

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y  
ELECTRÓNICA**

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA MULTIESPECTRAL  
PARA DETERMINAR EL ÍNDICE DE VEGETACIÓN DE  
DIFERENCIA NORMALIZADA DE LAS PLANTAS EN UN CULTIVO  
DE ÁREA MENOR A  $100m^2$**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN  
ELECTRÓNICA Y AUTOMATIZACIÓN**

**OLMEDO BECERRA RAÚL ANDRÉS**

**raul.olmedo@epn.edu.ec**

**DIRECTOR: FAUSTO EDUARDO ÁVALOS CASCANTE, Ph. D**

**eduardo.avalos@epn.edu.ec**

**DMQ, enero 2023**

## **DECLARACIÓN DE AUTORÍA**

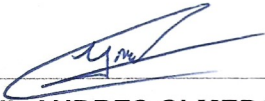
A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

RAUL ANDRÉS OLMEDO BECERRA

ÁVALOS CASCANTE, Ph. D

## CERTIFICACIONES

Yo, RAÚL ANDRÉS OLMEDO BECERRA declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



RAUL ANDRES OLMEDO BECERRA

Certifico que el presente trabajo de integración curricular fue desarrollado por Raúl Andrés Olmedo Becerra, bajo mi supervisión.



FAUSTO EDUARDO ÁVALOS CASCANTE, Ph. D  
DIRECTOR

## **DEDICATORIA**

A mi padre Raúl Olmedo y mi madre Nely Becerra, quienes con su ejemplo me han demostrado que no hay que rendirse frente a las dificultades que se presenten en la vida y también me han inculcado valores como humildad y perseverancia.

A mis hermanas, Giss y Gaby por su apoyo que me han brindado de forma incondicional, siempre alentándome y han estado presentes a través de todos estos años.

A Liz que me ha brindado apoyo durante varios años en mi formación profesional.

**Raúl Olmedo**

## **AGRADECIMIENTO**

Agradezco a Dios por darme la oportunidad de recuperarme de toda enfermedad y que su bendición ha cubierto a toda mi familia.

Agradezco a mis padres por su apoyo incondicional que han contribuido enormemente en mi realización profesional.

A toda mi familia por su constante guía y por apoyarme incondicionalmente en las decisiones que he tomado a lo largo de los años.

A mi tutor de tesis ÁVALOS CASCANTE, Ph. D que me apoyó en la realización del proyecto de titulación.

A mis amigos Jairo, Patricio, Bryan, Jorge, Naty, Cesar, Vero por el apoyo mutuo durante las materias de la carrera, compartiendo conocimiento y trabajando juntos para tener un mejor desempeño en cada semestre.

**Raúl Olmedo**

# ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VII
ABSTRACT .....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO .....	9
1.1 OBJETIVO GENERAL .....	10
1.2 OBJETIVOS ESPECÍFICOS .....	10
1.3 ALCANCE .....	11
1.4 MARCO TEÓRICO.....	11
1.4.1 Agricultura Inteligente.....	11
1.4.2 Espectro Electromagnético.....	12
1.4.3 Reflectancia de las plantas según su vitalidad .....	13
1.4.4 NDVI (Índice De vegetación de Diferencia Normalizada) .....	14
1.4.5 ArcGIS .....	15
1.4.5.1 Teledetección.....	15
1.4.5.2 LANDSAT 8.....	15
1.4.6 Python .....	16
1.4.6.1 Opencv.....	19
1.4.6.2 Numpy.....	20
1.4.6.3 PyQT5.....	20
1.4.6.4 Matplotlib.....	20
1.4.6.5 Procesamiento de Imágenes.....	21
1.4.7 Filtro Infrarrojo en cámaras comerciales.....	22
1.4.8 Software .....	23
1.4.8.1 Solid Works.....	23
1.4.8.2 Visual Studio Code.....	24
1.4.8.3 QT-Designer.....	24
1.4.8.4 Thonny.....	25
2 METODOLOGÍA.....	25

3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	30
3.1	PRUEBAS Y RESULTADOS .....	48
3.2	CONCLUSIONES.....	54
3.3	RECOMENDACIONES .....	56
4	REFERENCIAS BIBLIOGRÁFICAS .....	56
5	ANEXOS.....	62

## RESUMEN

En el presente Trabajo de Integración Curricular, se detalla la metodología para implementar un prototipo portátil capaz de identificar el Índice de Vegetación Normalizado (NDVI) de plantas. El prototipo consta de un sistema óptico, una Raspberry Pi, un sistema de carga de batería, y una pantalla táctil. El prototipo desarrollado captura fotografías espectrales, gracias a un sistema óptico, y las procesa para obtener el NDVI. El sistema óptico del prototipo se conforma por una cámara que ha sido modificada retirando su filtro infrarrojo (IR) y reemplazándolo con otro filtro especializado colocado al frente del sensor de la cámara con el propósito de adquirir una imagen espectral. Posteriormente, con el propósito de procesar la imagen espectral y obtener el índice NDVI, se ha desarrollado un código de programación en Python, el cual se ejecuta en una Raspberry Pi. El NDVI obtenido con el código de programación es validado mediante el uso del software ArcGIS. Adicionalmente, el prototipo implementado posee una interfaz gráfica de usuario (GUI) en una pantalla táctil que permite al usuario adquirir las imágenes espectrales y obtener el índice vegetativo con su respectiva imagen NDVI. Por otra parte, en este documento, se detallan las características técnicas de los módulos electrónicos comerciales utilizados para que el sistema pueda funcionar de forma portátil y sea recargable. De igual manera, se presenta el proceso para crear una interfaz gráfica que sea capaz de funcionar con programación en lenguaje Python. Por último, se complementa el estudio con investigación teórica sobre la reflectancia de las plantas y la teoría del procesamiento digital de imágenes.

**PALABRAS CLAVE:** NDVI, Python, NIR, RED, BLUE, Procesamiento, reflectancia, raster.



## **ABSTRACT**

In this Curricular Integration Work, the methodology for implementing a portable prototype capable of identifying the NDVI of plants is detailed, which consists of an optical system, a raspberry pi, a battery charging system, and a touch screen. The developed prototype captures spectral photographs, thanks to an optical system, and processes them for the obtaining of the NDVI vegetation index. The optical system of the prototype is made up of a camera which has been modified by removing its IR filter, replacing it with a specialized filter placed in front of the camera's sensor with the purpose of acquiring a spectral image. Subsequently, with the purpose of processing the spectral image and obtaining the NDVI index, a programming code has been developed in the Python language, which is executed on a raspberry pi. The NDVI index obtained with the programming code is validated through the use of the ArcGIS software. Additionally, the implemented prototype has a user interface (GUI) on a touch screen that allows the user to acquire spectral images, also obtain the vegetation index with its respective NDVI image. On the other hand, in this document, the technical characteristics of the commercial electronic modules used for the system to work in a portable way and that is rechargeable are detailed. Similarly, the process for creating a graphical interface that is capable of working with programming in the Python language is presented. Finally, the study is complemented with theoretical research on plant reflectance and the theory of digital image processing.

**KEYWORDS:** NDVI, Python, NIR, RED, BLUE, Processing, reflectance, raster

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El componente desarrollado permite adquirir imágenes espectrales mediante un sistema óptico implementado. Se ha retirado el filtro IR de la cámara comercial y se ha añadido un filtro especial en el frente del sensor de la cámara. Las imágenes espectrales adquiridas son procesadas digitalmente para determinar el índice vegetativo NDVI. Para que el prototipo pueda comenzar a adquirir las imágenes espectrales, se ha utilizado un pulsador asociado a una entrada digital de la Raspberry Pi 3 B+. Al momento de accionar el pulsador, se guarda el fotograma actual del video. Posteriormente, la imagen almacenada es procesada para segmentar en imágenes de un solo canal. La imagen correspondiente al canal "B" es equivalente a la imagen NIR, gracias al sistema óptico implementado, mientras que la imagen del canal "R" corresponde a la imagen RED. Una vez almacenadas las imágenes de un solo canal en variables internas en memoria, son utilizadas para realizar el cálculo del índice NDVI. Después de efectuar la ecuación del NDVI píxel a píxel, se procede a mostrar en pantalla el índice NDVI de acuerdo con la posición donde se encuentre el ratón sobre la imagen. Para llevar a cabo este trabajo se han utilizado librerías especializadas. La imagen NDVI se logra visualizar en una pantalla táctil, gracias a la programación desarrollada en código abierto.

Para determinar el NDVI con ArcGIS, se deben cargar las imágenes de un solo canal al programa ArcGIS y proceder con el cálculo del índice NDVI. Además, se ha construido la imagen NDVI completamente a partir de todos los índices NDVI que se almacenan en una matriz conocida como numpy array. Para la construcción de la imagen, se han empleado una ecuación adaptativa que permite determinar la intensidad de los píxeles de acuerdo a un análisis previo de los índices vegetativos calculados.

En la implementación de una interfaz gráfica de usuario (GUI), se ha utilizado el software QT-Designer, que permite crear interfaces gráficas para trabajar con lenguaje Python. La interfaz gráfica consta de dos pulsadores y un botón de opción. Un pulsador permite cargar la fotografía a procesar, el otro pulsador sirve para iniciar la función de procesamiento y visualización de la imagen NDVI, y finalmente, el botón de opción permite acceder al video de la cámara comercial para guardar los fotogramas del video.

Finalmente se ha agregado colores a la imagen NDVI de un solo canal para que se incremente el contraste de la imagen y de esta manera se pueda distinguir las regiones de la planta a estudiarse.

## **1.1 OBJETIVO GENERAL**

Diseñar y construir un sistema multiespectral para determinar el índice de vegetación de diferencia normalizada de las plantas en un cultivo de área menor a 100  $m^2$ .

## **1.2 OBJETIVOS ESPECÍFICOS**

Los objetivos necesarios para la construcción del componente se ha realizado los siguientes objetivos:

1. Revisión bibliográfica sobre la utilización de software ArcGIS para determinar el índice NDVI y sobre la programación en lenguaje Python para el procesamiento de imágenes.
2. Revisión bibliográfica sobre la reflectancia de las plantas cuando estas se encuentran sanas y enfermas.
3. Construcción de la estructura mecánica que permita la sujeción de todos los componentes del sistema del prototipo portátil desarrollado que dispondrá de una pantalla táctil.
4. Desarrollo de un sistema óptico a partir de una cámara comercial para que esta pueda captar la banda del infrarrojo cercano del espectro electromagnético.
5. Programación del algoritmo en lenguaje Python para el procesamiento digital de las fotografías adquiridas para la determinación del índice NDVI y su visualización en una pantalla táctil interactiva, así como en un ordenador.
6. Desarrollo de una interfaz gráfica con el programa QT-Designer para que trabaje en conjunto con la programación realizada en lenguaje Python.
7. Implementación de módulos electrónicos comerciales para que el prototipo sea portable y recargable.
8. Validación del índice NDVI obtenido mediante el algoritmo desarrollado en Python con el índice NDVI determinado por ArcGIS.

## **1.3 ALCANCE**

- Se presentará una revisión bibliográfica sobre la utilización del software ArcGIS para determinar el índice NDVI y sobre la programación en lenguaje Python para el procesamiento de imágenes.
- Se realizará una revisión bibliográfica sobre la reflectancia de las plantas cuando estas se encuentran sanas y enfermas.
- Se realizará la construcción de la estructura mecánica que permita la sujeción de todos los componentes del sistema del prototipo portátil desarrollado que dispondrá de una pantalla táctil.
- Se desarrollará a partir de una cámara comercial un sistema óptico para que esta pueda captar la banda del infrarrojo cercano del espectro electromagnético.
- Se desarrollará la programación del algoritmo en lenguaje Python para el procesamiento digital de las fotografías adquiridas para la determinación del índice NDVI y su visualización en una pantalla táctil interactiva, así como en un ordenador.
- Se desarrollará una interfaz gráfica con el programa QT-Designer para que trabaje en conjunto con la programación realizada en lenguaje Python
- Se implementará módulos electrónicos comerciales para que el prototipo sea portable y recargable.
- Se validará el índice NDVI obtenido mediante el algoritmo desarrollado en Python con el índice NDVI determinado por ArcGIS.

## **1.4 MARCO TEÓRICO**

### **1.4.1 Agricultura Inteligente**

La agricultura Inteligente consiste en un conjunto de elementos tecnológicos que permiten a los productores tener información actualizada sobre el estado de sus cultivos las 24 horas del día, los 7 días de la semana, así como operar una gama de equipos agrícolas de forma remota con el objetivo principal de incrementar la cantidad y calidad de la producción agrícola de sus cultivos.

La aplicación de la agricultura inteligente sobre un cultivo puede otorgar ciertos beneficios como ahorro en recursos, mayor rentabilidad, optimizar tiempos de los procesos de

producción, prevenir enfermedades detectando anticipadamente el desarrollo de las plagas y enfermedades [1].

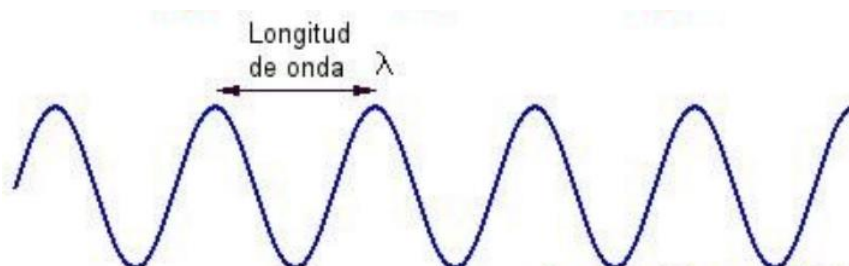


**Figura 1.1.** Agricultura Inteligente. [2]

Existen nuevas tecnologías que se han venido desarrollando en los últimos años que provocan grandes cambios en el sector agrícola, y surgen nuevos campos de investigación que se puede incursionar para mejorar procesos productivos. Entre estos campos de investigación se encuentran el manejo de Big Data para la predicción de la producción agrícola. Así como también, la medición de parámetros importantes del cultivo, tales como humedad, temperatura, conductividad, pH, la velocidad y la dirección del viento. Además, la utilización de drones para detección de plagas, enfermedades y elaboración de mapas geográficos de la productividad del terreno [2].

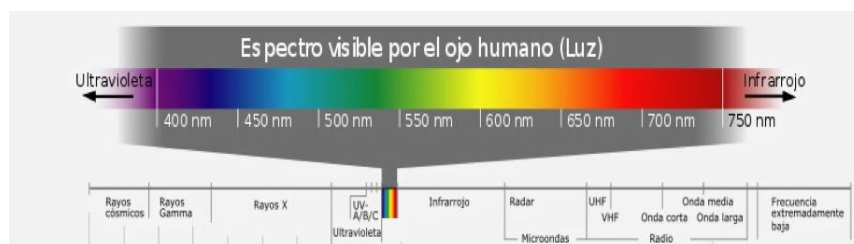
#### **1.4.2 Espectro Electromagnético.**

La luz, al tener una naturaleza dual, se puede llegar a comportar como partícula y como onda. El comportarse como onda conlleva a que esta tenga características como frecuencia, longitud de onda, y amplitud. La longitud de onda se define como la distancia que existe entre dos crestas de la misma onda, esta distancia puede visualizarse en la Figura 1.2.



**Figura 1.2.** Espectro Electromagnético. [3]

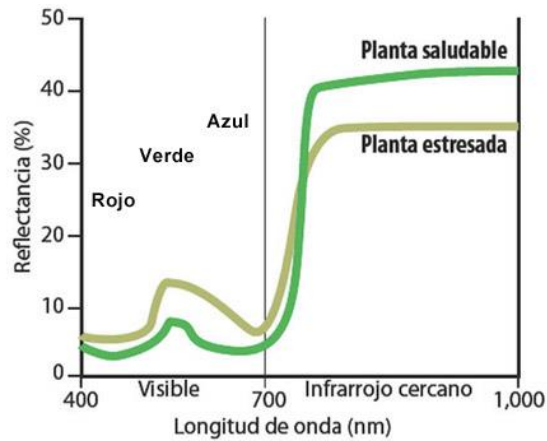
El espectro electromagnético se define como el conjunto de los tipos de luz, los cuales son clasificados de acuerdo con su frecuencia y su longitud de onda [3]. En la clasificación se encuentran los rayos x, ultravioleta, rayos gama, visible, infrarrojo, microondas y ondas de radio como se muestra en la Figura 1.3.



**Figura 1.3.** Espectro Electromagnético. [3]

### 1.4.3 Reflectancia de las plantas según su vitalidad

La energía que llega a las plantas proviene del sol, el cual emite longitudes de onda entre 200 hasta 3000 nm [4]. Parte de esta energía puede ser absorbida o reflejada, dependiendo de las características de la planta. En el proceso de la fotosíntesis la planta utiliza la banda del azul y del rojo del espectro visible para transformarla en alimento. Por lo tanto, al no utilizarse grandes cantidades de luz verde, esta luz se refleja cuando la longitud de onda varía entre 400nm y 700 nm (esto se puede visualizar en la Figura 1.4). Otra característica importante de las plantas es que, cuando la vegetación se encuentra saludable, en la banda del infrarrojo cercano del espectro electromagnético se tiene una gran cantidad de radiación [5]. Por lo tanto, con el análisis del espectro electromagnético se puede determinar si una planta se encuentra enferma o no.



**Figura 1.4.** Reflectancia de una planta según su vitalidad. [5]

#### 1.4.4 NDVI (Índice De vegetación de Diferencia Normalizada)

El índice NDVI es una herramienta cuantitativa para medir el vigor vegetativo de una planta. Se obtiene a través de imágenes satelitales. Cuando la vegetación es sana presenta índices cercanos al valor de 1, mientras que cuando está enferma se acercan al valor de -1. Esto permite cuantificar el estrés vegetativo de una planta [6]. El NDVI puede tomar valores entre -1 y 1 y se lo utiliza para predecir la producción agrícola, monitorear sequías y detectar zonas propensas a incendios, entre otras aplicaciones.

El NDVI es calculado con la expresión matemática expuesta en la ecuación 1.1 [6].

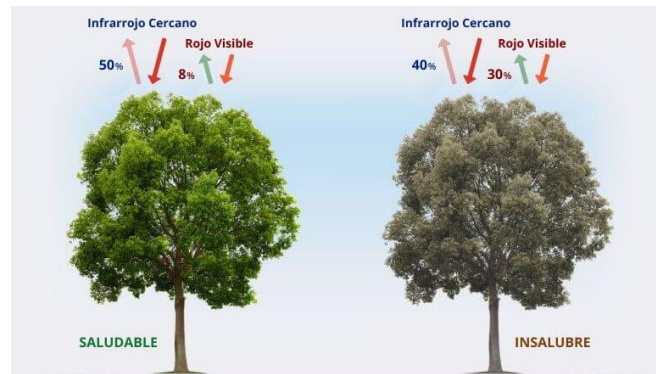
$$NDVI = \frac{NIR-RED}{NIR+RED} \quad [6] \quad (1.1)$$

Donde:

NIR: Imagen correspondiente a la banda del Infrarrojo cercano.

RED: Imagen correspondiente a la banda roja del espectro electromagnético.

La clorofila, en el proceso de la fotosíntesis, absorbe energía del sol para transformarla en alimento para la planta. En este proceso, si la planta se encuentra enferma, reflejará menos energía del infrarrojo cercano y más energía del rojo visible [6]. En la Figura 1.5 se puede observar que una planta sana, reflejará más energía del infrarrojo cercano (50%>40%), mientras que la luz roja reflejada será menor (8%<30%).



**Figura 1.5.** Porcentajes de reflectancia según la vitalidad de una planta. [6]

## 1.4.5 ArcGIS

El programa ArcGIS es una herramienta para analizar, administrar y compartir información geográfica para permitir la toma de decisiones. En este software se crean mapas para realizar análisis espaciales y administración de los datos [7].

Para la utilización del software ArcGis, en el cálculo de NDVI, se utilizan imágenes satelitales que se encuentran disponibles en la web. Es posible descargar imágenes satelitales de forma gratuita en sitios como Sentinel Playground, EO browser, Copernicus Open Access Hub, USGS Earth Explorer, entre otros [8].

### 1.4.5.1 Teledetección

La técnica de teledetección se basa en recolectar datos de la superficie terrestre mediante dispositivos electrónicos instalados en plataformas espaciales. Los sensores de estas plataformas reciben información a través de la interacción entre la radiación electromagnética que viaja desde el espacio hacia el sensor. Es importante procesar estos datos para poder interpretar la información recolectada [9].

### 1.4.5.2 LANDSAT 8

El satélite Landsat 8 es el octavo en su serie y es utilizado para la observación de la Tierra con una resolución media. Este satélite cuenta con dos instrumentos OLI, y TIRS los cuales permiten capturar imágenes correspondientes a diferentes regiones del espectro electromagnético. Además, las imágenes del Landsat 8 se pueden descargar para que sean cargadas al programa ArcGIS para su análisis.



En la Tabla 1.1 se presentan las regiones en las que el satélite Landsat 8 tiene cobertura. Es importante tener en cuenta que al descargar las imágenes en línea, se debe seleccionar la banda del satélite, por lo que es crucial conocer a que corresponde cada número de banda [10].

