resulta especialmente útil cuando el sistema trabaja en modo automático, porque permite seguir la evolución temporal de las mediciones. Durante las pruebas también se verificó

que los valores enviados desde la M5Stack TimerCamera-F coincidían con los obtenidos en la interfaz web local, lo que confirma que el cálculo se realiza en el propio dispositivo y que Thinger.io actúa como plataforma de visualización y almacenamiento.

El dashboard incorporó también un acceso a la última imagen capturada mediante la dirección IP local del ESP32. Durante las pruebas se comprobó que esta imagen solo podía visualizarse cuando el equipo desde el que se consultaba Thinger.io tenía acceso a la misma red Wi-Fi que el dispositivo. Desde una red diferente, las mediciones y las funciones de control continuaban disponibles, pero la imagen no podía cargarse.

Refresh Last 6 months

Date	auto_capture_count	auto_count	auto_deep_sleep_enabled	auto_interval	auto_interval_sec	auto_mode	awake_window_seconds	battery_percent	battery_voltage	detection_mode	detection_mode_name	exp_threshold	hsv_h_max	hsv_h_min
6/7/2025 17:01:36	6	6	true	120	120	true	60	100	4.20327373193359	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:58:10	5	5	true	120	120	true	60	100	4.19987938441162	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:54:44	4	4	true	120	120	true	60	100	4.20340200708008	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:51:18	3	3	true	120	120	true	60	100	4.207093729166259	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:47:52	2	2	true	120	120	true	60	100	4.208956144470216	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:44:26	1	1	true	120	120	true	60	100	4.217996125453881	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:41:00	28	28	false	60	60	true	60	100	4.199125986382468	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:40:39	27	27	false	60	60	true	60	100	4.2021473264942627	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:39:39	26	26	false	60	60	true	60	100	4.2038397789001485	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:38:39	25	25	false	60	60	true	60	100	4.2000661389558105	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:37:39	24	24	false	60	60	true	60	100	4.205727577205473	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:36:39	23	23	false	60	60	true	60	100	4.202896181640625	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:35:39	22	22	false	60	60	true	60	100	4.20836971903076	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:34:39	21	21	false	60	60	true	60	100	4.205727577205473	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:33:39	20	20	false	60	60	true	60	100	4.2021473264942627	0	Exceso de Verde normalizado	0.11999999731770099	150	80
6/7/2025 16:32:39	19	19	false	60	60	true	60	100	4.205349932180176	0	Exceso de Verde normalizado	0.11999999731770099	150	80

Viewing 0 to 99 items

Refresh Last 6 months

hsv_h_min	hsv_h_max	rci_bottom_percent	rci_enabled	rci_height_px	rci_left_percent	rci_px	rci_right_percent	rci_top_percent	rci_total_px	rci_width_px	total_area
50	40	20	true	288	20	110592	20	20	110592	384	00:07
50	40	20	true	288	20	110592	20	20	110592	384	00:06
50	40	20	true	288	20	110592	20	20	110592	384	00:04
50	40	20	true	288	20	110592	20	20	110592	384	00:03
50	40	20	true	288	20	110592	20	20	110592	384	00:01
50	40	20	true	288	20	110592	20	20	110592	384	00:00
50	40	20	true	288	20	110592	20	20	110592	384	00:30
50	40	20	true	288	20	110592	20	20	110592	384	00:25
50	40	20	true	288	20	110592	20	20	110592	384	00:20
50	40	20	true	288	20	110592	20	20	110592	384	00:17
50	40	20	true	288	20	110592	20	20	110592	384	00:16
50	40	20	true	288	20	110592	20	20	110592	384	00:15
50	40	20	true	288	20	110592	20	20	110592	384	00:14
50	40	20	true	288	20	110592	20	20	110592	384	00:13
50	40	20	true	288	20	110592	20	20	110592	384	00:12
50	40	20	true	288	20	110592	20	20	110592	384	00:11

Viewing 0 to 99 items

Refresh Last 6 months

rci_width_px	total_awake_formatted	total_awake_minutes	total_awake_seconds	uptime_formatted	uptime_minutes	uptime_seconds	veg_percent	vegetation_pixels	vegetation_px	wake_count
384	00:07:44	7	464	00:00:21	0	21	34.40552788723853	38117	38117	5
384	00:06:16	6	376	00:00:21	0	21	3.098033695220473	3393	3393	4
384	00:04:52	4	292	00:00:21	0	21	36.33987808227939	40189	40189	3
384	00:03:25	3	205	00:00:21	0	21	0	0	0	2
384	00:01:59	1	119	00:00:21	0	21	39.2078125	42416	42416	1
384	00:00:33	0	33	00:00:33	0	33	1.054325459015503	1166	1166	0
384	00:30:48	30	1848	00:18:54	19	1194	7.9345703125	8775	8775	6
384	00:29:48	29	1768	00:18:54	18	1134	6.929976483317871	7864	7864	6
384	00:28:48	28	1728	00:17:54	17	1074	11.326116833498094	12526	12526	6
384	00:27:48	27	1668	00:16:54	16	1014	1.7144097088797458	1896	1896	6
384	00:26:48	26	1608	00:15:54	15	954	39.48296358201172	45665	45665	6
384	00:25:48	25	1548	00:14:54	14	894	36.882415771484875	40789	40789	6
384	00:24:48	24	1488	00:13:54	13	834	34.81030479199219	36488	36488	6
384	00:23:48	23	1428	00:12:54	12	774	35.82347052001953	39619	39619	6
384	00:22:48	22	1368	00:11:54	11	714	35.8232825012207	39621	39621	6
384	00:21:48	21	1308	00:10:54	10	654	38.88527387973633	42004	42004	6

Viewing 0 to 99 items

Tabla 6.2. Registro de mediciones almacenadas en el bucket vegetacion.

Fuente: elaboración propia

Los resultados obtenidos confirmaron que Thinger.io permitió consultar los valores principales del sistema, conservar un histórico de las mediciones y modificar parámetros de funcionamiento de forma remota. La interfaz web local se mantuvo como medio de consulta de la imagen, la máscara y la superposición, mientras que Thinger.io se utilizó para el seguimiento temporal y el control del dispositivo.

6.4. Comparación entre ExG, HSV y Canopeo

Para valorar el comportamiento de los algoritmos implementados, los porcentajes obtenidos mediante Exceso de Verde normalizado y HSV se compararon con los proporcionados por Canopeo. Esta herramienta se tomó como referencia externa, aunque no como un valor exacto de verdad, ya que cada método utiliza criterios distintos para decidir qué píxeles pertenecen a la vegetación.

