

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PROYECTO: DESARROLLO DE UN SISTEMA DE DETECCIÓN DEL NIVEL DE RADIACIÓN UV DENTRO DE INVERNADEROS AGRÍCOLAS

**Componente: Desarrollo e implementación de un sistema de
procesamiento análisis e interpretación los datos obtenidos en
un prototipo UV**

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

CHRISTIAN JOHAO SÁNCHEZ JUMBO

christian.sanchez@epn.edu.ec

DIRECTOR: DR.-ING. PAÚL MARCELO POZO PALMA

marcelo.pozo@epn.edu.ec

DMQ, julio 2024

CERTIFICACIONES

Yo, Christian Johao Sánchez Jumbo 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.

JOHAO SÁNCHEZ

Certifico que el presente trabajo de integración curricular fue desarrollado por JOHAO SÁNCHEZ, bajo mi supervisión.

DR.-ING. MARCELO POZO
DIRECTOR

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.

CHRISTIAN JOHAO SÁNCHEZ JUMBO

DR. ING. MARCELO POZO

DEDICATORIA

Dedico este trabajo a mi mamá Teresa, a mi hermana Scarlet, a mi papá Christian y a mi abuelo José. Al ser apoyo incondicional durante mi vida universitaria y por ser testigos del esfuerzo invertido para lograr esta meta.

También dedico este trabajo a la familia Sánchez y familia Jumbo como muestra de afecto por su apoyo durante mi formación profesional.

Finalmente, quiero dedicar este trabajo a mí mismo como muestra de que la perseverancia y cariño a lo que hago me llevará a cumplir mis objetivos. Como primer politécnico en la familia espero inspirar a futuras generaciones en mi familia a cumplir sus sueños.

AGRADECIMIENTO

Agradezco a las siguientes personas por ser pilares fundamentales durante mi formación como persona, deportista, artista y profesional:

A mis tutores, Dr. Eduardo Ávalos, Dr. Marcelo Pozo, a los docentes Ing. Ma. Fernanda Trujillo e Ing. Víctor Santos por su guía y enseñanza durante el desarrollo de este trabajo.

Al MSc. Andrés Mena y MSc. Jorge Llumiquinga por su aporte a mi formación deportiva para continuar con mis pasiones en representación de la EPN: fútbol 11 y fútbol de salón.

Al Dr. Ramiro Analuisa y al Coro Mixto Politécnico por inspirarme a nuevas pasiones durante esta etapa: la música y el arte. Gracias por estar, me salvaron la vida.

A mis colegas ingenieros y futuros ingenieros que han sido parte de este viaje brindando experiencias, conocimiento y sobre todo su sincera amistad: Daniela, Carla, Gabriela, Ian, Nathaly, Andrés, Renato, Alejandro, Betty, Pablo, Henry, Josselyn, Karen B, Karen Y, Kevin, Luis, Ma. José, Mauricio, Ney, Samuel, Sebastián, Steve, Tamara, Erick, Wellington, Xavier H, Xavier E, Bryan, Joseph.

A mi compañero de este trabajo, Marco Salazar, por su compromiso, entrega y sobre todo amistad para finalizar nuestra carrera universitaria.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
RESUMEN	VIII
ABSTRACT	IX
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	1
1.2 Objetivos específicos	1
1.3 Alcance	1
1.4 Marco teórico	2
En este capítulo se detallarán partes esenciales del trabajo, ampliando teóricamente sus definiciones, características técnicas y ecuaciones matemáticas según el caso. Aquí se va a desarrollar además cualquier definición que se requiera revisar para entender los siguientes capítulos. 1.4.1 Espectrómetro de fibra óptica Ocean Optics USB-4000.....	2
1.4.2 Secciones transversales para absorción y dispersión de fotones	3
1.4.3 Suavizado SavitzkiGolay	4
1.4.4 App Designer de Matlab.....	5
1.4.5 Módulo HC-05	6
2 METODOLOGÍA	7
2.1 Diseño general	7
2.2 Conexión entre Matlab y módulo Bluetooth	8
2.3 Comunicación entre espectrómetro USB – 4000 y App Designer (Matlab)	10
2.4 Desarrollo de Interfaz para adquisición de espectros.	12
2.5 Cálculo de coeficiente de atenuación.....	15
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	17
3.1 Resultados	17
3.1.1 Validación de datos	17
3.1.2 Adquisición de espectros	20
3.1.3 Comunicación entre App Designer y Arduino por Bluetooth	26
3.2 Conclusiones	27
3.3 Recomendaciones	28
4 REFERENCIAS BIBLIOGRÁFICAS	28

ANEXO I Tutorial de uso del HMI.....	30
ANEXO II Toma de espectros de focos de colores.	32
ANEXO III Evidencias de toma de datos en Yachay Tech.	33
ANEXO IV Evidencias de toma de datos en Jardín Botánico de Quito.	34
ANEXO V Código de App Designer de Matlab.....	35

ÍNDICE DE FIGURAS

Figura 1 Espectrómetro USB 4000-UV-VIS-ES.....	3
Figura 2 Haz de luz ingresando a una placa [2]......	3
Figura 3 Interfaz de desarrollo en App Desginer.....	6
Figura 4 Módulo Bluetooth HC-05	6
Figura 5 Diseño general del proceso	8
Figura 6 Resultado de comando <i>instrhwinfo</i>	9
Figura 7 Nombres de dispositivos Bluetooth disponibles.....	9
Figura 8 IDs de dispositivos Bluetooth disponibles.....	9
Figura 9 Creación de dispositivo Bluetooth.	10
Figura 10 Abrir puerto serial	10
Figura 11 Escribir en puerto serial.....	10
Figura 12 Cerrar puerto y eliminar dispositivos.....	10
Figura 13 Vectores adquiridos y almacenados en el Work Space en Matlab.	11
Figura 14 Curva obtenida con el ejemplo de Matlab	11
Figura 15 Resultados de las pruebas realizadas del filtro digital	12
Figura 16 Acercamiento a señal filtrada.....	12
Figura 17 Campo de entrada para tiempo de integración	13
Figura 18 Ejes para graficar espectros.	13
Figura 19 Botones de adquisición y limpieza de ejes.....	13
Figura 20 Botones para comandar comunicación Bluetooth.....	14
Figura 21 Campos de visualización para coeficiente de atenuación.....	14
Figura 22 Editable de etiquetas.....	14
Figura 23 Datos exportados a Excel.	15
Figura 24 Vista general del HMI	15
Figura 25 Aproximación de área bajo la curva [11].	16
Figura 26 Vista general de curva con el área calculada	17
Figura 27 Comparación curva con el área calculada.....	17
Figura 28 SpectraSuite	18
Figura 29 Validación de color rojo.....	18
Figura 30 Validación de color azul.....	19
Figura 31 Validación de luz cálida	19
Figura 32 Validación luz fría	20
Figura 33 Espectro solar para tiempo de integración de 100 y 200 ms.....	21
Figura 34 Espectro solar para tiempo de integración de 300, 400, 500, 600, 700 y 800 ms.....	21
Figura 35 Curva de referencia y filtrada para el techo.....	22
Figura 36 Curva de referencia y filtrada para otro punto del techo.	23
Figura 37 Curva de referencia y filtrada para el medio.....	23
Figura 38 Muestras obtenidas en invernadero de plantas carnívoras.....	24

Figura 39 Muestras obtenidas en invernadero de orquídeas cálidas.	25
Figura 40 Muestras obtenidas en invernadero de orquídeas frías.	25
Figura 41 Prueba de filtro UV.	26
Figura 42 Elementos de control para PWM en Arduino.	26
Figura 43 PWM generada para 10 y 30 % de relación de trabajo.	27
Figura 44 PWM generada para 50 y 80 % de relación de trabajo.	27
Figura 45 PWM generada para 90 y 100 % de relación de trabajo.	27

RESUMEN

Este documento detalla el diseño, desarrollo e implementación de una interfaz gráfica para adquirir datos desde un espectrómetro USB 4000-UV-VIS-ES con el objetivo de analizar la calidad de filtros UV de invernaderos agrícolas.

El equipo junto a la interfaz tiene la capacidad de adquirir espectros de referencia y luego espectros con el plástico para superponer las curvas y comparar la diferencia de estas dos; además, en el HMI se mostrarán parámetros relevantes como espesor del plástico, coeficiente de atenuación, entre otros.

Para el desarrollo de la interfaz, los datos adquiridos deben ser procesados, pasando por un filtro digital y los debidos cálculos matemáticos con el objetivo de ser una herramienta de adquisición, cálculo y procesamiento de datos.

En cuanto a resultados. se obtuvieron datos de campo tomados en los invernaderos que tiene La Universidad de Investigación de Tecnología Experimental Yachay Tech, y en invernaderos del Jardín Botánico de Quito. Además, se realizaron pruebas con focos de colores, un led UV junto a filtros de determinada longitud de onda en donde el equipo funcionó correctamente y los datos se observaron en el HMI, así como se exportaron en un archivo de formato Excel para mayor accesibilidad.

PALABRAS CLAVE: espectrómetro, HMI, App Designer, comunicación, Procesamiento Digital de Señales.

ABSTRACT

This document details the design, development and implementation of a graphical interface to acquire data from a USB 4000-UV-VIS-ES spectrometer with the goal to analyze the quality of UV filters in agricultural greenhouses.

The equipment with the humane machine interface-HMI has the ability to acquire a first reference spectra and the spectra with the plastic in order to superimpose the curves and thus to compare the difference of both; in addition, the HMI can show relevant parameters such as plastic thickness, attenuation coefficient, among others.

For the development of the interface, the acquired data must be processed, passing through a digital filter and the appropriate mathematical calculations with the aim of a data acquisition, calculation and processing tool.

As for results, field data taken in the greenhouses of “La Universidad de Investigación de Tecnología Experimental Yachay Tech” were obtained, as well as in greenhouses of the Jardín Botánico in Quito. In addition, tests with colored spotlights were carried out, and also with a UV LED along with filters of a certain wavelength where the equipment worked correctly and the data was observed in the HMI, as well as exported in an Excel format file for greater accessibility.

KEYWORDS: spectrometer, HMI, App Designer, communication, Digital Signal Processing.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el presente proyecto se diseñará, desarrollará e implementará un prototipo para uso en la determinación del nivel de radiación ultravioleta que pasa por las cubiertas plásticas dentro de un invernadero. El prototipo se lo podrá utilizar tanto en laboratorio como y procesará la información recopilada para la visualización de la incidencia, respectivamente refracción UV debido al material de los techos de los invernaderos.

1.1 Objetivo general

Diseñar y desarrollar una interfaz gráfica que permita observar espectros de luz obtenidos desde un espectrómetro, el cual permite tomar diferentes muestras y comparar curvas de referencia junto a curvas filtradas por plásticos, además de procesar las señales obtenidas para posteriores cálculos.

1.2 Objetivos específicos

1. Realizar una recopilación bibliográfica sobre desarrollo de modelos, análisis, transmisión, procesamiento e interpretación de la información de sistemas de medición de radiación UV en especial en invernaderos agrícolas.
2. Diseñar e implementar un sistema para la captación y transmisión de datos desde el prototipo de sensado hacia el sistema de procesamiento de la información.
3. Realizar una interfaz gráfica para observar parámetros medidos por el espectrómetro y el estado energético de batería.
4. Desarrollar un HMI-interfaz hombre máquina que permita visualizar los resultados del procesamiento de datos.
5. Desarrollar un algoritmo para el procesamiento análisis e interpretación de los datos obtenidos en el prototipo UV.
6. Validar los resultados obtenidos con el HMI desarrollado comparando con datos obtenidos del software proporcionado por el fabricante del espectrómetro.

1.3 Alcance

- Se realizará un estudio y análisis de protocolos de comunicación para enlazar los datos entre el espectrómetro, microcontrolador y HMI.

- Se procesarán los datos obtenidos por el espectrómetro con el objetivo de mostrarlos en una interfaz gráfica.
- Se diseñará una interfaz gráfica para poder visualizar las curvas obtenidas, además se podrá cambiar el tiempo de integración del espectrómetro para la adquisición de espectros.
- Se realizará una validación de los resultados obtenidos, comparando las curvas obtenidas en el HMI desarrollado con las curvas obtenidas por el software del fabricante del espectrómetro.

1.4 Marco teórico

En este capítulo se detallarán partes esenciales del trabajo, ampliando teóricamente sus definiciones, características técnicas y ecuaciones matemáticas según el caso. Aquí se va a desarrollar además cualquier definición que se requiera revisar para entender los siguientes capítulos. 1.4.1 Espectrómetro de fibra óptica Ocean Optics USB-4000

El espectrómetro Ocean Optics USB-4000 (Figura 1) representa un sistema de fibra óptica de alto rendimiento en miniatura. Este espectrómetro, diseñado a partir del modelo USB-2000 popular, incorpora un detector avanzado y una electrónica de alta velocidad. Sus nuevas características incluyen un detector de 3648 elementos con obturador, electrónica de alta velocidad y capacidades de interfaz compatibles con los sistemas operativos Linux, Macintosh y Windows como indica la hoja de datos del fabricante [1]. El USB4000 abarca una respuesta espectral entre 200 y 1100 nm, aunque el rango y la resolución específicos dependen de las opciones de rejilla y hendidura de entrada seleccionadas.

La conexión del espectrómetro USB4000 a una computadora se realiza a través del puerto USB o puerto serie. Al conectarse mediante un USB 2.0 o 1.1, el espectrómetro se alimenta directamente desde la computadora anfitriona, eliminando la necesidad de una fuente de alimentación externa [1].



Figura 1 Espectrómetro USB 4000-UV-VIS-ES

Las características más relevantes del equipo son [1]:

- Sensibilidad: 130 fotones/cuenta a 400 nm
- Tiempos de integración: 10 μ s a 65 s
- Velocidad de transmisión de datos: 480 Mbps
- Consumo de potencia: 240 mA a 5 Vdc
- Convertidor A/D: 16 Bits, 3 MHz
- Rango de detección: 200 – 1100 nm
- Longitud focal: 42 mm
- Interfaz: USB 2.0, 115.2 K baudios

1.4.2 Secciones transversales para absorción y dispersión de fotones

Se considera que en la Figura 2 un haz paralelo de fotones atraviesa una placa. Los fotones del haz interactúan con los átomos de la placa en cuatro procesos diferentes: producción de pares, fotoeléctrico, Compton y Thomson. En todos estos procesos existe una eliminación de fotones [2].

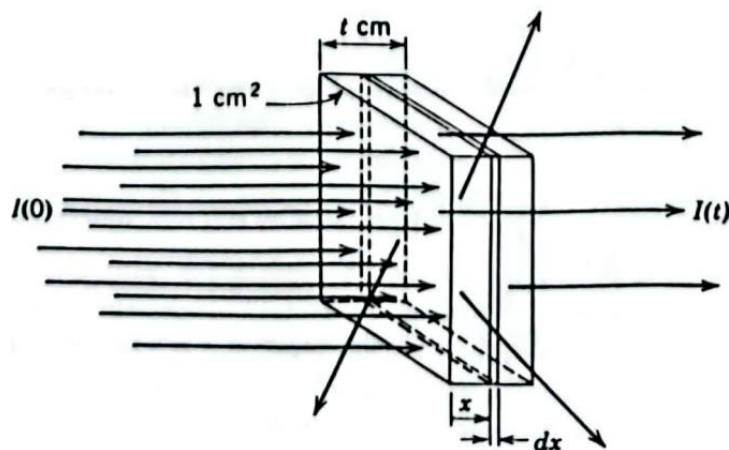


Figura 2 Haz de luz ingresando a una placa [2].

Cuando el haz de luz incide sobre la placa de espesor “t”, ingresa un número de fotones “ $I(0)$ ” y la sección recortada tiene ρ átomos por cm^3 . Debido a los procesos de absorción y dispersión, el haz paralelo a la salida tendrá $I(t)$ fotones después de haber atravesado en

la sección cortada “x” cm. Considerando una lámina muy delgada de espesor diferencial dx y situada en x . El número de átomos por cm^2 es ρ veces su volumen, es decir: ρdx . El número de fotones absorbidos o dispersados en la lámina está dado por $\sigma I(x)\rho dx$. Por tanto, los fotones que salen de la lámina $I(x+dx)$, son igual al número que entró menos a los que se perdieron [13]:

$$I(x + dx) = I(x) - \sigma I(x)\rho dx$$

Ecuación 1

$$dI(x) = I(x + dx) - I(x) = -\sigma I(x)\rho dx$$

Ecuación 2

Para hallar el número $I(t)$ de fotones se debe integrar:

$$\frac{dI(x)}{I(x)} = -\sigma\rho dx$$

Ecuación 3

$$\int_0^t \frac{dI(x)}{I(x)} = -\sigma\rho \int_0^t dx$$

Ecuación 4

$$I(t) = I(0)e^{-\sigma\rho t}$$

Ecuación 5 Relación de fotones de entrada y salida

Según la Ecuación 2 el número de fotones a la salida $I(t)$ decrece exponencialmente conforme el espesor de la placa “t” aumenta. El coeficiente de atenuación $\sigma\rho$ tiene dimensiones (cm^{-1}), despejando:

$$\sigma\rho = \frac{1}{t} \cdot \ln\left(\frac{I(0)}{I(t)}\right)$$

Ecuación 6 Coeficiente de atenuación

Este coeficiente va a determinar que tanta cantidad de fotones se perdió al impactar con el filtro a utilizar.