**Tabla 1.1** Bandas de trabajo del satélite Landsat 8 [10]

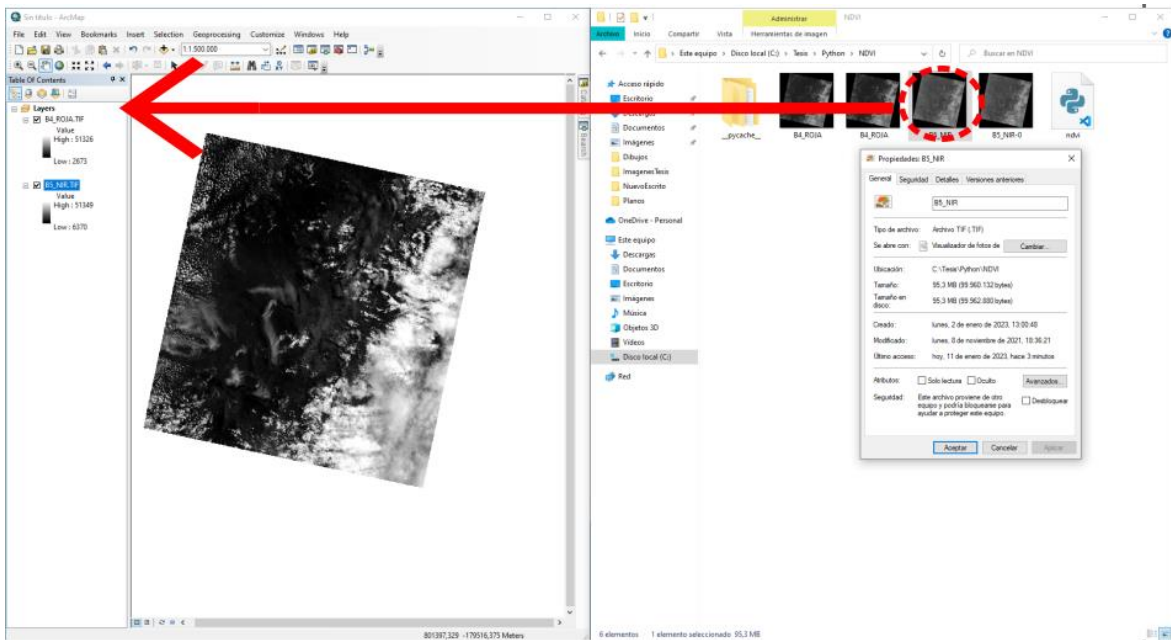
<b>Banda</b>	<b>Resolución (m)</b>	<b>Longitud de onda (um)</b>
Banda 1-Coastal Aerosol	30	[0.435-0.451]
Banda 2-Blue	30	[0.452-0.512]
Banda 3-Green	30	[0.533-0.590]
Banda 4-Red	30	[0.636-0.673]
Banda 5- Near Infrared (NIR)	30	[0.851-0.879]
Banda 6-Short-wave Infrared (SWIR) 1	30	[1.566-1.651]
Banda 7-Short-wave Infrared (SWIR) 2	30	[2.107-2.294]
Banda 8-Panchromatic	15	[0.503-0.676]
Banda 9- Cirrus	30	[1.363-1.384]
Banda 10- TIR 1	100	[10.60-11.19]
Banda 11-TIR 2	100	[11.50-12.51]

Para calcular el índice NDVI en el programa ArcGIS, se utilizan las bandas 5-NIR y 4-RED, las cuales se describen en la tabla 1.1. Es necesario descargar previamente las imágenes del satélite, y luego cargarlas en el programa ArcGIS para realizar el cálculo del índice NDVI.

En el presente trabajo, se pretende diseñar un prototipo que funcione con bandas similares a las del satélite Landsat 8, como se puede observar en los valores de las bandas presentados en la Tabla 1.1.

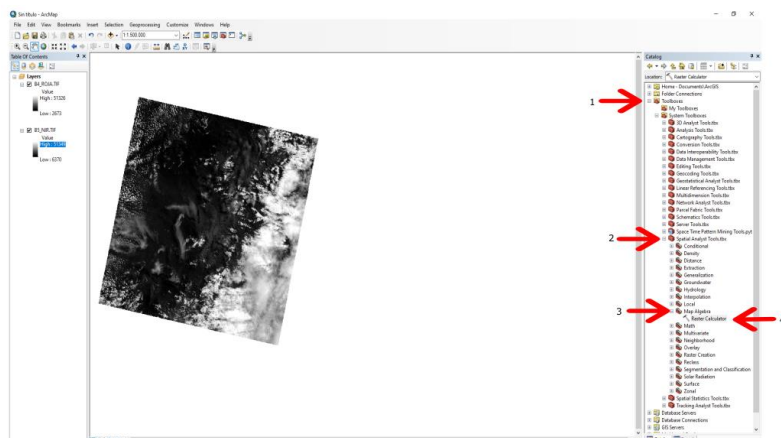
#### **1.4.5.3 Determinación del NDVI con ArcGIS.**

Para calcular el índice NDVI con ArcGIS es necesario descargar imágenes correspondientes a la banda 5-Near Infrared (NIR) y la banda 4-Red del satélite Landsat 8, como se describen en la Tabla 1.1. Luego, se deben cargar estas imágenes en el programa ArcGIS mediante el proceso de arrastrarlas al área de trabajo. Una vez se hayan cargado, se procede a realizar el cálculo del índice NDVI.



**Figura 1.6.** Cargar las imágenes formato .tif a ArcGIS [Fuente Propia]

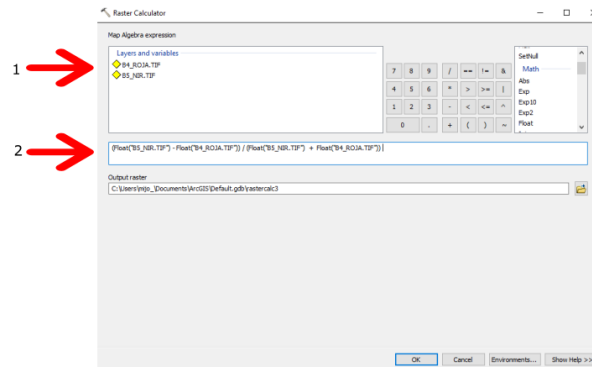
Una vez las dos imágenes en formato .tif han sido cargadas al programa de ArcGIS, se procede a realizar el cálculo pixel a pixel utilizando la herramienta de “Raster Calculation”, que es propia de ArcGIS. El cálculo del índice NDVI se lo realiza con la herramienta Raster Calculator. Esta herramienta se encuentra en el catálogo de ArcGIS en la siguiente ruta: Tolboxes->System Toolboxes->Spatial Analyst Tools-> Map Algebra-> Raster Calculator tal y como se observa en la Figura 1.7.



**Figura 1.7.** Ubicación de la herramienta Raster Calculator [Fuente Propia].

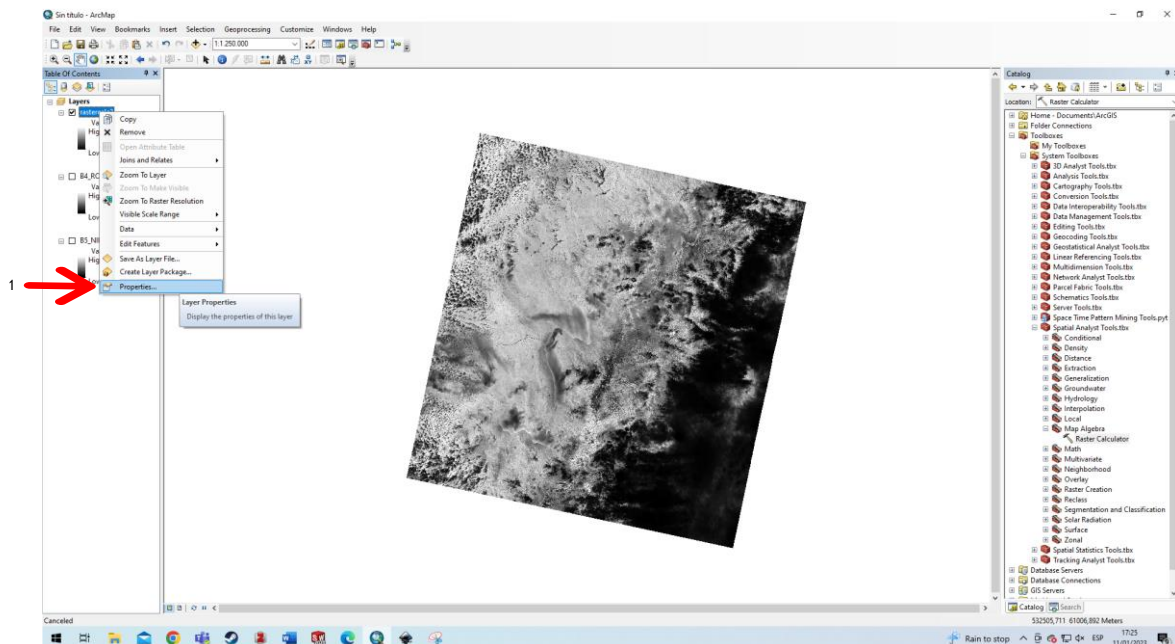
Al dar doble click sobre sobre la herramienta raster calculator se desplegará una ventana en dónde podrá ser ingresada la ecuación del ndvi tal y como se muestra en la Figura 1.8.

- 1) Se muestra los rasters previamente subidos al área de trabajo
- 2) Se visualiza la ecuación ingresada por teclado del NDVI que permitirá el computo pixel a pixel para observar el índice vegetativo NDVI.



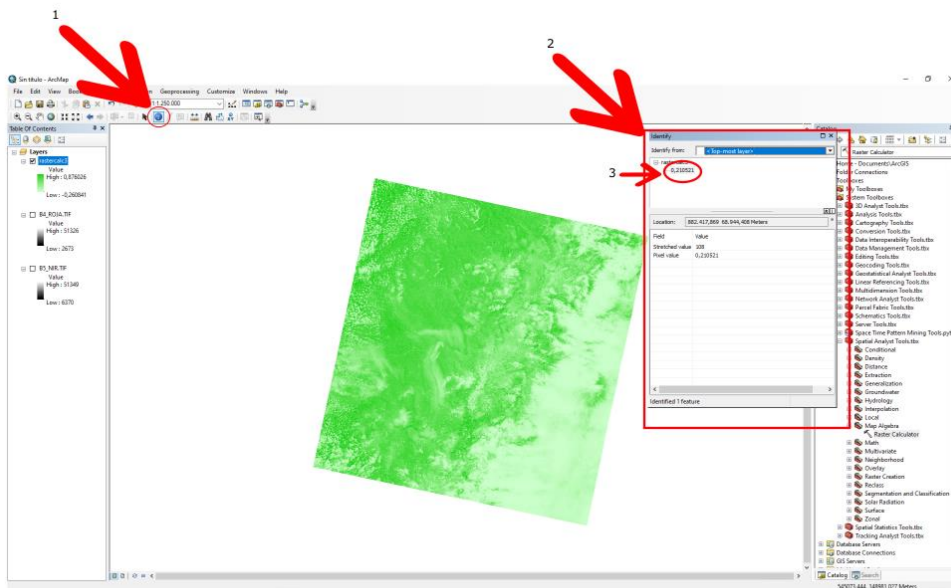
**Figura 1.8.** Calculo del índice NDVI con la herramienta raster Calculator de ArcGIS [6]

Después de dar click en la opción “ok” se creará un nuevo raster de un solo canal que corresponde a la imagen NDVI. Este nuevo raster puede ser configurado ajustando sus propiedades dando click, como se muestra en la Figura 1.9.



**Figura 1.9.** Modificar las propiedades del raster generado.[6]

Finalmente, para visualizar el índice NDVI se da click izquierdo en el ícono de “Identify” (1), lo que hará que se despliegue una ventana (2) con el índice vegetativo del área de interés (3) tal y como se muestra en la Figura 1.10.



**Figura 1.10.** Visualización del índice NDVI del área de interés. [6]

## 1.4.6 Python

Python es uno de los lenguajes de programación más eficientes y versátiles que existen en la actualidad. Es un lenguaje de código libre, lo que permite la creación de librerías para facilitar su uso en el desarrollo de aplicaciones. Existe una gran cantidad de complementos se adaptan a las necesidades de los proyectos, como por ejemplo Django que es utilizado para desarrollar páginas web, PyQt el cual es utilizado para el desarrollo de interfaces gráficas [11]. Además, Python es multiplataforma, lo que permite a los desarrolladores ejecutarlo en las principales plataformas [11].

### 1.4.6.1 OpenCV

OpenCV es una librería ampliamente utilizada para implementar la visión por computadora, la inteligencia artificial y el procesamiento de imágenes. Esta librería es ampliamente utilizada en sistemas inteligentes en dónde sea necesario el procesamiento de imágenes a tiempo real. Entre las aplicaciones de esta librería se encuentra la identificación de objetos, rostros humanos, etc. ya sea en imágenes o videos.

La librería de OpenCV, en conjunto con la librería Numpy, permite el procesamiento y análisis de la estructura de la matriz obtenida. Al obtener una imagen, se genera una variable con datos que van desde 0 a 255, independientemente del tipo de imagen. La matriz generada posee los 3 canales: rojo, verde y azul [12].

### 1.4.6.2 Numpy

Numpy es una de las librerías más importantes a comprender al momento de comenzar a programar en Python. Esta librería permite efectuar rápidamente varios tipos de operaciones con las variables a procesar. Numpy permite crear arreglos (arrays) de n dimensiones, realizar cálculos de algebra lineal, transformación de Fourier y también la implementación de números aleatorios. Por estas características, Numpy es ampliamente utilizada para computación científica.

La librería Numpy es ampliamente utilizada en computación científica y en el desarrollo de aplicaciones que necesitan manejar varios datos numéricos de forma eficiente. Una de las principales bondades es la capacidad de crear y manejar arreglos, una estructura de datos similar a las listas en Python, pero con una mayor velocidad y eficiencia en operaciones matemáticas y estadísticas. Numpy también ofrece funciones para realizar operaciones de algebra lineal y transformaciones de Fourier [13].

### 1.4.6.3 PyQT5

PyQT5 es una biblioteca gráfica de interfaz de usuario (GUI) que es utilizada para crear interfaces gráficas con Python de una forma rápida y sencilla. Esta biblioteca facilita el diseño y programación de la lógica de una interfaz gráfica completa [14].

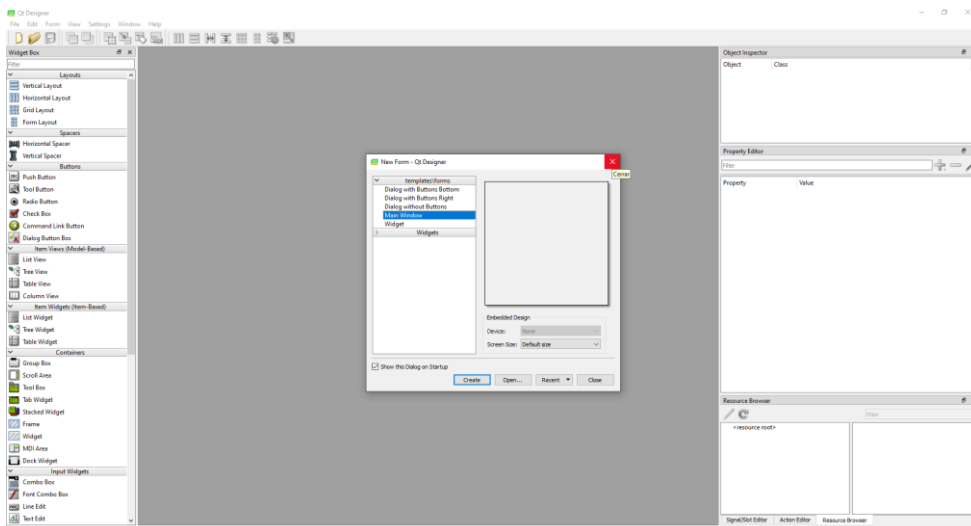


Figura 1.11. Creación de una Interfaz Gráfica [Fuente propia]

### 1.4.6.4 Matplotlib

Matplotlib es una librería de Python que permite la creación de visualizaciones animadas, interactivas o estáticas. Con ella se pueden crear varias figuras, realizar un zoom o desplazarse en ellas para hacerlas interactivas. Además, permite personalizar el estilo visual de las figuras [15]. En el presente trabajo es utilizada para visualizar el índice NDVI.

### 1.4.6.5 Procesamiento de Imágenes

El procesamiento de imágenes es un conjunto de algoritmos que se utilizan para transformar una imagen de entrada en otra imagen de salida con características diferentes. Sin embargo, aunque la salida puede tener características distintas a la imagen de entrada, sigue siendo una imagen. Por otro lado, la visión por computadora busca obtener información sobre la imagen o ciertas características de ella, en lugar de simplemente transformarla [16].



**Figura 1.12.** Creación de una Interfaz Gráfica [Fuente propia]

La imagen habitualmente obtenida directamente de una cámara comercial posee los 3 canales RGB y para que el programa pueda leer la imagen, se utiliza la función “imread()”. Esto genera una variable en donde se pueden extraer propiedades importantes de la imagen, como el número de píxeles y las dimensiones de la misma.

```
import cv2
import numpy as np
var_imagen = cv2.imread('Fotografia 1.jpg')
var_imagen = np.array(var_imagen , dtype=float)
print("Número de Píxeles: " + str(var_imagen.size))
print("Dimensiones: " + str(var_imagen.shape))
```

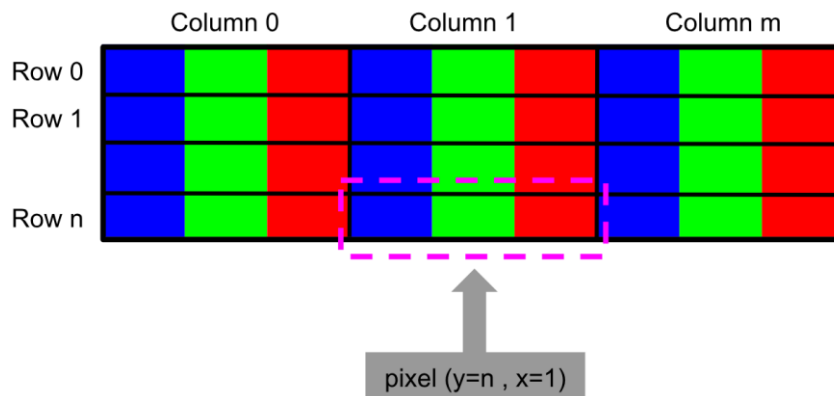
**Figura 1.13.** Adquisición de una imagen desarrollado en Python [Fuente propia]

Si ejecutamos el anterior código en cualquier entorno de programación (en este caso Vscod) se logran los siguientes resultados para una imagen.

```
Número de Píxeles: 2764800
Dimensiones: (720, 1280, 3)
```

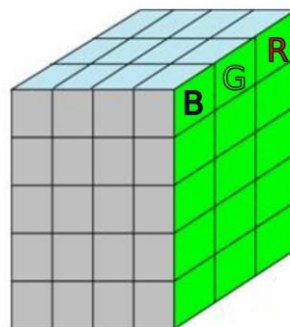
**Figura 1.14.** Salida en Consola a partir del código ejecutado [Fuente propia]

Existe varias formas de comprender cómo se genera la variable de la imagen adquirida. Una de ellas es crear una variable numpy array, equivalente a una matriz de 720x1280, dónde cada elemento de la matriz contiene una tupla de 3 datos (BGR). Cuando se utiliza la librería OpenCV, el orden de los canales es BGR debido a que de esta manera lo han implementado los desarrolladores [17].



**Figura 1.15.** Formato de la representación de toda una imagen en Opencv [17]

La segunda forma de comprender la creación de la variable es imaginarse un dato cúbico en dónde se tienen almacenados la intensidad de cada uno de los colores de cada píxel que conforma la imagen de la siguiente manera.



**Figura 1.16.** Creación de la variable numpy Array [Fuente propia]

### 1.4.7 Filtro Infrarrojo en cámaras comerciales (Filtro IR)

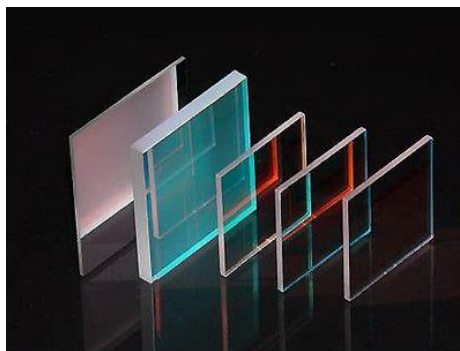
Para calcular el índice NDVI es esencial obtener una imagen de radiación infrarroja (NIR). tal y como se expone en la ecuación (1.1). Muchas cámaras comerciales vienen con un

filtro que bloquea esta radiación, por lo que es necesario retirar el filtro para permitir que el sistema óptico capte la radiación NIR.

Las cámaras suelen incluir un filtro que permite pasar la luz visible, pero bloquea la radiación infrarroja. La radiación infrarroja es reflejada o absorbida pero no pasa a través de él.

La luz de la banda de la radiación infrarroja permite que una cámara capture imágenes en condiciones de poca luz, es decir, tiene visión nocturna. Sin embargo, las imágenes capturadas por una cámara sin un filtro IR pueden presentar colores distorsionados en la fotografía. Además, la ausencia de un filtro IR no afecta la resolución de la imagen.

Muchas veces el filtro IR no es removible debido a que la construcción de una cámara con un filtro removible puede ser más costosa que una cámara sin esta característica. Además, las cámaras económicas a menudo no tienen la capacidad de remover el filtro con facilidad [18].



**Figura 1.17.** Filtro Infrarrojos [19]

## 1.4.8 Software

### 1.4.8.1 Solid Works

Solidworks es un software de tipo CAD, que permite crear modelos en 3D, dibujos y piezas ensambladas. Con él, es posible generar archivos .stl que son utilizados para la fabricación de productos. Solidworks cuenta con una amplia variedad de herramientas para garantizar que las dimensiones del producto se ajusten al diseño especificado en el programa [20].





