



UNIVERSIDAD DE
GUANAJUATO



Guía para las mediciones.

Agrifotónica: Aplicaciones de la óptica en la agricultura

Equipo de investigación:

Omar Jiménez Sánchez
Alejandro León Mendoza
Ricardo Brayan Juárez Valencia
Jesús Arath Cabrera Elías

Asesor del proyecto:

Dr. Oleksiy Shulika

Salamanca 2024

ÍNDICE

Contenido

RESUMEN	3
INTRODUCCIÓN	3
OBJETIVOS	3
JUSTIFICACIÓN.....	4
METODOLOGÍA.....	4
AMBIENTE CONTROLADO.....	4
ESPECTROSCOPIA	11
CONCLUSIONES	16

RESUMEN

En este proyecto, se presenta una técnica basada en la espectroscopía para caracterizar las propiedades ópticas de las plantas, enfocándose en hojas de planta de fresas cultivadas en un ambiente controlado dentro del laboratorio. Utilizando espectros de reflectancia y transmitancia, para evaluar eficazmente la salud vegetal, ahorrando tiempo y recursos.

Para llevar a cabo este estudio, se realizaron pruebas con un arreglo de una fuente de luz y fibras ópticas con hojas de fresa en diversos estados: desde verdes y sanas, hasta secas y enfermas. Las variaciones en los espectros obtenidos se atribuyen a cambios en la composición química y los pigmentos de las hojas, los cuales reflejan su estado de salud. Las hojas utilizadas en el experimento se obtuvieron en un ambiente controlado, donde se monitorearon parámetros como la humedad, iluminación y temperatura de las plantas.

INTRODUCCIÓN

La producción agrícola enfrenta una serie de desafíos significativos que afectan tanto la eficiencia como la calidad de los productos. Entre estos desafíos están la detección temprana de enfermedades y plagas, el manejo eficiente de los recursos y la mejora continua de la calidad de producción. Estos factores no solo impactan la rentabilidad del cultivo, sino también la sostenibilidad y competitividad del sector agrícola.

En el cultivo de fresas, la aparición de enfermedades y plagas puede causar pérdidas significativas si no se detectan y controlan a tiempo. Los métodos tradicionales de inspección visual pueden ser ineficaces y consumir mucho tiempo, lo que lleva a intervenciones tardías y mayores daños. Por ello, es crucial implementar técnicas más precisas y rápidas para la detección de estos problemas.

La mejora de la calidad de producción de las fresas es un objetivo primordial para los agricultores. La calidad se mide en apariencia y sabor, y en el valor nutritivo y la resistencia de las fresas. Los métodos tradicionales pueden no alcanzar los estándares de calidad exigidos por el mercado, afectando la competitividad y las ganancias de los productores.

OBJETIVOS

- Utilizar la espectroscopia para el estudio de las fresas en el ambiente controlado, obteniendo datos valiosos para la optimización de prácticas agrícolas.
- Realizar un estudio espectroscópico para facilitar el crecimiento óptimo de las plantas de fresas para están detectar enfermedades y plagas para un mejor manejo eficiente de recursos.

- Desarrollar un entorno controlado para el cultivo de fresas que permita el suministro de recursos en proporciones ideales, mejorando la calidad y consistencia de las muestras.
- Contribuir a una producción agrícola más eficiente y sostenible, asegurando la fiabilidad de los datos obtenidos y ofreciendo beneficios significativos en términos de rentabilidad.

JUSTIFICACIÓN

El objetivo principal de este proyecto es crear un ambiente controlado para dos plantas de fresa, asegurando que se mantengan en condiciones óptimas para su crecimiento para su posterior utilización en el estudio de espectroscopia. Este entorno controlado permitirá que las plantas reciban los recursos necesarios, tales como luz y agua, en proporciones ideales para su óptimo desarrollo. La creación de este ambiente controlado es importante para garantizar que las hojas de las plantas de fresa estén en las perfectas condiciones, para su estudio mediante técnicas de espectroscopia, y reflejen resultados precisos y fiables en el análisis espectroscópico.