Las pruebas se realizaron con dos plantas y bajo iluminación natural exterior. En ambos casos se utilizó la imagen completa como ROI, con unas dimensiones de 640×480 píxeles y un total de 307 200 píxeles. La primera planta se situó a unos 40 cm de la cámara y la segunda a 50 cm. Para ExG se emplearon umbrales de 0,130 y 0,110, respectivamente, mientras que en HSV se mantuvo el rango de tono entre 40° y 150° , con una saturación mínima de 72 y un valor mínimo de 10. Además del porcentaje final, se revisaron también la máscara binaria y la imagen de superposición generadas por el sistema, ya que dos métodos pueden dar valores parecidos y, sin embargo, clasificar zonas distintas de la escena.

En la Fig. 6.4. se muestra el resultado obtenido para la primera planta mediante Exceso de Verde normalizado: 13,9 % de cubierta vegetal, con un umbral ExG de 0,130.

Porcentaje de Vegetacion

13.9%

vegetacion detectada

Píxeles vegetacion: 39936

Píxeles ROI total: 307200

Algoritmo usado: Exceso de Verde normalizado

Foto analizada:



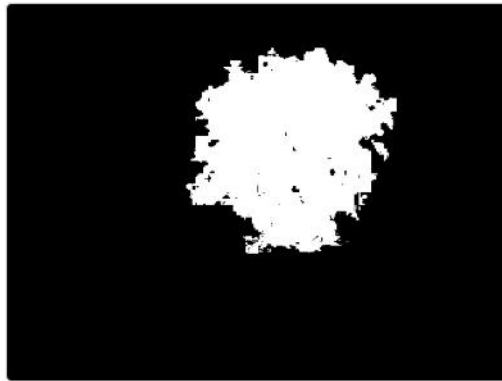
Visualizacion de deteccion

Los píxeles resaltados son exactamente los clasificados como vegetacion en el ultimo analisis.

Superposicion sobre la imagen original:



Mascara binaria:



Nueva foto y recalcular

Fig. 6.4. Resultado obtenido para la primera planta mediante Exceso de Verde.

Fuente: elaboración propia.

En la Fig. 6.5. se muestra el resultado obtenido para la primera planta mediante HSV: 15,0 % de cubierta vegetal, con un rango de tono de 40° a 150°, saturación mínima de 72 y valor mínimo de 10.

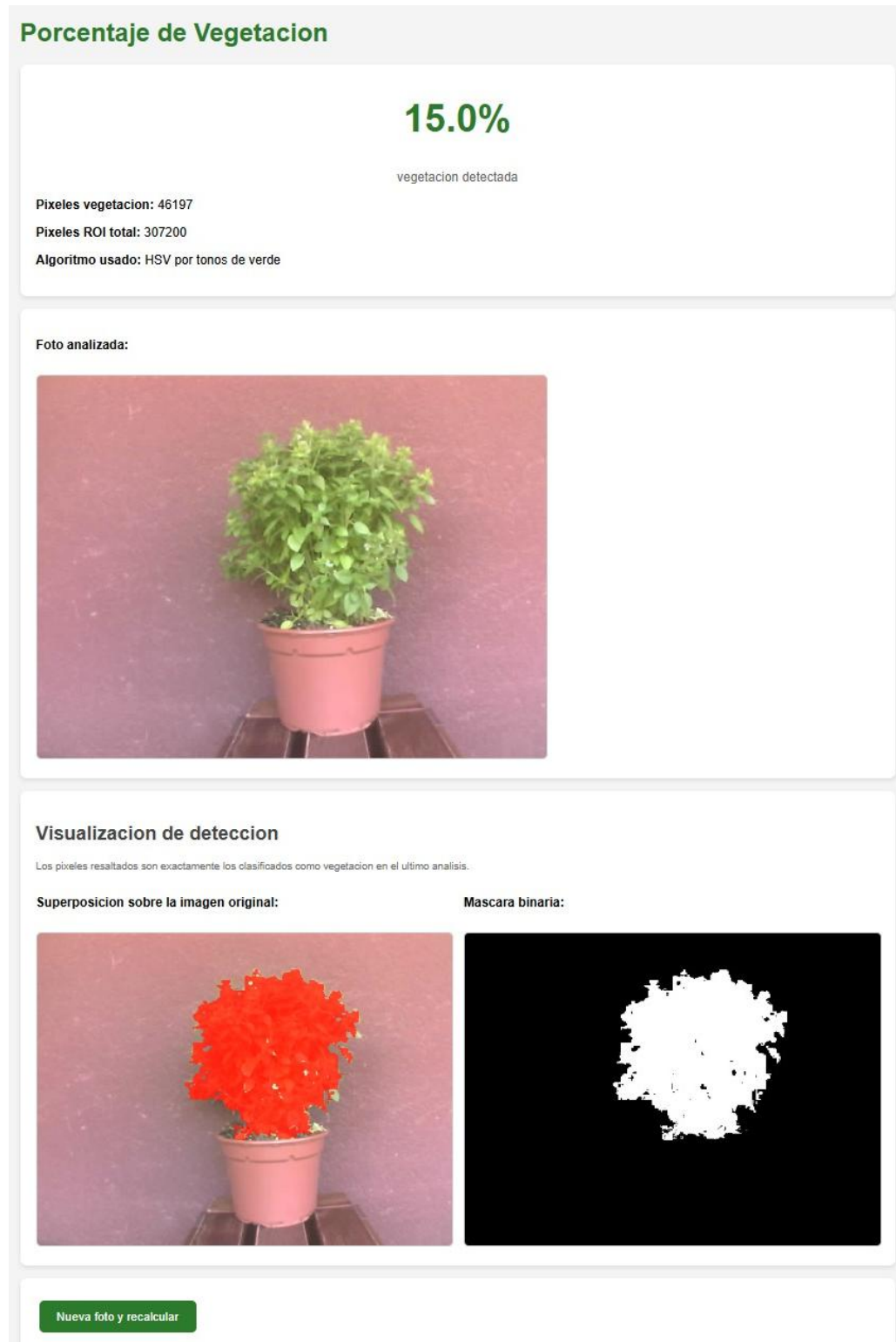


Fig. 6.5. Resultado obtenido para la primera planta mediante HSV.

Fuente: elaboración propia.

En la Fig. 6.6. se muestra el resultado obtenido para la primera planta mediante Canopeo: 13,3 % de cubierta vegetal.



Fig. 6.6. Resultado obtenido para la primera planta mediante Canopeo.

Fuente: elaboración propia.