**Figura 1.18.** Logo de Solidworks [20].

### **1.4.8.2 Visual Studio Code**

Visual Studio Code es un editor de código gratuito que permite a los usuarios a programar en varios lenguajes de programación como Python, Java, JavaScript, C++, entre otros. Una de las ventajas de utilizar esta herramienta es que permite al desarrollador identificar rápidamente el tipo de dato ya que Visual Studio Code utiliza diferentes colores para los identificadores. Además, permite completar líneas de código mientras el desarrollador está creando su aplicación. La aplicación Visual Studio Code permite un alto grado de personalización, permitiendo cambiar entre cientos de temas el color que más se acople a las preferencias del programador [21].



**Figura 1.19.** Logo de Visual Studio Code[21].

### **1.4.8.3 QT-Designer**

QT-Designer es una herramienta para la creación de interfaces gráficas de usuario (GUI) que permite a los desarrolladores mejorar la productividad y eficiencia de su aplicación. Con este software, se pueden arrastrar y soltar objetos llamados QWidgets, que pueden ser programados en Python para adquirir características específicas. Además, cada elemento de la interfaz gráfica se puede personalizar mediante la herramienta “stylesheet”.

Existen dos formas para crear una interfaz gráfica: el primer método consiste en utilizar la aplicación QT-Designer, en la cual se pueden arrastrar y configurar objetos QWidgets a los lugares deseados y configurarlos, o el otro método que consiste en programar manualmente todo desde Python, se programa desde la ubicación exacta de los objetos, hasta su apariencia. Cualquiera de los dos métodos es válido para la creación de un GUI(Interfaz Gráfica de Usuario) [22].

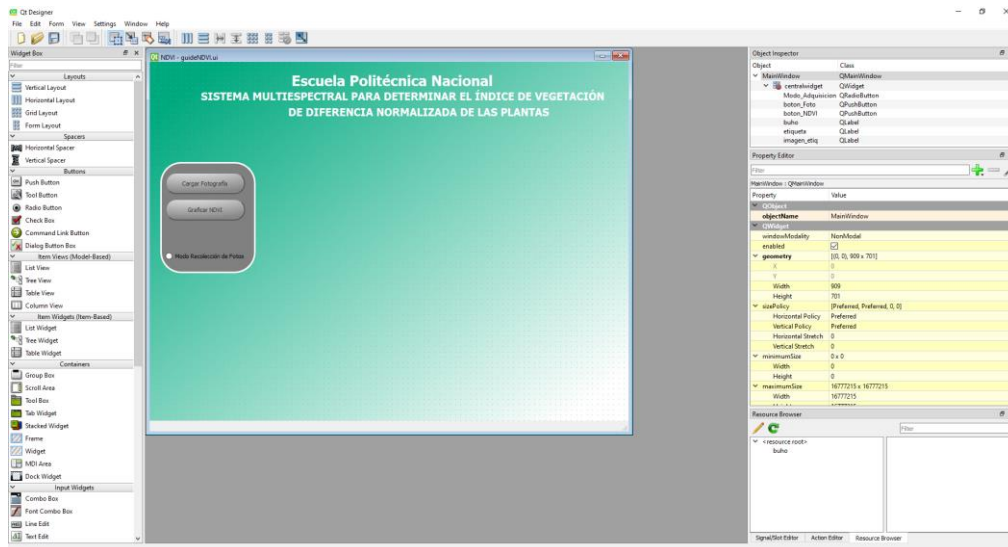


Figura 1.20. Programa QT-Designer [Fuente Propia].

#### 1.4.8.4 Thonny

Thonny es un entorno de desarrollo que permite a programadores sin mucha experiencia empezar a programar en lenguaje Python. Su estilo minimalista permite facilidad a principiantes comenzar a comprender el lenguaje de alto nivel Python. Este software está disponible para Windows, Linux y viene preinstalado en el sistema operativo Raspbian, el cual es instalado en una Raspberry Pi [23].

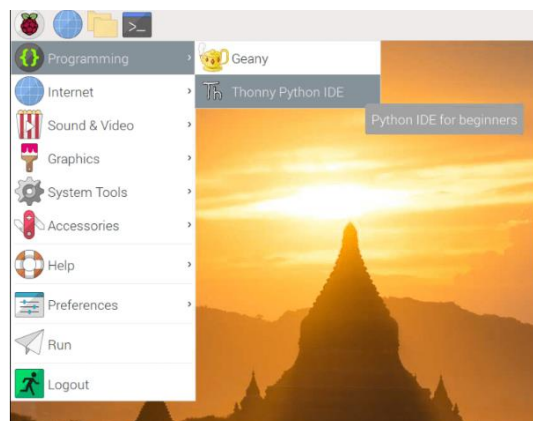


Figura 1.21. Thonny en el sistema operativo Raspbian [23].

### 1.4.9 Elementos del Sistema Multiespectral

### 1.4.9.1 Raspberry pi 3 B+

La raspberry pi 3 B+ es un pequeño ordenador de placa reducida, lo que hace compacto que un computador común. Los miniordenadores Raspberry Pi pueden realizar cualquier actividad o tarea básica que puede ejecutar un computador común, como navegar por internet, usar office, reproducir de archivos multimedia y también puede correr algunos videojuegos [24].

La Raspberry Pi, al ser un pequeño ordenador, posee el hardware similar al de un ordenador convencional, como memoria RAM un CPU, puertos USB, salida tanto de audio como de video, puerto Ethernet, espacio para una tarjeta microSD y puerto de alimentación. Sin embargo, no cuenta con un interruptor de encendido y apagado. Para que comience a funcionar es necesario instalar un sistema operativo, el más comúnmente utilizado es Raspbian en la tarjeta microSD. Una vez instalado, se deben conectar los periféricos como ratón, teclado, pantalla, etc. para que el dispositivo pueda funcionar correctamente [24].

**Tabla 1.2** Datos técnicos de la raspberry pi 3 B+ [24].

Modelo	FRECUENCIA DEL RELOJ	RAM	NÚMERO DE PUERTOS USB	WIFI/BLUETOOTH	PUERTO ETHERNET	Corriente a plena carga
Raspberry pi 3 B+	1500 Mhz	1 Gigabyte	4	Si	Si	2.5A

### 1.4.9.2 Filtros Ópticos

Según Mid Opt, especialista en filtros de visión artificial, es posible obtener el índice NDVI a través de una cámara económica retirando el filtro IR de la cámara. Luego, se extraen los canales rojo y azul de la imagen capturada para obtener los datos correspondientes a la imagen RED y a la imagen NIR para el cálculo del índice NDVI. Con la ayuda del filtro ubicado frente al sensor de la cámara, el canal azul ya no corresponde a la imagen azul, sino que resulta en la imagen NIR. De esta manera la manera del cálculo del índice NDVI cambia a la ecuación (1.2) [25].

$$NDVI = \frac{BLUE - RED}{BLUE + RED} [25] \quad (1.2)$$

### 1.4.9.3 Cargas de una batería ion Litio

Las baterías tienen un cátodo y un ánodo. En el caso de las baterías de iones de litio, el ánodo se realiza con grafito y cátodo con óxido de cobalto. Se puede observar en la Figura 1.22. que es necesario suministrar una corriente constante para cargar una batería. Cuando la batería alcanza el voltaje máximo, la corriente comienza a disminuir proporcionalmente hasta que la corriente es menor del 3% de la corriente nominal de la batería [26].

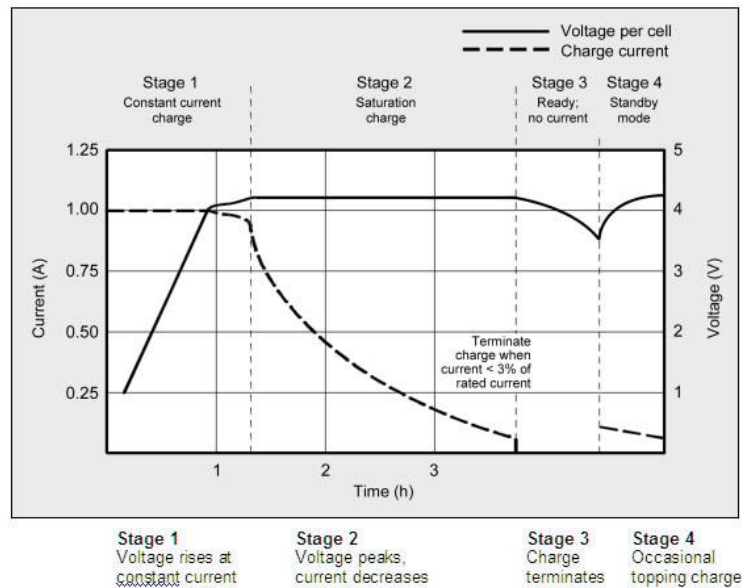
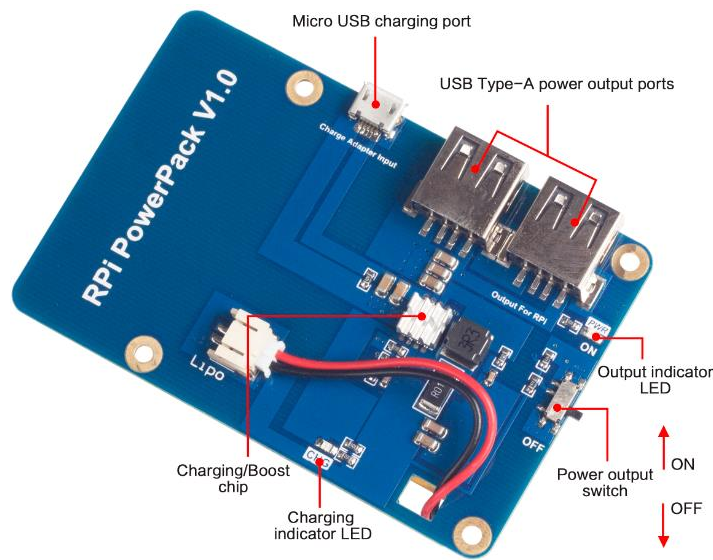


Figura 1.22. Gráfica de carga y descarga de una batería Litio[26] .

### 1.4.9.4 Módulo de carga para una batería de Litio

La Raspberry Pi puede funcionar de manera portátil mediante una batería de polímero de litio de 3.7 V, y su controlador de carga. El controlador de carga cuenta con funciones de encendido del sistema, un indicador LED de encendido del sistema y un indicador que permite conocer si la batería de polímero de litio de 3.7V se está cargando [27].



**Figura 1.23.** Módulo de carga para una batería de polímero de Litio[27] .

**Tabla 1.3** Datos técnicos del controlador de carga de la batería de polímero de Litio [27].

Capacidad de la batería	Máxima Corriente de descarga	Salida de Voltaje sin Carga	Voltaje de carga	Corriente de Carga	Voltaje de Corte a plena carga
3800 mAh	1.8A	5.1V $\pm$ 0.1V	5V	1A	[4.18-4.2]V

El controlador de carga viene con una batería de 3800mAh pero si es posible reemplazarla con una de mayor capacidad, entre 5000mAh y 10000mAh de acuerdo al fabricante [27].

#### 1.4.9.5 Cámara Digital

Para que el sistema multiespectral no tenga que extraer las fotografías y transferirlas a un computador, es necesario utilizar una cámara que pueda funcionar como cámara web. De esta manera, se guarda el fotograma actual del video cada vez que se presiona el botón conectado a una entrada digital de la Raspberry pi 3 B+. Debido a las prestaciones, una buena resolución y la funcionalidad de cámara web se ha seleccionado la cámara QY-08K.

**Tabla 1.4** Datos técnicos la cámara comercial QY-08 [28]

Resolución	Almacenamiento de una tarjeta microSD	Tipo de conector USB	Dimensiones mm	Peso con batería /sin batería	Capacidad de la batería	Resolución de las fotos
4k/2.7k/1080p/720p	1.8A	2.0	59x41x29mm	58g/44g	750 mAh	16m/12m/10m/8m/5m/3m/2m/1.3



**Figura 1.24. Cámara deportiva tipo QY-08K**

#### **1.4.9.6 Pantalla táctil de 3.5 pulgadas.**

Los miniordenadores de la serie de la raspberry pi 3 B+ son computadoras pequeñas que no cuentan con pantalla incorporada. Por lo tanto, se pueden utilizar de 3 maneras diferentes: conectando la raspberry pi 3 B+ a una red y utilizando SSH o VNC, conectando la Raspberry pi a un computador, o utilizando una pantalla táctil que vienen en diferentes tamaños.

Para el proyecto actual, se ha utilizado una pantalla de 3.5 pulgadas conectada directamente a los pines de la Raspberry Pi 3 B+. En la tabla 1.5 se muestran las características técnicas de la pantalla LCD de 3.5 pulgadas.



**Figura 1.8. Pantalla táctil de 3.5 pulgadas para la raspberry pi [29]**

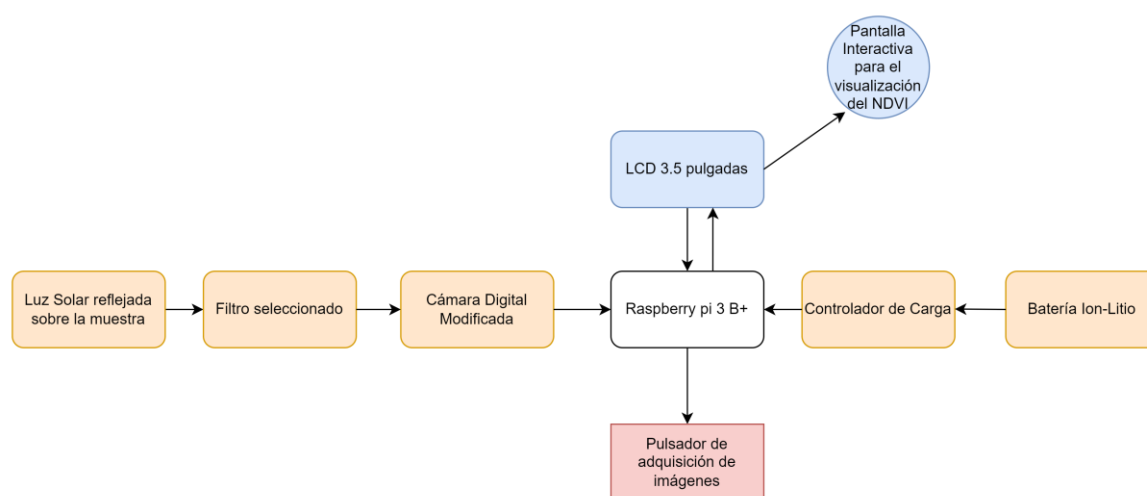
**Tabla 1.5** Especificaciones técnicas del display LCD 3.5”

Resolución	Táctil	Driver	Interfaz (SPI)	Dimensiones	Área efectiva	Lápiz óptico
480x320	resistivo	XPT2046	59x41x29mm	86*55*20mm	74*50 mm	Si

## 2 METODOLOGÍA

### 2.1 Esquema del Sistema General

El componente desarrollado cuenta con un sistema óptico que permite capturar la banda del infrarrojo cercano y la banda del rojo del espectro electromagnético. La información es almacenada en una raspberry pi 3 B+. Gracias a la interfaz desarrollada, el usuario puede cargar la imagen adquirida por el prototipo y realizar el cálculo del índice NDVI, el cual es mostrado en la pantalla táctil de 3.5 pulgadas conectada a la raspberry pi 3 B+. Para finalizar el trabajo, se ha diseñado e implementado una estructura que sostiene todos los componentes del sistema multiespectral.



**Figura 2.1.** Esquema General de todo el sistema multiespectral.

### 2.2 Sistema Óptico

Se ha desarrollado un sistema óptico para que una cámara convencional pueda capturar imágenes en la luz infrarroja. Gracias a esto, la cámara es capaz de adquirir imágenes en diferentes bandas del espectro electromagnético. La fotografía adquirida tendrá 3 canales, incluyendo las bandas RED y NIR. Posteriormente, se llevará a cabo un proceso de

segmentación para separar estas dos bandas, y se calculará el índice vegetativo NDVI. Para lograr esto, se modificó una cámara comercial retirando el filtro IR que impedía que la cámara capturara la banda del infrarrojo cercano (NIR) [18]. Además, se añadió un filtro seleccionado mediante pruebas con un espectrómetro, que se ubica delante del sensor de la cámara para que el canal BLUE de la imagen RGB adquirida corresponda al canal NIR y el canal RED corresponda a la imagen RED [25].



**Figura 2.2.** Sistema Óptico del prototipo multispectral.

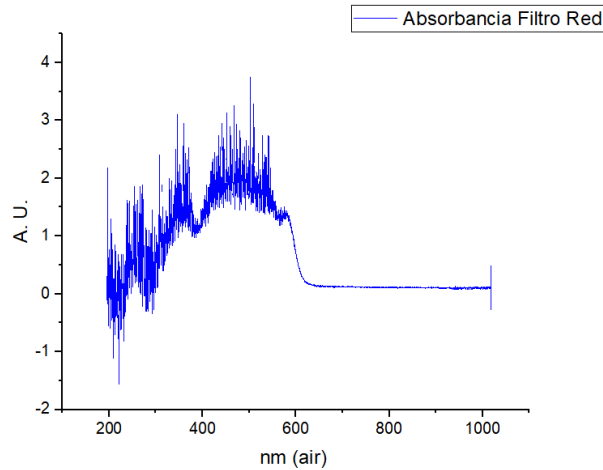
En la Figura 2.2 se presenta un esquema general del prototipo implementado que permite la captura de las fotografías multispectrales y almacenamiento de estas imágenes en la memoria de un miniordenador. Posteriormente, estas imágenes se utilizan para calcular el índice NDVI en el mismo dispositivo.

### 2.2.1 Filtro Seleccionado

Para determinar el índice NDVI con una cámara comercial que no está diseñada para este propósito, según los estudios realizados en el presente trabajo, es necesario utilizar un filtro rojo que permita pasar la banda del infrarrojo cercano, ya que de esta manera se reemplazará la información del canal azul con la información sobre el NIR [25].

Con el objetivo de aplicar la ecuación 1.1 del NDVI se han realizado pruebas con varios filtros y el filtro rojo de la marca Yadsux, mostrado en la Figura 2.4, ha proporcionado resultados más precisos cercanos a la banda 4 del satélite Landsat 8, que se muestra en la Figura 2.3, ya que este permite el paso del NIR. Este filtro es esencial para realizar el proceso de segmentación. Los resultados de las pruebas con un espectrómetro se han graficado en el programa Origin 8 Pro.





**Figura 2.3.** Curva de absorción del Filtro RED.

En la Figura 2.3 se muestra la curva de absorción del Filtro RED, donde se puede ver cómo el filtro absorbe ondas del espectro electromagnético con longitudes de onda desde 195.78 nm hasta 626.308 nm. El valor 626.308nm se acerca al valor de 636 nm utilizado en la Banda 4-Red del satélite Landsat 8 según se indica en la Tabla 1.1. Este filtro permite el paso de longitudes de onda desde 626.308 nm.



**Figura 2.4.** Filtro rojo con montura de 58mm de la marca Yadsux [30] .

### **2.2.2 Modificación de la cámara digital comercial para que esta capte el infrarrojo cercano.**

Para cualquier cámara digital para capturar la banda NIR, es necesario retirar su filtro IR ubicado frente al sensor de la cámara. Este proceso se lo realizó en la cámara QY-08k. Al reemplazar el filtro IR con un filtro NIR. La prueba para determinar la capacidad del sensor para detectar la radiación infrarroja produjo imágenes que se muestran en la Figura 2.5, que muestran una alta densidad de vegetación.

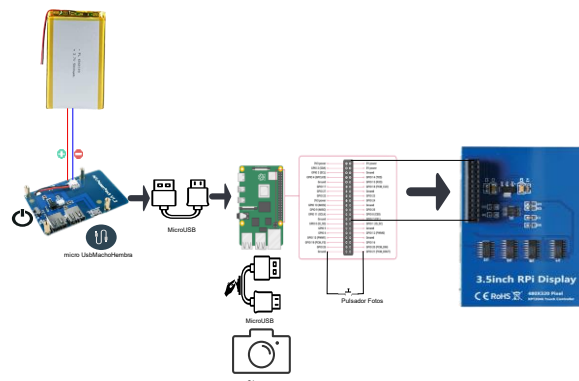


**Figura 2.5. a)** Fotografía con un Filtro NIR de la Figura 2.5, **b)** Fotografía sin ningún Filtro. [Fuente Propia]

Tras las pruebas con el espectrómetro en el filtro NIR y la adquisición de una imagen NIR, se puede afirmar que el sistema multispectral, debido a la eliminación del filtro IR, es capaz de capturar imágenes en el infrarrojo cercano.

## 2.3 Sistema Electrónico

En el sistema electrónico hay un botón para tomar fotos y dos interruptores. Uno enciende el sistema y el otro enciende la cámara web.

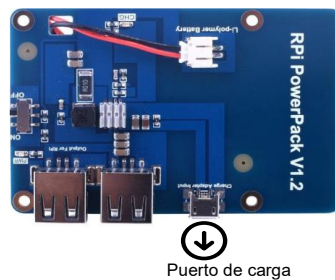


**Figura 2.6. Esquema Técnico.** [Fuente Propia]

En la Figura 2.6 se puede visualizar todos los componentes del sistema electrónico implementado para que el prototipo funcione de forma portátil y sea recargable.

## 2.4 Sistema para cargar la batería

Se ha utilizado un controlador de carga comercial para hacer que la Raspberry Pi 3 B+ funcione de forma portátil. Este controlador de carga está diseñado específicamente para los modelos de la Raspberry Pi y tiene un puerto micro-USB, como se ve en la Figura 2.7. Se conecta un cargador a este puerto micro-USB para cargar la batería de Ion-Litio [27].