La implementación de un ambiente controlado no solo mejora la calidad y la consistencia de las muestras, sino que también contribuye a un mejor entendimiento de cómo factores ambientales específicos que influyen en el crecimiento y desarrollo de las fresas. Esto, a su vez, puede proporcionar información valiosa para la optimización de prácticas agrícolas y la mejora de la producción de fresas en condiciones controladas.

La justificación del proyecto es la necesidad de obtener hojas de fresa de alta calidad y uniformidad para su estudio espectroscópico, que ayudará a conocer la salud de la planta para mejorar los procesos agrícolas en un ambiente controlado. Este enfoque asegura la fiabilidad de los datos obtenidos y que puede ofrecer beneficios significativos en cuanto a eficiencia y calidad en la producción agrícola de fresas y potencialmente de otras verduras.

METODOLOGÍA

AMBIENTE CONTROLADO

1. Selección del Entorno

El ambiente controlado se ubicó dentro del laboratorio de optoelectrónica.

2. Condiciones Ambientales



Figura 01: Ambiente controlado completo.

Para la creación del ambiente controlado se decidió utilizar dos plantas de fresa para su uso conforme a los parámetros ambientales que se controlaron, los cuales son:

- Temperatura ambiental: Controlada con una lámpara incandescente.
- Humedad del suelo (planta): Monitorizada mediante sensores de humedad.
- Temperatura del suelo (planta): Monitorizada mediante sensores de temperatura.
- Luz ambiental: Proporcionada por la lámpara incandescente que simula el ciclo día-noche.
- Los rangos específicos establecidos en el programa son:
 - Humedad del suelo para Planta 1: Valores crudos de 538 a 255 mapeados a 0% a 100% de humedad. *
 - Humedad del suelo para Planta 2: Valores crudos de 489 a 216 mapeados a 0% a 100% de humedad. *
 - Riego activado cuando la humedad del suelo cae por debajo del 40%.

* Los valores censados cambian debido a la configuración de fábrica del sensor.

```
// Lee la humedad del suelo del primer sensor capacitivo
int soilMoisture1 = analogRead(SOIL_SENSOR_1); // Lee el valor analógico del
primer sensor capacitivo
// Convierte la lectura a un porcentaje
float soilMoisturePercent1 = map(soilMoisture1, 538, 255, 0, 100); // Mapea el
valor leído a un porcentaje
```

```

soilMoisturePercent1 = constrain(soilMoisturePercent1, 0, 100); // Asegura que el
valor esté entre 0 y 100

// Lee la humedad del suelo del segundo sensor capacitivo
int soilMoisture2 = analogRead(SOIL_SENSOR_2); // Lee el valor analógico del
segundo sensor capacitivo






// Convertir la lectura a un porcentaje
float soilMoisturePercent2 = map(soilMoisture2, 489, 216, 0, 100); // Mapea el
valor leído a un porcentaje
soilMoisturePercent2 = constrain(soilMoisturePercent2, 0, 100); // Asegura que el
valor esté entre 0 y 100

```

Figura 02: Parte del código donde se mapean los valores crudos de los sensores de humedad del suelo a porcentajes.

3. Equipos y Herramientas

- Lámpara incandescente: controla la temperatura y proporciona luz.
- Componentes utilizados:

Cantidad	Componente	Imagen	Descripción
1	Arduino UNO		Microcontrolador para ejecutar el programa y coordinar los sensores y actuadores.
1	DHT22		Sensor de temperatura y humedad ambiental.
1	MicroSD		Módulo para tarjetas microSD para almacenamiento de datos.
1	RTC (Real Time Clock)		Módulo para obtener la fecha y hora actual.
2	DS18B20		Sensor de temperatura de suelo.

2	SEN-HS-CAP		Sensor capacitivo de humedad de suelo.
2	Relevadores		Módulo para controlar bombas de agua en base a la humedad del suelo.
2	Mini bomba de agua		Controladas por los relés para regar las plantas según la humedad del suelo.