En la Fig. 6.7. se muestra el resultado obtenido para la segunda planta mediante Exceso de Verde normalizado: 19,6 % de cubierta vegetal, con un umbral ExG de 0,110.

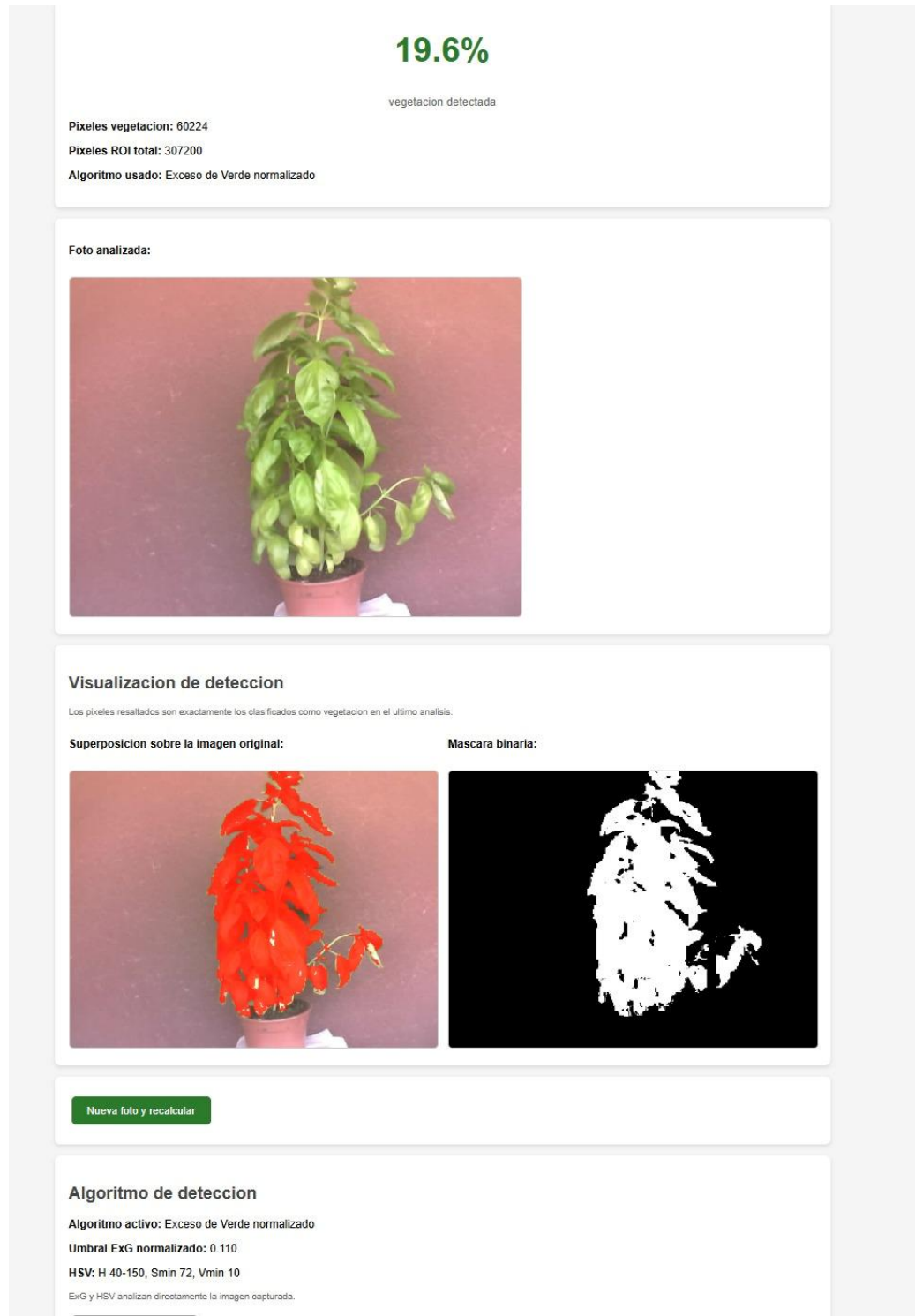


Fig. 6.7. Resultado obtenido para la segunda planta mediante Exceso de Verde.

Fuente: elaboración propia.

En la Fig. 6.8. se muestra el resultado obtenido para para la segunda planta mediante HSV: 21,0 % de cubierta vegetal, con un rango de tono de 40° a 150°, saturación mínima de 72 y valor mínimo de 10.