1.4.3 Suavizado SavitzkiGolay

Los procesos de medición se someten a variaciones atípicas en la señal; este fenómeno se conoce como ruido. Para poder suavizar o reducir este efecto se realiza un proceso conocido como filtrado [3]. El suavizado SavitzkyGolay está basado en la aproximación del

polinomio de mínimos cuadrados por una ventana móvil dentro del conjunto de datos a lo largo del dominio del tiempo [4]. Este filtrado permite mantener las características de los datos como ancho y altura del pico, que usualmente se ven atenuados por un filtro de promedio móvil. Este filtro para una forma de onda se puede calcular mediante la ecuación Ecuación 6:

$$g_m := \sum_{k=-n_L}^{n_R} c_{k+n_L} \cdot s_{m+k}$$

Ecuación 6

En donde s_{m+k} es la señal para filtrar y g_m es la salida filtrada. n_L es el número de muestras utilizadas a la izquierda de un dato m , mientras que n_R es el número utilizado a la derecha. Los coeficientes de SavitzkyGolay se calculan con la Ecuación 7:

$$c_{M+n_L} := [(A^T \cdot A)^{-1} \cdot (A^T)]_{0, M+n_L}$$

Ecuación 7

En donde:

$$M := -n_L \cdot n_R$$

Ecuación 8

$$m := n_L \cdot N - n_R$$

Ecuación 9

$$j := 0..n$$

Ecuación 10

$$A_{M+n_L, j} := M^j$$

Ecuación 11

1.4.4 App Designer de Matlab

App Designer es una extensión de Matlab destinada a la creación de aplicaciones, la cual permite arrastrar los elementos que se requieran dentro de una pantalla para posicionarlos de la manera en la que el usuario requiera como se observa en la Figura 3 Interfaz de desarrollo en App Designer. Además, el programa permite programar el comportamiento de cada uno de estos elementos dentro de la interfaz. Permite ejecutar apps comprimiéndolas en los archivos del instalador directamente creando una aplicación de escritorio [5].

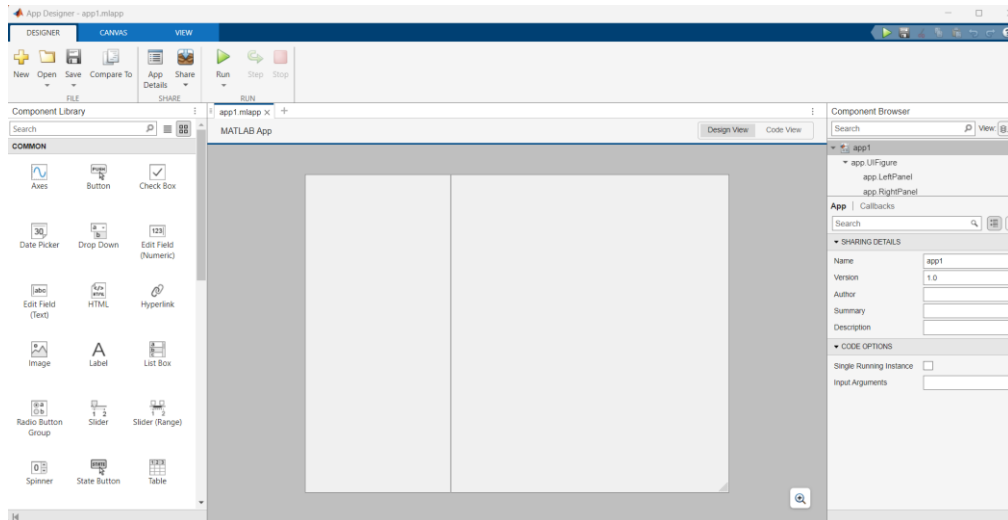


Figura 3 Interfaz de desarrollo en App Desginer

1.4.5 Módulo HC-05

Este módulo Bluetooth (Figura 4 Módulo Bluetooth HC-05) se maneja con el protocolo de puerto serie (SPP), diseñado para comunicar dispositivos seriales de manera inalámbrica. Este módulo maneja velocidad de datos mejorada de 3Mbps y un transceptor de radio completo (2.4GHz) [6].

Características:

- Voltaje de operación: 3.6 V a 5 V
- Interfaz UART con velocidades de 9600, 19200, 38400, 57600, 115200, 230400, 460800.
- Código PIN: 0000
- Sensibilidad: -80 dBm



Figura 4 Módulo Bluetooth HC-05

2 METODOLOGÍA

2.1 Diseño general

Para el diseño del presente proyecto, se muestra en la Figura 5 Diseño general del proceso un esquema general de la metodología a implementar en esta aplicación. Para la comunicación Bluetooth se requiere tener encendido el servicio dentro del ordenador, además de conectarlo con la ID asociada al dispositivo, después se tiene que abrir o cerrar el puerto serial para permitir o restringir la comunicación. Para la adquisición de datos con el espectrómetro se debe conectar a la computadora a través de un cable USB. Para que puedan intercambiar datos se requiere que se instalen los omnidrivvers del fabricante, posteriormente se hará definición de parámetros y finalmente la adquisición. Con los datos guardados entrarán al procesamiento (filtros, gráficos y cálculos) para finalmente ser exportados en formato Excel.

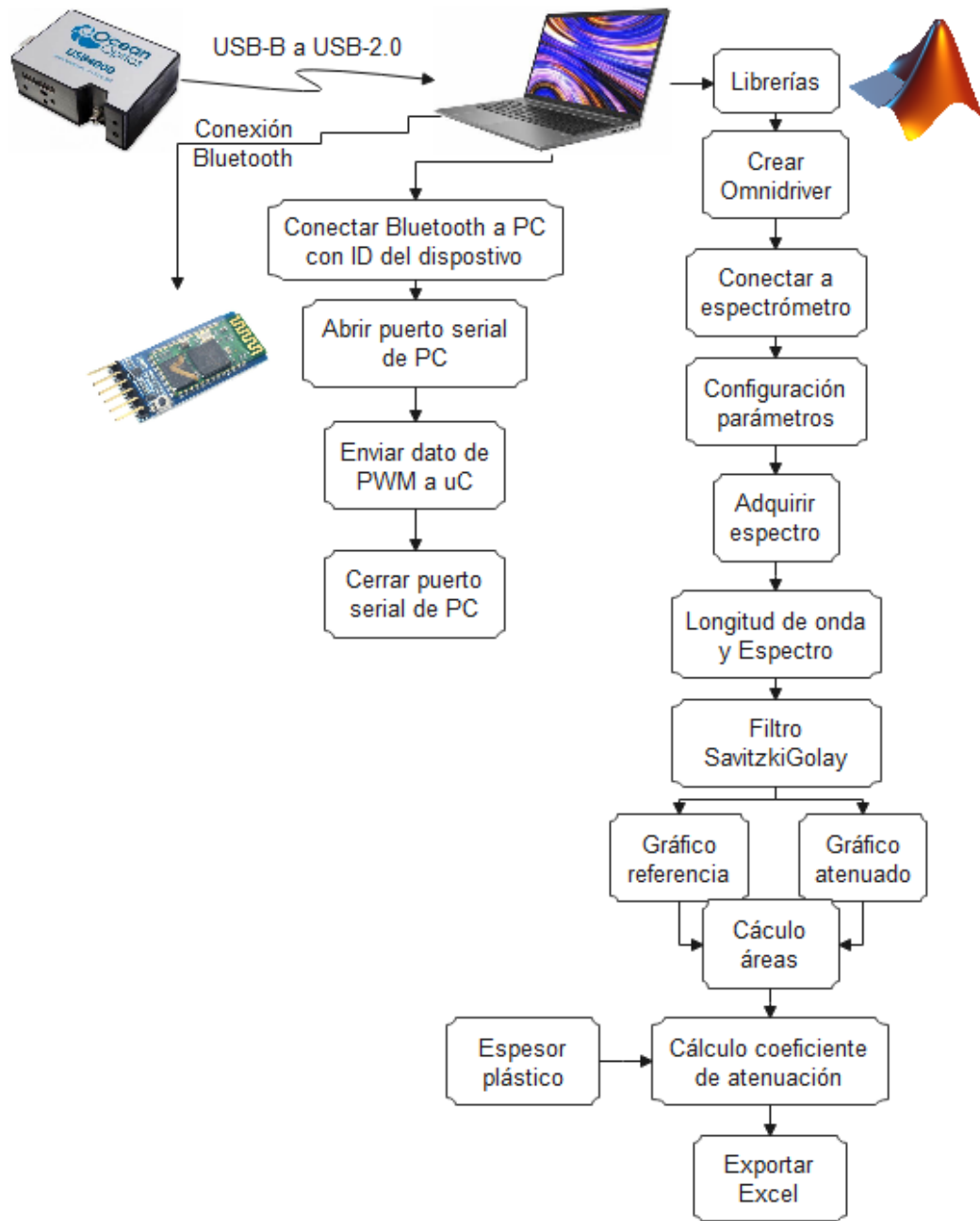


Figura 5 Diseño general del proceso

2.2 Conexión entre Matlab y módulo Bluetooth

Para la comunicación entre el módulo Bluetooth y Matlab se utilizaron comandos propios de la librería de Matlab.

Para ello, el Bluetooth del ordenador debe estar encendido y se deben ingresar los siguientes comandos para detectar el módulo.

El comando *instrhwinfo* que se muestra en la Figura 6 Resultado de comando *instrhwinfo* Figura 6 permite acceder a la información del dispositivo Bluetooth en donde está instalado Matlab.

```
>> info=instrhwinfo('Bluetooth')
Warning: instrhwinfo('Bluetooth') will be removed in a future release. Use bluetoothlist instead.

info =

HardwareInfo with properties:

    RemoteNames: []
    RemoteIDs: []
    BluecoveVersion: 'BlueCove-2.1.1-SNAPSHOT'
    JarFileVersion: 'Version 23.2'

Access to your hardware may be provided by a support package. Go to the Support Package Installer to learn more.
```

Figura 6 Resultado de comando *instrhwinfo*

Al ingresar el comando *info.RemoteNames* se observarán todos los nombres de los dispositivos Bluetooth cercanos al ordenador como se muestra en la Figura 7.

```
ans =

7x1 cell array

    {'MIBT'          }
    {'Echo Dot-WST'  }
    {'Bose Color SoundLink'}
    {'Galaxy Tab A8'  }
    {'HC-05'         }
    {'BM6'           }
    {'SC-HC29'       }
```

Figura 7 Nombres de dispositivos Bluetooth disponibles

Al ingresar el comando *info.RemoteIDs* se observarán todas las IDs Bluetooth de los dispositivos cercanos al ordenador como se muestra en la Figura 8.

```
ans =

7x1 cell array

    {'btspp://002300006309'}
    {'btspp://90F82EE099F9'}
    {'btspp://000C8AFB6C36'}
    {'btspp://6C55635D8C13'}
    {'btspp://98D30296C567'}
    {'btspp://325408E1F125'}
    {'btspp://000B9735893D'}
```

Figura 8 IDs de dispositivos Bluetooth disponibles

Como las IDs aparecen en el mismo orden que los nombres, se establece que para *HC-05* corresponde la dirección *btspp://98D30296C56*.

Por lo que para iniciar la conexión entre Matlab y el módulo HC-05 se debe ejecutar la línea de código de la Figura 9, en donde se establece que un objeto Bluetooth llamado *comu*.

```
comu = Bluetooth('btspp://98D30296C567',1);
```

Figura 9 Creación de dispositivo Bluetooth.

Luego ese objeto creado debe abrir el puerto serial para que pueda iniciar la comunicación con el comando *fopen* de la Figura 10.

```
>> fopen(comu)
```

Figura 10 Abrir puerto serial

Para escribir en el puerto serial se utiliza el comando de la en donde se envía un caracter para establecer órdenes hacia el microcontrolador.

```
>> fwrite(comu,'p');
```

Figura 11 Escribir en puerto serial

Cuando se termina de escribir el dato deseado se debe cerrar el puerto de comunicación y además eliminar el objeto Bluetooth creado mediante el comando de la línea

```
>> fclose(app.comu);  
delete(app.comu);
```

Figura 12 Cerrar puerto y eliminar dispositivos

2.3 Comunicación entre espectrómetro USB – 4000 y App Designer (Matlab)

Los espectrómetros de Ocean Optics tienen soporte en Matlab para plataformas Windows y Linux. Instrument Control Toolbox es compatible con los espectrómetros, permitiendo a Matlab tener un control integral de estos instrumentos, pudiendo realizar:

- Adquisición de espectros
- Configurar tiempo de integración
- Habilitar correcciones espectrales de corriente oscura y no lineales
- Ver dispositivos conectados

Para utilizar el espectrómetro en Matlab se tiene que instalar el [OmniDriver](#), el cual, cumple el papel de driver para crear la compatibilidad entre el espectrómetro y el equipo. Una vez instalado, Matlab ofrece un ejemplo para adquirir el espectro, el cual, se muestra en el

Anexo I. Este ejemplo permite extraer los vectores de dimensiones 3648 x 1 como se observa en la Figura 13 de la, los cuales son el espectro y la longitud de onda.



Figura 13 Vectores adquiridos y almacenados en el Work Space en Matlab.

En la Figura 14 se observa el espectro obtenido desde un Live Script de Matlab, para una luz de linterna blanca.

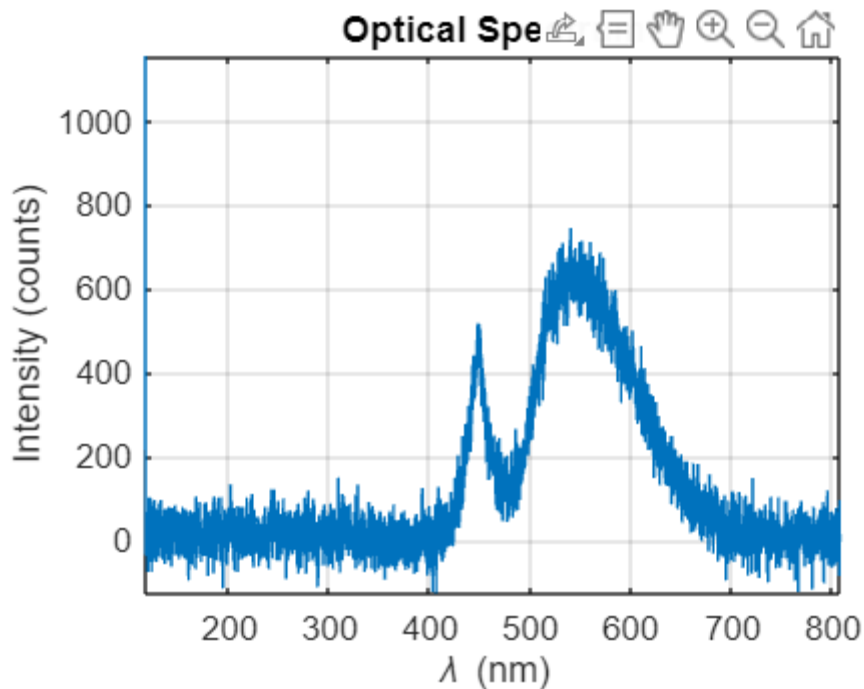


Figura 14 Curva obtenida con el ejemplo de Matlab

1

La Figura 14 muestra el espectro obtenido con la linterna blanca de un celular. En esta gráfica se aprecia que tiene ruido, por lo que para el procesamiento de datos se va a aplicar un filtro a los vectores obtenidos para dicha gráfica, estos se aplican mediante el comando: `sgolayfilt`, el cual se encarga de aplicar un filtro de suavizado FIR de Savitzky-Golay de orden polinomial [7].

En donde a sintaxis del filtro está dada por:

`y = sgolayfilt(x,order,framelen)`

- `x`: es el vector para filtrar
- `order`: indica el orden del filtro

- framelen: longitud del cuadro

En la interfaz realizada se tiene como fin comparar dos espectros, antes del filtro y después por lo que se graficarán las dos curvas sobrepuestas de la siguiente manera: Luego, filtrando cada vector se tiene una curva mejorada como se muestra en la Figura 15:

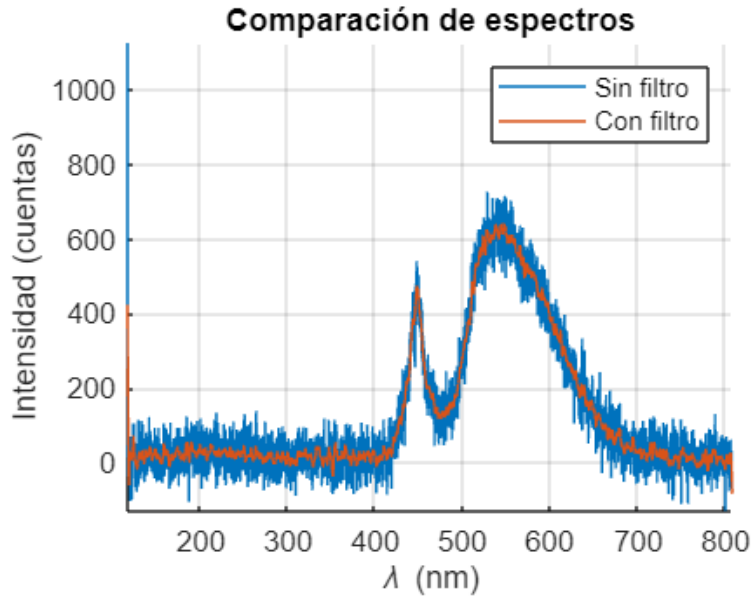


Figura 15 Resultados de las pruebas realizadas del filtro digital

Al hacer una ampliación en la gráfica se puede observar en la Figura 16 que es evidente la acción del filtro de ruido para las señales obtenidas. La curva varía mucho menos que la anterior eliminando gran cantidad de ruido, por lo que la medición es más limpia para poder hacer análisis posteriores, cumpliendo así el objetivo de colocar un filtro a la señal medida por el espectrómetro.

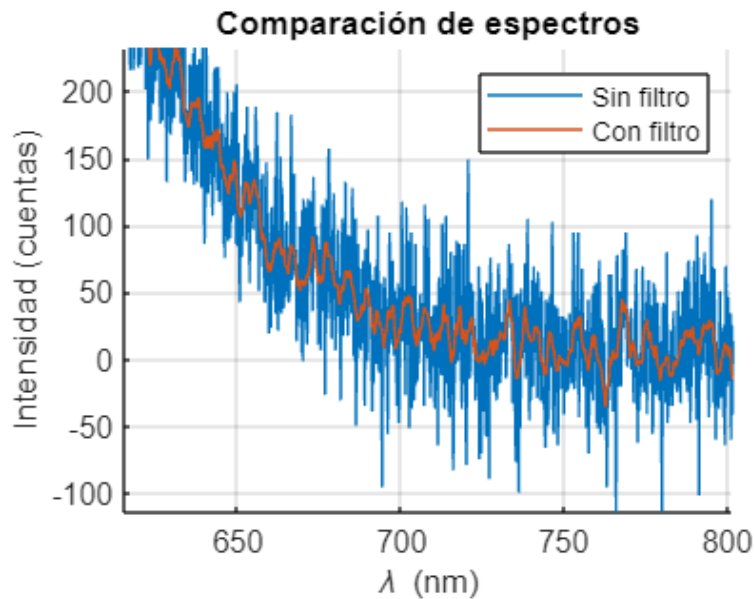


Figura 16 Acercamiento a señal filtrada

2.4 Desarrollo de Interfaz para adquisición de espectros.

Para el diseño de la interfaz se debe tomar en cuenta que el diseño debe ser intuitivo para el usuario y a su vez debe mostrar los parámetros de interés tanto para control como para visualización.