Tabla 01: Componentes utilizados en el ambiente controlado

4. Configuración y Mantenimiento

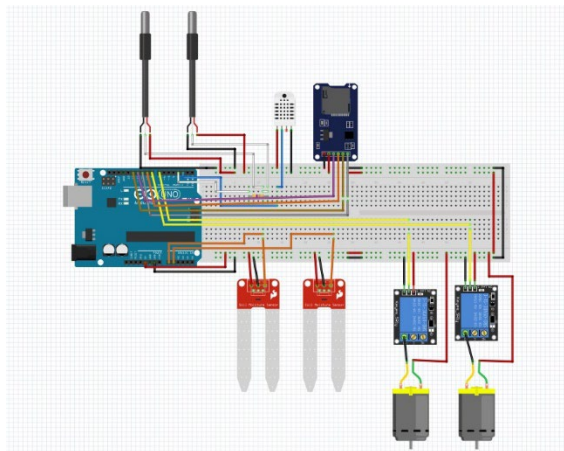


Figura 03: Diagrama de las conexiones para el ambiente controlado.

- Se usó un Arduino Uno programado por un ordenador para controlar y monitorear las condiciones ambientales del invernadero.
- La humedad del suelo se controla mediante sensores capacitivos de humedad y un sistema de riego automatizado, el cual utiliza relés que activan una bomba de agua para regar para cada una de las dos plantas.
- La temperatura y la luz se controlan mediante una lámpara incandescente que simula el ciclo día-noche.
- Las verificaciones se realizaron diariamente de lunes a viernes en diferentes horarios extrayendo la memoria microSD para el procesamiento de los datos recopilados, mientras que los ajustes de humedad se hicieron automáticamente conforme las lecturas de los sensores y su programación.

```

• #include "DHT.h" // Incluye la biblioteca DHT para trabajar con sensores DHT
• #include <OneWire.h> // Incluye la biblioteca OneWire para trabajar con el bus OneWire
• #include <DallasTemperature.h> // Incluye la biblioteca DallasTemperature para trabajar con sensores DS18B20
• #include <SD.h> // Incluye la biblioteca SD para trabajar con tarjetas SD
• // Definiciones para el sensor DHT22
• #define DHTPIN 2 // Pin digital donde está conectado el sensor DHT22
• #define DHTTYPE DHT22 // Define el tipo de sensor DHT (DHT22)
• DHT dht(DHTPIN, DHTTYPE); // Crea una instancia del sensor DHT22
• // Definiciones para el sensor DS18B20
• #define ONE_WIRE_BUS 3 // Pin digital donde está conectado el bus OneWire para el DS18B20
• OneWire oneWire(ONE_WIRE_BUS); // Crea una instancia del bus OneWire
• DallasTemperature sensors(&oneWire); // Crea una instancia del sensor DS18B20 usando el bus OneWire
• DeviceAddress sensor1Address; // Dirección del primer sensor DS18B20
• DeviceAddress sensor2Address; // Dirección del segundo sensor DS18B20
• // Definiciones para los sensores de humedad de suelo capacitivo
• #define SOIL_SENSOR_1 A0 // Pin analógico donde está conectado el primer sensor de humedad de suelo capacitivo
• #define SOIL_SENSOR_2 A1 // Pin analógico donde está conectado el segundo sensor de humedad de suelo capacitivo
• // Definiciones para los relés
• #define RELAY_PIN_1 7 // Pin digital donde está conectado el primer relé (Planta 1)
• #define RELAY_PIN_2 8 // Pin digital donde está conectado el segundo relé (Planta 2)
• // Definiciones para el módulo microSD
• #define SD_CS_PIN 10 // Pin digital donde está conectado el pin CS del módulo SD
• void setup() {
•     Serial.begin(9600); // Inicializa la comunicación serie a 9600 bps
•     Serial.println("Sensores: DHT22, DS18B20, Sensores capacitivos, microSD y relés!"); // Imprime un mensaje de
inicio
•     // Inicializa el sensor DHT22
•     dht.begin();
•     // Inicializa los sensores DS18B20
•     sensors.begin();
•     // Busca y asigna las direcciones de los sensores DS18B20
•     if (!sensors.getAddress(sensor1Address, 0)) {
•         Serial.println("No se encontró el primer sensor DS18B20!");
•     }
•     if (!sensors.getAddress(sensor2Address, 1)) {
•         Serial.println("No se encontró el segundo sensor DS18B20!");
•     }
•     // Configura la precisión de los sensores DS18B20
•     sensors.setResolution(sensor1Address, 9);
•     sensors.setResolution(sensor2Address, 9);
•     // Inicializa la tarjeta SD
•     if (!SD.begin(SD_CS_PIN)) {
•         Serial.println("Error inicializando la tarjeta SD!");
•         return;
•     }
•     Serial.println("Tarjeta SD inicializada correctamente.");
•     // Configura los pines de los relés como salidas
•     pinMode(RELAY_PIN_1, OUTPUT);
•     pinMode(RELAY_PIN_2, OUTPUT);
•     // Apaga las bombas inicialmente
•     digitalWrite(RELAY_PIN_1, LOW);
•     digitalWrite(RELAY_PIN_2, LOW);
• }
• void loop() {
•     // Línea de espacio entre bloques de mediciones
•     Serial.println();
•     delay(2000); // Espera 2 segundos entre lecturas
•     // Abre el archivo para escritura
•     File dataFile = SD.open("dataLog.txt", FILE_WRITE);
•     // Leer la humedad y la temperatura del DHT22
•     float humidity = dht.readHumidity(); // Lee la humedad del DHT22
•     float temperatureDHT = dht.readTemperature(); // Lee la temperatura del DHT22
•     // Verificar si hay fallos en la lectura del DHT22