**Figura 2.7.** Puerto de carga. [Fuente Propia]

El controlador de carga funciona cambiando la posición del interruptor de encendido y proporciona 5 voltios en sus puertos USB con una corriente de salida de 1.8A. Según el fabricante, la batería incluida de fábrica puede proporcionar hasta 9 horas de autonomía a la Raspberry Pi [27].

### 2.4.1 Consumo de Corriente de los componentes

La raspberry pi 3 B+ puede requerir diferentes corrientes dependiendo de la velocidad del procesador, por ello en la Tabla 1.6 se ha utilizado el valor del corriente más alto que se puede presentar en la raspberry pi 3 B+.

**Tabla 1.6** Consumo individual de cada componente del prototipo implementado extraído de los datasheets.

Elemento	Corriente
Cámara Digital	65mA
Controlador de carga	3.5mA
Raspberry pi 3 B+	2500mA
Pantalla LCD 3.5"	250mA
<b>Corriente Total</b>	<b>2818.5mA</b>

$$Autonomía = \frac{\text{Capacidad que posea la batería de Ion – Litio}}{\text{Corriente total de sistema multispectral}} \quad (1.3)$$

$$Autonomía_{\text{minutos}} = \frac{5000 \text{ mAh}}{2818.5 \text{ mA}} = 1.77399 \text{ h} * \frac{60 \text{ minutos}}{1 \text{ hora}} \quad (1.4)$$

$$= 106.4395 \text{ minutos}$$

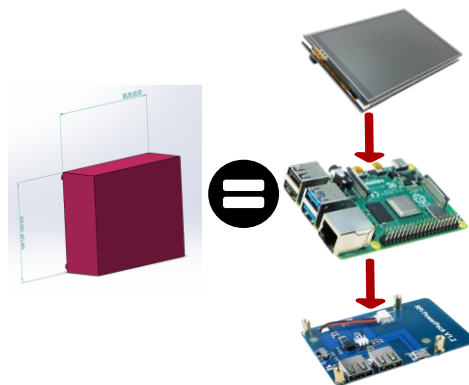
$$Autonomía = 106 \text{ minutos} \quad (1.5)$$

## 2.4.2 Selección de la batería

Se ha reemplazado la batería de ion litio de 3000mAh del cargador comercial que viene con el controlador de carga de la Figura 1.23 con una de 5000mAh, según los consumos energéticos de cada componente. Este cambio es posible gracias a que el fabricante indica que puede cambiarse por una batería hasta de 10000mAh [27].

## 2.5 Sistema Mecánico

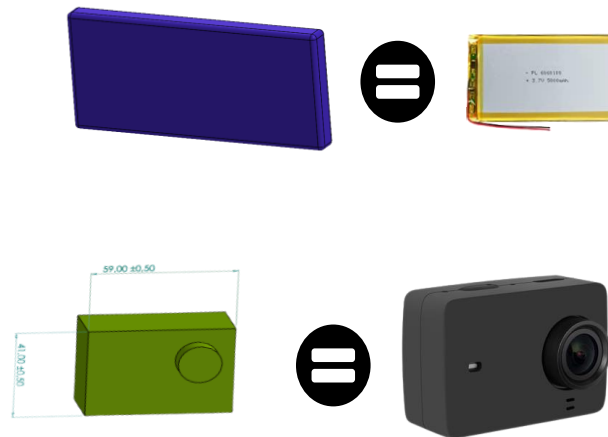
Se ha creado un objeto cúbico, el cual no va a ser impreso, pero sirve para conocer la ubicación final de la pantalla de 3.5 pulgadas en el prototipo portátil. Los planos de este objeto son presentados en el Anexo A. En la Figura 2.8 se muestran los 3 componentes que son llevados al software de diseño de piezas 3D.



**Figura 2.8.** Pantalla de 3.5 pulgadas llevada solidworks

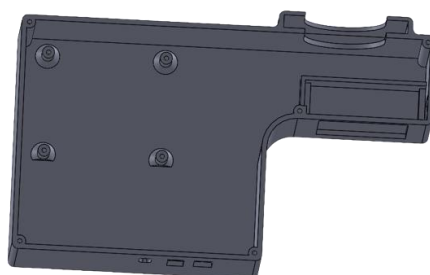
Se han creado objetos cúbicos en SolidWorks que representan los componentes reales, como la batería y la cámara dentro para conocer la posición que tendrán en el prototipo desarrollado, como también las dimensiones que estos ocuparán en el prototipo. Estos

objetos poseen las mismas dimensiones que los componentes reales y estas dimensiones se presentan en el Anexo A.



**Figura 2.9.** Batería de ion Litio y cámara llevada a solidworks.

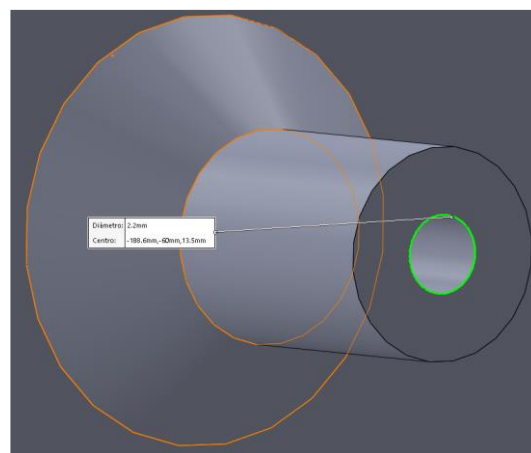
El prototipo cuenta con un sistema mecánico del prototipo asegura la sujeción de todos los componentes del sistema multispectral. Es necesario la sujeción de: la raspberry pi 3 B+, el controlador de carga, la pantalla LCD 3.5”, la batería de ion de litio de 5000 mAh, la cámara comercial modificada, los interruptores, pulsadores y el cable hembra macho hembra micro-usb para la carga de la batería de ion de litio. Para la sujeción de todos los componentes se diseñó la estructura que se muestra en la Figura 2.10.



**Figura 2.10.** Estructura de sujeción de todos los componentes del sistema multispectral. El sistema posee una cubierta en dónde se sitúan un interruptor, los espacios para manipular la cámara web y la pantalla táctil de 3.5”. Además de eso unos ganchos para lograr sujetar el lápiz óptico que es necesario para interactuar con la interfaz desarrollada en QT-Designer. En la impresión de las piezas en 3d, se imprimió con boquilla 0.4 que

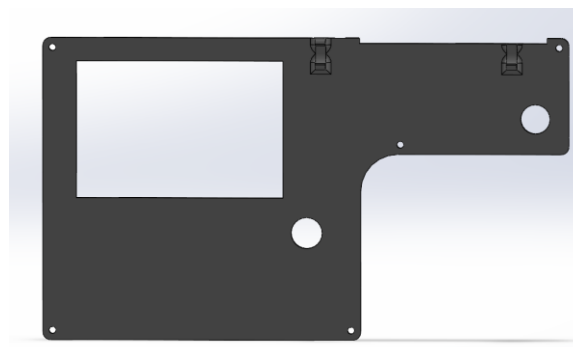
brinda mejores acabados en el diseño, y a una velocidad lenta para no tener problemas con la resistencia del material PLA premium de 1.75mm.

Se han utilizado separadores para sujetar el conjunto pantalla-Raspberry Pi-controlador de carga y estos están sujetos a cuatro agujeros de la base de la estructura. Se ha realizado un chaflán en la base del cilindro para aumentar la robustez de la pieza donde se aplicará esfuerzo al colocar los sujetadores, como se muestra en la Figura 2.11.



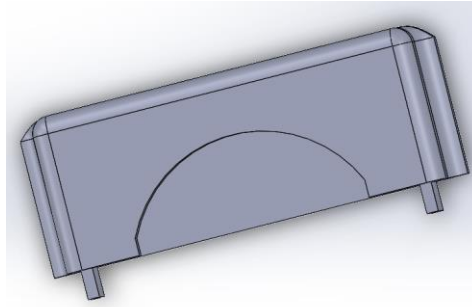
**Figura 2.11.** Chaflan para los sujetadores.

También se ha diseñado la tapa del prototipo portátil que cubre los componentes electrónicos y permite acceder a la interfaz gráfica, sujetar el lápiz óptico y usar los botones de OK de la cámara y el pulsador de captura de imágenes. La tapa se atornilla a la base gracias a los agujeros de 2.2 mm para permitir que pasen los tornillos de 2.5mm. Se requiere presión inicial para lograr formar la rosca.



**Figura 2.12.** Estructura para la manipulación del prototipo portátil.

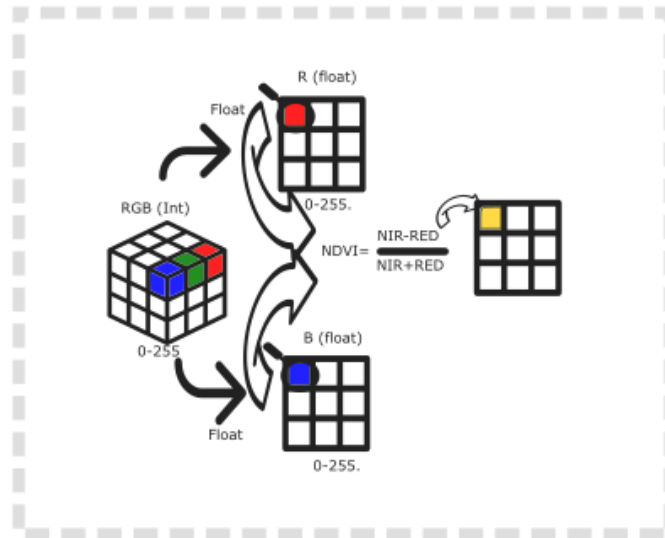
Con el objetivo de cubrir el filtro óptico seleccionado, se diseñó una estructura que permite cubrir los rayos solares, para evitar que las ondas de luz se reflejen en el lente y cause la aparición de objetos no deseados en la fotografía final. Así, el prototipo queda totalmente cerrado con los 3 componentes.



**Figura 2.13.** Tapa del Filtro Óptico.

## 2.6 Programación.

En la figura 2.14 se muestra el flujo de la información y cómo se transforma para calcular el índice vegetativo NDVI a partir de la imagen multiespectral y almacenar los índices vegetativos en la memoria. Primero, la imagen se almacena como una variable cúbica de 3 dimensiones en el miniordenador. Luego, se extraen dos variables de 2 dimensiones correspondientes a imágenes de un solo canal, eliminando información de otros canales. En este proceso tanto las filas como las columnas son invariables, pero se elimina la información de los otros canales (la profundidad del cubo). La librería numpy se utiliza para operar las matrices y almacenar los valores NDVI correspondientes a la imagen. Es importante saber que Opencv ordena los valores de los pixeles en BGR, no RGB como lo hace matplotlib. La transición de la matriz cúbica a matrices de dos dimensiones también implica la conversión de valores enteros a flotantes. Dicha conversión de datos también lo realiza ArcGIS con la herramienta de raster calculation.



**Figura 2.14.** Flujo de Información con el programa implementado en Python [Fuente Propia].

Se utilizó el entorno de desarrollo VScode en un computador y Thonny en una Raspberry Pi 3 B+ para ejecutar la interfaz gráfica y su programación. VScode, puede ser instalado en la raspberry pi 3 B+, pero no es utilizable debido a las limitaciones de hardware. En ambos entornos requieren la instalación de las librerías y complementos necesarios para ejecutar el código. Para la instalación de dichos complementos y librerías se debe utilizar el símbolo del sistema.

Para la extracción de los 3 canales, de los cuales solo van a ser utilizado dos, se ha utilizado sintaxis de Python que resulta similar a Matlab. La figura 2.15 se indica la sintaxis para la extracción de los canales BGR [32]. En este punto se verifica que el orden en opencv de los canales es BGR más no RGB.

```
B = img_OpenCV[:, :, 0]
G = img_OpenCV[:, :, 1]
R = img_OpenCV[:, :, 2]
```

**Figura 2.15.** Extracción de los canales BGR de la imagen.



### 2.6.1 Adquisición de la Imagen.

El prototipo incluye una cámara web y con ella se pueden tomar fotos. En este documento se investigaron la sintaxis y líneas de código en Python para controlar la cámara y tomar fotos. Para que el prototipo sea capaz de detectar una cámara web es necesario la utilización de la librería Opencv y la instalación del controlador de la cámara. El driver, llamado fswebcam, se instala desde la consola de la Raspberry Pi[33]. Con ayuda de este driver se puede acceder a cualquier cámara web que se conecte a un puerto USB de la raspberry pi, no solamente las picams. Se utiliza el método `cv2.VideoCapture()` para acceder a la cámara, como se muestra en las siguientes líneas de código.

```
argumento = '/dev/video0'  
objeto_videoCapture=cv2.VideoCapture(argumento)
```

En el caso de un ordenador convencional, el argumento para instanciar el objeto de cámara debe ser 1 o 0, mientras que en la Raspberry Pi se utiliza “/dev/video0”. Al crear la variable objeto, se tiene acceso a los métodos de la clase padre, lo que facilita su uso.

Para la adquisición de la imagen BGR se utiliza el método `read()` de la clase padre, este método tiene como retorno dos variables que son almacenadas para su uso posterior.

```
ret, frame= objeto_videoCapture.read()
```

En este punto, la variable “frame” tendrá una estructura de 3 dimensiones con los canales BGR. Para abstraer este tipo de variable visualice la Figura 1.16.

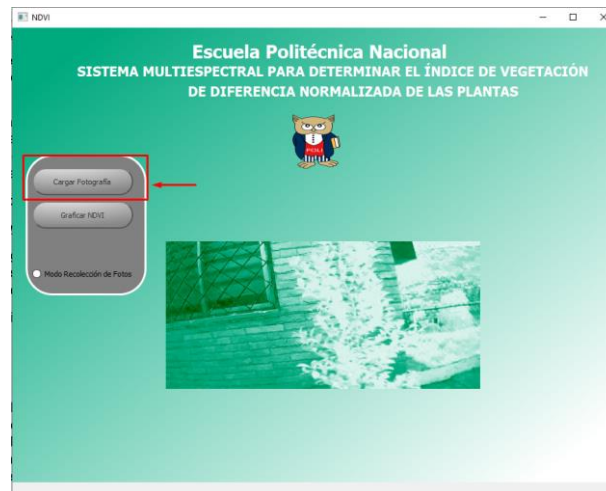
Para visualizar el video de la cámara web, se puede mostrar la imagen actual almacenada en la variable “frame” con el método `imshow()`. De esta manera se mostrará el video, el cual puede verse limitado por la velocidad del procesador, tanto en Raspberry Pi como en un ordenador. Además, para evitar que la imagen salga de los límites de la pantalla, se utiliza el método `cv2.pyrDown(frame)`, que permite reducir su tamaño. Finalmente, la imagen es mostrada en pantalla.

```
frame_mostrar=cv2.pyrDown(frame)  
cv2.imshow('Video',frame_mostrar)
```

Para capturar una imagen en formato .jpg se utiliza el método `cv2.write()`. Este método se lo utilizará cada vez que el pulsador sea presionado, considerando el fenómeno de los rebotes y activando la resistencia pull up interna de la Raspberry Pi 3 B+.

Una vez creada la imagen .jpg, se puede cargar en el programa a través de un botón en la interfaz gráfica. Al presionar el botón se invoca una función previamente creada en el

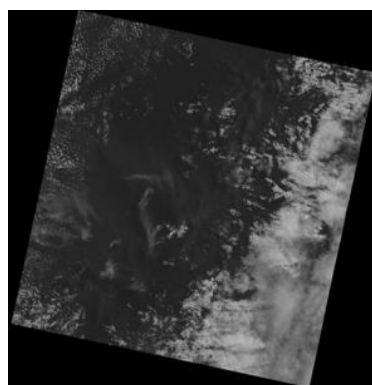
programa. Esta función devuelve la ruta de la imagen seleccionada, que luego es utilizada como argumento por el método `cv2.imread()` para obtener la variable cúbica.



**Figura 2.16.** Botón para cargar la Fotografía adquirida por el equipo.

## 2.6.2 Segmentación de la Imagen

La imagen capturada por el prototipo es una imagen de 3 canales, lo que significa que contiene información sobre la intensidad de azul, del verde, y rojo de cada uno de los píxeles de la imagen. Para calcular el índice NDVI, es necesario segmentar la imagen de 3 canales en imágenes de un solo canal. En la Figura 2.17 se observa la representación de una imagen de un solo canal.

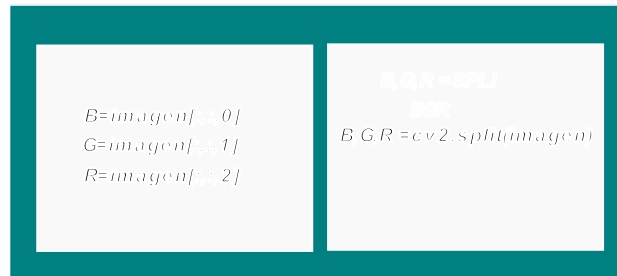


**Figura 2.17.** Raster de Imbabura del satélite Landsat 8 [34]

En la Figura 2.17 muestra una imagen en escala de grises con un solo canal, donde el valor máximo del píxel corresponde a blanco y el valor 0 a negro. Las imágenes satelitales

tienen mucha información gracias a su formato .tif, en comparación con las imágenes .jpg o .png que tienen un valor máximo de 255 (8bits).

La Figura 2.18 muestra dos métodos para extraer los canales RGB en Python. En el presente trabajo se ha utilizado el método del recuadro izquierdo.



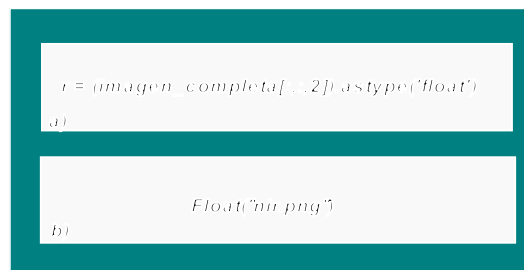
**Figura 2.18.** Métodos para obtener los canales de una Imagen BGR en Python

Se debe tener en cuenta que OpenCV utiliza el orden BGR en lugar de RGB, por lo que el orden de las variables es tal y como se ilustra en la Figura 2.18.

Para el cálculo del índice NDVI, es necesario tener dos imágenes de un solo canal: una obtenida del canal azul y otra del canal rojo. La imagen del canal azul corresponde a la imagen NIR, mientras que la imagen roja corresponde a la imagen RED en la ecuación (1.1).

### 2.6.3 Cálculo del Índice NDVI

ArcGIS convierte los valores de la imagen de un solo canal en valores flotantes antes de realizar el computo del índice NDVI. La programación en lenguaje Python también lo hace, como se muestra en la Figura 2.19. En el caso a) se extrae el canal rojo y se convierten los valores a flotantes. En el caso b) se muestra como ArcGIS convierte una imagen de un solo canal a flotantes.



**Figura 2.19.** a) Transformación de enteros a valores flotantes en Python b) Transformación de valores enteros a valores flotantes usando ArcGIS.

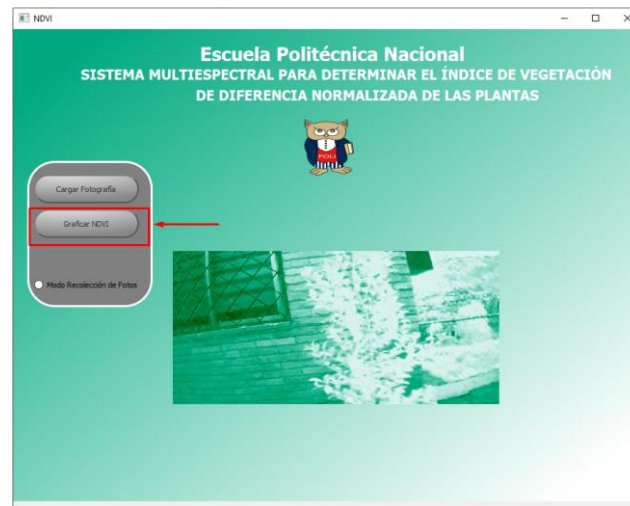
En la Figura 2.19 se puede observar como Python puede realizar el mismo proceso que ArcGIS, que es convertir los valores enteros de la información de los píxeles de una imagen de un canal en valores flotantes.

Una vez que se tienen las dos matrices de dos dimensiones de cada canal (ROJO y NIR), se realiza la ecuación (1.1) píxel a píxel con ayuda de la librería numpy.

$$ndvi = np.true\_divide(np.subtract(NIR, RED), np.add(NIR, RED))$$

De esta manera se almacenan todos los índices en una variable numpy array (matriz) con las mismas dimensiones que las matrices de las imágenes de un solo canal (720x1280 en el caso del prototipo multispectral).

Todo el cálculo del índice NDVI se ha sido asignado a un botón en la interfaz de usuario. Este botón está bloqueado hasta que se cargue una imagen, de lo contrario, permanecerá bloqueado para evitar un uso inadecuado del equipo por parte del usuario.