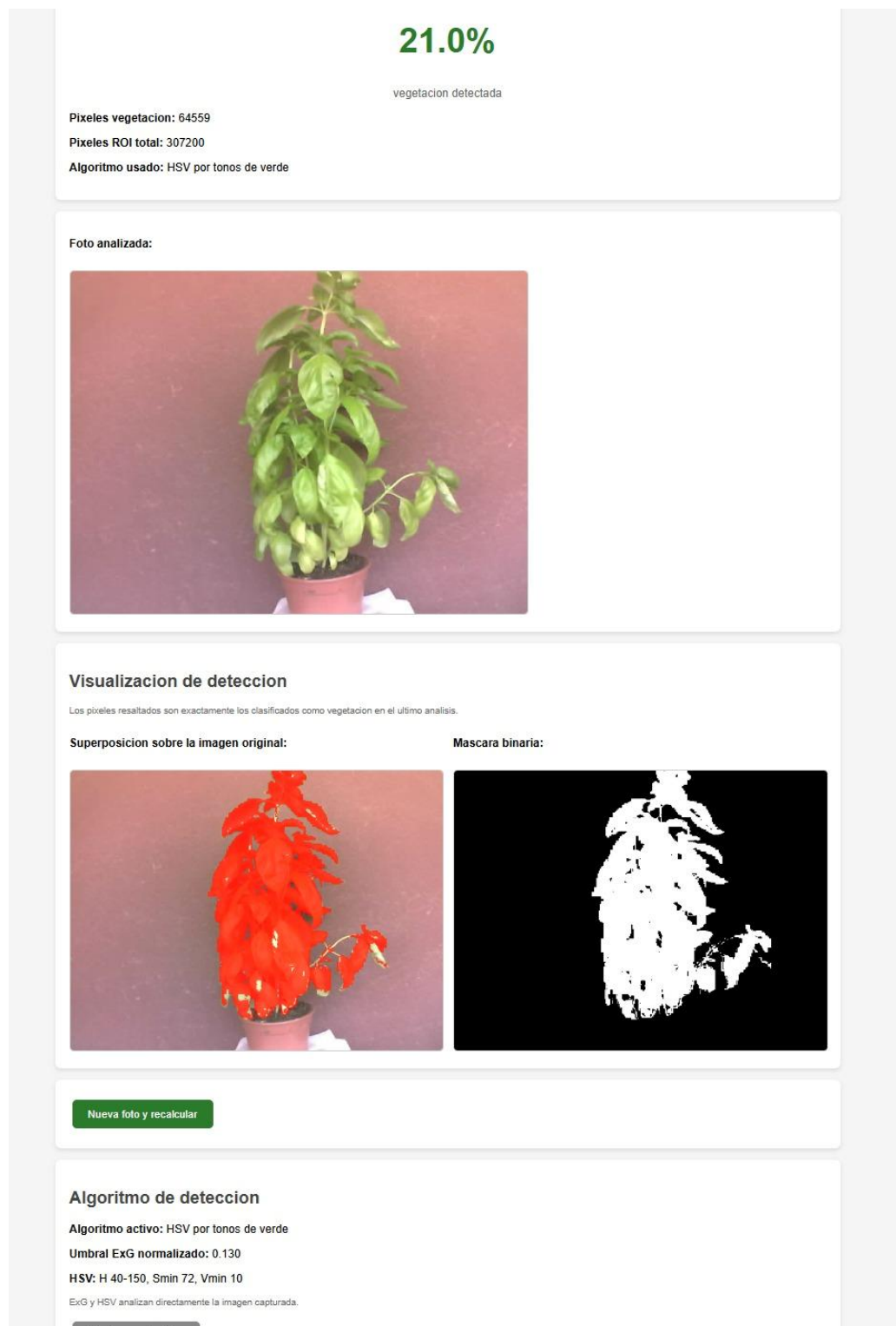


Fig. 6.8. Resultado obtenido para la segunda planta mediante HSV.

Fuente: elaboración propia.

En la Fig. 6.9. se muestra el resultado obtenido para la segunda planta mediante Canopeo: 15,8 % de cubierta vegetal.

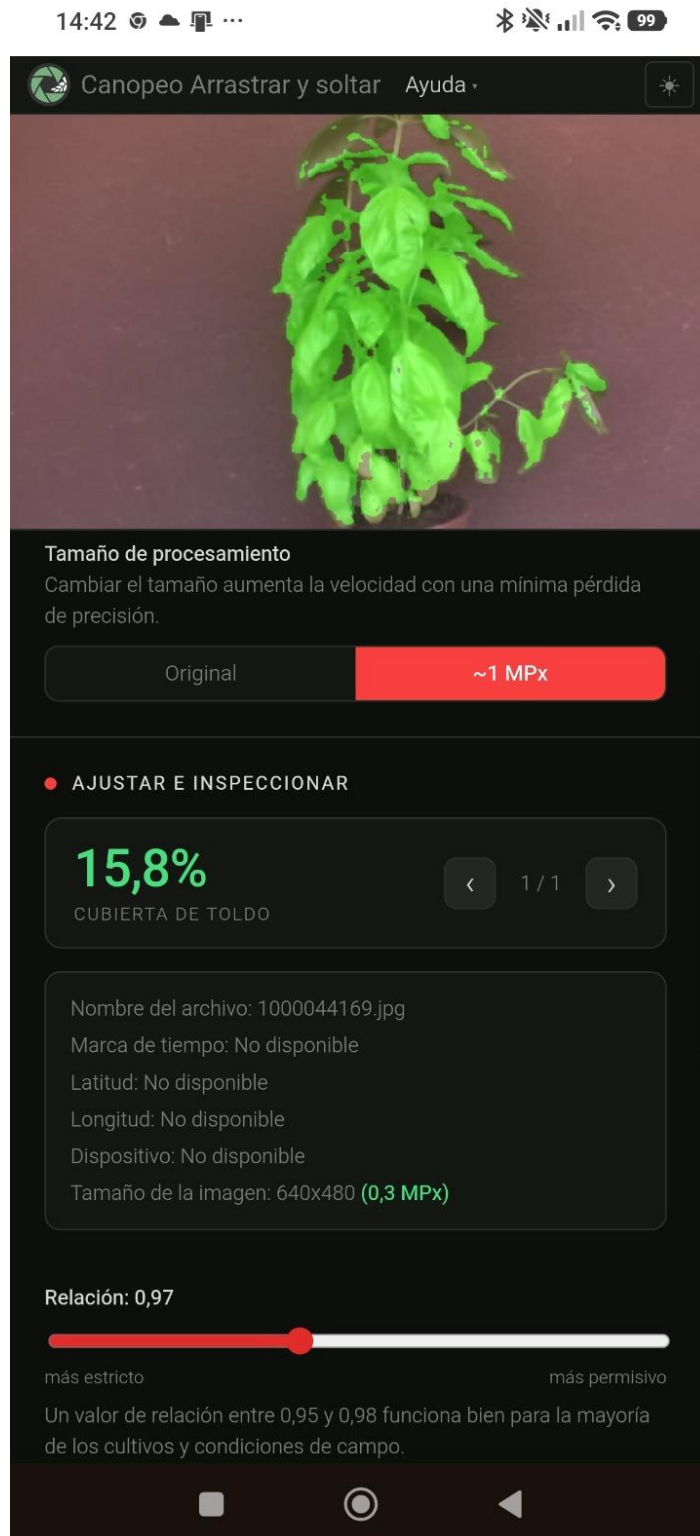


Fig. 6.9. Resultado obtenido para la segunda planta mediante Canopeo.

Fuente: elaboración propia.

En la primera escena, correspondiente a una maceta con una presencia abundante de hojas, ExG calculó una cubierta vegetal del 13,9 %, mientras que HSV alcanzó el 15,0 % y Canopeo proporcionó un 13,3 %. La diferencia entre ExG y la referencia fue de solo 0,6 puntos porcentuales. En el caso de HSV, la diferencia fue de 1,7 puntos por encima de Canopeo, lo que indica que este algoritmo clasificó como vegetación una superficie ligeramente mayor. Aun así, los tres resultados se mantuvieron próximos y ofrecieron una estimación similar para la escena analizada.

En la segunda prueba, la planta presentaba una distribución foliar menos compacta. El porcentaje obtenido mediante ExG fue del 19,6 %, equivalente a 60 224 píxeles clasificados como vegetación. HSV detectó 64 559 píxeles y alcanzó el 21,0 %, mientras que Canopeo mostró un valor inferior, del 15,8 %. En este caso, la separación entre métodos fue más evidente, con una diferencia de 3,8 puntos porcentuales entre ExG y Canopeo y de 5,2 puntos entre HSV y Canopeo.T

Prueba	Distancia	ExG	HSV	Canopeo	Diferencia ExG–Canopeo	Diferencia HSV–Canopeo
Planta 1	40 cm	13,9 %	15,0 %	13,3 %	0,6 p.p.	1,7 p.p.
Planta 2	50 cm	19,6 %	21,0 %	15,8 %	3,8 p.p.	5,2 p.p.