El tiempo de integración es un parámetro que necesita ser ingresado al espectrómetro para que abra el lente y adquiera los datos durante ese tiempo configurado. Por lo que se agregará un campo de entrada numérico [8] y además un menú desplegable [9] para elegir las unidades que podrán ser microsegundos, milisegundos y segundos como se observa en la Figura 17

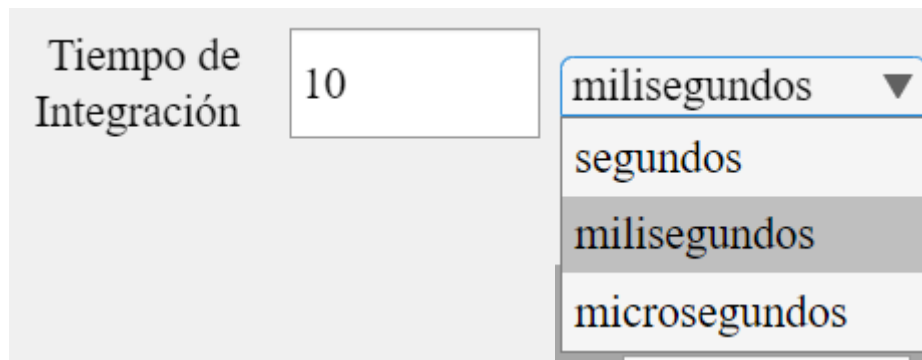


Figura 17 Campo de entrada para tiempo de integración

Se requieren ejes para graficar los espectros tomados, por lo que se añadieron tres UI AXES para diferentes muestras que se requieran tomar con etiqueta de longitud de onda 0..en el eje de abscisas y con la magnitud del espectro en el eje de las ordenadas. Además, deben mostrar la leyenda de cada curva como se observa en la Figura 18.

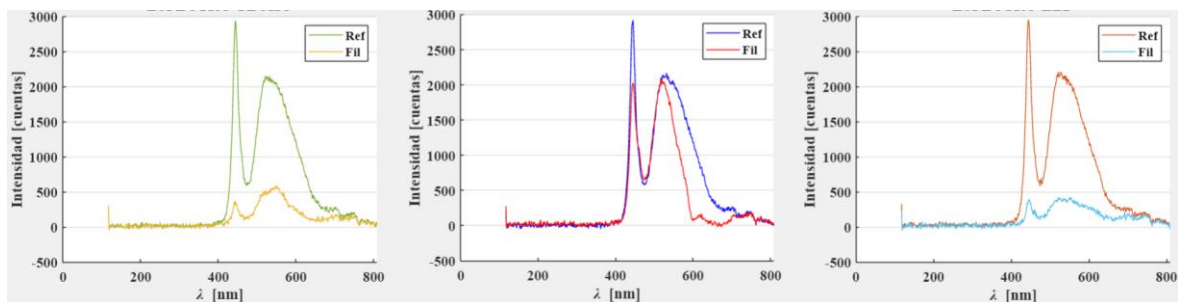


Figura 18 Ejes para graficar espectros.

Para adquirir o limpiar curvas se requiere de dos botones [10] como los que se muestran en la Figura 19 para cada gráfico en donde uno se va a encargar de adquirir los espectros y graficarlos en los ejes correspondientes, mientras que el otro botón de limpiar se encargará de borrar el espectro previamente adquirido para permitir al usuario volver a tomar muestras en caso de que por razones inesperadas se tomen incorrectamente.

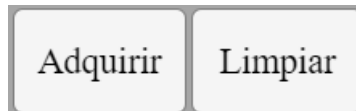


Figura 19 Botones de adquisición y limpieza de ejes

Para comandar la comunicación Bluetooth se necesitan 2 botones [10] para abrir y cerrar el puerto serial, además de un switch para encender o apagar un led UV, el cual será utilizado para pruebas en laboratorio, y de una perilla que va a variar el ancho de pulso de una PWM como se muestra en Figura 20.

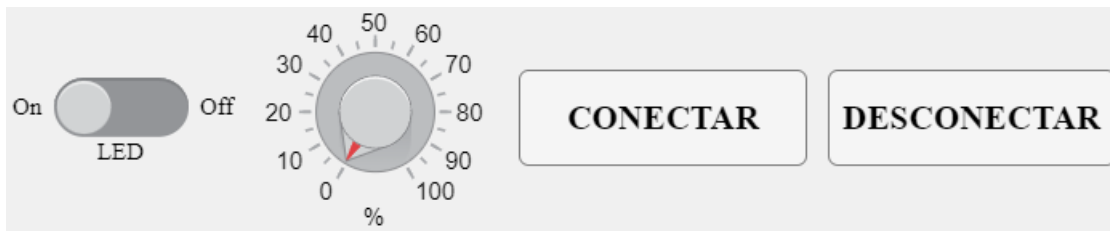


Figura 20 Botones para comandar comunicación Bluetooth

Para el cálculo del coeficiente de atenuación se requieren tres parámetros, estos son el espesor del plástico que está filtrando, mismo que va a utilizar un campo de entrada numérico y por defecto va a tener un valor de 1 [8]. Además, se requiere el área bajo las curvas para interpretar la cantidad de fotones que están ingresando al filtro y la cantidad que salen, por lo que se van a colocar dos indicadores numéricos para visualizar dichas áreas. Finalmente, otro indicador para observar el coeficiente de atenuación como en la Figura 21.

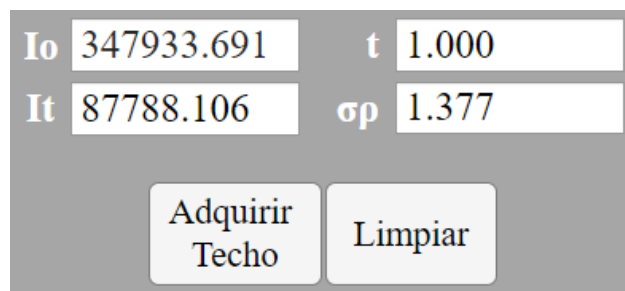


Figura 21 Campos de visualización para coeficiente de atenuación.

Se podrá etiquetar cada una de las muestras con un espacio de entrada de texto con el objetivo de que el usuario identifique a qué escenario pertenece la gráfica deseada como se observa en la Figura 22.

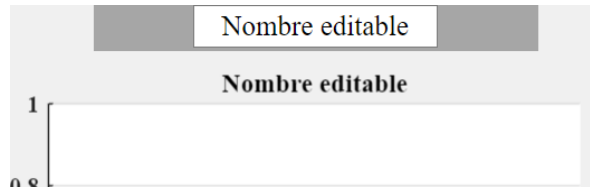


Figura 22 Editable de etiquetas

Se tendrá también un botón para exportar todos los datos hacia un archivo con formato Excel, con el objetivo de poder visualizar toda la misma información que en el HMI (Figura 23 Datos exportados a Excel., con el objetivo de poder tener registrados todos los parámetros de las pruebas y poder acceder a los datos desde cualquier equipo sin la necesidad de tener instalado Matlab, esto beneficia a los usuarios finales en poder emitir documentos como informes, casos de estudio, entre otros.

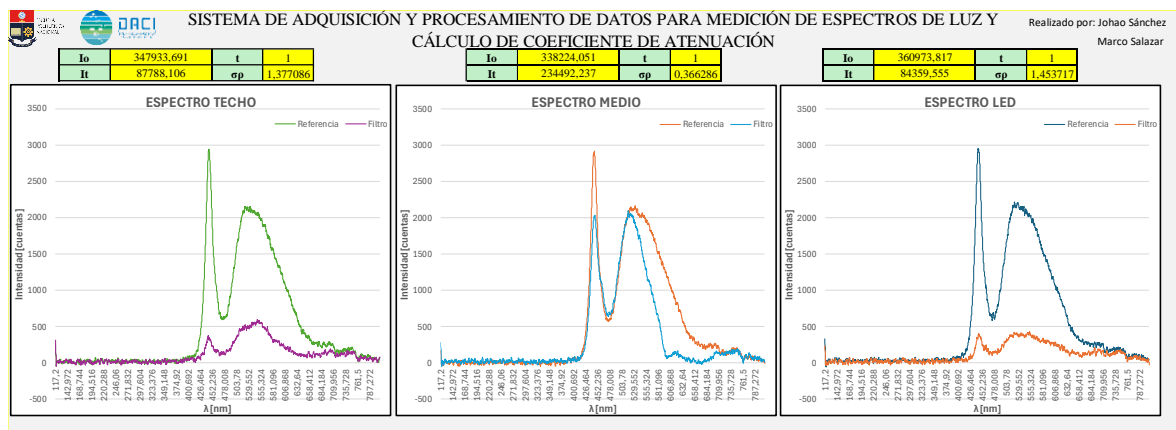


Figura 23 Datos exportados a Excel.

La vista general del HMI quedó como la Figura 24 recalcando el orden de la distribución de elementos de control y visualización además de su armonía de colores.

Legenda

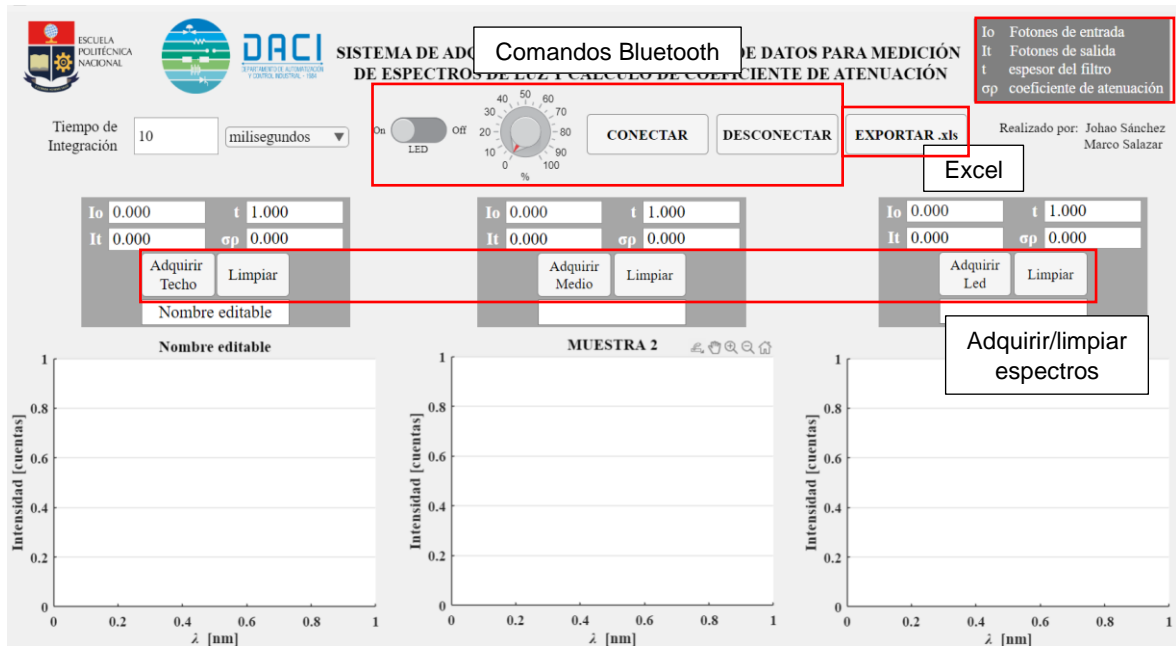


Figura 24 Vista general del HMI

2.5 Cálculo de coeficiente de atenuación.

Para el cálculo de este coeficiente se requiere utilizar la Ecuación . Para la obtención de los fotones de entrada y salida se debe realizar el cálculo del área bajo la curva, para ello se desarrolló un algoritmo en donde se calcula el área de rectángulos relativamente pequeños y se suman sus áreas al final aproximadamente como se observa en la Figura 25 Aproximación de área bajo la curva [11].

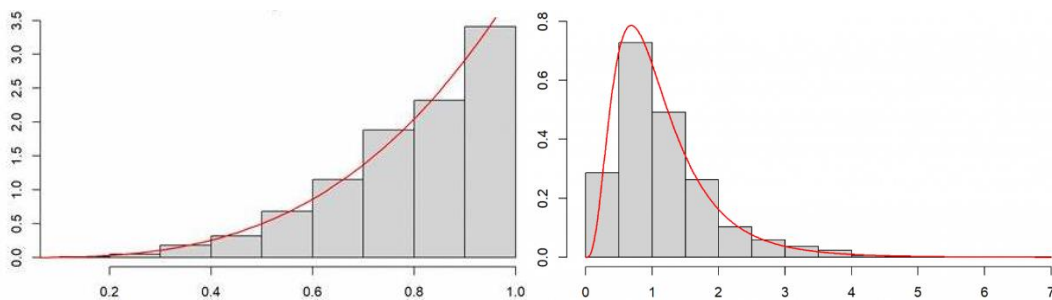


Figura 25 Aproximación de área bajo la curva [11].

El algoritmo se basa en obtener las dimensiones de cada pequeño rectángulo, tomando la diferencia de dos elementos consecutivos para la base y la diferencia dividida entre dos más el valor del eje de ordenadas para la altura. Posteriormente a esto se hace la multiplicación de base por altura y se suma con el área anterior. Al ser un ciclo for, se ejecutará el ciclo 3647 veces debido a que esa es la dimensión de los vectores obtenidos como se menciona en la Figura 13.

```

for i = 1:1:3647
    dx=(a(i+1,1)-a(i,1));
    dy=b(i,1)+(b(i+1,1)-b(i,1))/2;
    A = (dx*dy)+A;
    Vy(i) = dy;
end

```

Donde:

- i: contador del ciclo FOR
- a: es el vector de longitud de onda
- b: es el vector de espectro
- A: es el área bajo la curva

Se grafica el vector Vy con el comando “stem”, el cual sirve para gráficas discretas, obtenido del cálculo anterior para corroborar que el algoritmo es correcto “pintando” el área bajo las curvas, se muestra en la Figura 26 las imágenes superpuestas.

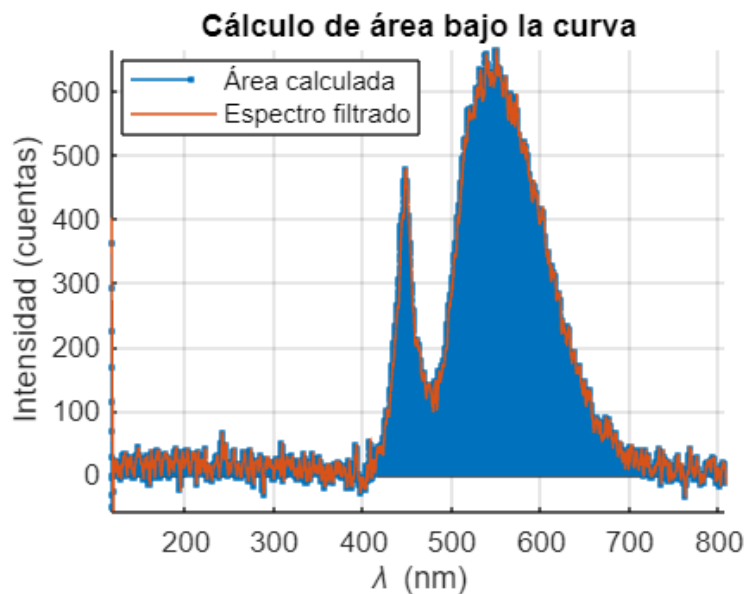


Figura 26 Vista general de curva con el área calculada

Luego, se realiza un acercamiento (Figura 27) para confirmar que el cálculo es correcto, como se puede observar, el área calculada es muy similar al espectro adquirido, validando así el cálculo.

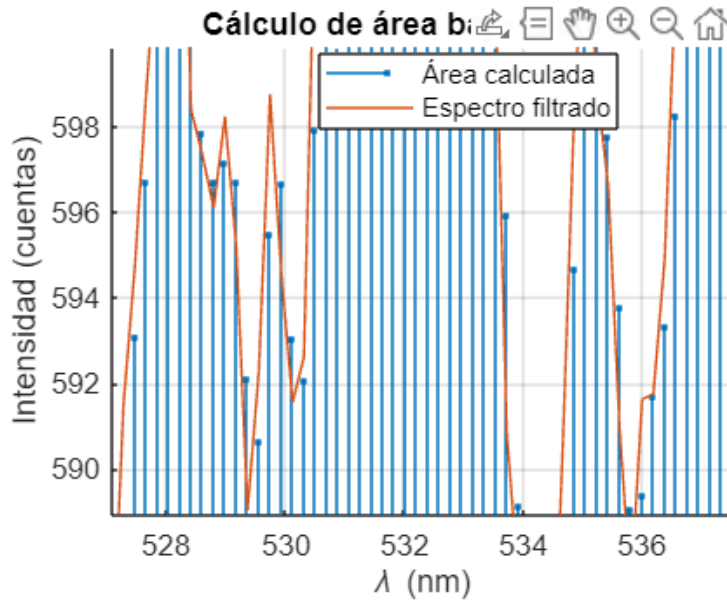


Figura 27 Comparación curva con el área calculada

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

3.1.1 Validación de datos

Para la validación de las curvas se utilizará el software SpectraSuite (Figura 28) del fabricante Ocean Optics para observar los espectros, sin embargo, cuenta con algunas limitaciones, por ejemplo, super posición de gráficas, cálculo de coeficientes, exportación de datos entre otros.