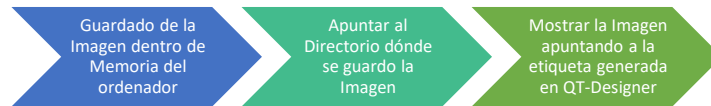


**Figura 2.20.** Botón para realizar el cálculo del índice NDVI

#### **2.6.4 Visualización de la Imagen NDVI dentro de la Interfaz gráfica.**

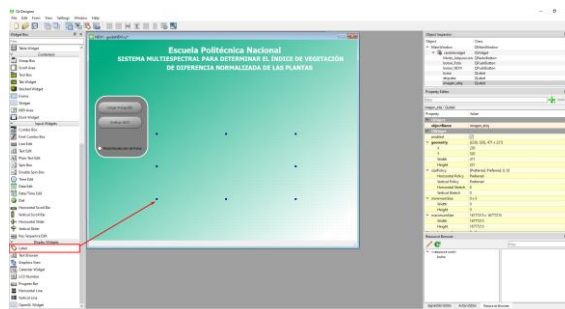
La visualización de la imagen NDVI se realizará utilizando dos librerías: OpenCV y Matplotlib. OpenCV se utilizará exclusivamente para visualizar la imagen, mientras que con Matplotlib permitirá mostrar el valor del índice NDVI en pantalla. Para visualizar la imagen, primero se debe generar un archivo .jpg en el computador a través de un algoritmo de programación que guarde la imagen en formato .jpg y apunte al directorio correspondiente. Es importante destacar que la librería de OpenCV trabaja con el orden BGR, mientras que

Matplotlib utiliza el orden RGB, por lo que se ha realizado transformaciones de las matrices para obtener las mismas imágenes en ambos casos.



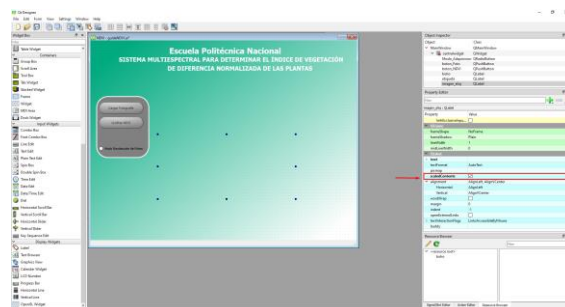
**Figura 2.21.** Pasos para visualizar la Imagen NDVI dentro de la Interfaz gráfica con OpenCV.

En QT-Designer, la herramienta “Label” se arrastra al área de trabajo y se redimensiona el objeto según las necesidades. Además, se asigna un nombre en el campo objectName que será usado para hacer referencia a la etiqueta en el código Python, como se muestra en la Figura 2.22.



**Figura 2.22.** Uso del objeto QLabel para la visualización de Imágenes .

Posteriormente, en la ventana de propiedades, se debe habilitar la opción “Scaled Content” para que la imagen se ajuste a las dimensiones del objeto “Label”, tal como se muestra en la Figura 2.23.



**Figura 2.23.** Configuración del objeto QLabel para reescalar el contenido de la imagen.

Una vez completados estos pasos, es posible hacer referencia al objeto creado en QT-Designer desde la programación en Python utilizando el mismo nombre especificado en el campo "Object Name". Para mostrar una imagen se debe utilizar la función `setPixmap` y proporcionar la ruta completa de la imagen a mostrar.

```
self.imagen_etiq.setPixmap(QtGui.QPixmap(ruta_img))
```

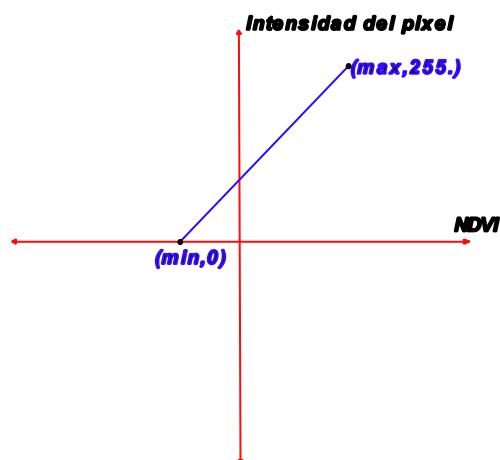
#### 2.6.4.1 Construcción de la imagen NDVI a partir de los índices

En ArcGIS, al realizar el cálculo del índice NDVI, se obtiene no solo los índices, sino también un ráster correspondiente a la imagen NDVI, la cual es una imagen de un solo canal. En el presente trabajo se ha construido la imagen NDVI a partir de los índices vegetativos que se encuentran almacenados en memoria. La matriz con los índices NDVI tiene las mismas dimensiones que una imagen de un solo canal, es decir, de 2 dimensiones. Por lo tanto, solo se requiere cambiar el contenido de esta variable y asignarle un nuevo nombre. Además, se utilizó un método de mejoramiento de imagen para obtener la imagen NDVI resultante. Se han utilizado métodos de la librería Numpy, (`np` en la programación) con el objetivo de determinar el valor máximo y el mínimo entre los 921600 datos que posee la variable `numpy.array`.

```
maximo_ndvi=np.max(ndvi_index)
```

```
minimo_ndvi=np.min(ndvi_index)
```

Una vez se han obtenido estos parámetros, se construye la imagen cambiando los 921600 datos de la matriz a valores enteros desde 0 y 255. Para posteriormente utilizar el método `cv2.imshow()`.



**Figura 2.24.** Gráfica de la recta para la reconstrucción de la imagen NDVI.

A partir de la Figura 2.24 se obtiene la ecuación de la recta, resultando en la ecuación 2.1 la cual es utilizada en la programación.

$$ndvi = (255/(maximo\_ndvi - minimo\_ndvi)) * (ndvi - minimo\_ndvi) \quad (2.1)$$

En la variable *ndvi* de la ecuación (2.1) se almacenan las intensidades de los píxeles. Estos valores se redondean con el método *round ()*, ya que es necesario que todos los valores de la matriz sean enteros para que se pueda proyectar una imagen con la librería OpenCV.

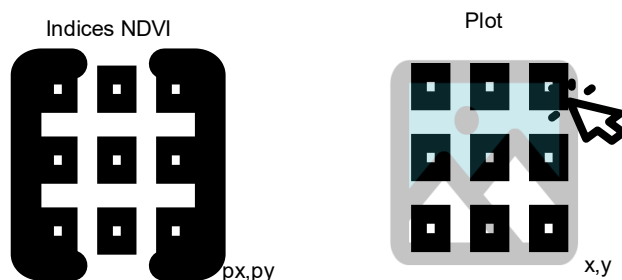
#### **2.6.4.2 Visualización con la librería matplotlib**

Para visualizar el índice vegetativo NDVI junto con la imagen NDVI, se utilizó la librería Matplotlib, que permite la creación de gráficos, gráficos de barras, rectas, o cargar imágenes provenientes de matrices almacenadas en memoria.

La imagen se carga al gráfico mediante la sintaxis investigada en este trabajo, y el usuario puede interactuar con ella[35]. Para mostrar el índice vegetativo NDVI en el punto exacto, se ejecuta una función cada vez que el mouse se mueve por la ventana creada. En esta función se conocen las coordenadas del ratón sobre la imagen NDVI. Como las dimensiones tanto de la imagen NDVI como de la matriz con los índices NDVI son iguales, es posible pasarle directamente la posición del mouse a variables auxiliares (*col* y *row*) que permiten apuntar a la matriz con los índices NDVI [36]. Al mover el mouse por el área de trabajo, los valores que cambiarán dentro de las variables internas del computador son “*x*” y “*y*”, y se hace uso de esta variación para apuntar a la matriz dónde se encuentran los índices vegetativos. Este procedimiento es posible gracias a que ambas matrices tienen las mismas dimensiones. En caso de que se tenga una imagen reescalada con dimensiones diferentes, será necesario hacer una regla de 3 entre las dimensiones de la imagen cargada en matplotlib y la matriz que almacena los índices NDVI. Este fenómeno puede ocurrir cuando se usa solo OpenCV sin Pyqt5. Esto se debe a que al querer mostrar una imagen con OpenCV, se visualizarán imágenes demasiado grandes que se salgan de pantalla, lo que requiere reescalar la imagen. Este proceso implica la pérdida de información de los píxeles, por lo que es importante conocer siempre las dimensiones de las matrices con las que trabaja. Se ha utilizado el comando `print(variable.shape)` para poder visualizar las dimensiones de la matriz utilizada.

```
def format_coord(x, y):
    #La posición tanto de el mouse dentro del área de trabajo, como de la matriz ndvi son las mismas.
    col = x
    row = y
    #Si me encuentro dentro del área de trabajo
    if col>=0 and col<width and row>=0 and row<height:
        z = ndvi_index[int(y),int(x)]
        return 'x=%1.4f, y=%1.4f, NDVI=%1.4f'%(x, y, z)
    else:
        return 'x=%1.4f, y=%1.4f'%(x, y)
```

**Figura 2.25.** Función para determinar el índice NDVI según la posición del ratón [Fuente Propia].



**Figura 2.26.** Figura con la imagen y los índices vegetativos. [Fuente Propia].

En la Figura 2.26, se presentan tanto la matriz con los índices vegetativos NDVI como el plot en el que se carga la imagen NDVI. Ambas variables tienen las mismas dimensiones y para presente trabajo estas son de dimensiones de (720,1280,1) debido a la resolución de la cámara web utilizada.

### 2.6.5 Creación de la Interfaz Gráfica.

La creación de la interfaz gráfica en Python para trabajar con PyQt se puede realizar mediante dos métodos. El primero consiste en crear toda la interfaz gráfica solamente con código, especificando la ubicación exacta de todos los objetos en el área de trabajo. El segundo método, más sencillo, es crear la interfaz gráfica mediante el arrastre de objetos hacia el área de trabajo y la asignación de una etiqueta a estos objetos en el programa QT-Designer, para luego apuntar a estas etiquetas y modificar los atributos del objeto ya creado.

En el programa QT-Designer, la opción “changeStylesheet” permite modificar el aspecto de un objeto al hacer clic derecho en él. Esta herramienta se utilizó para cambiar colores, formas e inclusive animaciones de los objetos en la interfaz gráfica. Por ejemplo, se cambió la animación del botón agregando un color verde al momento de presionarlo.



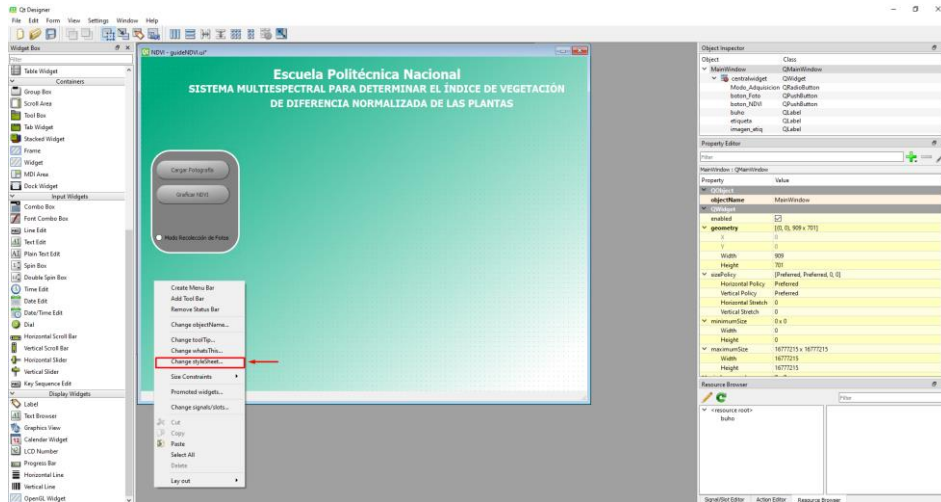


Figura 2.27. Herramienta ChangeStyleSheet [Fuente Propia].

### 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En este capítulo, se presentan los índices vegetativos obtenidos por el sistema multiespectral desarrollado y se comparan con los índices proporcionados por el software ArcGIS para validar los índices NDVI obtenidos en el cultivo. Las pruebas se llevaron a cabo mediante la toma de fotografías en el mismo prototipo y el análisis posterior de los índices vegetativos de la zona enferma de la planta.

#### 3.1 PRUEBAS Y RESULTADOS.

Se han realizado capturas de muestras del prototipo implementado en Quito-Pichincha y en un invernadero de Mira-Carchi. Se puede visualizar como las fotografías obtenidas por el prototipo pueden ser procesadas tanto con ArcGIS como con el prototipo multiespectral. Además, se llevaron a cabo pruebas de autonomía del prototipo, superando la hora prevista.

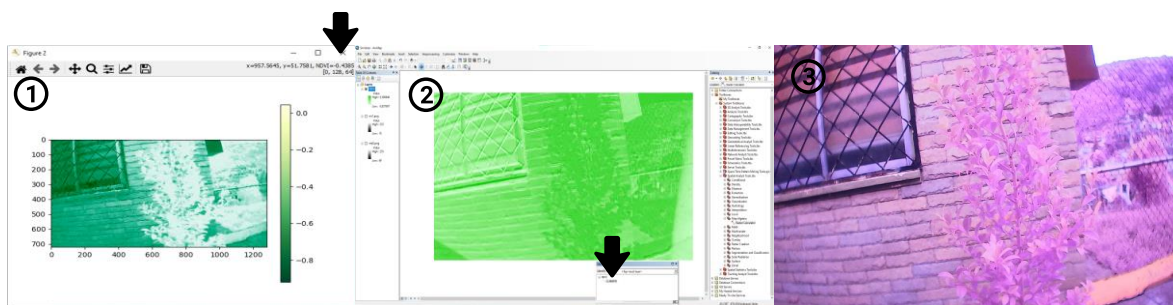


Figura 3.1. Fotografías tomadas por el sector la Vicentina en Quito. [Fuente Propia].

**Tabla 3.1** Descripción de las fotografías tomadas en Quito

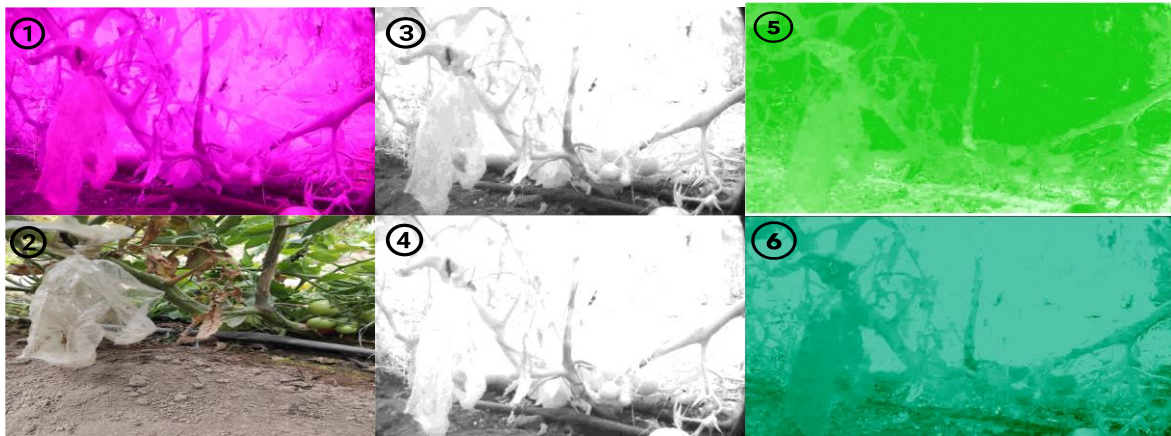
Número	Fotografía	Descripción
1	NDVI procesada por el sistema multiespectral	Esta imagen, de un solo canal, es la representación del Índice Vegetativo de Diferencia Normalizada. Se puede observar cómo el sistema implementado permite diferenciar efectivamente una planta de los objetos inanimados
2	NDVI procesada por ArcGIS	Esta imagen representa la imagen NDVI, de un solo canal, Esta fotografía es procesada por el software ArcGIS.
3	Banda Visible	Esta imagen es una representación de los tres canales de la imagen obtenida por la cámara. No se ha aplicado ningún tipo de filtro en la imagen.

### 3.1.1 Validación de la Imagen NDVI

Se llevó a cabo un viaje a una plantación en Mira, Ecuador para validar el Índice de Vegetación de Diferencia Normalizada (NDVI). La plantación cuenta con invernaderos para cultivar exclusivamente tomates de árbol y otros productos, y está experimentando una infección por Botritis.



**Figura 3.2.** Sistema Multiespectral en Invernadero Mira, Ecuador.



**Figura 3.3.** Planta enferma de Boytritis [Fuente Propia].

Cómo se puede apreciar en la Figura 3.3 las imágenes 3 y 4 representan imágenes de un solo canal. El color negro se logra cuando el valor del píxel se acerca a 0, mientras que el blanco se representa cuando el valor del píxel se acerca a 255.

**Tabla 3.2** Descripción de las fotografías tomadas en Mira

Número	Fotografía	Descripción
1	Adquirida por el sistema multiespectral	Esta imagen posee los 3 canales de color: rojo, verde y azul (RGB) y fue adquirida por el sistema multiespectral implementado.
2	Banda Visible	Esta imagen posee los 3 canales de color: rojo, verde y azul (RGB) y fue adquirida por una cámara que no ha sido modificada.
3	Banda NIR digital	Esta imagen posee un canal, el canal corresponde al canal azul y gracias al sistema óptico implementado representa la imagen NIR.
4	Banda Red digital	Esta fotografía posee un canal, el canal corresponde al canal rojo y representa la imagen RED.
5	NDVI ArcGIS	Esta fotografía de un solo canal corresponde a la imagen NDVI, la cual es generada por el programa ArcGIS.
6	NDVI por el Sistema multiespectral	Esta fotografía de un solo canal corresponde a la imagen NDVI, la cual es generada por el software ArcGIS.

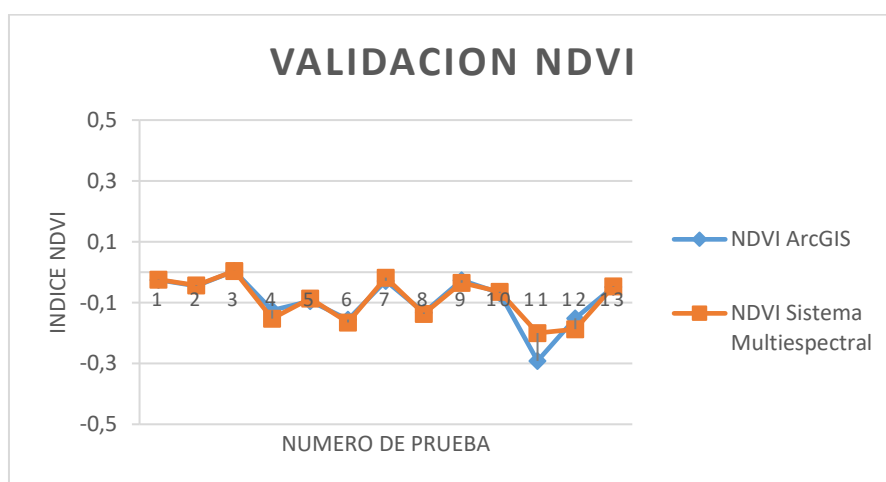
La fotografía 5 y 6 corresponden, respectivamente, a la imagen NDVI producida por ArcGIS, y a la imagen NDVI producida por el sistema multiespectral. A simple vista, ambas imágenes presentan similitudes evidentes. A continuación, se comparan los índices NDVI obtenidos.

### 3.1.2 Validación del índice NDVI obtenido.

Para esta prueba se ha realizado el computo que realiza ArcGIS con el computo que realiza el sistema multiespectral implementado para una misma muestra, y se han tomado diferentes índices vegetativos de una misma fotografía.

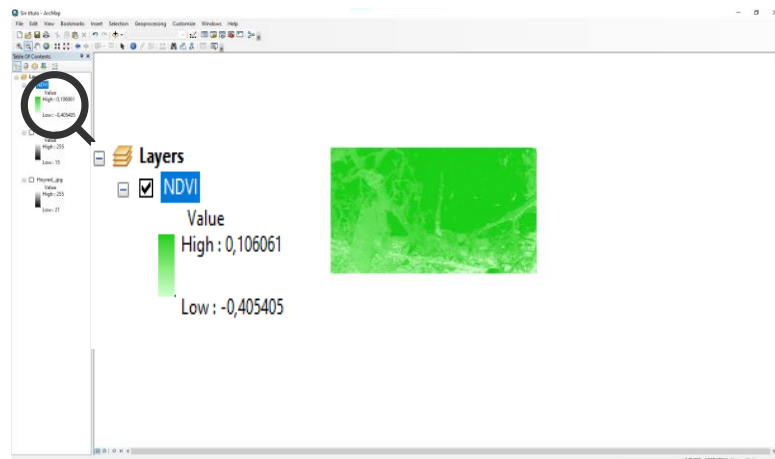
**Tabla 3.3** Comparación de los índices vegetativos NDVI con ArcGIS y el sistema multiespectral

Número	NDVI ArcGIS	NDVI Sistema Multiespectral	Desviación estándar ArcGIS	Desviación estándar del sistema multiespectral
1	-0.025751	-0.0244	0.00432176	0.0041543
2	-0.045	-0.0429	0.00216142	0.00211176
3	0.004016	0.0039	0.0091216	0.008603
4	-0.125	-0.1525	0.00112285	0.00405
5	-0.095023	-0.086	0.00001247	0.000008144
6	-0.15555	-0.1648	0.004103	0.0057678
7	-0.028226	-0.018	0.004	0.00502
8	-0.132203	-0.137	0.001657	0.0023
9	-0.026596	-0.0352	0.00421	0.00287
10	-0.068884	-0.0647	0.000511	0.00058341
11	-0.291667	-0.2	0.0400	0.01235347
12	-0.1515	-0.1875	0.0036	0.00973
13	-0.048	-0.046	0.00189147	0.001836



**Figura 3.4.** Comparación de los índices NDVI Obtenidos con el Sistema Multiespectral y los índices NDVI de ArcGIS [Fuente Propia].