```



```

•   if (isnan(humidity) || isnan(temperatureDHT)) {
•       Serial.println("Error leyendo el sensor DHT22!"); // Imprime un mensaje de error si la lectura falla
•   } else {
•       // Mostrar los valores del DHT22 en el monitor serie
•       Serial.print("Humedad ambiente (DHT22): "); // Imprime el texto "Humedad: "
•       Serial.print(humidity); // Imprime el valor de la humedad
•       Serial.print(" %\t"); // Imprime el símbolo de porcentaje y una tabulación
•       Serial.print("Temperatura ambiente (DHT22): "); // Imprime el texto "Temperatura (DHT22): "
•       Serial.print(temperatureDHT); // Imprime el valor de la temperatura
•       Serial.println(" *C"); // Imprime el símbolo de grados Celsius y una nueva línea
•       // Guardar los valores del DHT22 en la tarjeta SD
•       if (dataFile) {
•           dataFile.print("Humedad ambiente (DHT22): ");
•           dataFile.print(humidity);
•           dataFile.print(" %\t");
•           dataFile.print("Temperatura ambiente (DHT22): ");
•           dataFile.print(temperatureDHT);
•           dataFile.println(" *C");
•       }
•   }
•   // Solicitar la temperatura de los sensores DS18B20
•   sensors.requestTemperatures();
•   // Leer la temperatura del suelo del primer DS18B20
•   float temperatureDS18B20_1 = sensors.getTempC(sensor1Address);
•   if (temperatureDS18B20_1 == DEVICE_DISCONNECTED_C) {
•       Serial.println("Error leyendo el primer sensor DS18B20!"); // Imprime un mensaje de error si la lectura
falla
•   } else {
•       Serial.print("Temperatura de suelo Planta 1 (DS18B20): ");
•       Serial.print(temperatureDS18B20_1);
•       Serial.println(" *C");
•       // Guardar los valores del primer DS18B20 en la tarjeta SD
•       if (dataFile) {
•           dataFile.print("Temperatura de suelo Planta 1 (DS18B20): ");
•           dataFile.print(temperatureDS18B20_1);
•           dataFile.println(" *C");
•       }
•   }
•   // Leer la temperatura del suelo del segundo DS18B20
•   float temperatureDS18B20_2 = sensors.getTempC(sensor2Address);
•   if (temperatureDS18B20_2 == DEVICE_DISCONNECTED_C) {
•       Serial.println("Error leyendo el segundo sensor DS18B20!"); // Imprime un mensaje de error si la lectura
falla
•   } else {
•       Serial.print("Temperatura de suelo Planta 2 (DS18B20): ");
•       Serial.print(temperatureDS18B20_2);
•       Serial.println(" *C");
•       // Guardar los valores del segundo DS18B20 en la tarjeta SD
•       if (dataFile) {
•           dataFile.print("Temperatura de suelo Planta 2 (DS18B20): ");
•           dataFile.print(temperatureDS18B20_2);
•           dataFile.println(" *C");
•       }
•   }
•   //*****Calibración de sensor de humedad*****
•   // Leer la humedad del suelo del primer sensor capacitivo
•   int soilMoisture1 = analogRead(SOIL_SENSOR_1); // Lee el valor analógico del primer sensor capacitivo
•   Serial.print("Valor crudo del sensor 1: ");
•   //Serial.println(soilMoisture1); // Imprime el valor crudo del sensor 1
•   //*****
•   // Convertir la lectura a un porcentaje (opcional, depende de tu configuración y calibración)
•   float soilMoisturePercent1 = map(soilMoisture1, 538, 255, 0, 100); // Mapea el valor leído a un porcentaje
•   soilMoisturePercent1 = constrain(soilMoisturePercent1, 0, 100); // Asegura que el valor esté entre 0 y 100