Figura 28 SpectraSuite

Se van a utilizar focos de colores (Anexo II) para recibir luz en determinado ancho de banda y para que los datos sean válidos del HMI diseñado tienen que estar las ondas en la misma longitud de onda.

Para el color rojo de la Figura 29 se puede observar que en ambos gráficos la apertura del pico del espectro se encuentra aproximadamente entre 592 y 650 [nm].

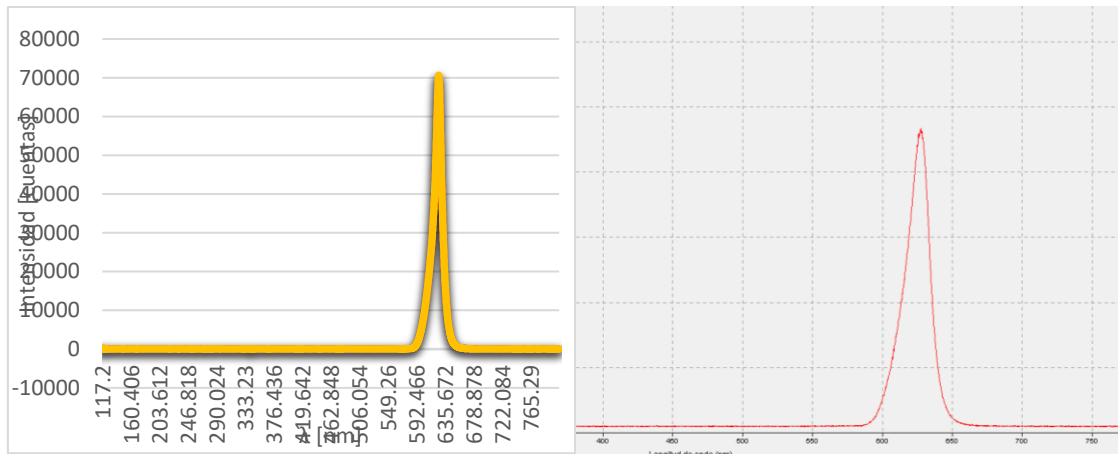


Figura 29 Validación de color rojo.

Para el color azul de la Figura 30 se puede observar que en ambos gráficos el pico del espectro se encuentra aproximadamente entre 425 y 500 [nm].

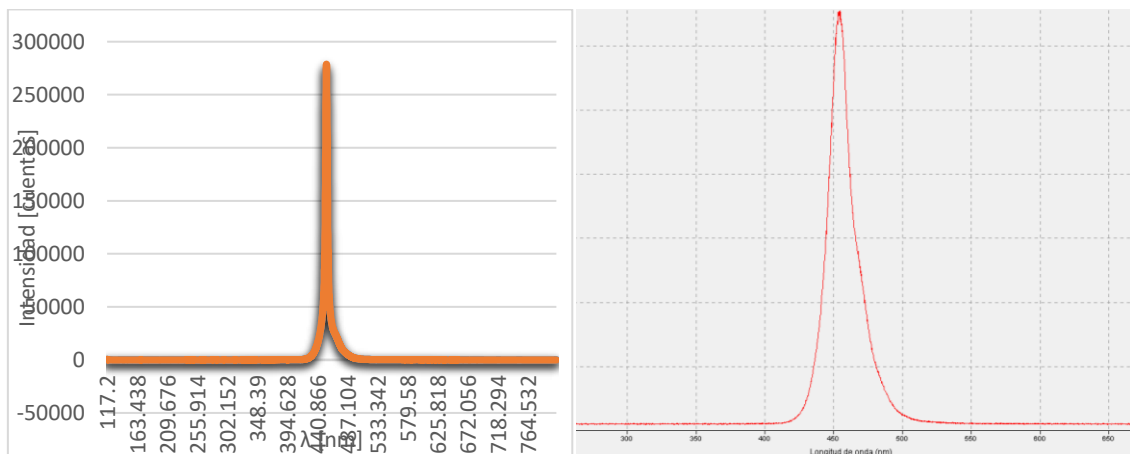


Figura 30 Validación de color azul

Para un foco de luz cálida Figura 31 se puede observar que en ambos gráficos el pico del espectro se encuentra aproximadamente entre 460 y 680 [nm]. Nótese que la forma de onda varía ligeramente, esto es debido a que se tomó diferente tiempo de integración para cada caso para hacer pruebas, por lo que el espectro también va a variar, sin embargo, están dentro de la similitud esperada tanto en ancho de banda como en forma de la curva.

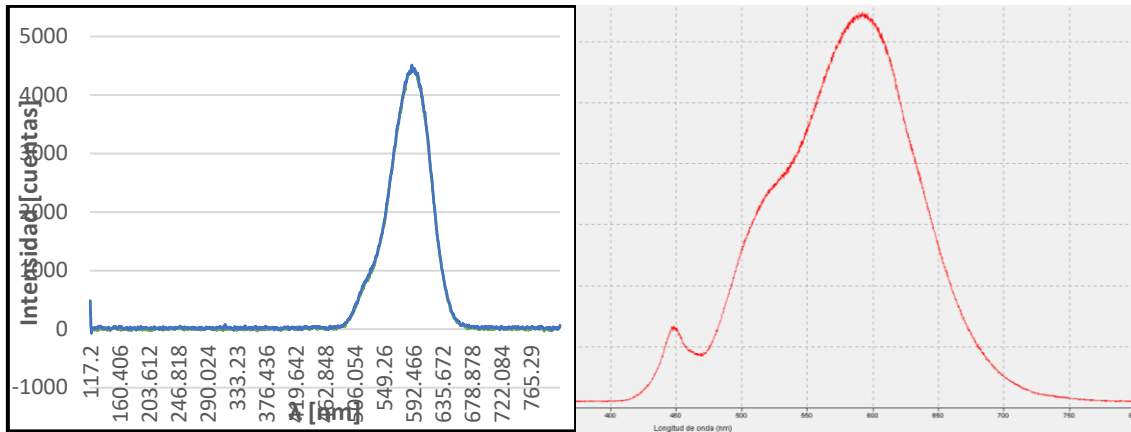


Figura 31 Validación de luz cálida

Para un foco de luz cálida Figura 32 se puede observar que en ambos gráficos el pico del espectro se encuentra aproximadamente entre 420 y 650 [nm]. Nótese que la forma de onda varía ligeramente, esto es debido a que se tomó diferente tiempo de integración para cada caso para hacer pruebas, por lo que el espectro también va a variar, sin embargo, están dentro de la similitud esperada tanto en ancho de banda como en forma de la curva.

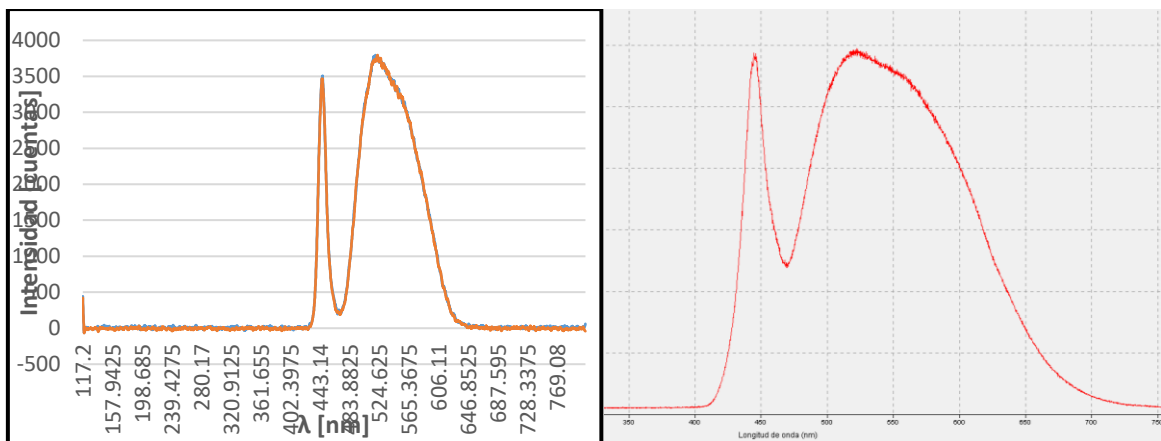


Figura 32 Validación luz fría

3.1.2 Adquisición de espectros

Para la obtención de resultados se realizaron pruebas de campo en La Universidad de Investigación de Tecnología Experimental Yachay Tech, ubicada en San Miguel de Urcuquí, Hacienda San José (Imbabura - Ecuador), en el invernadero que tiene dicha universidad (Anexo III).

Para ello se hizo la medición del espectro solar con distintos tiempos de integración como se observa en la Figura 33 y Figura 34, determinando que el tiempo adecuado de integración para la toma de datos es de 600 ms debido que, a partir de 600 ms, la curva del espectro solar no aumenta ni disminuye como con tiempos menores. Y eso se ve

reflejado en cómo las últimas muestras de la Figura 34 se superponen, además de que el área bajo la curva de cada una es numéricamente similar de $3553135.912 \approx 3511599.461$.

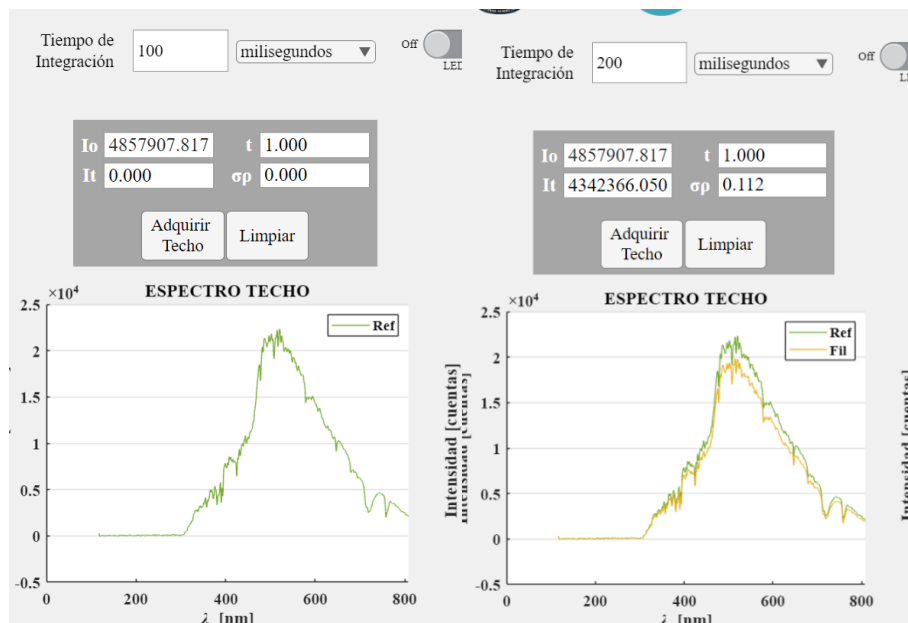


Figura 33 Espectro solar para tiempo de integración de 100 y 200 ms.

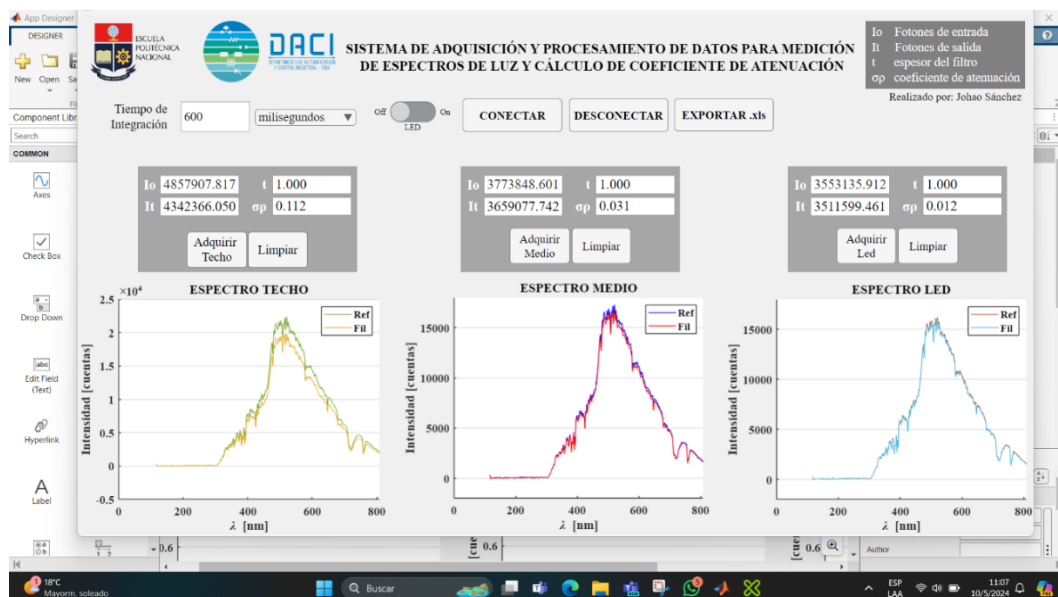


Figura 34 Espectro solar para tiempo de integración de 300, 400, 500, 600, 700 y 800 ms.

Una vez, encontrado el tiempo de integración adecuado para las mediciones, se muestra en la Figura 35 las curvas obtenidas en el techo, donde la línea anaranjada es la referencia del sol y la celeste es después del plástico del invernadero. Además, en la Tabla 1 se observan los valores obtenidos para el coeficiente de atenuación y la cantidad de fotones de entrada/salida.

Tabla 1 Resultados de pruebas realizados en Techo de invernadero

lo	2602199,499	t	1
lt	2424820,441	$\sigma\rho$	0,0706

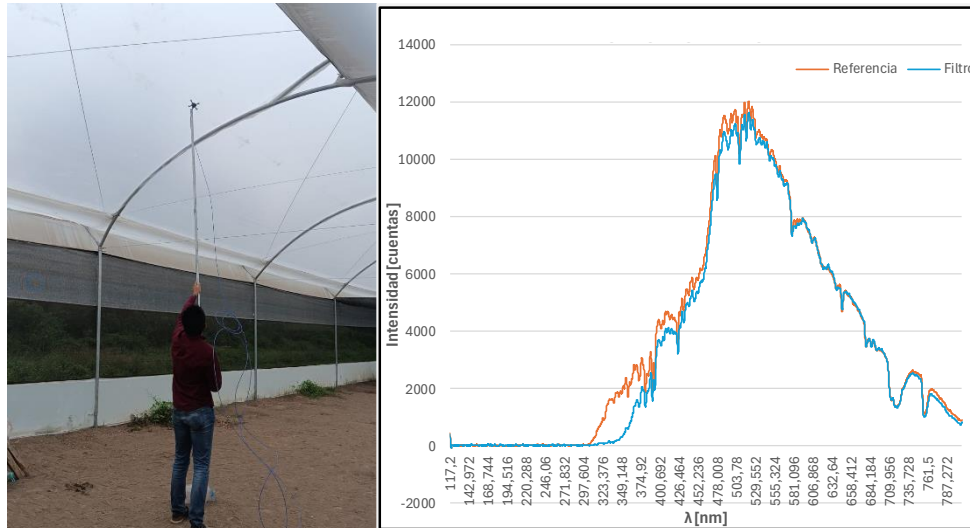


Figura 35 Curva de referencia y filtrada para el techo.

Se realizó la medición en otro punto del techo para poder observar que tanto varía el coeficiente de atenuación y la curva del filtro. Como se puede observar en la Figura 36, existe una sombra en el techo por lo que en consecuencia la curva atenuada es mucho más pequeña que la medición anterior. Por otro lado, el coeficiente de atenuación va a ser mayor como se observa en la Tabla 2.

Tabla 2 Resultados de las pruebas realizadas en otro punto del techo.

lo	1281341,167	t	1
lt	895615,547	$\sigma\rho$	0,358151

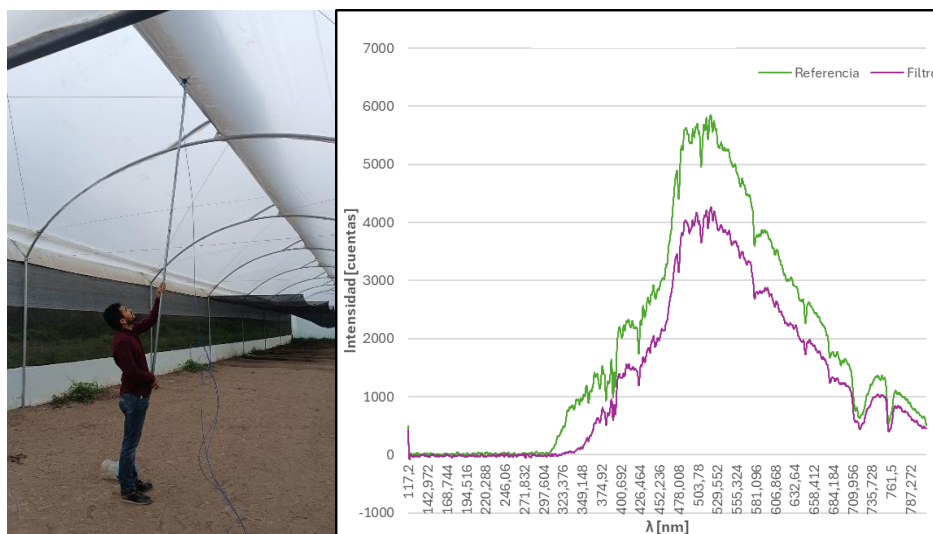


Figura 36 Curva de referencia y filtrada para otro punto del techo.

Se realizó otra medición en el medio cerca del suelo para poder observar que tanto variaría el coeficiente de atenuación y la curva del filtro. Como se puede observar en la Figura 37, existe mayor distancia del techo al piso por lo que en consecuencia la curva atenuada es mucho más pequeña que las mediciones anteriores. Por otro lado, el coeficiente de atenuación va a ser mucho mayor como se observa en la Tabla 3.

Tabla 3 Resultados de las pruebas realizadas en el medio.