Tabla 6.3. Comparación de los porcentajes obtenidos mediante ExG, HSV y Canopeo.

Fuente: elaboración propia.

A primera vista puede resultar llamativo que la segunda planta, visualmente menos compacta, presente un porcentaje mayor. Sin embargo, la cubierta vegetal no mide la densidad botánica real de la planta, sino la proporción de píxeles identificados como vegetación dentro de la imagen. Por tanto, el resultado también depende de la distancia, del encuadre y de la superficie aparente ocupada por las hojas.

En las dos escenas analizadas, ExG produjo los valores más próximos a Canopeo. HSV ofreció porcentajes superiores y clasificó como vegetación una superficie mayor de la imagen. La revisión visual ayudó a interpretar esas diferencias, ya que la máscara y la superposición permiten valorar si la clasificación resulta coherente con las hojas visibles o si se han incorporado partes del fondo, sombras o zonas de la maceta. Al haberse utilizado únicamente dos escenas, estos resultados deben entenderse como una comparación experimental del prototipo y no como una validación general de la precisión de los algoritmos.

6.5. Prueba de repetibilidad

Una vez comparados los algoritmos con Canopeo, se realizó una prueba adicional para comprobar si el sistema ofrecía resultados estables al repetir el análisis en condiciones similares. El objetivo no era determinar la exactitud absoluta de la medición, sino observar cuánto variaban los porcentajes entre capturas sucesivas de una misma escena. Para ello se efectuaron cinco mediciones consecutivas con cada algoritmo, manteniendo el encuadre y la configuración de la cámara. En ExG se utilizó un umbral de 0,090, mientras que para HSV se mantuvo el rango de tono entre 40° y 150°, con una saturación mínima de 72 y un valor mínimo de 10. Los resultados obtenidos se recogen en la Tabla 6.4.

Captura	ExG	HSV
1	20,7 %	21,1 %
2	20,6 %	21,1 %
3	20,5 %	21,1 %
4	20,5 %	21,2 %
5	20,6 %	21,1 %
Media	20,58 %	21,12 %
Rango	0,2 p.p.	0,1 p.p.
Desviación estándar muestral	0,084 p.p.	0,045 p.p.

Tabla 6.4. Resultados de la prueba de repetibilidad.

Fuente: elaboración propia.

En ExG, los valores se situaron entre el 20,5 % y el 20,7 %, con una media del 20,58 %. La diferencia entre el valor máximo y el mínimo fue de 0,2 puntos porcentuales. HSV presentó una variación todavía menor: cuatro de las cinco capturas dieron un 21,1 % y solo una alcanzó el 21,2 %, por lo que el rango quedó reducido a 0,1 puntos. Las desviaciones estándar obtenidas, de 0,084 puntos porcentuales para ExG y 0,045 puntos porcentuales para HSV, muestran una dispersión muy reducida entre las mediciones, ligeramente menor para HSV en esta prueba concreta.

Estos resultados indican que el conjunto de captura y procesamiento mantuvo una respuesta estable cuando las condiciones de la escena permanecieron prácticamente constantes. Aun así, la repetibilidad no debe confundirse con la exactitud: la prueba demuestra que el prototipo ofrece resultados consistentes, pero no que el porcentaje calculado coincida necesariamente con el valor real de cubierta vegetal.

6.6. Funcionamiento del modo automático y Deep Sleep

El último bloque funcional comprobado fue la realización de mediciones sin intervención manual. Durante las pruebas se activaron tanto el modo automático convencional como la modalidad con Deep Sleep desde la interfaz

web local y desde Thingier.io. En el modo automático convencional, la M5Stack TimerCamera-F permanece encendida y conectada a la red. Al cumplirse el intervalo seleccionado, realiza una nueva captura, procesa la imagen con el algoritmo configurado, actualiza el porcentaje de cubierta vegetal e incrementa el contador de capturas automáticas. El resultado se envía también a Thingier.io, donde queda disponible para su visualización y almacenamiento. Los intervalos configurados en la versión final fueron de 60, 120 y 300 segundos, además de una opción de 24 horas.

La modalidad con Deep Sleep sigue un funcionamiento diferente. Después de cada despertar, el dispositivo recupera la configuración conservada en la memoria RTC, realiza una captura automática y actualiza tanto el resultado como los contadores del sistema. A continuación, permanece activo durante una ventana de tiempo para permitir la conexión Wi-Fi, el envío de la medición y la consulta de los datos. Cuando esa ventana termina, el ESP32 desconecta la cámara y la comunicación inalámbrica y entra en Deep Sleep hasta que el temporizador provoca el siguiente despertar. Gracias a la memoria RTC, se conservan el modo de detección, los umbrales del algoritmo, la configuración de la ROI y el número acumulado de capturas y despertares entre ciclos consecutivos.

Esta secuencia pudo repetirse sin pérdida de configuración, lo que confirma que el sistema mantiene el estado necesario para continuar funcionando tras cada ciclo de sueño profundo. La página dedicada al funcionamiento automático permitió comprobar el modo activo, el intervalo seleccionado, el número de capturas realizadas, el contador de despertares y la duración de la ventana de actividad. También mostró la última imagen procesada, por lo que fue posible verificar que cada ciclo generaba una nueva medición y no se limitaba únicamente a despertar el dispositivo.

En el modo automático convencional, la interfaz web local y la comunicación con Thingier.io permanecieron disponibles entre capturas. En la modalidad con Deep Sleep, estos servicios dejaron de estar accesibles durante el periodo de reposo y volvieron a estar disponibles después de cada despertar, durante la ventana de actividad configurada. La principal diferencia observada

entre ambos modos fue, por tanto, la disponibilidad del dispositivo entre mediciones.

El intervalo de 24 horas resulta especialmente útil cuando se pretende realizar un seguimiento diario desde una posición fija, ya que permite obtener las imágenes aproximadamente a la misma hora y reducir parte de las diferencias asociadas a la posición del sol. Sin embargo, esta programación no garantiza unas condiciones de captura idénticas, puesto que la nubosidad, las sombras, los reflejos y otros cambios ambientales pueden modificar la iluminación de la escena entre un día y otro.