```

```

• Serial.print("Humedad del suelo Planta 1 (Capacitivo): "); // Imprime el texto "Humedad del suelo Sensor 1
(Capacitivo): "
• Serial.print(soilMoisturePercent1); // Imprime el valor de la humedad del suelo en porcentaje
• Serial.println(" %"); // Imprime el símbolo de porcentaje y una nueva línea
• // Guardar los valores del primer sensor capacitivo en la tarjeta SD
• if (dataFile) {
•     dataFile.print("Humedad del suelo Planta 1 (Capacitivo): ");
•     dataFile.print(soilMoisturePercent1);
•     dataFile.println(" %");
• }
• //*****Calibración de sensor de humedad*****
• // Leer la humedad del suelo del segundo sensor capacitivo
• int soilMoisture2 = analogRead(SOIL_SENSOR_2); // Lee el valor analógico del segundo sensor capacitivo
• Serial.print("Valor crudo del sensor 2: ");
• //Serial.println(soilMoisture2); // Imprime el valor crudo del sensor 2
• //*****
• // Convertir la lectura a un porcentaje (opcional, depende de tu configuración y calibración)
• float soilMoisturePercent2 = map(soilMoisture2, 489, 216, 0, 100); // Mapea el valor leído a un porcentaje
• soilMoisturePercent2 = constrain(soilMoisturePercent2, 0, 100); // Asegura que el valor esté entre 0 y 100
• Serial.print("Humedad del suelo Planta 2 (Capacitivo): "); // Imprime el texto "Humedad del suelo Sensor 2
(Capacitivo): "
• Serial.print(soilMoisturePercent2); // Imprime el valor de la humedad del suelo en porcentaje
• Serial.println(" %"); // Imprime el símbolo de porcentaje y una nueva línea
• // Guardar los valores del segundo sensor capacitivo en la tarjeta SD
• if (dataFile) {
•     dataFile.print("Humedad del suelo Planta 2 (Capacitivo): ");
•     dataFile.print(soilMoisturePercent2);
•     dataFile.println(" %");
• }
• // Control de las bombas basándose en los valores de humedad del suelo
• if (soilMoisturePercent1 > 40) {
•     digitalWrite(RELAY_PIN_1, HIGH); // Enciende la bomba para la Planta 1 si la humedad está por debajo del
umbral
• } else {
•     digitalWrite(RELAY_PIN_1, LOW); // Apaga la bomba para la Planta 1 si la humedad está por encima del umbral
• }
• if (soilMoisturePercent2 > 40) {
•     digitalWrite(RELAY_PIN_2, HIGH); // Enciende la bomba para la Planta 2 si la humedad está por debajo del
umbral
• } else {
•     digitalWrite(RELAY_PIN_2, LOW); // Apaga la bomba para la Planta 2 si la humedad está por encima del umbral
• }
• // Cierra el archivo después de escribir los datos
• if (dataFile) {
•     dataFile.close();
• }
• }

```

Figura 04: Código completo utilizado para la configuración del Arduino y sensores.