Io	1288544,318	t	1
I_t	638382,945	σ_p	0,70233

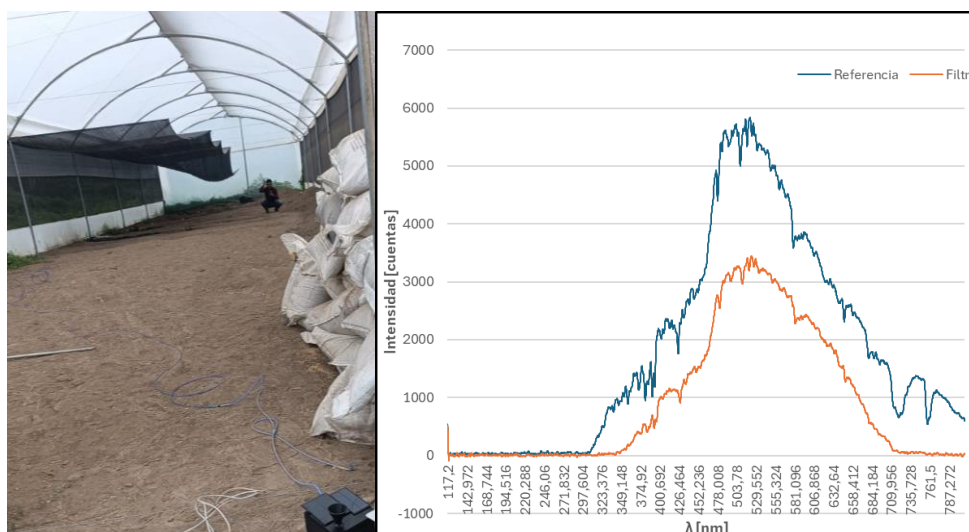


Figura 37 Curva de referencia y filtrada para el medio.

También realizaron pruebas de campo en el Jardín Botánico de Quito, ubicado en el interior del parque La Carolina, pasaje # 34, Rumipamba E6-264 Y, Quito 170135, (Quito - Ecuador), en tres invernaderos que tiene el jardín: plantas carnívoras, orquídeas frías y orquídeas cálidas (Anexo IV).

Para el invernadero de plantas carnívoras se tomaron tres muestras en diferentes puntos del invernadero, los datos mostrados en la Figura 38 en donde se evidencia que estas plantas requieren una atenuación de aproximadamente la mitad del sol que hay afuera del invernadero. Por lo que las curvas filtradas evidentemente van a ser menores en la referencia. De igual manera se muestran los coeficientes de atenuación obtenidos para cada muestra.

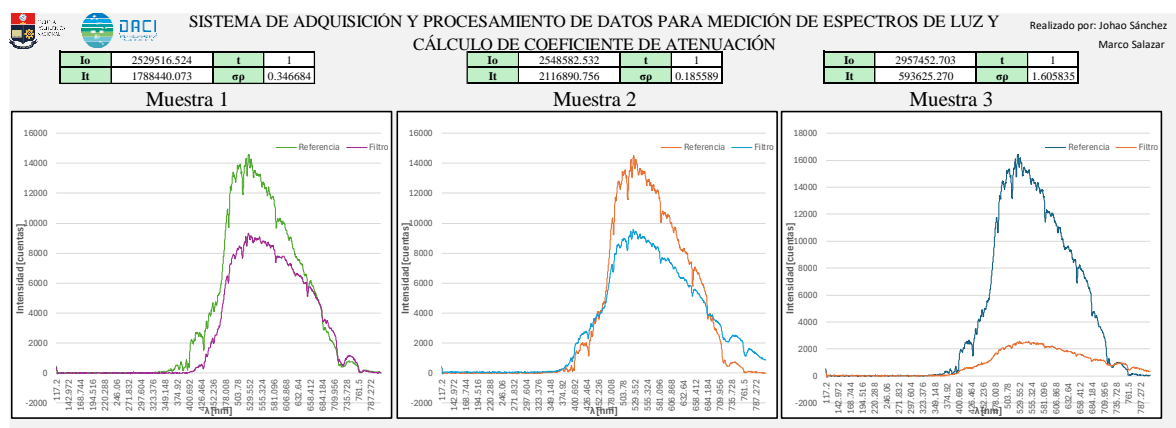


Figura 38 Muestras obtenidas en invernadero de plantas carnívoras.

Para el invernadero de orquídeas cálidas se tomaron tres muestras en diferentes puntos del invernadero, los datos mostrados en la Figura 39 en donde se evidencia que estas plantas requieren una atenuación baja del sol que hay afuera del invernadero. Por lo que las curvas filtradas van a ser similares con respecto a la referencia. Se observa también que en un ancho de longitud de onda el plástico del invernadero tiende a amplificar un rango del espectro para concentrar más luz dentro del invernadero. De igual manera se muestran los coeficientes de atenuación obtenidos para cada muestra.

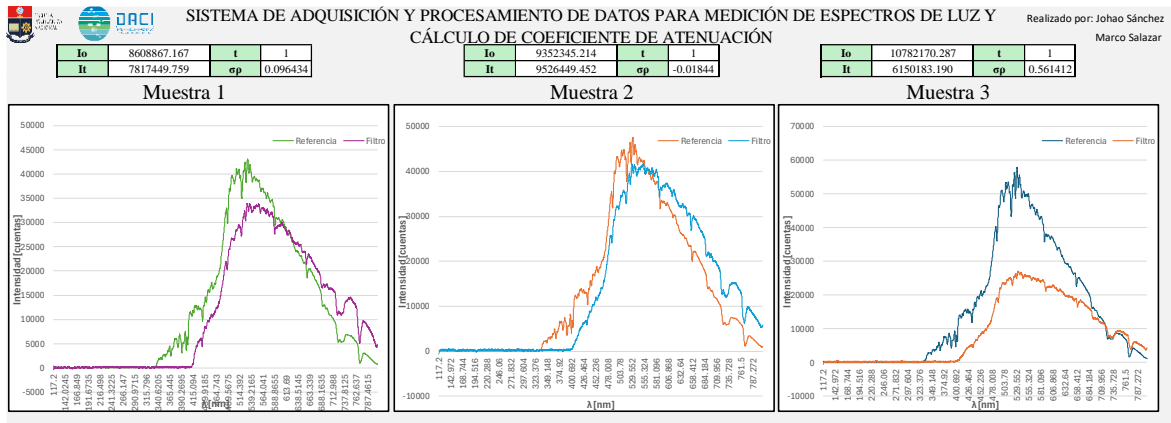


Figura 39 Muestras obtenidas en invernadero de orquídeas cálidas.

Para el invernadero de orquídeas frías se tomaron tres muestras en diferentes puntos del invernadero, los datos mostrados en la Figura 40 en donde se evidencia que estas plantas requieren una atenuación muy baja del sol que hay afuera del invernadero. Por lo que las curvas filtradas van a ser mucho más pequeñas con respecto a la referencia. Se observa también que en un ancho de longitud de onda el plástico del invernadero tiende a amplificar un rango del espectro para concentrar más luz dentro del invernadero, no obstante, reduce el espectro para tener una temperatura más fría dentro del invernadero. De igual manera se muestran los coeficientes de atenuación obtenidos para cada muestra.

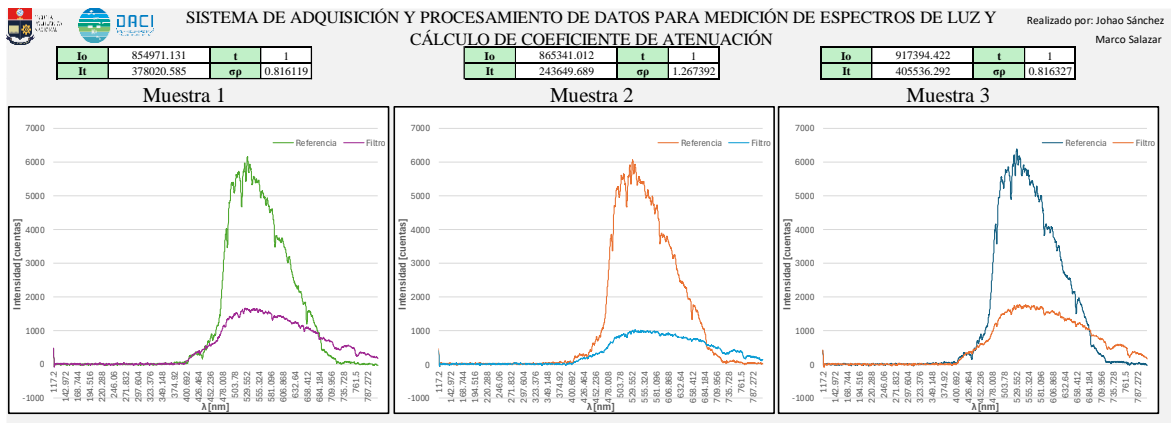


Figura 40 Muestras obtenidas en invernadero de orquídeas frías.

Se realizó una prueba adicional en laboratorio con un led UV para captar la curva de referencia y después se volvió a captar el espectro con un filtro de invernadero especial para filtrar la longitud de onda UV como se muestra en Figura 41. Se observa que la banda UV desaparece, pero el resto de luz sí pasa. Además, el coeficiente de atenuación de la Tabla 4 es alto.

Tabla 4 Resultados de las pruebas realizadas con LED UV.

I_o	263508.428	t	1
I_t	88073.523	σ_p	1.095913

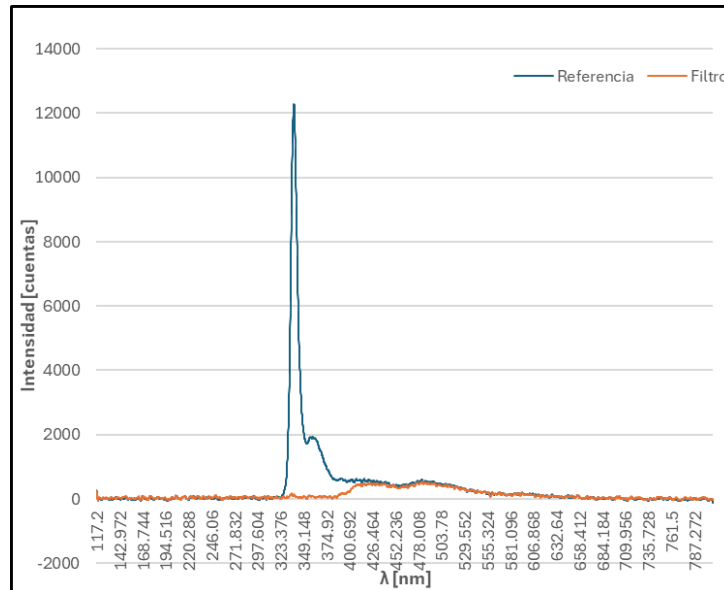


Figura 41 Prueba de filtro UV.

3.1.3 Comunicación entre App Designer y Arduino por Bluetooth

A continuación, se mostrarán los resultados en cuanto al control de la PWM por Bluetooth entre el HMI y el módulo HC-05.

Después de pulsar el botón de conectar, se modifica la perilla para variar el ancho de pulso de la PWM (Figura 42).

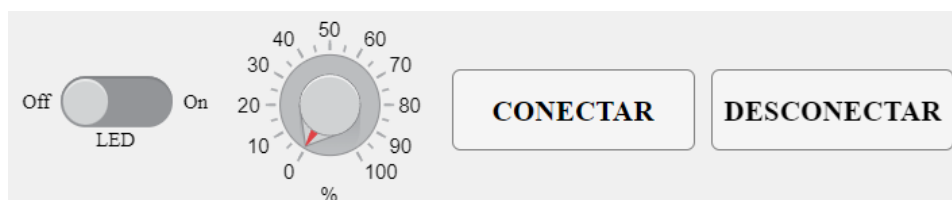


Figura 42 Elementos de control para PWM en Arduino.

Para ver cómo varía la PWM se midió la señal con un osciloscopio como se muestra en las Figura 43, Figura 44 y Figura 45 para distintos porcentajes de relación de trabajo. Variar la relación de trabajo de la PWM sirve para controlar la intensidad del LED UV para las pruebas de laboratorio.

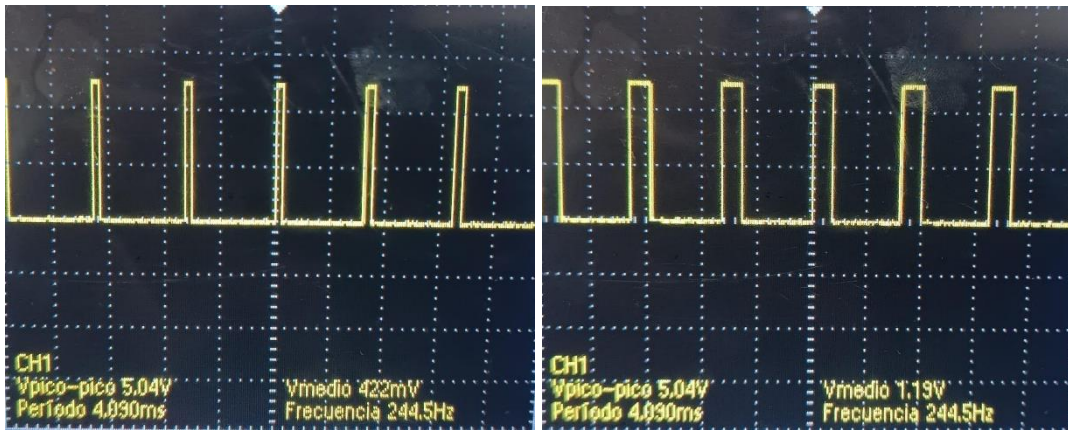


Figura 43 PWM generada para 10 y 30 % de relación de trabajo.

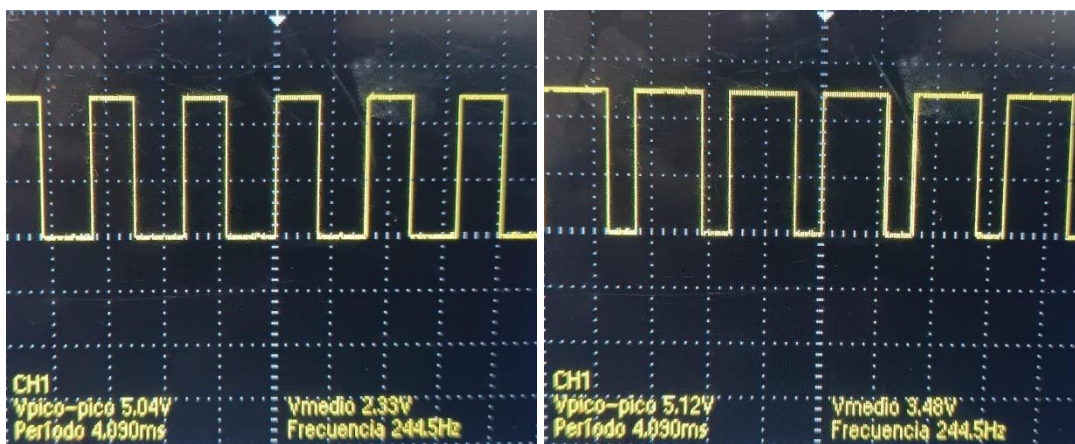


Figura 44 PWM generada para 50 y 80 % de relación de trabajo.

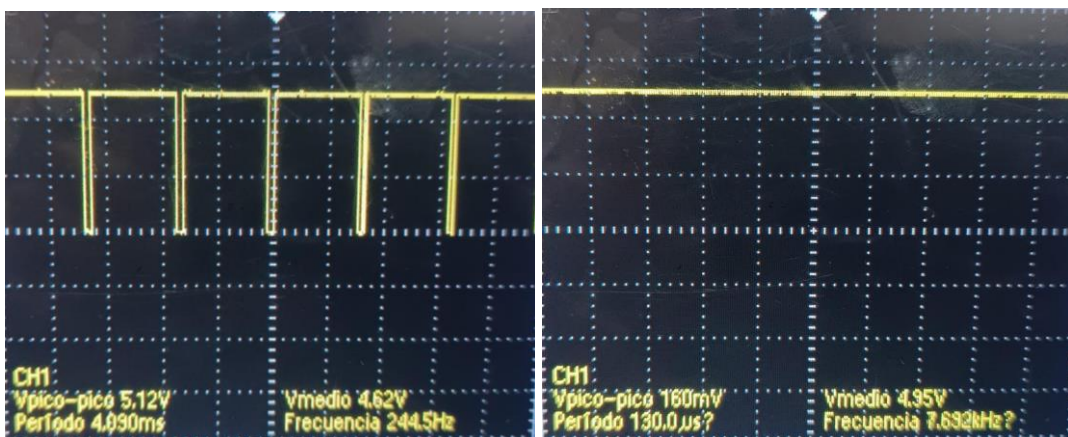


Figura 45 PWM generada para 90 y 100 % de relación de trabajo.

3.2 Conclusiones

El diseño de la interfaz utilizada cumplió con los requerimientos planteados, siendo una herramienta de control de cambios, visualización y procesamiento de datos. Además, la facilidad de uso permite al usuario utilizar el programa de forma sencilla e intuitiva.

La interfaz permite solventar distintas necesidades o aplicaciones que requieran expertos en el campo de la espectroscopía, agrícola, física, debido a que el espectrómetro tiene un amplio rango de medición y la misma interfaz permite obtener datos procesados, los cuales en la interfaz comercial no permite, es decir el presente trabajo personaliza los resultados según requerimientos del usuario.

Los datos obtenidos por la aplicación desarrollada fueron validados con el software proporcionado por el fabricante del espectrómetro, lo cual garantiza la confiabilidad de la aplicación desarrollada para hacer estudios, experimentos o control de calidad de plásticos para invernaderos.

3.3 Recomendaciones

La distribución de los elementos del HMI es importante que sea simétrica y correcta para obtener una interfaz agradable evitando saturación de elementos o que no sea intuitivo el uso de la misma.

El procesamiento de datos debe realizarse correctamente para evitar perder información cuando se aplique filtros a las señales, debido a que las lecturas pueden ser erróneas y en consecuencia los cálculos de coeficientes también.