En la Figura 3.5 se ilustra cómo ArcGIS presenta el índice máximo y mínimo entre todos los índices NDVI calculados debajo del ráster correspondiente a la imagen NDVI. Esta imagen NDVI posee solo 1 canal y aparece en el área de trabajo dónde fueron cargadas las imágenes de un solo canal que se utilizaron para realizar el cálculo del índice NDVI con el software ArcGIS.



**Figura 3.5.** Máximos y mínimos valores del índice NDVI que procesa ArcGIS [Fuente Propia].

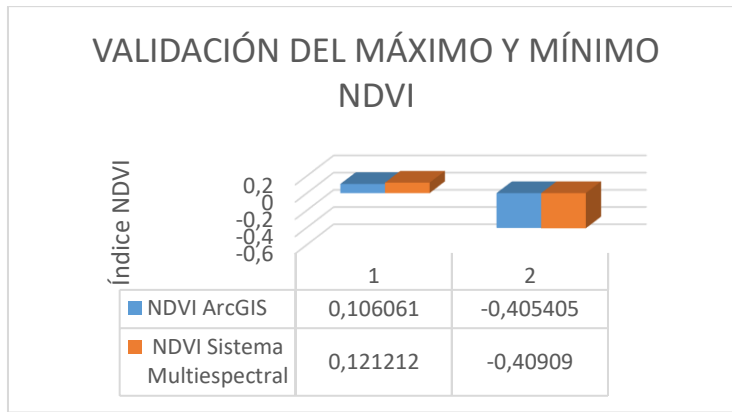
La determinación de los valores máximos y mínimos en la matriz que contiene los 921600 índices vegetativos es posible a través de programación. Se utilizan las funciones “max” y “min” propias del objeto “numpy”, donde se almacenan los índices vegetativos. La implementación se muestra en la Figura 3.6

```
maximo_ndvi=np.max(ndvi_index)
minimo_ndvi=np.min(ndvi_index)
```

**Figura 3.6.** Código para extraer el máximo y el mínimo valor de los índices vegetativos.

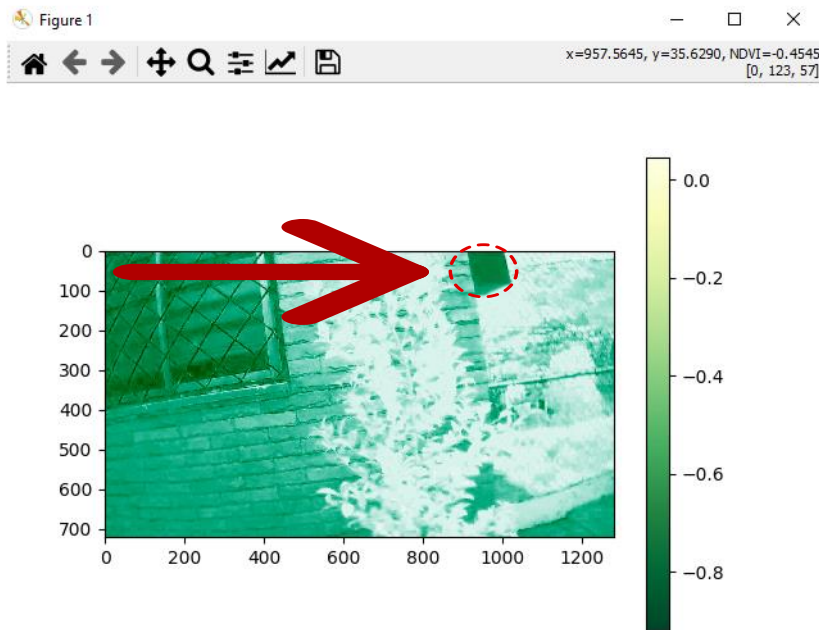
**Tabla 3.4** Validación de los máximos y mínimos índices NDVI entre ArcGIS y el sistema multispectral implementado

NDVI	NDVI ArcGIS	NDVI Sistema Multiespectral	Error Relativo [%]
<b>Máximo</b>	-0.106061	0.121212	14.28517551
<b>Mínimo</b>	-0.405405	-0.40909	0.908967



**Figura 3.7.** Comparación de los índices NDVI máximos y mínimos con el programa ArcGIS con el sistema multiespectral desarrollado.

En la Figura 3.7 se observa que tanto el máximo como el mínimo índice NDVI proporcionados por ArcGIS se aproximan considerablemente a los índices NDVI calculados por el sistema multiespectral.

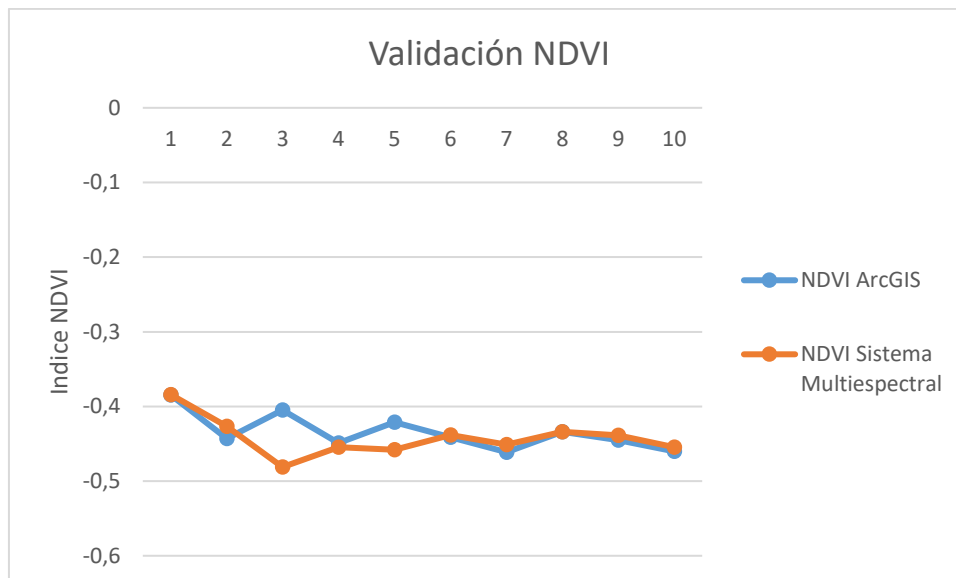


**Figura 3.8.** Imagen NDVI de una planta adquirida por el sistema Multiespectral.

En la Figura 3.8 se muestra la imagen NDVI de una planta, donde el área dentro del círculo discontinuo presenta índices NDVI cercanos a -0.4. Tanto en ArcGIS como el prototipo implementado presentan valores similares, ya que ambos se acercan a -0.4.

**Tabla 3.5** Validación de índices NDVI entre ArcGIS y el sistema multiespectral implementado de la Figura 8.

Número	NDVI ArcGIS	NDVI Sistema Multiespectral	Desviación estándar ArcGIS	Desviación estándar del sistema multiespectral
1	-0.384615	-0.3841	0.0024	0.000006173
2	-0.443182	-0.4268	0.00007605	0.00000005784
3	-0.404959	-0.4812	0.00087037	0.00000075755
4	-0.449	-0.4545	0.00021138	0.0000000446813
5	-0.4212	-0.458	0.00017586	0.000000030925
6	-0.4413	-0.4384	0.00004677	0.000000002187
7	-0.461255	-0.4511	0.00071791	0.000000515399
8	-0.4338	-0.4338	0.00000043705	0.00000000000019101
9	-0.4453	-0.4385	0.00011748	0.0000000138
10	-0.46	-0.4545	0.00065224	0.000000425411



**Figura 3.9.** Comparación de los índices NDVI obtenidos con el sistema multiespectral y los índices NDVI de ArcGIS en el área del círculo discontinuo en la Figura 3.8.

### 3.2 CONCLUSIONES

- Con base en las pruebas realizadas y los datos obtenidos, se puede concluir que el sistema multiespectral es capaz de extraer, almacenar y procesar imágenes multiespectrales con el fin de calcular el índice NDVI de manera portátil, cumpliendo con los objetivos del trabajo de integración curricular.
- Se concluye que el procesamiento de la imagen es efectivo, ya que los errores relativos al comparar los índices de ArcGIS con el prototipo desarrollado son

inferiores al 15% lo que garantiza un cálculo preciso del índice de Vegetación de Diferencia Normalizada (NDVI).

- Se construyó un sistema óptico mediante la eliminación del filtro IR de una cámara convencional y colocación de un filtro seleccionado de acuerdo con su curva en un espectrómetro, lo que permitió desarrollar un sistema óptico capaz de obtener información en el infrarrojo cercano y determinar cuantitativamente el estrés de una planta.
- Se construyó una estructura eficiente para soportar los componentes del sistema multispectral utilizando software especializado, garantizando una operación cómoda y precisa gracias a la entrada digital de la Raspberry Pi 3 B+.
- Se investigó el lenguaje programación Python con el fin de crear un sistema portátil que simplifique la extracción y procesamiento de información de la cámara utilizando técnicas de visión por computadora.
- En este trabajo se combinaron todas las funciones, incluyendo la adquisición de imágenes, carga de fotografías y procesamiento del índice NDVI, en una interfaz de fácil uso y con medidas de seguridad para prevenir su uso inadecuado, como el bloqueo del botón de cálculo del índice NDVI hasta que se haya cargado una imagen.
- Python es un lenguaje de programación que busca simplificar el trabajo para los desarrolladores, por lo que cuenta con una amplia variedad de librerías preexistentes que permiten agilizar el proceso de creación de prototipos, incluyendo aplicaciones para el sector agrícola.
- Se demostró como el cálculo preciso del índice NDVI permite generar imágenes que permiten distinguir la vegetación de objetos inanimados, y también como se puede resaltar las enfermedades de las plantas, lo que será útil en futuros trabajos de detección.
- Se integraron dos sistemas en uno, el primero para la captura de imágenes y el segundo para el cálculo y procesamiento de imágenes multispectrales. Esto reduce el tiempo de toma de muestras para los agricultores, lo que hace que el prototipo sea más atractivo para ellos y les permita aumentar la producción de sus cultivos.



- En este trabajo se desarrolló un prototipo que realiza el cálculo del índice NDVI de manera similar a ArcGIS, e incluso con capacidades adicionales. Se utilizó una cámara comercial de bajo costo y programación de código abierto, lo que permite una fácil modificación en el futuro. Además, el prototipo permite el procesamiento de imágenes multiespectrales con cualquier tipo de cámara web, sin necesidad de que sea especializada para la determinación del NDVI.
- En el presente trabajo se utilizó del software QT-Designer para desarrollar la interfaz hombre-máquina, que brinda la posibilidad de utilizar diferentes herramientas en función de las necesidades del usuario. Esto facilitó la creación de una interfaz bien diseñada con una ubicación adecuada de los componentes.

### **3.3 RECOMENDACIONES**

- Se recomienda prestar atención a las dimensiones de la matriz que representa una imagen al momento de realizar operaciones entre matrices, para garantizar un correcto cumplimiento de las leyes de álgebra lineal.
- Se recomienda utilizar el prototipo en las horas de mayor luminosidad para asegurar una captura de imagen clara y precisa, lo cual puede ser en la mañana o en la tarde, dependiendo de las condiciones ambientales.
- Se recomienda considerar la opción de una Raspberry de mayor costo y prestaciones para aumentar la velocidad de procesamiento y utilizar una cámara web de mayor resolución para obtener resultados más precisos en la determinación del índice NDVI y el estrés vegetativo de la planta.
- Se recomienda revisar la sintaxis utilizada para la declaración de variables globales y locales dentro de la programación en lenguaje Python, a fin de asegurarse de que el código sea claro, fácil de entender y pueda ser reutilizado para otros proyectos.

### **3.4 Trabajos Futuros**

- En este trabajo, se podría hacer una modificación para que la imagen a procesar ya no sea una fotografía, sino un fotograma de un video. De esta manera, se podría procesar el video NDVI, lo cual reduciría aún más los tiempos en la toma de decisiones.

- Mediante la separación de los canales de una imagen, se puede reducir su tamaño en memoria, lo que permitiría implementar una transmisión de video, enviando primero las imágenes de un solo canal para luego reconstruir la imagen completa a partir de las tres imágenes de canal individual en una tarjeta receptora compuesta por un ordenador personal.
- Con el presente trabajo se puede ampliar los periodos de prueba con el objetivo de detectar plagas, sequías, entre otros factores relevantes para el cultivo.
- Para determinar la proliferación de hongos en una plantación es necesario evaluar tanto el índice NDVI como la humedad de la misma. Por lo tanto, en trabajos futuros, podría ser considerado el desarrollo de un módulo que permita medir esta variable para una evaluación más completa.

#### 4 REFERENCIAS BIBLIOGRÁFICAS

- [1] Mextic, «¿Qué es la Agricultura Inteligente?» [Online] Disponible: <https://www.agriculturainteligente.com/novedades/index.php/59-que-es-la-agricultura-inteligente> (accedido 22 de noviembre de 2022).
- [2] «Smart Agro, nuevas tecnologías para la Agricultura», Tractores y Máquinas, 26 de noviembre de 2017. [Online] Disponible: <https://www.tractoresymaquinas.com/smart-agro-nuevas-tecnologias-agrarias/> (accedido 22 de noviembre de 2022).
- [3] H. instruments, «Espectro electromagnético» HANNA® instruments Ecuador, HANNA® instruments Ecuador, 22 de junio de 2019. [Online] Disponible: <https://hannainst.ec/blog/agricultura-e-hidroponiablog/industria-alimenticiablog/laboratorio/espectro-electromagnetico/> (accedido 22 de noviembre de 2022).
- [4] «RADIACIÓN SOLAR - IDEAM». [Online] Disponible: <http://www.ideam.gov.co/web/tiempo-y-clima/radiacion-solar-ultravioleta> (accedido 1 de diciembre de 2022).
- [5] S. User, «Estructura de la hoja», droncontrol. [Online] Disponible: <http://www.droncontrol.com/index.php/es/servicios/aplicaciones-agricolas/51-agricola/165-estructura-de-la-hoja> (accedido 22 de noviembre de 2022).

- [6] «NDVI: Fórmula Y Uso Del Índice De Vegetacion Na Agricultura», 27 de septiembre de 2021. [Online] Disponible: <https://eos.com/es/make-an-analysis/ndvi/> (accedido 22 de noviembre de 2022).
- [7] «ArcGIS Desktop | Desktop GIS Software Suite». [Online] Disponible: <https://www.esri.com/en-us/arcgis/products/arcgis-desktop/overview> (accedido 23 de noviembre de 2022).
- [8] gisadminbeers, «Todo lo que deberías saber sobre imágenes Landsat», Gis&Beers, 22 de febrero de 2020. [Online] Disponible: <http://www.gisandbeers.com/lo-deberias-saber-imagenes-landsat/> (accedido 26 de septiembre de 2022).
- [9] «Teledetección: Tipos Y Aplicaciones De Los Sensores Remotos», 9 de noviembre de 2021. [Online] Disponible: <https://eos.com/es/blog/teledeteccion/> (accedido 23 de noviembre de 2022).
- [10] «Imágenes De Satélite Gratis: Fuentes Y Sus Características», 2 de noviembre de 2020. [Online] Disponible: <https://eos.com/es/blog/imagenes-de-satelite-gratis/> (accedido 23 de noviembre de 2022).
- [11] «Python: qué es, para qué sirve y cómo se programa | Informática Industrial», aula21 | Formación para la Industria, 8 de octubre de 2020. [Online] Disponible: <https://www.cursosaula21.com/que-es-python/> (accedido 23 de noviembre de 2022).
- [12] «OpenCV - Overview», GeeksforGeeks, 23 de septiembre de 2019. [Online] Disponible: <https://www.geeksforgeeks.org/opencv-overview/> (accedido 23 de noviembre de 2022).
- [13] «La guía definitiva del paquete NumPy para computación científica en Python», freeCodeCamp.org, 20 de marzo de 2021. [Online] Disponible: <https://www.freecodecamp.org/espanol/news/la-guia-definitiva-del-paquete-numpy-para-computacion-cientifica-en-python/> (accedido 23 de noviembre de 2022).
- [14] Hebert, «PyQT5 interfáces gráficas con Python», ▷ Cursos de Programación de 0 a Experto © Garantizados, 8 de mayo de 2020. [Online] Disponible:

- <https://unipython.com/pyqt5-interfaces-graficas-con-python/> (accedido 23 de noviembre de 2022).
- [15] «Matplotlib — Visualización con Python».[Online] Disponible: <https://matplotlib.org/> (accedido 23 de noviembre de 2022).
- [16] R. de la Vega, «▷ Introducción al procesamiento de imágenes en Python con OpenCV | Pharos», 11 de noviembre de 2022. [Online] Disponible: <https://pharos.sh/introduccion-al-procesamiento-de-imagenes-en-python-con-opencv/> (accedido 24 de noviembre de 2022).
- [17] «BGR order in OpenCV | Mastering OpenCV 4 with Python», Packt. [Online] Disponible: <https://subscription.packtpub.com/book/data/9781789344912/3/ch03lvl1sec23/bgr-order-in-opencv> (accedido 24 de noviembre de 2022).
- [18] «Que es el filtro mecánico de infrarrojos?», Securame, 13 de junio de 2012. [Online] Disponible: <https://www.securame.com/blog/que-es-el-filtro-mecanico-de-infrarrojos/> (accedido 24 de noviembre de 2022).
- [19] «Filtro óptico de luz visible by CreatorOptics | DirectIndustry». [Online] Disponible: <https://www.directindustry.es/prod/creatoroptics/product-204781-2124815.html> (accedido 24 de noviembre de 2022).
- [20] «¿Qué es SolidWorks? - ADR Formación». [Online] Disponible: [https://www.adrformacion.com/knowledge/ingenieria-y-proyectos/\\_que\\_es\\_solidworks\\_.html](https://www.adrformacion.com/knowledge/ingenieria-y-proyectos/_que_es_solidworks_.html) (accedido 24 de noviembre de 2022).
- [21] «Get Started with Visual Studio Code».[Online] Disponible: <https://code.visualstudio.com/learn/overview> (accedido 24 de noviembre de 2022).
- [22] R. Python, «Qt Designer and Python: Build Your GUI Applications Faster – Real Python». [Online] Disponible: <https://realpython.com/qt-designer-python/> (accedido 24 de noviembre de 2022).
- [23] ed, «Get Started with Thonny IDE on Raspberry Pi OS», The Robotics Back-End, 14 de diciembre de 2020. [Online] Disponible: <https://roboticsbackend.com/thonny-ide-raspberry-pi-os/> (accedido 24 de noviembre de 2022).

- [24] E. R. de Luis, «De cero a maker: todo lo necesario para empezar con Raspberry Pi», Xataka, 15 de julio de 2018. [Online] Disponible: <https://www.xataka.com/makers/cero-maker-todo-necesario-para-empezar-raspberry-pi> (accedido 28 de noviembre de 2022).
- [25] «MidOpt DB660/850 | Dual Bandpass Red & 850nm NIR», MidOpt, 18 de junio de 2015. [Online] Disponible: <https://midopt.com/filters/db660850/> (accedido 28 de noviembre de 2022).
- [26] «Carga y descarga baterías de Li-Ion - Blog recambios portátiles». [Online] Disponible: [https://www.portatilmovil.com/blog/54\\_carga-de-baterias-li-ion-los-mejores-metodos-de-carga.html](https://www.portatilmovil.com/blog/54_carga-de-baterias-li-ion-los-mejores-metodos-de-carga.html) (accedido 30 de noviembre de 2022).
- [27] «Raspberry Pi Lithium Battery Power Pack - Wiki». [Online] Disponible: [http://wiki.sunfounder.cc/index.php?title=Raspberry\\_Pi\\_Lithium\\_Battery\\_Power\\_Pack](http://wiki.sunfounder.cc/index.php?title=Raspberry_Pi_Lithium_Battery_Power_Pack) (accedido 30 de noviembre de 2022).
- [28] «ACTION CAM 4K ULTRA HD ANDOWL AN-QY-08K», Tapandaola.gr, 22 de octubre de 2021. [Online] Disponible: <https://tapandaola.gr/product/action-cam-4k-ultra-hd-andowl-an-qy-08k/> (accedido 7 de diciembre de 2022).
- [29] «Display LCD 3.5" táctil HAT para Raspberry Pi», Naylamp Mechatronics - Perú. [Online] Disponible: <https://naylampmechatronics.com/raspberry-pi/547-display-lcd-3p5-tactil-hat-para-raspberry-pi.html> (accedido 7 de diciembre de 2022).
- [30] «Amazon.com: Filtro rojo de 2.283 in, filtro de color de 2.283 in: Electrónica». [Online] Disponible: [https://www.amazon.com/dp/B09KY92H2D/ref=cm\\_sw\\_r\\_awdo\\_navT\\_g\\_ER0C7F8QGFCBWCQGR3TH?th=1](https://www.amazon.com/dp/B09KY92H2D/ref=cm_sw_r_awdo_navT_g_ER0C7F8QGFCBWCQGR3TH?th=1) (accedido 6 de diciembre de 2022).
- [31] «Buy GREEN.L 67mm Infrared Filter, Multi-Resistant Nano Coating HD 67mm X-Ray IR 850nm Filter for Camera Lens Online at desertcart Botswana». [Online] Disponible: <https://botswana.desertcart.com/products/162733395-green-l-infrared-filter-67-mm-ir-850-nm-x-ray-filter-optical-glass-filter> (accedido 6 de diciembre de 2022).
- [32] «Mastering OpenCV 4 with Python». Packt, 10 de enero de 2023. [Online] Disponible: [https://github.com/PacktPublishing/Mastering-OpenCV-4-with-Python/blob/0fd966a58e6b48b46d39a13f92835661031fbb23/Chapter02/01-chapter-content/03\\_displaying\\_images.py](https://github.com/PacktPublishing/Mastering-OpenCV-4-with-Python/blob/0fd966a58e6b48b46d39a13f92835661031fbb23/Chapter02/01-chapter-content/03_displaying_images.py) (accedido 11 de enero de 2023).