Aunque no se realizó una medición instrumental del consumo eléctrico, se efectuó una estimación teórica de la autonomía del prototipo. Se partió de la batería nominal de la TimerCamera-F, $C_{\text{nominal}} = 270 \text{ mAh}$, y se consideró aprovechable el 80 % de su capacidad. La capacidad útil, $C_{\text{útil}}$, fue:

$$C_{\text{útil}} = 0,80 \cdot C_{\text{nominal}} = 0,80 \cdot 270 = 216 \text{ mAh}$$

Para el funcionamiento continuo se adoptó una corriente media activa $I_{\text{activo}} = 150 \text{ mA}$. La autonomía T_{continuo} se estimó mediante:

$$T_{\text{continuo}} = \frac{C_{\text{útil}}}{I_{\text{activo}}} = \frac{216}{150} = 1,44 \text{ h}$$

Este valor equivale aproximadamente a 1 hora y 26 minutos.

Para el modo Deep Sleep se consideró un ciclo cada 24 horas, con un tiempo activo $t_{\text{activo}} = 90 \text{ s} = 0,025 \text{ h}$ y un tiempo de reposo $t_{\text{reposo}} = 23,975 \text{ h}$. Adoptando una corriente en reposo $I_{\text{reposo}} = 0,12 \text{ mA}$, el consumo total por ciclo, Q_{ciclo} , fue:

$$Q_{\text{ciclo}} = I_{\text{activo}} \cdot t_{\text{activo}} + I_{\text{reposo}} \cdot t_{\text{reposo}}$$

$$Q_{\text{ciclo}} = 150 \cdot 0,025 + 0,12 \cdot 23,975 = 6,627 \text{ mAh/ciclo}$$

El número estimado de ciclos disponibles fue:

$$N_{\text{ciclos}} = \frac{C_{\text{útil}}}{Q_{\text{ciclo}}} = \frac{216}{6,627} = 32,59 \text{ ciclos}$$

Como se realiza un ciclo cada 24 horas, la autonomía teórica resultante es de aproximadamente 32,6 días. Estos valores son estimaciones basadas en consumos medios y pueden variar respecto al funcionamiento real del prototipo.

Las principales diferencias observadas entre ambos modos de funcionamiento se resumen en la Tabla 6.5.

Aspecto comprobado	Modo automático normal	Modo con Deep Sleep
Capturas periódicas	Correctas	Correctas después de cada despertar
Acceso web entre capturas	Disponible	No disponible durante el reposo
Comunicación con Thinger.io	Continua	Disponible durante la ventana despierta
Conservación de configuración	Durante el funcionamiento	Conservada entre ciclos mediante memoria RTC
Actualización de contadores	Correcta	Correcta entre despertares
Autonomía	1,44 h	32,59 días

Tabla 6.5. Comparación funcional de los modos automáticos comprobados.

Fuente: elaboración propia.

6.7. Limitaciones observadas

Las pruebas realizadas permitieron comprobar el funcionamiento del prototipo, aunque también pusieron de manifiesto algunas limitaciones propias del método empleado. La detección se basa principalmente en el color, por lo que los resultados pueden variar cuando cambia la iluminación, aparecen sombras o reflejos, o el fondo presenta tonos parecidos a los de la vegetación.

A esto se suma que los algoritmos trabajan con umbrales configurados previamente, sin reajustarlos de forma automática en cada captura, lo que puede dar lugar a falsos positivos o falsos negativos.

La instalación fija de la cámara favorece la repetibilidad, ya que mantiene constantes la posición, la distancia, la orientación y la región de interés, pero no permite controlar las condiciones ambientales de la escena. Aunque el intervalo de 24 horas ayuda a que las capturas se realicen aproximadamente a la misma hora, factores como la nubosidad, la posición del sol o el propio movimiento de las plantas pueden alterar los valores registrados. El encuadre y la región de interés influyen también de forma directa en el porcentaje calculado, ya que un cambio en la distancia a la planta o en la superficie visible puede modificar el resultado, aunque se esté analizando la misma vegetación.

Por otro lado, la comparación con Canopeo se realizó únicamente sobre dos escenas y la repetibilidad se comprobó en una situación concreta, por lo que estas pruebas permiten observar el comportamiento general del sistema, pero no constituyen una validación exhaustiva de la precisión de los algoritmos. Tampoco se midió de forma instrumental el consumo eléctrico ni el ahorro energético del modo Deep Sleep, aunque sí se verificó su correcto funcionamiento.

Estas limitaciones delimitan el alcance de los resultados obtenidos y deben tenerse en cuenta al interpretar las mediciones y plantear futuras mejoras del sistema.

7. CONCLUSIONES

Este Trabajo Fin de Grado partía del objetivo de diseñar, implementar y validar un sistema capaz de estimar el porcentaje de cubierta vegetal mediante el procesamiento local de imágenes RGB. Puede considerarse que este objetivo se ha cumplido dentro del alcance experimental planteado, ya que el prototipo desarrollado sobre la M5Stack TimerCamera-F reúne en un mismo dispositivo la captura, el análisis y la comunicación de las mediciones, sin depender de un ordenador externo para realizar el procesamiento. La versión final adquiere imágenes en resolución VGA, las convierte a formato RGB888 y analiza una región de interés mediante dos procedimientos de segmentación: Exceso de Verde normalizado y HSV. A partir de la clasificación de los píxeles se obtiene el porcentaje de cubierta vegetal, junto con una máscara binaria y una imagen de superposición. Estas dos salidas visuales han resultado casi tan importantes como el propio valor numérico, porque permiten comprobar de forma inmediata si el porcentaje calculado se corresponde con la vegetación visible o si la detección se ha visto afectada por el fondo, las sombras o un encuadre poco adecuado.

Los resultados pueden consultarse de dos formas complementarias. La interfaz web local proporciona acceso inmediato a la fotografía, la máscara, la superposición y los parámetros utilizados, mientras que Thinger.io permite visualizar los valores a distancia y conservar un histórico de las mediciones. A estas funciones se añade el modo automático, tanto en funcionamiento continuo como combinado con Deep Sleep, con el que el sistema puede capturar y procesar imágenes de forma periódica sin intervención directa del usuario.

En cuanto a los algoritmos, las pruebas mostraron estimaciones próximas entre ExG y HSV, aunque no idénticas. ExG se aproximó más a los valores proporcionados por Canopeo en las escenas comparadas, mientras que HSV tendió a clasificar como vegetación una superficie algo mayor. No puede afirmarse que uno de los métodos sea mejor de forma general, pero sí que la elección del algoritmo y de sus umbrales influye directamente en el resultado. La prueba de repetibilidad mostró, además, variaciones de solo 0,2 puntos porcentuales en ExG y de 0,1 en HSV entre capturas sucesivas, por lo que el

sistema mantuvo una respuesta prácticamente constante cuando la escena permaneció sin cambios.