Para mejoras en el programa y ampliar el alcance del proyecto se pueden añadir ventanas para realizar el cálculo de otros coeficientes relacionados con esta aplicación.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] O. Optics, «USB4000 Fiber Optic Spectrometer,» [En línea]. Available: <https://www.oceaninsight.com/globalassets/catalog-blocks-and-images/manuals--instruction-ocean-optics/spectrometer/usb4000operatinginstructions.pdf>. [Último acceso: 22 mayo 2024].
- [2] R. R. Robert Eisberg, Física Cuántica, Mexico: Limusa, 1978.
- [3] D. Donnelly, «The fast Fourier transform for experimentalist, part V: Filters,» *Computing in Science&Engineering*, vol. 1, nº 92, p. 8, 2006.
- [4] S. A. G. M., «Smoothing and differentiation of data by simplified least squares procedures,» *Analytical Chemistry*, vol. 2, nº 1627, p. 36, 1964.
- [5] Math Works, «Math Works,» Matlab App Designer, [En línea]. Available: <https://la.mathworks.com/help/matlab/app-designer.html>. [Último acceso: 06 17 2024].

- [6] ITead Studio, «HC-05 - Bluetooth to Serial Port Module,» 2010. [En línea]. Available: https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf. [Último acceso: 21 mayo 2024].
- [7] MathWorks, «Help Center,» 2006. [En línea]. Available: <https://la.mathworks.com/help/signal/ref/sgolayfilt.html>. [Último acceso: 15 abril 2024].
- [8] MathWorks, «Help Center,» 2016. [En línea]. Available: <https://la.mathworks.com/help/matlab/ref/uieditfield.html>. [Último acceso: 15 abril 2024].
- [9] MathWorks, «Help Center,» 2016. [En línea]. Available: <https://la.mathworks.com/help/matlab/ref/uidropdown.html>. [Último acceso: 15 abril 2024].
- [10] MathWorks, «Help Center,» 2016. [En línea]. Available: <https://la.mathworks.com/help/matlab/ref/uiaxes.html>. [Último acceso: 25 abril 2024].
- [11] MathWorks, «Help Center,» 2020. [En línea]. Available: <https://la.mathworks.com/help/matlab/bluetooth-communication.html>. [Último acceso: 16 abril 2024].
- [12] MathWorks, «Help Center,» 2019. [En línea]. Available: <https://la.mathworks.com/help/matlab/ref/writematrix.html>. [Último acceso: 16 abril 2024].

ANEXO I Tutorial de uso del HMI

Antes de abrir la aplicación se debe conectar el espectrómetro físicamente al puerto USB 2.0 del ordenador, además el Bluetooth del ordenador debe estar encendido.


Adquisición de espectros

El tiempo de integración debe ser configurado con números enteros, además, se debe escoger la unidad de tiempo (microsegundos, milisegundos o segundos).



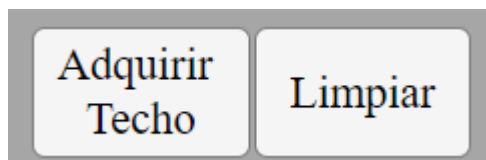
The image shows a user interface for setting the integration time. It consists of a label 'Tiempo de Integración' on the left, a text input field in the middle containing the number '10', and a dropdown menu on the right currently set to 'milisegundos' with a downward-pointing arrow.

En el campo de entrada se debe ingresar el grosor del filtro a utilizar, este puede ser en cualquier unidad de medida debido a que el coeficiente tendrá como unidad el inverso de la longitud.



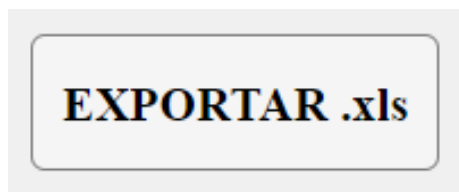
The image shows a text input field with a grey border. To the left of the field is a small 't' symbol. Inside the field, the number '1.000' is entered.

Con el botón de Adquirir en cada gráfico se obtendrá la curva de referencia cuando se lo pulse por primera vez, en caso de tener una medida no válida se puede borrar esta curva con el botón de limpiar. Así mismo cuando sea correcta la referencia, al pulsar nuevamente adquirir se graficará la segunda curva filtrada, de igual manera si no es válida, con el botón de limpiar se borrará la última curva.



The image shows two rectangular buttons side-by-side. The left button is labeled 'Adquirir Techo' and the right button is labeled 'Limpiar'. Both buttons have a light grey background and a thin border.

Con el botón de Exportar .xls se exportará el Excel con los datos obtenidos.



The image shows a single rectangular button with rounded corners and a light grey background. The text 'EXPORTAR .xls' is centered on the button in a bold, black, sans-serif font.

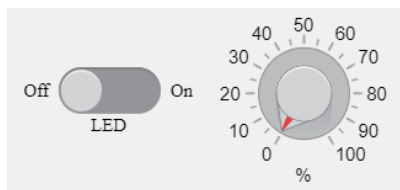
Conexión Bluetooth

Con el botón de Conectar se inicia la comunicación, para saber si ya están conectados los dispositivos el led rojo del módulo HC-05 debe parpadear lento, mientras que cuando está desconectado parpadea rápido.



The image shows a single rectangular button with rounded corners and a light grey background. The text 'CONECTAR' is centered on the button in a bold, black, sans-serif font.

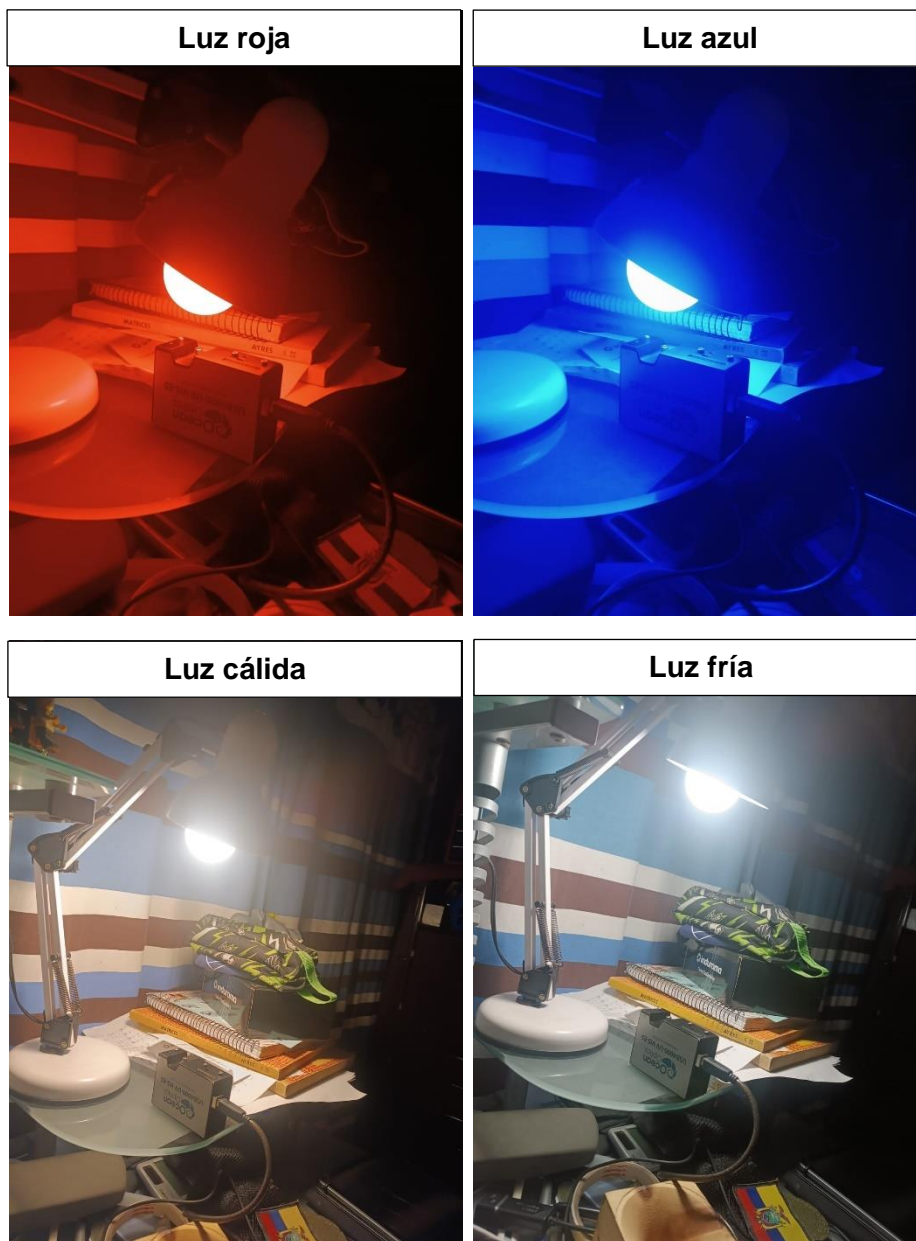
Con el switch se prende o apaga el Led por completo y con la perilla se varía el ancho de pulso de la PWM.



Con el botón de Desconectar se finaliza la comunicación.



ANEXO II Toma de espectros de focos de colores.



ANEXO III Evidencias de toma de datos en Yachay Tech.



ANEXO IV Evidencias de toma de datos en Jardín Botánico de Quito.



ANEXO V Código de App Designer de Matlab.

```
classdef APP_Espectrometro < matlab.apps.AppBase

% Properties that correspond to app components
properties (Access = public)
    UIFigure                matlab.ui.Figure
    Titulo3                 matlab.ui.control.EditField
    Titulo2                 matlab.ui.control.EditField
    Titulo1                 matlab.ui.control.EditField
    Knob                   matlab.ui.control.Knob
    Label                   matlab.ui.control.Label
    DESCONNECTARButton    matlab.ui.control.Button
    CONECTARButton        matlab.ui.control.Button
    LEDSwitch              matlab.ui.control.Switch
    LEDSwitchLabel        matlab.ui.control.Label
    RealizadoporJohaoSnchezMarcoSalazarLabel matlab.ui.control.Label
    Titulo                 matlab.ui.control.Label
    EFFotIn3               matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_12 matlab.ui.control.Label
    EFFotOut3              matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_11 matlab.ui.control.Label
    EFTled                 matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_10 matlab.ui.control.Label
    EFColed                matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_9 matlab.ui.control.Label
    EFFotIn2               matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_8 matlab.ui.control.Label
    EFFotOut2              matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_7 matlab.ui.control.Label
    EFTmedio               matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_6 matlab.ui.control.Label
    EFCoMedio              matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_5 matlab.ui.control.Label
    Leyenda                matlab.ui.control.Label
    EFFotIn1               matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_4 matlab.ui.control.Label
    EFFotOut1              matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_3 matlab.ui.control.Label
    EFTtecho               matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel_2 matlab.ui.control.Label
    EFCoTecho              matlab.ui.control.NumericEditField
    CoeficienteatenuacinLabel matlab.ui.control.Label
    Unidades                matlab.ui.control.DropDown
    EXPORTARxls            matlab.ui.control.Button
    Limpiar3                matlab.ui.control.Button
    AdquirirLed            matlab.ui.control.Button
    TiempodeIntegracion    matlab.ui.control.NumericEditField
    TiempodeIntegracinLabel matlab.ui.control.Label
    Limpiar2                matlab.ui.control.Button
    AdquirirMedio          matlab.ui.control.Button
    Image2                  matlab.ui.control.Image
    Image                   matlab.ui.control.Image
    Limpiar1                matlab.ui.control.Button
    AdquirirTecho          matlab.ui.control.Button
    Fondo1                  matlab.ui.control.Label
    Fondo2                  matlab.ui.control.Label
    Fondo3                  matlab.ui.control.Label
end
```

```

GraficoTecho           matlab.ui.control.UIAxes
GraficoLED             matlab.ui.control.UIAxes
GraficoMedio           matlab.ui.control.UIAxes
end

properties (Access = private)
    Tintegracion = 0;
    AUX1 = 0; % AUXILIAR PARA GRAFICO TECHO
    AUX2 = 0; % AUXILIAR PARA GRAFICO MEDIO
    AUX3 = 0; % AUXILIAR PARA GRAFICO LED
    LongitudREF1 = 0; % Longitud de onda SOL   TECHO
    EspectroREF1 = 0; % Espectro             SOL   TECHO
    LongitudFIL1 = 0; % Longitud de onda FIL   TECHO
    EspectroFIL1 = 0; % Espectro             FIL   TECHO
    LongitudREF2 = 0; % Longitud de onda SOL   MEDIO
    EspectroREF2 = 0; % Espectro             SOL   MEDIO
    LongitudFIL2 = 0; % Longitud de onda FIL   MEDIO
    EspectroFIL2 = 0; % Espectro             FIL   MEDIO
    LongitudREF3 = 0; % Longitud de onda SOL   LED
    EspectroREF3 = 0; % Espectro             SOL   LED
    LongitudFIL3 = 0; % Longitud de onda FIL   LED
    EspectroFIL3 = 0; % Espectro             FIL   LED

    Aref1 = 0; %Area referencia   techo
    Afil1 = 0; %Area filtrada     techo
    Coftecho = 0; %Coeficiente     techo
    T_techo = 1; %espesor coeficiente techo

    Aref2 = 0; %Area referencia   medio
    Afil2 = 0; %Area filtrada     medio
    Cofmedio = 0; %Coeficiente     medio
    T_medio = 1; %espesor coeficiente techo

    Aref3 = 0; %Area referencia   LED
    Afil3 = 0; %Area filtrada     LED
    CofLED = 0; %Coeficiente     medio
    T_LED = 1; %espesor plastico led
    comu % Description

end

% Callbacks that handle component events
methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)
        %comunicacion bluetooth
        app.comu = Bluetooth('btspp://98D30296C567',1);
    end

    % Button pushed function: AdquirirTecho
    function AdquirirTechoPushed(app, event)
        global spectrometerObj wavelengths spectralData dxr1 dyr1 dxf1 dyf1
        %%%%%%%%% GRAFICAR REFERENCIA %%%%%%%%%
        app.AUX1 = app.AUX1 + 1;
        if app.AUX1 == 1

```

```

spectrometerObj = icdevice('OceanOptics_OmniDriver.mdd');
connect(spectrometerObj);
% integration time for sensor.
integrationTime = app.Tintegracion;
% Spectrometer index to use (first spectrometer by default).
spectrometerIndex = 0;
% Channel index to use (first channel by default).
channelIndex = 0;
% Enable flag.
enable = 1;
% Set integration time.
invoke(spectrometerObj, 'setIntegrationTime', spectrometerIndex,
channelIndex, integrationTime);
% Enable correct for detector non-linearity.
invoke(spectrometerObj, 'setCorrectForDetectorNonlinearity',
spectrometerIndex, channelIndex, enable);
% Enable correct for electrical dark.
invoke(spectrometerObj, 'setCorrectForElectricalDark',
spectrometerIndex, channelIndex, enable);
wavelengths = invoke(spectrometerObj, 'getWavelengths',
spectrometerIndex, channelIndex);
% Get the wavelengths of the first spectrometer and save them in a
double
% array.
spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

app.LongitudREF1 = sgolayfilt(wavelengths,2,21);
app.EspectroREF1 = sgolayfilt(spectralData,2,21);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disconnect(spectrometerObj);
delete (spectrometerObj);
hold(app.GraficoTecho,"on")
plot(app.GraficoTecho, app.LongitudREF1,app.EspectroREF1);
legend(app.GraficoTecho,"Ref","Fil")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:1:3647
dxr1=(app.LongitudREF1(i+1,1)-app.LongitudREF1(i,1));
dyr1=app.EspectroREF1(i,1) + (app.EspectroREF1(i+1,1) -
app.EspectroREF1(i,1))/2;
app.Aref1 = (dxr1*dyr1) + app.Aref1;
end
app.EFCoTecho.Value = 0;
app.EFFotIn1.Value = app.Aref1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end

%%%%%%%%%% GRAFICAR FILTRADA %%%%%%%%%%%

if app.AUX1 == 2

```

```

spectrometerObj = icdevice('OceanOptics_OmniDriver.mdd');
connect(spectrometerObj);
% integration time for sensor.
integrationTime = app.Tintegracion;
% Spectrometer index to use (first spectrometer by default).
spectrometerIndex = 0;
% Channel index to use (first channel by default).
channelIndex = 0;
% Enable flag.
enable = 1;
% Set integration time.
invoke(spectrometerObj, 'setIntegrationTime', spectrometerIndex,
channelIndex, integrationTime);
% Enable correct for detector non-linearity.
invoke(spectrometerObj, 'setCorrectForDetectorNonlinearity',
spectrometerIndex, channelIndex, enable);
% Enable correct for electrical dark.
invoke(spectrometerObj, 'setCorrectForElectricalDark',
spectrometerIndex, channelIndex, enable);
wavelengths = invoke(spectrometerObj, 'getWavelengths',
spectrometerIndex, channelIndex);
% Get the wavelengths of the first spectrometer and save them in a
double
% array.
spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
app.LongitudFIL1 = sgolayfilt(wavelengths,2,21);
app.EspectroFIL1 = sgolayfilt(spectralData,2,21);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disconnect(spectrometerObj);
delete (spectrometerObj);
plot(app.GraficoTecho, app.LongitudFIL1,app.EspectroFIL1);
hold(app.GraficoTecho,"off")
legend(app.GraficoTecho,"Ref","Fil")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:1:3647
    dxf1=(app.LongitudFIL1(i+1,1) - app.LongitudFIL1(i,1));
    dyf1=app.EspectroFIL1(i,1) + (app.EspectroFIL1(i+1,1) -
app.EspectroFIL1(i,1))/2;
    app.Afil1 = (dxf1*dyf1) + app.Afil1;
end
app.EFFotOut1.Value = app.Afil1;
app.Coftecho = (1/app.T_techo)*log(app.Aref1/app.Afil1);
app.EFCoTecho.Value = app.Coftecho;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
end

% Button pushed function: Limpiar1