5. Protocolo de Riego

1. El sensor de humedad realiza mediciones cada 500 milisegundos.
2. Si la humedad del suelo de la respectiva planta baja del 40%, activa la bomba de agua para el riego automático de la misma, hasta que se eleve por encima de ese umbral.
3. Los datos se guardan en la memoria microSD para su posterior procesamiento.

6. Monitoreo del Crecimiento

- Métodos Visuales:

Se utilizó la observación visual diaria para evaluar el crecimiento y desarrollo de las plantas. Esto incluye la inspección de hojas, tallos y raíces para detectar signos visibles de salud, así como coloración adecuada para las posteriores mediciones

- **Métodos Espectroscópicos:**

Se realizaron mediciones espectroscópicas para verificar la salud de las plantas. Analizando las diferentes coloraciones de las hojas, lo cual puede permitir el posible análisis de la composición química de las hojas y detectar posibles deficiencias nutricionales, estrés hídrico o presencia de enfermedades.

- **Frecuencia de Medición:**

Las mediciones de humedad del suelo se hicieron cada 500 milisegundos automáticamente mediante sensores capacitivos de humedad, asegurando un monitoreo constante y preciso de las condiciones del suelo. Esta alta frecuencia de medición le evita un mayor estrés hídrico a la planta, lo que permite detectar y corregir rápidamente cualquier desviación en los niveles de humedad, garantizando un riego adecuado y oportuno, así como su buen crecimiento.

ESPECTROSCOPIA

1. Preparación de las Muestras:

- **Selección de Hojas:**

Se seleccionaron tres tipos diferentes de hojas de las plantas de fresa para los estudios espectroscópicos:

- Una hoja verde y sana.
- Una hoja amarilla y enferma.
- Una hoja seca.



Figura 05: Hojas seleccionadas para el experimento; a la izquierda inferior, hoja seca, hoja enferma y hoja sana.

- Preparación:

Las hojas seleccionadas se prepararon adecuadamente, asegurando que cada hoja representaría su salud y características deseadas para obtener datos precisos y comparables, además, se aplanaron para obtener mejores resultados en la medición espectroscópica de su reflectancia y transmitancia.

2. Equipo de Espectroscopia

Cantidad	Componente	Imagen	Descripción
1	Fuente de luz IR SLS202		La Fuente de Luz IR SLS202 emite luz infrarroja estable y de alta intensidad (450 - 5500 nm).
1	Espectrómetro Aurora4000 GE-UV-NIR		El Espectrómetro Aurora4000 GE-UV-NIR es un dispositivo utilizado para analizar luz en las regiones UV, visible e infrarroja cercana.
1	Fibra óptica		Fibra óptica multimodo utilizada para permitir el paso de la luz al experimento.
1	Programa Spectral Analysis		Programa de procesamiento de datos Spectral Analysis.

Tabla 02: Componentes utilizados en la parte espectroscópica.

Este equipo permitió la obtención de datos precisos sobre la reflectancia y transmitancia de las hojas en diferentes condiciones de salud.

3. Protocolo de Medición

- Procedimiento:

1. Se configuró el arreglo experimental y se abrió el programa para el procesamiento de datos.

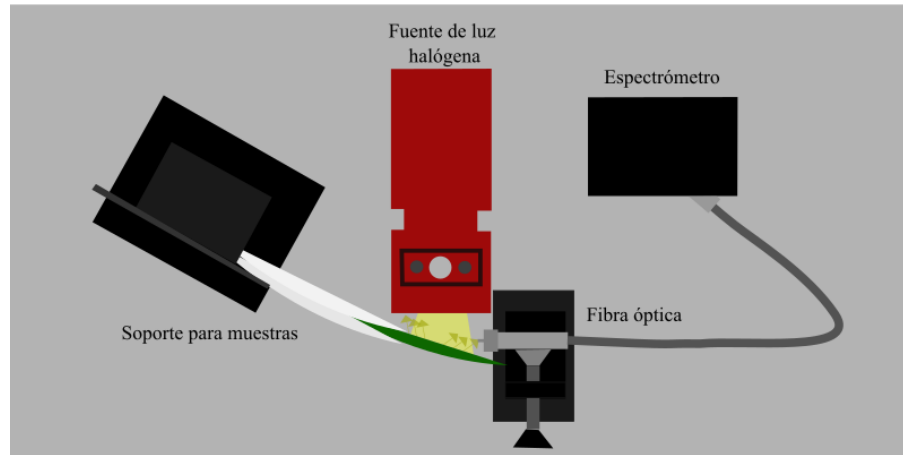


Figura 06: Arreglo experimental utilizado

2. Se seleccionaron las hojas a estudiar: una hoja totalmente verde, otra hoja amarilla o con partes amarillentas, y una hoja seca.



Figura 07: Hojas seleccionadas para el estudio; de izquierda a derecha, hoja seca, hoja enferma y hoja sana.

3. Se colocó una hoja blanca en la base del arreglo para medir su reflectancia a un ángulo de 45° , para tener una base con la cual comparar sus resultados con las hojas de la planta de fresa.



Figura 08: Arreglo realizado con la hoja blanca.

4. Cada hoja se colocó en la base del arreglo para medir su reflexión a un ángulo de 45° , tomando varias muestras de cada una.

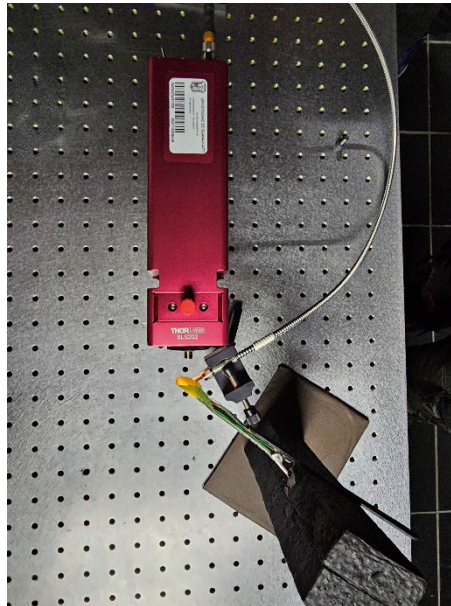


Figura 09: Arreglo realizado con la hoja de fresa.

5. Los datos obtenidos se guardaron para su posterior procesamiento en Matlab.

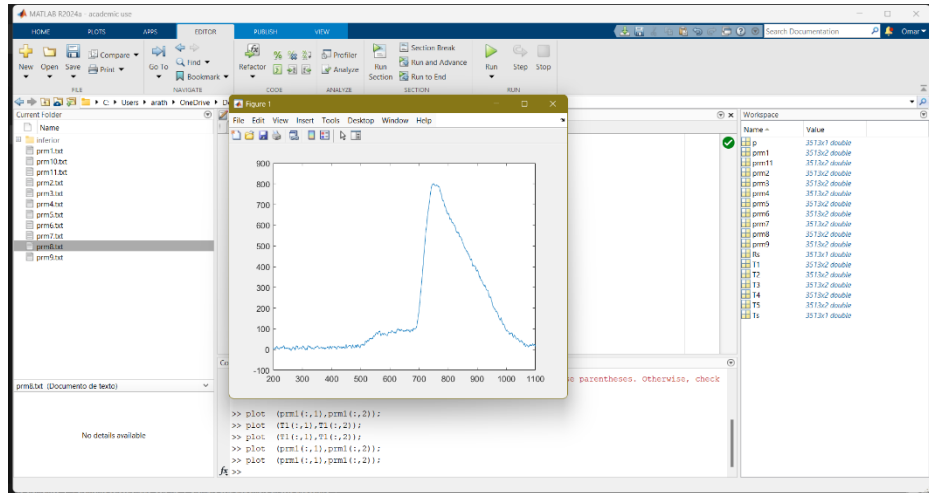


Figura 10: Espectro de onda de la reflectancia de una hoja de fresa fresca y verde.