- [33] «Cómo conectar una Cámara USB a una Raspberry Pi», 26 de septiembre de 2019. [Online] Disponible: <https://descubrearduino.com/conectar-camara-raspberry-pi/> (accedido 11 de enero de 2023).
- [34] «EarthExplorer». [Online] Disponible: <https://earthexplorer.usgs.gov/> (accedido 8 de diciembre de 2022).
- [35] «matplotlib.axes.Axes.imshow — Matplotlib 3.6.3 documentation». [Online] Disponible:  
[https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.axes.Axes.imshow.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.axes.Axes.imshow.html)  
(accedido 12 de enero de 2023).
- [36] «Modifying the coordinate formatter — Matplotlib 3.1.2 documentation». [Online] Disponible:  
[https://matplotlib.org/3.1.1/gallery/images\\_contours\\_and\\_fields/image\\_zcoord.html#sphx-glr-gallery-images-contours-and-fields-image-zcoord-py](https://matplotlib.org/3.1.1/gallery/images_contours_and_fields/image_zcoord.html#sphx-glr-gallery-images-contours-and-fields-image-zcoord-py) (accedido 9 de diciembre de 2022).
- [37] Downey, A., Elkner, J., Meyers, C., Vilella, M. A., Arnal, A., Juanes, I., Amurrio, L., Andia, E., Ballardini, C. (2002). "Aprenda a Pensar Como un Programador con Python." Green Tea Press, Wellesley, Massachusetts.

5 ANEXOS

ANEXO A (Planos del sistema multiespectral).

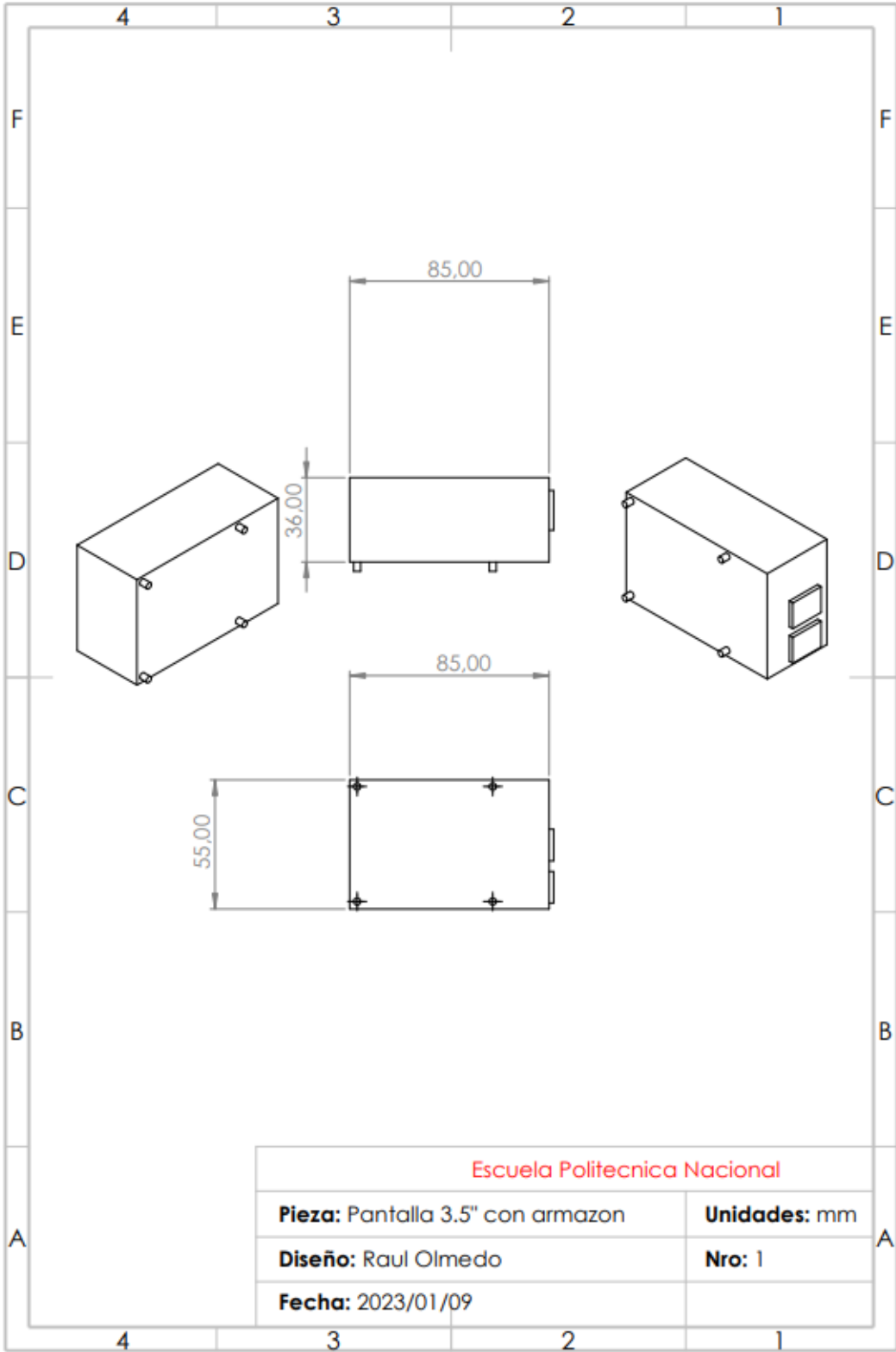
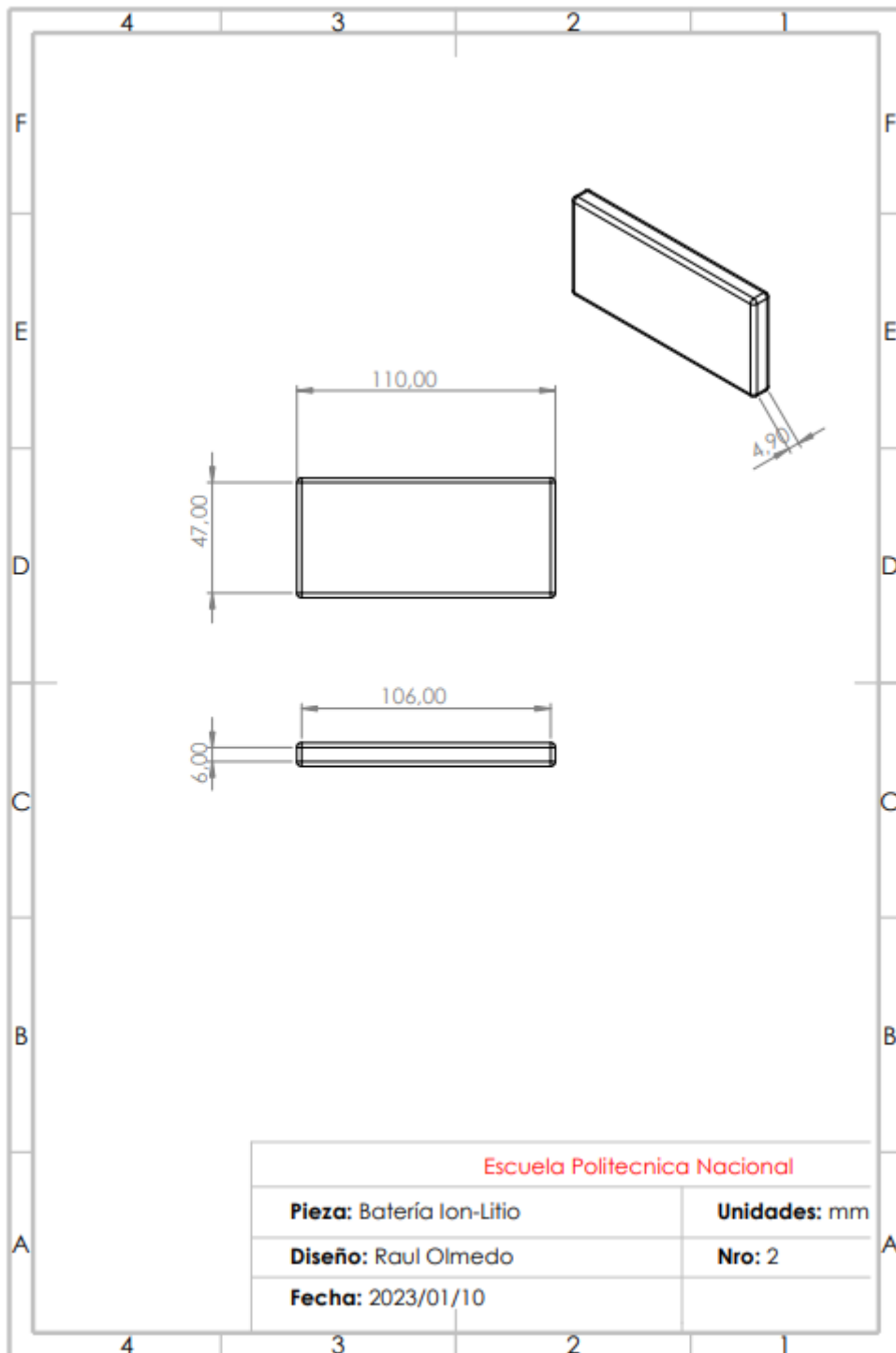


Figura A.1 Planos de la Figura de la pantalla 3.5" con su armazón.



**Figura A.2** Batería de ion Litio.



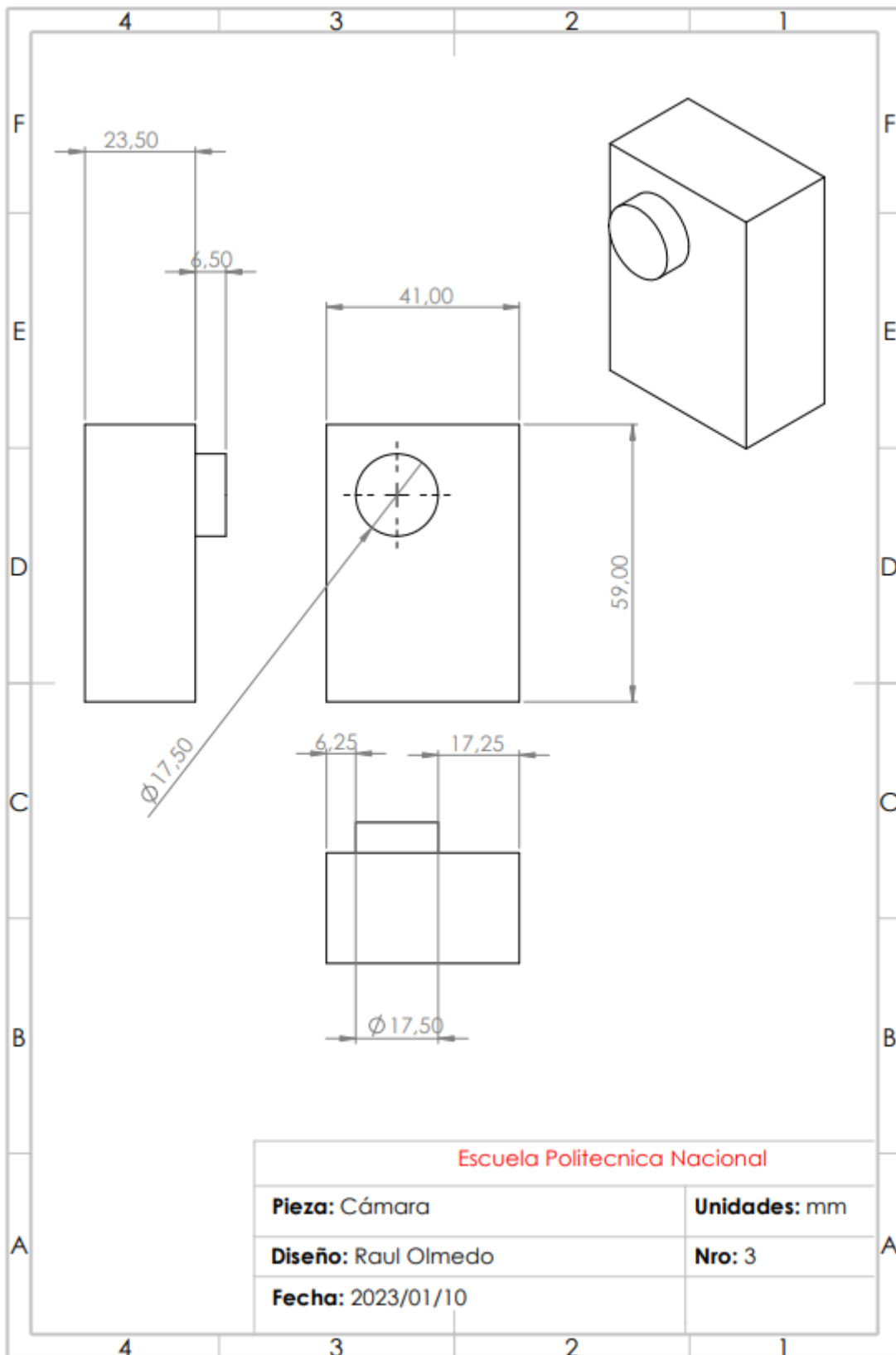
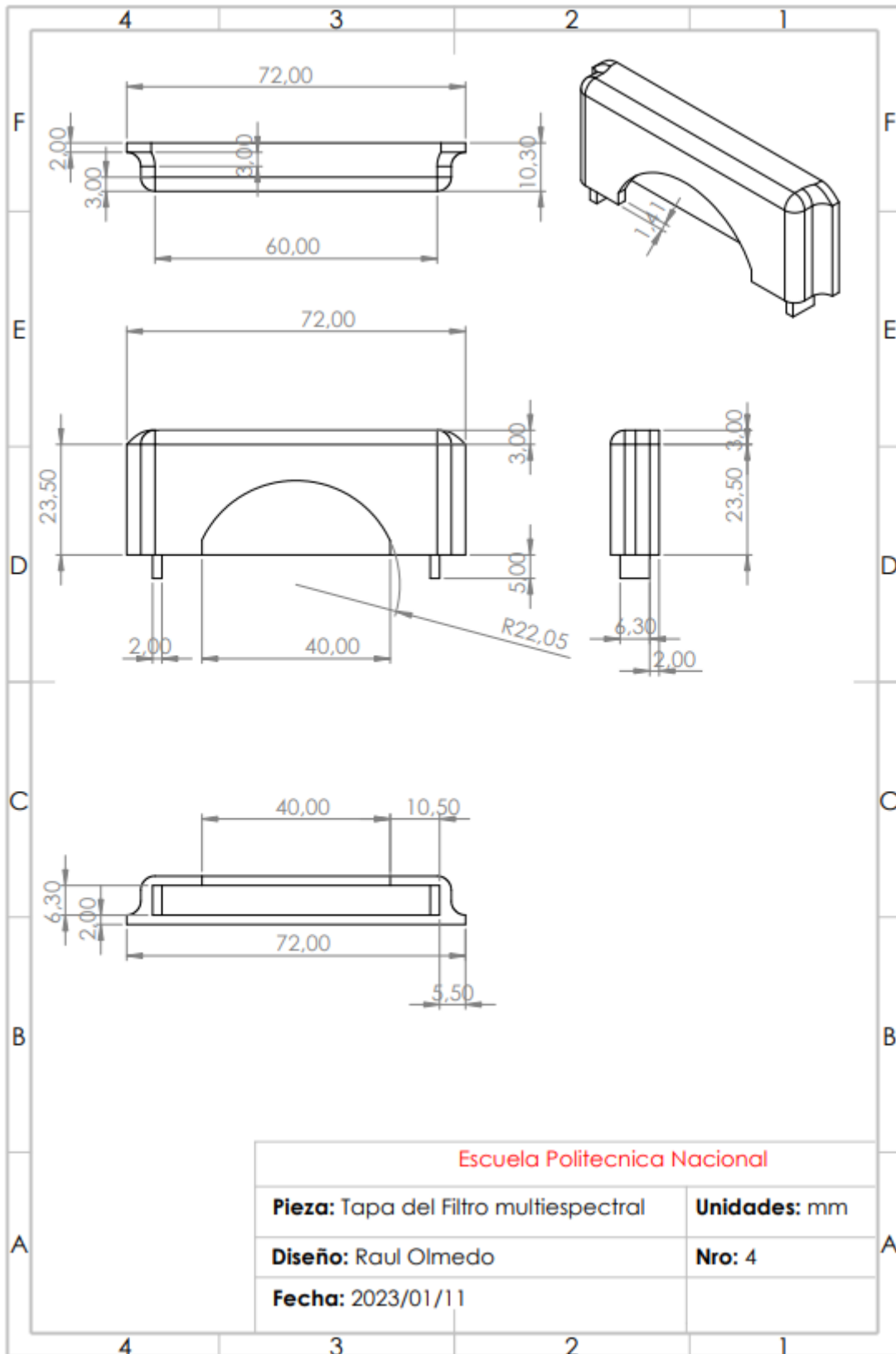


Figura A.3 Cámara Digital.



**Figura A.4** Tapa del Filtro Multiespectral.

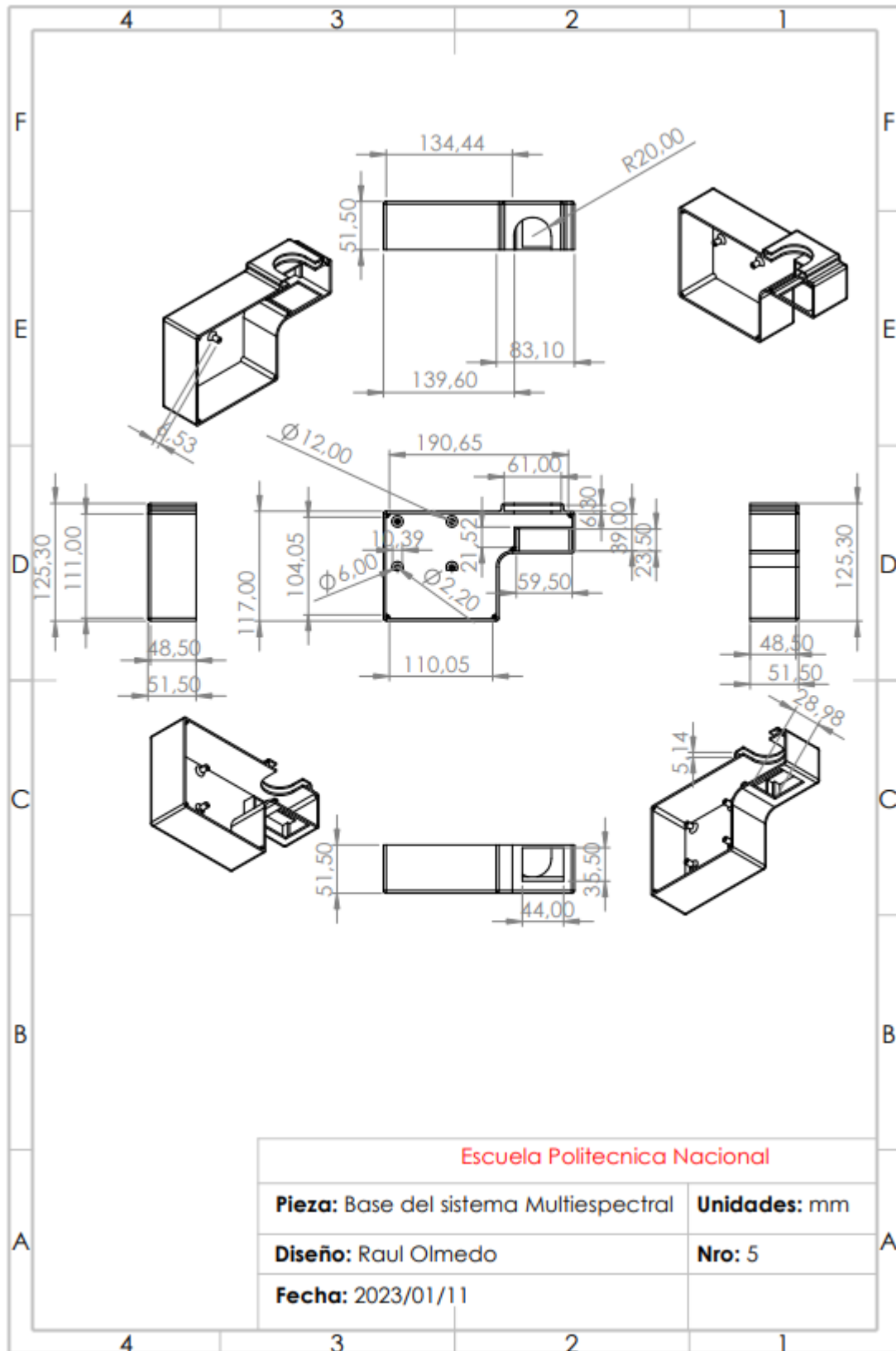


Figura A.5 Base del sistema Multiespectral.