```



```

function Limpiar1Pushed(app, event)
app.AUX1 = app.AUX1 - 1;
if app.AUX1 == 0
hold(app.GraficoTecho,"off")
cla(app.GraficoTecho)
legend(app.GraficoTecho,"Ref","Fil")
app.EFCoTecho.Value = 0;
app.Aref1 = 0;
app.Afil1 = 0;
app.Coftecho = 0;
app.EFFotOut1.Value = 0;
app.EFFotIn1.Value = 0;
end

if app.AUX1 == 1
cla(app.GraficoTecho)
hold(app.GraficoTecho,"on")
plot(app.GraficoTecho, app.LongitudREF1,app.EspectroREF1);
legend(app.GraficoTecho,"Ref","Fil")
app.EFCoTecho.Value = 0;
app.Afil1 = 0;
app.EFFotOut1.Value = 0;
end
end

% Button pushed function: AdquirirMedio
function AdquirirMedioPushed(app, event)
global spectrometerObj wavelengths spectralData dxr2 dyr2 dxf2 dyf2
%%%%%%%%%%%%% GRAFICAR REFERENCIA %%%%%%%%%%%%%%
app.AUX2 = app.AUX2 + 1;
if app.AUX2 == 1

spectrometerObj = icdevice('OceanOptics_OmniDriver.mdd');
connect(spectrometerObj);
% integration time for sensor.
integrationTime = app.Tintegracion;
% Spectrometer index to use (first spectrometer by default).
spectrometerIndex = 0;
% Channel index to use (first channel by default).
channelIndex = 0;
% Enable flag.
enable = 1;
% Set integration time.
invoke(spectrometerObj, 'setIntegrationTime', spectrometerIndex,
channelIndex, integrationTime);
% Enable correct for detector non-linearity.
invoke(spectrometerObj, 'setCorrectForDetectorNonlinearity',
spectrometerIndex, channelIndex, enable);
% Enable correct for electrical dark.
invoke(spectrometerObj, 'setCorrectForElectricalDark',
spectrometerIndex, channelIndex, enable);
wavelengths = invoke(spectrometerObj, 'getWavelengths',
spectrometerIndex, channelIndex);
% Get the wavelengths of the first spectrometer and save them in a
double
% array.
spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    app.LongitudREF2 = sgolayfilt(wavelengths,2,21);
    app.EspectroREF2 = sgolayfilt(spectralData,2,21);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    disconnect(spectrometerObj);
    delete (spectrometerObj);
    hold(app.GraficoMedio,"on")
    plot(app.GraficoMedio, app.LongitudREF2,app.EspectroREF2);
    legend(app.GraficoMedio,"Ref","Fil")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for i = 1:1:3647
        dxr2=(app.LongitudREF2(i+1,1)-app.LongitudREF2(i,1));
        dyr2=app.EspectroREF2(i,1) + (app.EspectroREF2(i+1,1) -
app.EspectroREF2(i,1))/2;
        app.Aref2 = (dxr2*dyr2) + app.Aref2;
    end
    app.EFCoMedio.Value = 0;
    app.EFFotIn2.Value = app.Aref2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% GRAFICAR FILTRADA %%%%%%%%%

    if app.AUX2 == 2
        spectrometerObj = icdevice('OceanOptics_OmniDriver.mdd');
        connect(spectrometerObj);
        % integration time for sensor.
        integrationTime = app.Tintegracion;
        % Spectrometer index to use (first spectrometer by default).
        spectrometerIndex = 0;
        % Channel index to use (first channel by default).
        channelIndex = 0;
        % Enable flag.
        enable = 1;
        % Set integration time.
        invoke(spectrometerObj, 'setIntegrationTime', spectrometerIndex,
channelIndex, integrationTime);
        % Enable correct for detector non-linearity.
        invoke(spectrometerObj, 'setCorrectForDetectorNonlinearity',
spectrometerIndex, channelIndex, enable);
        % Enable correct for electrical dark.
        invoke(spectrometerObj, 'setCorrectForElectricalDark',
spectrometerIndex, channelIndex, enable);
        wavelengths = invoke(spectrometerObj, 'getWavelengths',
spectrometerIndex, channelIndex);
        % Get the wavelengths of the first spectrometer and save them in a
double
        % array.
        spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    app.LongitudFIL2 = sgolayfilt(wavelengths,2,21);
    app.EspectroFIL2 = sgolayfilt(spectralData,2,21);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    disconnect(spectrometerObj);
    delete (spectrometerObj);
    plot(app.GraficoMedio, app.LongitudFIL2,app.EspectroFIL2);
    hold(app.GraficoMedio,"off")
    legend(app.GraficoMedio,"Ref","Fil")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for i = 1:1:3647
        dx2=(app.LongitudFIL2(i+1,1) - app.LongitudFIL2(i,1));
        dy2=app.EspectroFIL2(i,1) + (app.EspectroFIL2(i+1,1) -
app.EspectroFIL2(i,1))/2;
        app.Afil2 = (dx2*dy2) + app.Afil2;
    end
    app.EFFotOut2.Value = app.Afil2;
    app.Cofmedio = (1/app.T_medio)*log(app.Aref2/app.Afil2);
    app.EFCoMedio.Value = app.Cofmedio;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end
end

% Button pushed function: Limpiar2
function Limpiar2Pushed(app, event)
app.AUX2 = app.AUX2 - 1;
if app.AUX2 == 0
hold(app.GraficoMedio,"off")
cla(app.GraficoMedio)
legend(app.GraficoMedio,"Ref","Fil")
app.EFCoMedio.Value = 0;
app.Aref2 = 0;
app.Afil2 = 0;
app.Cofmedio = 0;
app.EFFotOut2.Value = 0;
app.EFFotIn2.Value = 0;
end

if app.AUX2 == 1
cla(app.GraficoMedio)
hold(app.GraficoMedio,"on")
plot(app.GraficoMedio, app.LongitudREF2,app.EspectroREF2);
legend(app.GraficoMedio,"Ref","Fil")
app.EFCoMedio.Value = 0;
app.Afil2 = 0;
app.EFFotOut2.Value = 0;
end
end

% Button pushed function: AdquirirLed

```

```

function AdquirirLedPushed(app, event)
    global spectrometerObj wavelengths spectralData dxr2 dyr2 dxf3 dyf3
    %%%%%%%%%% GRAFICAR REFERENCIA %%%%%%%%%%
    app.AUX3 = app.AUX3 + 1;
    if app.AUX3 == 1

        spectrometerObj = icdevice('OceanOptics_OmniDriver.mdd');
        connect(spectrometerObj);
        % integration time for sensor.
        integrationTime = app.Tintegracion;
        % Spectrometer index to use (first spectrometer by default).
        spectrometerIndex = 0;
        % Channel index to use (first channel by default).
        channelIndex = 0;
        % Enable flag.
        enable = 1;
        % Set integration time.
        invoke(spectrometerObj, 'setIntegrationTime', spectrometerIndex,
channelIndex, integrationTime);
        % Enable correct for detector non-linearity.
        invoke(spectrometerObj, 'setCorrectForDetectorNonlinearity',
spectrometerIndex, channelIndex, enable);
        % Enable correct for electrical dark.
        invoke(spectrometerObj, 'setCorrectForElectricalDark',
spectrometerIndex, channelIndex, enable);
        wavelengths = invoke(spectrometerObj, 'getWavelengths',
spectrometerIndex, channelIndex);
        % Get the wavelengths of the first spectrometer and save them in a
double
        % array.
        spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        app.LongitudREF3 = sgolayfilt(wavelengths,2,21);
        app.EspectroREF3 = sgolayfilt(spectralData,2,21);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        disconnect(spectrometerObj);
        delete (spectrometerObj);
        hold(app.GraficoLED, "on")
        plot(app.GraficoLED, app.LongitudREF3, app.EspectroREF3);
        legend(app.GraficoLED, "Ref", "Fil")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        for i = 1:1:3647
            dxr2=(app.LongitudREF3(i+1,1)-app.LongitudREF3(i,1));
            dyr2=app.EspectroREF3(i,1) + (app.EspectroREF3(i+1,1) -
app.EspectroREF3(i,1))/2;
            app.Aref3 = (dxr2*dyr2) + app.Aref3;
        end
        app.EFColed.Value = 0;
        app.EFFotIn3.Value = app.Aref3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% GRAFICAR FILTRADA %%%%%%%%%

if app.AUX3 == 2
spectrometerObj = icdevice('OceanOptics_OmniDriver.mdd');
connect(spectrometerObj);
% integration time for sensor.
integrationTime = app.Tintegracion;
% Spectrometer index to use (first spectrometer by default).
spectrometerIndex = 0;
% Channel index to use (first channel by default).
channelIndex = 0;
% Enable flag.
enable = 1;
% Set integration time.
invoke(spectrometerObj, 'setIntegrationTime', spectrometerIndex,
channelIndex, integrationTime);
% Enable correct for detector non-linearity.
invoke(spectrometerObj, 'setCorrectForDetectorNonlinearity',
spectrometerIndex, channelIndex, enable);
% Enable correct for electrical dark.
invoke(spectrometerObj, 'setCorrectForElectricalDark',
spectrometerIndex, channelIndex, enable);
wavelengths = invoke(spectrometerObj, 'getWavelengths',
spectrometerIndex, channelIndex);
% Get the wavelengths of the first spectrometer and save them in a
double
% array.
spectralData = invoke(spectrometerObj, 'getSpectrum',
spectrometerIndex);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
app.LongitudFIL3 = sgolayfilt(wavelengths,2,21);
app.EspectroFIL3 = sgolayfilt(spectralData,2,21);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
disconnect(spectrometerObj);
delete (spectrometerObj);
plot(app.GraficoLED, app.LongitudFIL3,app.EspectroFIL3);
hold(app.GraficoLED,"off")
legend(app.GraficoLED,"Ref","Fil")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:1:3647
dx3=(app.LongitudFIL3(i+1,1) - app.LongitudFIL3(i,1));
dy3=app.EspectroFIL3(i,1) + (app.EspectroFIL3(i+1,1) -
app.EspectroFIL3(i,1))/2;
app.Afil3 = (dx3*dy3) + app.Afil3;
end
app.EFFotOut3.Value = app.Afil3;
app.CofLED = (1/app.T_LED)*log(app.Aref3/app.Afil3);
app.EFColed.Value = app.CofLED;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
end

% Button pushed function: Limpiar3
function Limpiar3Pushed(app, event)
app.AUX3 = app.AUX3 - 1;
if app.AUX3 == 0
hold(app.GraficoLED,"off")
cla(app.GraficoLED)
legend(app.GraficoLED,"Ref","Fil")
app.EFColed.Value = 0;
app.Aref3 = 0;
app.Afil3 = 0;
app.CofLED = 0;
app.EFFotOut3.Value = 0;
app.EFFotIn3.Value = 0;
end

if app.AUX3 == 1
cla(app.GraficoLED)
hold(app.GraficoLED,"on")
plot(app.GraficoLED, app.LongitudREF3,app.EspectroREF3);
legend(app.GraficoLED,"Ref","Fil")
app.EFColed.Value = 0;
app.Afil3 = 0;
app.EFFotOut3.Value = 0;
end

end

% Button pushed function: EXPORTARxls
function EXPORTARxlsButtonPushed(app, event)
global datos parametros
datos = [app.LongitudREF1'; app.EspectroREF1'; app.LongitudFIL1';
app.EspectroFIL1'; app.LongitudREF2'; app.EspectroREF2'; app.LongitudFIL2';
app.EspectroFIL2'; app.LongitudREF3'; app.EspectroREF3'; app.LongitudFIL3';
app.EspectroFIL3'];
parametros = [app.Aref1 app.Afil1 app.Coftecho app.T_techo;
app.Aref2 app.Afil2 app.Cofmedio app.T_medio; app.Aref3 app.Afil3 app.CofLED
app.T_LED];
%titulos = [app.Titulo1.Value; app.Titulo2.Value;
app.Titulo3.Value];
writematrix(datos,'datos.xlsx','Sheet',2,'Range','A3')
writematrix(parametros,'datos.xlsx','Sheet',2,'Range','M3')
%writematrix(titulos,'datos.xlsx','Sheet',2,'Range','M6')
end

% Value changed function: Unidades
function UnidadesValueChanged(app, event)
value = app.Unidades.Value;
switch value
case 'segundos'
app.Tintegracion = app.TiempodeIntegracion.Value * 1000000;
case 'milisegundos'
app.Tintegracion = app.TiempodeIntegracion.Value * 1000;
case 'microsegundos'

```

```

app.Tintegracion = app.TiempodeIntegracion.Value * 1;
end
end

% Value changed function: EFCoTecho
function EFCoTechoValueChanged(app, event)
    app.EFCoTecho.Value = app.Coftecho;
end

% Value changed function: EFTtecho
function EFTtechoValueChanged(app, event)
    app.T_techo = app.EFTtecho.Value;
    app.Coftecho = (1/app.T_techo)*log(app.Aref1/app.Afil1);
    app.EFCoTecho.Value = app.Coftecho;
end

% Value changed function: EFFotIn1
function EFFotIn1ValueChanged(app, event)
    app.EFFotIn1.Value = app.Aref1;
end

% Value changed function: EFFotOut1
function EFFotOut1ValueChanged(app, event)
    app.EFFotOut1.Value = app.Afil1;
end

% Value changed function: EFCoMedio
function EFCoMedioValueChanged(app, event)
    app.EFCoMedio.Value = app.Cofmedio;
end

% Value changed function: EFTmedio
function EFTmedioValueChanged(app, event)
    app.T_medio = app.EFTmedio.Value;
    app.Cofmedio = (1/app.T_medio)*log(app.Aref2/app.Afil2);
    app.EFCoMedio.Value = app.Cofmedio;
end

% Value changed function: EFFotIn2
function EFFotIn2ValueChanged(app, event)
    app.EFFotIn2.Value = app.Aref2;
end

% Value changed function: EFFotOut2
function EFFotOut2ValueChanged(app, event)
    app.EFFotOut2.Value = app.Afil2;
end

% Value changed function: EFTled
function EFTledValueChanged(app, event)
    app.T_LED = app.EFTled.Value;
    app.CofLED = (1/app.T_LED)*log(app.Aref3/app.Afil3);
    app.EFColed.Value = app.CofLED;
end

% Value changed function: EFColed
function EFColedValueChanged(app, event)
    app.EFColed.Value= app.CofLED;
end

```

```

end

% Value changed function: EFFotIn3
function EFFotIn3ValueChanged(app, event)
    app.EFFotIn3.Value = app.Aref3;
end

% Value changed function: EFFotOut3
function EFFotOut3ValueChanged(app, event)
    app.EFFotOut3.Value= app.Afil3;
end

% Value changed function: LEDSwitch
function LEDSwitchValueChanged(app, event)
    global estadoLED
    value = app.LEDSwitch.Value;
    estadoLED = value;
    if strcmp(estadoLED, 'On')
        fwrite(app.comu, 'e'); %led encendido
        fwrite(app.comu, 'a');
        % inteLEDr = round(app.intLED);
        % fwrite(app.comu, inteLEDr);
    elseif strcmp(estadoLED, 'Off')
        fwrite(app.comu, 'p'); %led apagado
        fwrite(app.comu, 'a');
        % inteLEDr = round(app.intLED);
        %fwrite(app.comu, inteLEDr);
    end
end

% Close request function: UIFigure
function UIFigureCloseRequest(app, event)
    fclose(app.comu);
    delete(app.comu);
    delete(app)
end

% Button pushed function: CONECTARButton
function CONECTARButtonPushed(app, event)
    fopen(app.comu); %abrir puerto serial
end

% Button pushed function: DESCONECTARButton
function DESCONECTARButtonPushed(app, event)
    fwrite(app.comu, 'p');
    fwrite(app.comu, 'a');
    fclose(app.comu);
    clear all;
end

% Callback function
function KnobValueChanging(app, event)
    changingValue = event.Value;
    app.intLED = changingValue;
end

% Value changed function: Knob
function KnobValueChanged(app, event)

```



```

    if (app.Knob.Value <= 10)
        fwrite(app.comu, 'q');
    end
    if (app.Knob.Value <= 20) && (app.Knob.Value > 10)
        fwrite(app.comu, 'r');
    end
    if (app.Knob.Value <= 30) && (app.Knob.Value > 20)
        fwrite(app.comu, 's');
    end
    if (app.Knob.Value <= 40) && (app.Knob.Value > 30)
        fwrite(app.comu, 't');
    end
    if (app.Knob.Value <= 50) && (app.Knob.Value > 40)
        fwrite(app.comu, 'u');
    end
    if (app.Knob.Value <= 60) && (app.Knob.Value > 50)
        fwrite(app.comu, 'v');
    end
    if (app.Knob.Value <= 70) && (app.Knob.Value > 60)
        fwrite(app.comu, 'w');
    end
    if (app.Knob.Value <= 80) && (app.Knob.Value > 70)
        fwrite(app.comu, 'x');
    end
    if (app.Knob.Value <= 90) && (app.Knob.Value > 80)
        fwrite(app.comu, 'y');
    end
    if (app.Knob.Value <= 100) && (app.Knob.Value > 90)
        fwrite(app.comu, 'z');
    end
end

% Value changing function: Titulo1
function Titulo1ValueChanging(app, event)
    title(app.GraficoTecho,event.Value);
end

% Value changing function: Titulo2
function Titulo2ValueChanging(app, event)
    title(app.GraficoMedio,event.Value);
end

% Value changing function: Titulo3
function Titulo3ValueChanging(app, event)
    title(app.GraficoLED,event.Value);
end
end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

    % Create UIFigure and hide until all components are created
    app.UIFigure = uifigure('Visible', 'off');
    app.UIFigure.Position = [100 100 1389 757];
    app.UIFigure.Name = 'MATLAB App';