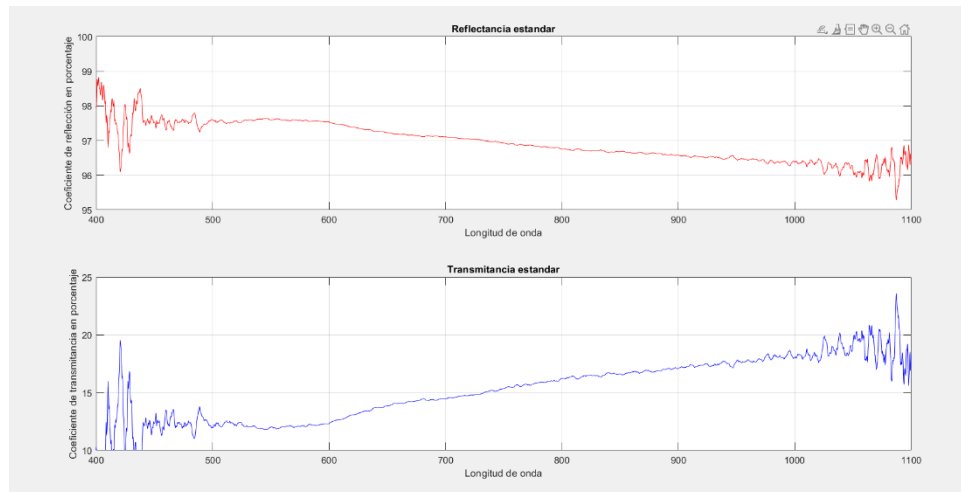


Figura 11: Reflectancia y transmitancia de una hoja de papel blanca.

- Frecuencia de Medición:

Se realizaron múltiples mediciones diarias por cada miembro del equipo para obtener una variedad de resultados. Estos datos, en promedio, permiten la repetición del experimento en futuras investigaciones y aseguran la fiabilidad de los resultados.

4. Análisis de Datos

- Los datos espectroscópicos obtenidos se analizaron para calcular los coeficientes de reflectancia y transmitancia de las hojas. Estos coeficientes proporcionan información sobre la cantidad de luz reflejada y transmitida por las hojas, lo cual es crucial para evaluar su salud y condición.

5. Control de Variables

- Minimización del Ruido:
 - Se grabó el ruido ambiental para minimizarlo y restarlo de la señal principal, obteniendo así una señal más limpia y precisa.
- Condiciones Constantes:
 - Se aseguraron condiciones constantes de iluminación y ángulo de reflexión durante todas las mediciones.
 - Se controlaron la temperatura, iluminación y la humedad del ambiente para evitar que estas variables afectaran los resultados, garantizando la precisión y fiabilidad de las mediciones espectroscópicas.

CONCLUSIONES

Al finalizar este proyecto de verano científico, se ha adquirido una comprensión profunda sobre cómo la luz interactúa con la materia. Se ha aprendido a identificar cómo cambia la materia orgánica a lo largo del tiempo y en respuesta a deficiencias de nutrientes. Este conocimiento ha sido clave para desarrollar métodos no invasivos para medir estos cambios, utilizando técnicas espectroscópicas para obtener datos precisos sin dañar las plantas.

Además, se ha logrado integrar esta información espectroscópica con programación en Matlab, lo que permite un análisis espectral de la salud de las plantas y la gestión de nutrientes. Este enfoque es fundamental para implementar procesos que optimicen el desempeño de las cosechas en volúmenes grandes. Al reducir el gasto en recursos para el mantenimiento y mejorar las condiciones de crecimiento. La capacidad de detectar deficiencias de nutrientes en etapas tempranas es crucial para evitar pérdidas efectivas y asegurar una producción de alta calidad. Este proyecto ha demostrado cómo la tecnología puede ser utilizada para mejorar las prácticas agrícolas, proporcionando una herramienta valiosa para el manejo eficiente y sostenible de cultivos.