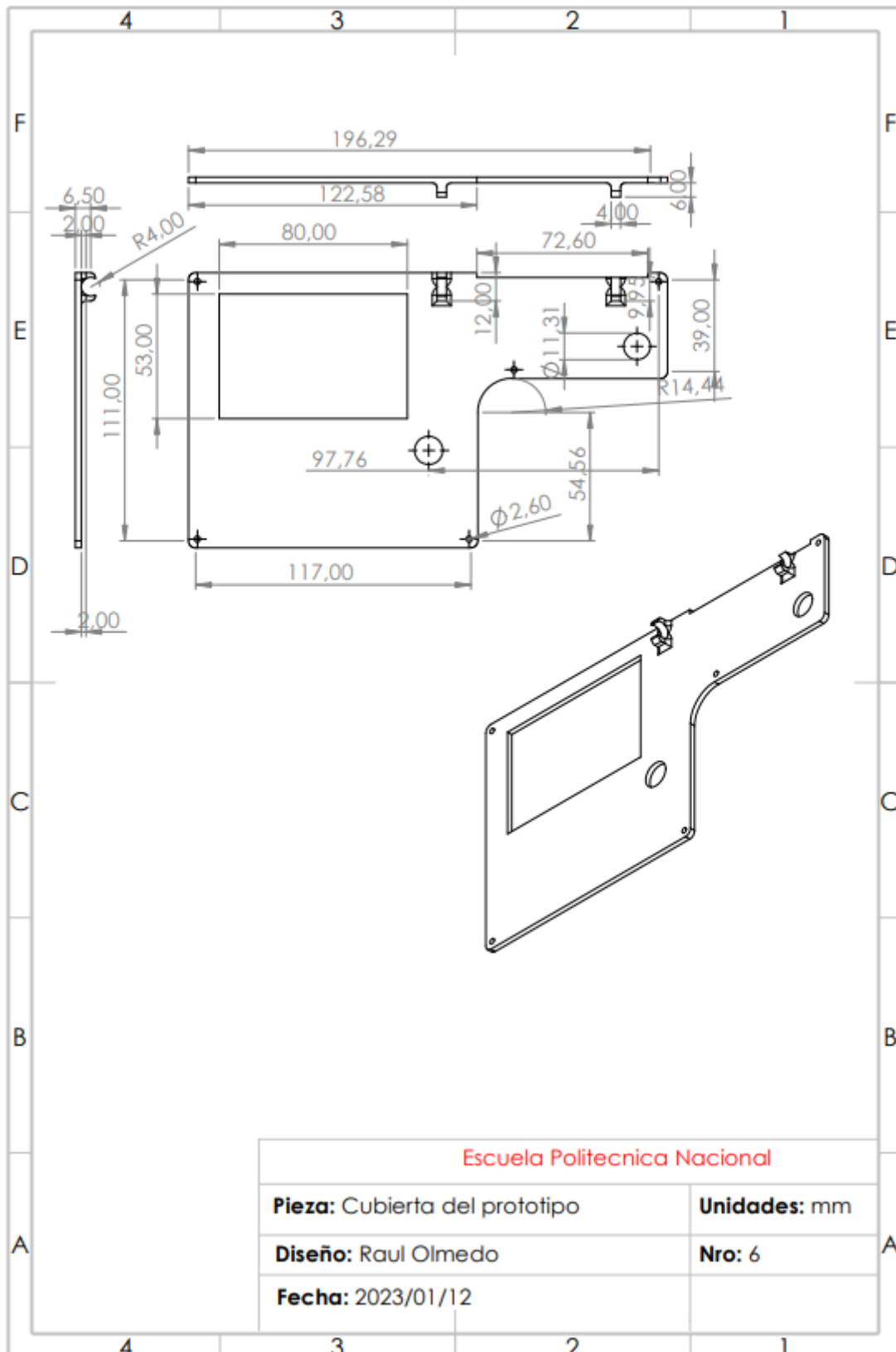


Figura A.6 Cubierta del prototipo implementado.

## Anexo B Programación

2/2/23, 11:09

c:\...\Python\Nueva\_Tesis\_V1.py

```
import cv2
import numpy as np
from numpy import *
from os.path import join

import sys
import os
from PyQt5 import uic, QtGui
from PyQt5.QtWidgets import QMainWindow, QApplication, QFileDialog

#Librerias agregadas para manejar la ventana del matplotlib
import matplotlib
import matplotlib.pyplot as plt

#Cursor
import matplotlib.cm as cm

destination_path = r'C:\Tesis\Python'

class PrimerGui(QMainWindow):

    def __init__(self):
        super().__init__()
        uic.loadUi("guideNDVI.ui", self)
        # Programación de los botones del guide
        self.boton_Foto.clicked.connect(self.fn_Red_NIR)
        self.boton_NDVI.clicked.connect(self.fn_NDVI)
        #Botón para cambiar a modo recolección de datos
        self.Modo_Adquisicion.toggled.connect(self.fn_modos_adquisicion)
        #Inializacion
        self.url_directorio = None
        self.img_RED_NIR=None
        self.im_color=None
        self.var=1
        self.indice_g=0
        #Camara
        camera_number = 1 #Es 0 o 1
        self.cap = cv2.VideoCapture( camera_number + cv2.CAP_DSHOW)

        #Para graficar el buho
        buho_ruta=r'C:\Tesis\Python\BUHO_EPN_big.png'
        self.buho.setPixmap(QtGui.QPixmap(buho_ruta))

        #Se desactiva el boton NDVI para esperar que se cargue una imagen
        self.boton_NDVI.setEnabled(False)

    def fn_Red_NIR(self):          #Funcion del RED
        #Activo el boton NDVI
        self.boton_NDVI.setEnabled(True)
        ##Desactivar
        self.url_directorio=self.getFileName()
        print(self.url_directorio)
        self.img_RED_NIR = cv2.imread(self.url_directorio,1)
        #print(type(self.img_RED))
        #print(self.img_RED)
        #print(self.img_RED.shape)
```

```

def fn_NDVI(self):                                #Funcion del NDVI

    self.etiqueta.setText("Imagen NDVI")

    #"Se convierten las imágenes en arreglos flotantes antes de las operaciones
matemáticas"
    imagen_completa = array(self.img_RED_NIR , dtype=float)
    print(type(imagen_completa))
    print(imagen_completa)
    #Se procede al proceso de segmentación
    #Se obtiene la imagen de 1 solo canal correspondiente al canal Rojo
    r = (imagen_completa[:, :, 2]).astype('float')
    print("dimensiones")
    print(r.shape)
    #Se obtiene la imagen de 1 solo canal correspondiente al canal Azul que gracias al
filtro corresponde a la imagen NIR.
    nir=(imagen_completa[:, :, 0]).astype('float')
    print("nir data",nir)
    print(nir.shape)
    #Imagenes
    r_foto=self.img_RED_NIR[:, :, 2]
    nir_foto=self.img_RED_NIR[:, :, 0]
    #cv2.imshow("Imagen roja",r_foto)
    #cv2.imshow("Imagen azul",nir_foto)
    #Guardado de las Imagenes.
    cv2.imwrite(r'C:\Tesis\Python\ndvi_casa\Hoynir_.jpg' , nir_foto)
    cv2.imwrite(r'C:\Tesis\Python\ndvi_casa\Hoyred_.jpg' , r_foto)

    #"Se calcula el índice ndvi"
    ndvi = np.true_divide(np.subtract(nir, r), np.add(nir, r))
    ndvi_index = ndvi
    maximo_ndvi=np.max(ndvi_index)
    minimo_ndvi=np.min(ndvi_index)
    #maximo_ndvi=ndvi_index[ndvi_index==1.]
    print("maximo valor ndvi:",maximo_ndvi)
    print("minimo valor ndvi:",minimo_ndvi)
    print("ndvi_index",ndvi_index)
    print("ndvi_index_type",type(ndvi_index))
    print("ndvi_index_dim: ",ndvi_index.ndim)

    #Se realiza una operación para visualizar la imagen ndvi
    ndvi = (255/(maximo_ndvi-minimo_ndvi))*(ndvi-minimo_ndvi)
    ndvi = ndvi.round()
    print("ndvi:",ndvi)
    print("ndvi shape:",ndvi.shape)

    ndvi_image = np.array(ndvi, dtype=np.uint8)
    #cv2.imshow("NDVI",ndvi_image)
    print("ndvi_image:",ndvi_image)
    print("ndvi_image shape:",ndvi_image.shape)

    #A continuación se aplica un cambio de colores para visualizar de mejor manera los
resultados
    self.im_color = cv2.applyColorMap(ndvi_image, cv2.COLORMAP_DEEPPGREEN)

    #En la siguiente línea se transforma de BGR a RGB con el objetivo de apuntar
correctamente a la columna deseada
    NIR_RED = cv2.cvtColor(self.img_RED_NIR, cv2.COLOR_BGR2RGB)
    #Convertir to HSV para crear la máscara binaria posteriormente
    hsv_NIR_RED = cv2.cvtColor(NIR_RED, cv2.COLOR_RGB2HSV)

```

```

#cv2.imshow("Imagen en el HSV",hsv_NIR_RED)

#Se crean los límites para crear la máscara binaria
low_red = np.array([160, 105, 84])
high_red = np.array([179, 255, 255])
#Creación de la máscara binaria para la obtención de la imagen ndvi quitando las
plantas sanas (inRange(ImagenHSV,liminf,limsuperior))
red_mask = cv2.inRange(hsv_NIR_RED, low_red, high_red)
red_mask = cv2.bitwise_not(red_mask)
cv2.imshow("Mascara binaria",red_mask)
#keep every pixel the same as the original image
#Transformación de matrices
NIR_RED = cv2.cvtColor(NIR_RED, cv2.COLOR_HSV2BGR)
print("red mask shape",red_mask.shape)
print("data red mask",red_mask)
result = cv2.bitwise_and(self.im_color, self.im_color, mask=red_mask)
cv2.imshow("resultado",result)
#En la siguiente línea se realiza el paso de la imagen para no visualizar la parte
sana en la imagen.
#self.im_color=result

#Se realiza la transformación de BGR a RGB para que se pueda dibujar con los mismos
colores
self.im_color=cv2.cvtColor(self.im_color, cv2.COLOR_BGR2RGB)
#Graficar la Imagen NDVI en un plot

#Se crea el objeto figura
fig=plt.figure()
#Se añade uno y solo un gráfico a la figura
ax=fig.add_subplot(111)
#Se crea el objeto image y se muestra al mismo tiempo la imagen previamente procesada
image=ax.imshow(self.im_color, cmap= cm.YlGn_r, interpolation='none')
#Se crea las variables plot_height y plot_width con el objetivo de que el área de
trabajo del mouse coincida con las dimensiones de la imagen
plot_height,plot_width,=self.im_color.shape
#Para la creación de la barra de colores:
#norm = matplotlib.colors.Normalize(vmin=-1., vmax=1.)
norm = matplotlib.colors.Normalize(vmin= minimo_ndvi, vmax= maximo_ndvi)
image.norm=norm
fig.colorbar(image,ax=ax,norm=norm)
height,width=ndvi_index.shape

def format_coord(x, y):
#La posición tanto de el mouse dentro del área de trabajo, como de la matriz ndvi
son las mismas.
    col = x
    row = y
    #Si me encuentro dentro del área de trabajo
    if col>=0 and col<width and row>=0 and row<height:
        z = ndvi_index[int (y),int(x)]
        return 'x=%1.4f, y=%1.4f, NDVI=%1.4f'%(x, y, z)
    else:
        return 'x=%1.4f, y=%1.4f'%(x, y)

ax.format_coord = format_coord

#Cursor
#Para guardar multiples imagenes
self.indice_g += 1
savefigura= "Foto " + str(self.indice_g) + '.jpg'
plt.savefig(savefigura)

```

```

plt.show()
cv2.waitKey(0)

#Antes de guardar volvemos al sistema BGR con el cual trabaja Opencv
self.im_color=cv2.cvtColor(self.im_color, cv2.COLOR_RGB2BGR)

ruta_img=join(destination_path , "Fotografia "+ str(self.indice_g) + ".jpg")
cv2.imwrite(ruta_img , self.im_color)
self.var=self.var+1
self.imagen_etiq.setPixmap(QtGui.QPixmap(ruta_img))

return ndvi_index, ndvi_image

##Función para obtener el directorio
def getFileName(self):
    file_filter = 'Data File (*.png *.jpg *.dat);'
    response = QFileDialog.getOpenFileName(
        parent=self,
        caption='Escoja la fotografía',
        directory=os.getcwd(),
        filter=file_filter,
        initialFilter='Excel File (*.xlsx *.xls)'
    )
    print(response)
    return response[0]

def fn_modos_adquisicion(self):
    if self.Modo_Adquisicion.isChecked()==1:
        while(True):
            if self.Modo_Adquisicion.isChecked()==0:
                cv2.destroyAllWindows()
                break
            Tecla=cv2.waitKey(1) & 0xFF
            # Captura el frame de la cámara web
            ret, frame = self.cap.read()
            # Our operations on the frame come here
            cv2.imshow('Video',frame)
            if Tecla==ord('q'):
                break

if __name__ == '__main__':
    app = QApplication(sys.argv)
    GUI = PrimerGui() # inicializar la clase
    GUI.show() # muestra la clase
    sys.exit(app.exec_())

```



## ANEXO C Manual de usuario

### Prototipo electrónico receptor (manual de usuario)

En este manual, se describen las operaciones de la interfaz gráfica desarrollada para la adquisición de fotografías y el cargado de estas para realizar el cálculo del índice NDVI, así como también se describen las configuraciones iniciales para poner en funcionamiento el prototipo implementado.

- **Descripción:**

El prototipo electrónico implementado permite tanto la captura de las imágenes multispectrales como su procesamiento para adquirir la imagen NDVI. Además, ofrece la visualización del índice NDVI en función de la posición del mouse, la cual es cambiada por un lápiz óptico ubicado en la tapa del prototipo.

- **Configuraciones iniciales:**

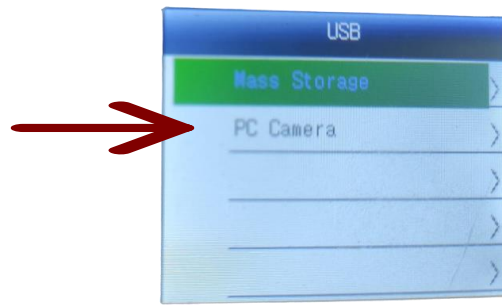
En la Figura C.1 se muestra la vista Lateral del sistema multispectral, en la cual se encuentra el interruptor de encendido del sistema (2), que permite poner en marcha el equipo.



**Figura C.1** Vista Lateral del sistema multispectral [34].

**Tabla C.1** Descripción de los componentes situados en el lateral del prototipo implementado.

Número	Elemento	Descripción
1	Puerto Micro-Usb	Mediante este puerto Micro-Usb, es posible cargar el prototipo implementado.
2	Interruptor de Encendido	Este Interruptor enciende todo el Sistema
3	Interruptor de la cámara	Este interruptor controla exclusivamente la alimentación de la cámara web y debe estar en la posición "1" para encender la cámara.



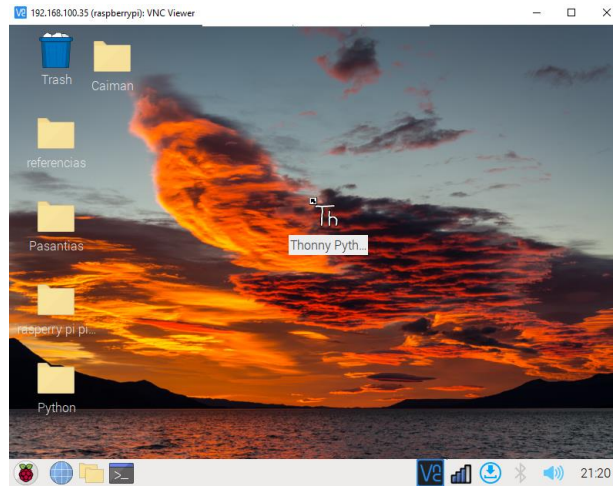
**Figura C.2** Menú de la Cámara Web [34].

La cámara será detectada por la Raspberry Pi al encenderse el sistema. Posteriormente, se debe elegir la opción "PC Camera", y presionar el botón "OK" de la cámara. El prototipo cuenta con orificios cercanos a estos botones para facilitar el acceso. Después de seleccionar correctamente la opción "PC Camera", se visualizará en la pantalla de la cámara web, como se muestra en la Figura C.3.

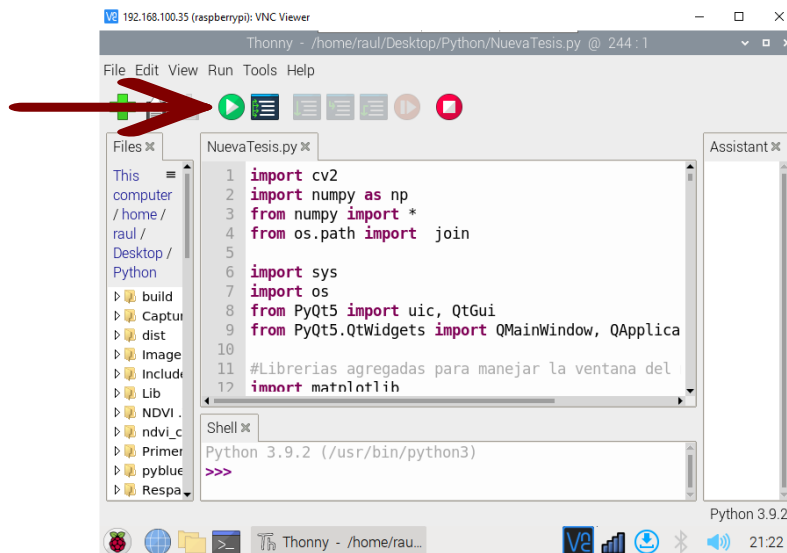


**Figura C.3** Funcionamiento de la cámara como cámara Web [34].

Luego, se inicia Thonny, el entorno de desarrollo de Python, y se ejecuta el programa predeterminado en Thonny al presionar el botón de ejecutar, como se muestra en la Figura C.5.

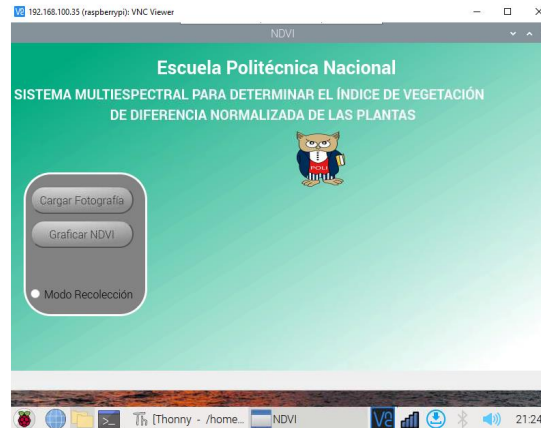


**Figura C.4** Thonny, entorno de desarrollo de Python



**Figura C.5** Ejecutar el programa desarrollado

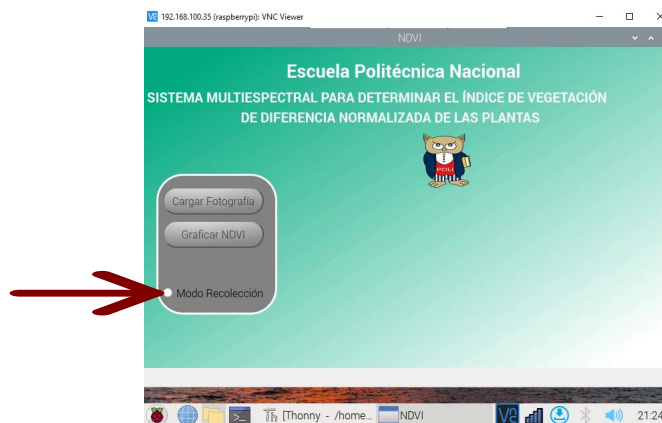
Al completar estos pasos, se desplegará la interfaz gráfica desarrollada y estará lista para la adquisición de imágenes multispectrales y la carga de imágenes adquiridas por el prototipo para calcular el índice NDVI.



**Figura C.6** Ventana con la Interfaz Gráfica.

- **Toma de Fotografías:**

Para adquirir imágenes multiespectrales, se hace clic en la opción “Modo de recolección”, lo que despliega otra ventana que muestra lo que se encuentra frente a la cámara web.



**Figura C.7** Modo de recolección de Fotografías.

Se desplegará una ventana que muestra la vista de la cámara web. Para guardar el fotograma actual en formato .jpg, se debe aplastar el botón correspondiente, como se muestra en la Figura C.8. Las imágenes se guardarán numeradas desde “Fotografía 0”.



Figura C.8 Botón para adquirir fotografías de la cámara web.

- Partes del prototipo:



Figura C.9 Componentes del prototipo multispectral.

**Tabla C.2** Descripción de todos los componentes del sistema multiespectral.

Número	Elemento	Descripción
1	Pantalla 3.5 pulgadas	En esta pantalla se puede visualizar la interfaz gráfica desarrollada.
2	Pulsador para Capturar Imágenes	Este pulsador permite la adquisición de las imágenes multiespectrales.
3	Soporte del lápiz óptico	En este soporte, se coloca el lápiz óptico para el uso de la interfaz grafica táctil.
4	Botón "OK" de la cámara	Este botón permite confirmar la acción mientras se navega por el menú de la cámara.
5	Filtro óptico seleccionado	Este filtro permite la adquisición de las imágenes multiespectrales, gracias al sistema óptico implementado.
6	Botones de navegación de la cámara.	Estos botones permiten navegar a través del menú de la cámara Web
7	Puerto de carga micro-USB	Puerto de carga Micro-Usb que permite cargar al sistema para que sea portátil.
8	Interruptor de encendido del sistema.	Este Interruptor permite encender el sistema multiespectral.
9	Interruptor de la cámara web	Este Interruptor permite energizar exclusivamente la cámara web.

- **Calcular el Índice NDVI:**

La Figura C.10 muestra el botón "Graficar NDVI", que permite desplegar la figura con la imagen NDVI y sus correspondientes índices. Este botón está bloqueado hasta que se cargue una fotografía RGB adquirida por el sistema multiespectral.



**Figura C.10** Botón Para Graficar la Imagen NDVI y los Índices correspondientes.