Estos resultados deben interpretarse teniendo en cuenta las limitaciones del trabajo. Aunque la posición fija de la cámara favorece la comparación entre imágenes y el sistema permite programar capturas aproximadamente cada 24 horas, este intervalo no las sincroniza con una hora fija. Además, el sistema no controla las condiciones ambientales ni reajusta automáticamente los umbrales de segmentación. Por ello, los cambios de iluminación, las sombras, los reflejos, el movimiento de las plantas o las variaciones del fondo pueden provocar falsos positivos o falsos negativos. Asimismo, la comparación se realizó sobre un número reducido de escenas y Canopeo se utilizó como referencia orientativa, no como una medida exacta del valor real. Tampoco se midió de forma instrumental el consumo eléctrico durante el funcionamiento con Deep Sleep.

7.1. Líneas futuras

Como continuación del trabajo, sería conveniente ampliar las pruebas a un conjunto mayor de plantas, fondos y condiciones de iluminación. También resultaría útil comparar los resultados con segmentaciones manuales de referencia e incorporar mecanismos que permitan reducir o detectar las variaciones de iluminación, como un sensor de luminosidad, una visera para limitar la luz directa o una iluminación controlada. Además, podrían estudiarse umbrales adaptativos, la detección automática de capturas no válidas y el empleo de modelos TinyML.

Otra mejora pendiente sería medir el consumo eléctrico real en cada modo de funcionamiento. Del mismo modo, el almacenamiento de las imágenes en la nube permitiría consultarlas desde Thinger.io sin depender de la red local y facilitaría el seguimiento remoto de las mediciones.

Asimismo, el sistema podría incorporar sensores ambientales, como temperatura, humedad o radiación, así como nuevos algoritmos de análisis de imagen, con el fin de complementar la estimación de la cubierta vegetal con otras variables relacionadas con el estado del cultivo.

Por último, sería adecuado realizar pruebas prolongadas en condiciones reales de campo e incorporar un sistema de sincronización horaria que permitiera programar las capturas a una hora fija cada día.

8. REFLEXIÓN Y VALORACIÓN PERSONAL

La realización de este Trabajo Fin de Grado me ha permitido poner en práctica conocimientos que hasta ahora había trabajado por separado durante la carrera. En un mismo proyecto he tenido que unir un microcontrolador, una cámara, procesamiento de imágenes, programación, comunicaciones e integración con una plataforma IoT, y al principio no resultaba tan evidente cómo encajaba cada parte. Con el avance del trabajo fui entendiendo mejor la relación entre ellas, hasta llegar a un sistema en el que una imagen capturada por la cámara se procesa en el propio ESP32, se obtiene un porcentaje de cubierta vegetal y el resultado se muestra tanto en la interfaz web como en Thinger.io. Ver ese funcionamiento completo ha sido, sin duda, una de las partes más satisfactorias del proyecto.

Más allá del desarrollo del código, este trabajo me ha obligado a probar, corregir y replantear decisiones en varias ocasiones. La memoria disponible, el formato de las imágenes, la influencia de la iluminación o el comportamiento del modo automático fueron aspectos que hicieron necesario revisar el sistema más de una vez. Este proceso me ha permitido comprobar que en un proyecto real no todo funciona de forma definitiva a la primera y que muchas mejoras aparecen después de observar con calma dónde falla el prototipo y por qué.

También valoro especialmente haber conseguido una versión funcional que puedo explicar y defender con seguridad. El resultado final no pretende ser un sistema profesional, pero sí refleja todo el recorrido seguido desde la idea inicial hasta un dispositivo capaz de capturar, procesar y comunicar sus propias mediciones. A nivel personal, considero que he aprendido más durante el desarrollo del trabajo que si me hubiera limitado a estudiar estas tecnologías por separado, y esa es probablemente la parte que más me llevo de todo el proceso.

9. REFERENCIAS

[1] A. Patrignani y T. E. Ochsner, “Canopeo: A Powerful New Tool for Measuring Fractional Green Canopy Cover,” *Agronomy Journal*, vol. 107, n.º 6, pp. 2312–2320, 2015, doi: 10.2134/agronj15.0150.

[2] A. Ashapure, J. Jung, A. Chang, S. Oh, M. Maeda y J. A. Landivar, “A Comparative Study of RGB and Multispectral Sensor-Based Cotton Canopy Cover Modelling Using Multi-Temporal UAS Data,” *Remote Sensing*, vol. 11, n.º 23, art. 2757, 2019, doi: 10.3390/rs11232757.

[3] M5Stack, “TimerCamera-F,” M5Stack Docs. [En línea]. Disponible en: https://docs.m5stack.com/en/unit/timercam_f. [Consultado: 15-jun-2026].

[4] Food and Agriculture Organization of the United Nations, *The State of Food and Agriculture 2022: Leveraging Automation in Agriculture for Transforming Agrifood Systems*. Rome, Italy: FAO, 2022, doi: 10.4060/cb9479en.

[5] R. P. Sishodia, R. L. Ray and S. K. Singh, “Applications of Remote Sensing in Precision Agriculture: A Review,” *Remote Sensing*, vol. 12, no. 19, art. 3136, 2020, doi: 10.3390/rs12193136.

[6] Z. Cao, S. Sun and X. Bao, “A Review of Computer Vision and Deep Learning Applications in Crop Growth Management,” *Applied Sciences*, vol. 15, no. 15, art. 8438, 2025, doi: 10.3390/app15158438.

[7] R. Gong, H. Zhang, G. Li and J. He, “Edge Computing-Enabled Smart Agriculture: Technical Architectures, Practical Evolution, and Bottleneck Breakthroughs,” *Sensors*, vol. 25, no. 17, art. 5302, 2025, doi: 10.3390/s25175302.

[8] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA, USA: O'Reilly Media, 2019.

[9] L. Büchi, M. Wendling, P. Mouly and R. Charles, "Comparison of Visual Assessment and Digital Image Analysis for Canopy Cover Estimation," *Agronomy Journal*, vol. 110, no. 4, pp. 1289–1295, 2018, doi: 10.2134/agronj2017.11.0679.

[10] T. R. Tenreiro, M. García-Vila, J. A. Gómez, J. A. Jiménez-Berni and E. Fereres, "Using NDVI for the assessment of canopy cover in agricultural crops within modelling research," *Computers and Electronics in Agriculture*, vol. 182, art. 106038, 2021, doi: 10.1016/j.compag.2021.106038.

[11] A. Kior, L. Yudina, Y. Zolin, V. Sukhov and E. Sukhova, "RGB Imaging as a Tool for Remote Sensing of Characteristics of Terrestrial Plants: A Review," *Plants*, vol. 13, no. 9, art. 1262, 2024, doi: 10.3390/plants13091262.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Harlow, United Kingdom: Pearson Education, 2018.

[13] ITU-T, *Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines*, Recommendation T.81, International Telecommunication Union, 1992.

[14] Espressif Systems, "Camera Application," ESP-FAQ Programming Guide. [En línea]. Disponible: <https://docs.espressif.com/projects/esp-faq/en/latest/application-solution/camera-application.html>. [Consultado: 15-jun-2026].

[15] Espressif Systems, "ESP32 Camera Driver," ESP Component Registry. [En línea]. Disponible: <https://components.espressif.com/components/espressif/esp32-camera>. [Consultado: 15-jun-2026].

[16] L. Li, Q. Zhang and D. Huang, "A Review of Imaging Techniques for Plant Phenotyping," *Sensors*, vol. 14, no. 11, pp. 20078–20111, 2014, doi: 10.3390/s141120078.

[17] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979, doi: 10.1109/TSMC.1979.4310076.

[18] D. M. Woebbecke, G. E. Meyer, K. Von Bargen and D. A. Mortensen, "Color Indices for Weed Identification Under Various Soil, Residue, and Lighting Conditions," *Transactions of the ASAE*, vol. 38, no. 1, pp. 259–269, 1995, doi: 10.13031/2013.27838.

[19] G. E. Meyer and J. Camargo Neto, "Verification of Color Vegetation Indices for Automated Crop Imaging Applications," *Computers and Electronics in Agriculture*, vol. 63, no. 2, pp. 282–293, 2008, doi: 10.1016/j.compag.2008.03.009.

[20] M. Guijarro et al., "Automatic Segmentation of Relevant Textures in Agricultural Images," *Computers and Electronics in Agriculture*, vol. 75, no. 1, pp. 75–83, 2011, doi: 10.1016/j.compag.2010.09.013.

[21] M. Beltrán-Escobar, T. E. Alarcón, J. Y. Rumbo-Morales, S. López, G. Ortiz-Torres and F. D. J. Sorcia-Vázquez, "A Review on Resource-Constrained Embedded Vision Systems-Based Tiny Machine Learning for Robotic Applications," *Algorithms*, vol. 17, no. 11, art. 476, 2024, doi: 10.3390/a17110476.

[22] A. Choudhary, "Internet of Things: a comprehensive overview, architectures, enabling technologies, applications and challenges," *Discover Internet of Things*, vol. 4, art. 4, 2024, doi: 10.1007/s43926-024-00084-3.

[23] S. H. Abdulhussain et al., "A Comprehensive Review of Sensor Technologies in IoT: Technical Aspects, Challenges, and Future Directions," *Computers*, vol. 14, no. 8, art. 342, 2025, doi: 10.3390/computers14080342.

[24] Thingier.io, "Overview," Thingier.io Documentation. [En línea]. Disponible: <https://docs.thinger.io/>. [Consultado: 15-jun-2026].

[25] Thingier.io, "Data Buckets," Thingier.io Documentation. [En línea]. Disponible: <https://docs.thinger.io/features/buckets>. [Consultado: 15-jun-2026].

[26] Thingier.io, "Dashboards," Thingier.io Documentation. [En línea]. Disponible: <https://docs.thinger.io/features/dashboards>. [Consultado: 15-jun-2026].

[27] R. De Bei et al., “VitiCanopy: A Free Computer App to Estimate Canopy Vigor and Porosity for Grapevine,” *Sensors*, vol. 16, no. 4, art. 585, 2016, doi: 10.3390/s16040585.

[28] Copernicus Data Space Ecosystem, “Sentinel-2.” [En línea]. Disponible en: <https://dataspace.copernicus.eu/data-collections/copernicus-sentinel-missions/sentinel-2> [Consultado: 9-jul-2026].

[29] SkyWatch, “Data Pricing.” [En línea]. Disponible en: <https://skywatch.com/data-pricing/>. [Consultado: 9-jul-2026].

[30] SkyWatch Knowledge Base, “Layer Size & Price Requirements.” [En línea]. Disponible en: <https://intercom.help/skywatch-ab92e82e1a0a/en/articles/9580185-layer-size-price-requirements> [Consultado: 9-jul-2026].

[31] W. Zhu, Z. Feng, S. Dai, P. Zhang and X. Wei, “Using UAV Multispectral Remote Sensing with Appropriate Spatial Resolution and Machine Learning to Monitor Wheat Scab,” *Agriculture*, vol. 12, no. 11, art. 1785, 2022, doi: 10.3390/agriculture12111785.

[32] Sentera, “FieldAgent: Field Health and Variability.” [En línea]. Disponible en: <https://sentera.com/fieldagent/> [Consultado: 9-jul-2026].

[33] European Union Aviation Safety Agency, “Open Category—Low Risk—Civil Drones.” [En línea]. Disponible en: <https://www.easa.europa.eu/en/domains/drones-air-mobility/operating-drone/open-category-low-risk-civil-drones> [Consultado: 9-jul-2026].

[34] C. A. Schneider, W. S. Rasband and K. W. Eliceiri, “NIH Image to ImageJ: 25 years of image analysis,” *Nature Methods*, vol. 9, pp. 671–675, 2012, doi: 10.1038/nmeth.2089.

[35] W. Yang, S. Wang, X. Zhao, J. Zhang and J. Feng, “Greenness identification based on HSV decision tree,” *Information Processing in Agriculture*, vol. 2, no. 3–4, pp. 149–160, 2015, doi: 10.1016/j.inpa.2015.07.003.

10. ANEXOS

10.1. Anexo 1. Repositorio GitHub y código fuente

El código fuente completo desarrollado para este Trabajo Fin de Grado se encuentra disponible en el siguiente repositorio público de GitHub:

Enlace: <https://github.com/alvaromartinezlopez14-cmd/tfg-cubierta-vegetal-esp32>

Archivo principal: `codigo_base_funcional_v9.ino`

Rama: `main`

Descripción: El repositorio contiene la versión final del programa utilizado en el prototipo y el archivo `README.md`, donde se incluyen la descripción general del proyecto, el hardware empleado, las librerías necesarias y las indicaciones básicas de configuración y carga del código.