```

```

app.UIFigure.CloseRequestFcn = createCallbackFcn(app,
@UIFigureCloseRequest, true);

% Create GraficoMedio
app.GraficoMedio = uiaxes(app.UIFigure);
title(app.GraficoMedio, 'MUESTRA 2')
xlabel(app.GraficoMedio, '\lambda [nm]')
ylabel(app.GraficoMedio, 'Intensidad [cuentas]')
zlabel(app.GraficoMedio, 'Z')
app.GraficoMedio.FontName = 'Times New Roman';
app.GraficoMedio.FontWeight = 'bold';
app.GraficoMedio.YGrid = 'on';
app.GraficoMedio.ColorOrder = [0 0 1;1 0 0;1 0.411764705882353
0.16078431372549;0 1 0;0.466666666666667 0.674509803921569
0.188235294117647;0.494117647058824 0.184313725490196
0.556862745098039;0.635294117647059 0.0784313725490196 0.184313725490196];
app.GraficoMedio.FontSize = 17;
app.GraficoMedio.Position = [472 3 448 372];

% Create GraficoLED
app.GraficoLED = uiaxes(app.UIFigure);
title(app.GraficoLED, 'MUESTRA 3')
xlabel(app.GraficoLED, '\lambda [nm]')
ylabel(app.GraficoLED, 'Intensidad [cuentas]')
zlabel(app.GraficoLED, 'Z')
app.GraficoLED.FontName = 'Times New Roman';
app.GraficoLED.FontWeight = 'bold';
app.GraficoLED.YGrid = 'on';
app.GraficoLED.ColorOrder = [0.850980392156863 0.325490196078431
0.0980392156862745;0.301960784313725 0.745098039215686 0.933333333333333;0 1
0;0.929411764705882 0.694117647058824 0.125490196078431;0.466666666666667
0.674509803921569 0.188235294117647;0.301960784313725 0.745098039215686
0.933333333333333;0.635294117647059 0.0784313725490196 0.184313725490196];
app.GraficoLED.FontSize = 17;
app.GraficoLED.Position = [941 1 448 372];

% Create GraficoTecho
app.GraficoTecho = uiaxes(app.UIFigure);
title(app.GraficoTecho, 'MUESTRA 1')
xlabel(app.GraficoTecho, '\lambda [nm]')
ylabel(app.GraficoTecho, 'Intensidad [cuentas]')
zlabel(app.GraficoTecho, 'Z')
app.GraficoTecho.FontName = 'Times New Roman';
app.GraficoTecho.FontWeight = 'bold';
app.GraficoTecho.YGrid = 'on';
app.GraficoTecho.ColorOrder = [0.466666666666667 0.674509803921569
0.188235294117647;0.929411764705882 0.694117647058824 0.125490196078431;1
0.0745098039215686 0.650980392156863;0 0 0;0 0.447058823529412
0.741176470588235;0.301960784313725 0.745098039215686
0.933333333333333;0.850980392156863 0.325490196078431 0.0980392156862745];
app.GraficoTecho.FontSize = 17;
app.GraficoTecho.Position = [1 2 448 371];

% Create Fondo3
app.Fondo3 = uilabel(app.UIFigure);
app.Fondo3.BackgroundColor = [0.651 0.651 0.651];
app.Fondo3.Position = [1033 384 318 154];
app.Fondo3.Text = '';

```

```

% Create Fondo2
app.Fondo2 = uilabel(app.UIFigure);
app.Fondo2.BackgroundColor = [0.651 0.651 0.651];
app.Fondo2.Position = [558 384 318 154];
app.Fondo2.Text = '';

% Create Fondo1
app.Fondo1 = uilabel(app.UIFigure);
app.Fondo1.BackgroundColor = [0.651 0.651 0.651];
app.Fondo1.Position = [89 384 318 154];
app.Fondo1.Text = '';

% Create AdquirirTecho
app.AdquirirTecho = uibutton(app.UIFigure, 'push');
app.AdquirirTecho.ButtonPushedFcn = createCallbackFcn(app,
@AdquirirTechoPushed, true);
app.AdquirirTecho.FontName = 'Times New Roman';
app.AdquirirTecho.FontSize = 18;
app.AdquirirTecho.Position = [160 420 87 52];
app.AdquirirTecho.Text = {'Adquirir '; 'Techo'};

% Create Limpiar1
app.Limpiar1 = uibutton(app.UIFigure, 'push');
app.Limpiar1.ButtonPushedFcn = createCallbackFcn(app,
@Limpiar1Pushed, true);
app.Limpiar1.FontName = 'Times New Roman';
app.Limpiar1.FontSize = 18;
app.Limpiar1.Position = [249 420 86 52];
app.Limpiar1.Text = 'Limpiar';

% Create Image
app.Image = uiimage(app.UIFigure);
app.Image.Position = [17 662 135 84];
app.Image.ImageSource = 'Logo EPN.png';

% Create Image2
app.Image2 = uiimage(app.UIFigure);
app.Image2.Position = [178 660 204 89];
app.Image2.ImageSource = 'Logo DACI.png';

% Create AdquirirMedio
app.AdquirirMedio = uibutton(app.UIFigure, 'push');
app.AdquirirMedio.ButtonPushedFcn = createCallbackFcn(app,
@AdquirirMedioPushed, true);
app.AdquirirMedio.FontName = 'Times New Roman';
app.AdquirirMedio.FontSize = 17;
app.AdquirirMedio.Position = [629 420 87 52];
app.AdquirirMedio.Text = {'Adquirir'; 'Medio'};

% Create Limpiar2
app.Limpiar2 = uibutton(app.UIFigure, 'push');
app.Limpiar2.ButtonPushedFcn = createCallbackFcn(app,
@Limpiar2Pushed, true);
app.Limpiar2.FontName = 'Times New Roman';
app.Limpiar2.FontSize = 17;
app.Limpiar2.Position = [719 420 86 52];
app.Limpiar2.Text = 'Limpiar';

```

```

% Create TiempodeIntegracinLabel
app.TiempodeIntegracinLabel = uilabel(app.UIFigure);
app.TiempodeIntegracinLabel.HorizontalAlignment = 'right';
app.TiempodeIntegracinLabel.FontName = 'Times New Roman';
app.TiempodeIntegracinLabel.FontSize = 18;
app.TiempodeIntegracinLabel.Position = [45 588 91 44];
app.TiempodeIntegracinLabel.Text = {'Tiempo de '; 'Integración '};

% Create TiempodeIntegracion
app.TiempodeIntegracion = uieditfield(app.UIFigure, 'numeric');
app.TiempodeIntegracion.Limits = [0 800000];
app.TiempodeIntegracion.RoundFractionalValues = 'on';
app.TiempodeIntegracion.ValueDisplayFormat = '%.0f';
app.TiempodeIntegracion.HorizontalAlignment = 'left';
app.TiempodeIntegracion.FontName = 'Times New Roman';
app.TiempodeIntegracion.FontSize = 18;
app.TiempodeIntegracion.Position = [151 588 100 44];
app.TiempodeIntegracion.Value = 10;

% Create AdquirirLed
app.AdquirirLed = uibutton(app.UIFigure, 'push');
app.AdquirirLed.ButtonPushedFcn = createCallbackFcn(app,
@AdquirirLedPushed, true);
app.AdquirirLed.FontName = 'Times New Roman';
app.AdquirirLed.FontSize = 17;
app.AdquirirLed.Position = [1104 421 86 52];
app.AdquirirLed.Text = {'Adquirir'; 'Led'};

% Create Limpiar3
app.Limpiar3 = uibutton(app.UIFigure, 'push');
app.Limpiar3.ButtonPushedFcn = createCallbackFcn(app,
@Limpiar3Pushed, true);
app.Limpiar3.FontName = 'Times New Roman';
app.Limpiar3.FontSize = 17;
app.Limpiar3.Position = [1193 421 87 52];
app.Limpiar3.Text = 'Limpiar';

% Create EXPORTARxls
app.EXPORTARxls = uibutton(app.UIFigure, 'push');
app.EXPORTARxls.ButtonPushedFcn = createCallbackFcn(app,
@EXPORTARxlsButtonPushed, true);
app.EXPORTARxls.FontName = 'Times New Roman';
app.EXPORTARxls.FontSize = 17;
app.EXPORTARxls.FontWeight = 'bold';
app.EXPORTARxls.Position = [994 588 144 48];
app.EXPORTARxls.Text = 'EXPORTAR .xls';

% Create Unidades
app.Unidades = uidropdown(app.UIFigure);
app.Unidades.Items = {'segundos', 'milisegundos', 'microsegundos'};
app.Unidades.ValueChangedFcn = createCallbackFcn(app,
@UnidadesValueChanged, true);
app.Unidades.FontName = 'Times New Roman';
app.Unidades.FontSize = 18;
app.Unidades.Position = [259 597 147 24];
app.Unidades.Value = 'milisegundos';

```

```

% Create CoeficienteatenuacinLabel
app.CoficienteatenuacinLabel = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel.FontSize = 20;
app.CoficienteatenuacinLabel.FontWeight = 'bold';
app.CoficienteatenuacinLabel.FontColor = [1 1 1];
app.CoficienteatenuacinLabel.Position = [249 476 27 26];
app.CoficienteatenuacinLabel.Text = 'op';

% Create EFCoTecho
app.EFCoTecho = uieditfield(app.UIFigure, 'numeric');
app.EFCoTecho.ValueDisplayFormat = '%.3f';
app.EFCoTecho.ValueChangedFcn = createCallbackFcn(app,
@EFCoTechoValueChanged, true);
app.EFCoTecho.HorizontalAlignment = 'left';
app.EFCoTecho.FontName = 'Times New Roman';
app.EFCoTecho.FontSize = 20;
app.EFCoTecho.Position = [284 476 115 26];

% Create CoeficienteatenuacinLabel_2
app.CoficienteatenuacinLabel_2 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_2.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_2.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_2.FontSize = 20;
app.CoficienteatenuacinLabel_2.FontWeight = 'bold';
app.CoficienteatenuacinLabel_2.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_2.Position = [255 510 21 24];
app.CoficienteatenuacinLabel_2.Text = 't';

% Create EFTtecho
app.EFTtecho = uieditfield(app.UIFigure, 'numeric');
app.EFTtecho.Limits = [1e-05 Inf];
app.EFTtecho.ValueDisplayFormat = '%.3f';
app.EFTtecho.ValueChangedFcn = createCallbackFcn(app,
@EFTtechoValueChanged, true);
app.EFTtecho.HorizontalAlignment = 'left';
app.EFTtecho.FontName = 'Times New Roman';
app.EFTtecho.FontSize = 20;
app.EFTtecho.Position = [284 507 115 27];
app.EFTtecho.Value = 1;

% Create CoeficienteatenuacinLabel_3
app.CoficienteatenuacinLabel_3 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_3.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_3.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_3.FontSize = 20;
app.CoficienteatenuacinLabel_3.FontWeight = 'bold';
app.CoficienteatenuacinLabel_3.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_3.Position = [88 476 25 26];
app.CoficienteatenuacinLabel_3.Text = 'It';

% Create EFFotOut1
app.EFFotOut1 = uieditfield(app.UIFigure, 'numeric');
app.EFFotOut1.ValueDisplayFormat = '%.3f';
app.EFFotOut1.ValueChangedFcn = createCallbackFcn(app,
@EFFotOut1ValueChanged, true);
app.EFFotOut1.HorizontalAlignment = 'left';

```

```

app.EFFotOut1.FontName = 'Times New Roman';
app.EFFotOut1.FontSize = 20;
app.EFFotOut1.Position = [121 475 115 27];

% Create CoeficienteatenuacinLabel_4
app.CoficienteatenuacinLabel_4 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_4.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_4.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_4.FontSize = 20;
app.CoficienteatenuacinLabel_4.FontWeight = 'bold';
app.CoficienteatenuacinLabel_4.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_4.Position = [90 507 25 26];
app.CoficienteatenuacinLabel_4.Text = 'Io';

% Create EFFotIn1
app.EFFotIn1 = uieditfield(app.UIFigure, 'numeric');
app.EFFotIn1.ValueDisplayFormat = '%.3f';
app.EFFotIn1.ValueChangedFcn = createCallbackFcn(app,
@EFFotIn1ValueChanged, true);
app.EFFotIn1.HorizontalAlignment = 'left';
app.EFFotIn1.FontName = 'Times New Roman';
app.EFFotIn1.FontSize = 20;
app.EFFotIn1.FontColor = [0.149 0.149 0.149];
app.EFFotIn1.Position = [121 507 115 27];

% Create Leyenda
app.Leyenda = uilabel(app.UIFigure);
app.Leyenda.BackgroundColor = [0.502 0.502 0.502];
app.Leyenda.FontName = 'Times New Roman';
app.Leyenda.FontSize = 18;
app.Leyenda.FontColor = [1 1 1];
app.Leyenda.Position = [1146 651 232 98];
app.Leyenda.Text = {' Io Fotonos de entrada'; ' It Fotonos
de salida'; ' t espesor del filtro'; ' op coeficiente de atenuación';
''};

% Create CoeficienteatenuacinLabel_5
app.CoficienteatenuacinLabel_5 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_5.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_5.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_5.FontSize = 20;
app.CoficienteatenuacinLabel_5.FontWeight = 'bold';
app.CoficienteatenuacinLabel_5.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_5.Position = [719 476 27 26];
app.CoficienteatenuacinLabel_5.Text = 'op';

% Create EFCoMedio
app.EFCoMedio = uieditfield(app.UIFigure, 'numeric');
app.EFCoMedio.ValueDisplayFormat = '%.3f';
app.EFCoMedio.ValueChangedFcn = createCallbackFcn(app,
@EFCoMedioValueChanged, true);
app.EFCoMedio.HorizontalAlignment = 'left';
app.EFCoMedio.FontName = 'Times New Roman';
app.EFCoMedio.FontSize = 20;
app.EFCoMedio.Position = [754 476 115 26];

% Create CoeficienteatenuacinLabel_6
app.CoficienteatenuacinLabel_6 = uilabel(app.UIFigure);

```

```

app.CoficienteatenuacinLabel_6.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_6.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_6.FontSize = 20;
app.CoficienteatenuacinLabel_6.FontWeight = 'bold';
app.CoficienteatenuacinLabel_6.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_6.Position = [725 510 21 24];
app.CoficienteatenuacinLabel_6.Text = 't';

% Create EFTmedio
app.EFTmedio = uieditfield(app.UIFigure, 'numeric');
app.EFTmedio.Limits = [1e-05 Inf];
app.EFTmedio.ValueDisplayFormat = '%.3f';
app.EFTmedio.ValueChangedFcn = createCallbackFcn(app,
@EFTmedioValueChanged, true);
app.EFTmedio.HorizontalAlignment = 'left';
app.EFTmedio.FontName = 'Times New Roman';
app.EFTmedio.FontSize = 20;
app.EFTmedio.Position = [754 507 115 27];
app.EFTmedio.Value = 1;

% Create CoficienteatenuacinLabel_7
app.CoficienteatenuacinLabel_7 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_7.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_7.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_7.FontSize = 20;
app.CoficienteatenuacinLabel_7.FontWeight = 'bold';
app.CoficienteatenuacinLabel_7.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_7.Position = [558 476 25 26];
app.CoficienteatenuacinLabel_7.Text = 'It';

% Create EFFotOut2
app.EFFotOut2 = uieditfield(app.UIFigure, 'numeric');
app.EFFotOut2.ValueDisplayFormat = '%.3f';
app.EFFotOut2.ValueChangedFcn = createCallbackFcn(app,
@EFFotOut2ValueChanged, true);
app.EFFotOut2.HorizontalAlignment = 'left';
app.EFFotOut2.FontName = 'Times New Roman';
app.EFFotOut2.FontSize = 20;
app.EFFotOut2.Position = [591 475 115 27];

% Create CoficienteatenuacinLabel_8
app.CoficienteatenuacinLabel_8 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_8.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_8.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_8.FontSize = 20;
app.CoficienteatenuacinLabel_8.FontWeight = 'bold';
app.CoficienteatenuacinLabel_8.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_8.Position = [560 507 25 26];
app.CoficienteatenuacinLabel_8.Text = 'Io';

% Create EFFotIn2
app.EFFotIn2 = uieditfield(app.UIFigure, 'numeric');
app.EFFotIn2.ValueDisplayFormat = '%.3f';
app.EFFotIn2.ValueChangedFcn = createCallbackFcn(app,
@EFFotIn2ValueChanged, true);
app.EFFotIn2.HorizontalAlignment = 'left';
app.EFFotIn2.FontName = 'Times New Roman';
app.EFFotIn2.FontSize = 20;

```

```

app.EFFotIn2.FontColor = [0.149 0.149 0.149];
app.EFFotIn2.Position = [591 507 115 27];

% Create CoeficienteatenuacinLabel_9
app.CoficienteatenuacinLabel_9 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_9.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_9.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_9.FontSize = 20;
app.CoficienteatenuacinLabel_9.FontWeight = 'bold';
app.CoficienteatenuacinLabel_9.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_9.Position = [1194 477 27 26];
app.CoficienteatenuacinLabel_9.Text = 'op';

% Create EFColed
app.EFColed = uieditfield(app.UIFigure, 'numeric');
app.EFColed.ValueDisplayFormat = '%.3f';
app.EFColed.ValueChangedFcn = createCallbackFcn(app,
@EFColedValueChanged, true);
app.EFColed.HorizontalAlignment = 'left';
app.EFColed.FontName = 'Times New Roman';
app.EFColed.FontSize = 20;
app.EFColed.Position = [1229 477 115 26];

% Create CoeficienteatenuacinLabel_10
app.CoficienteatenuacinLabel_10 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_10.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_10.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_10.FontSize = 20;
app.CoficienteatenuacinLabel_10.FontWeight = 'bold';
app.CoficienteatenuacinLabel_10.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_10.Position = [1200 511 21 24];
app.CoficienteatenuacinLabel_10.Text = 't';

% Create EFTled
app.EFTled = uieditfield(app.UIFigure, 'numeric');
app.EFTled.Limits = [1e-05 Inf];
app.EFTled.ValueDisplayFormat = '%.3f';
app.EFTled.ValueChangedFcn = createCallbackFcn(app,
@EFTledValueChanged, true);
app.EFTled.HorizontalAlignment = 'left';
app.EFTled.FontName = 'Times New Roman';
app.EFTled.FontSize = 20;
app.EFTled.Position = [1229 508 115 27];
app.EFTled.Value = 1;

% Create CoeficienteatenuacinLabel_11
app.CoficienteatenuacinLabel_11 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_11.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_11.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_11.FontSize = 20;
app.CoficienteatenuacinLabel_11.FontWeight = 'bold';
app.CoficienteatenuacinLabel_11.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_11.Position = [1033 477 25 26];
app.CoficienteatenuacinLabel_11.Text = 'It';

% Create EFFotOut3
app.EFFotOut3 = uieditfield(app.UIFigure, 'numeric');
app.EFFotOut3.ValueDisplayFormat = '%.3f';

```



```

app.EFFotOut3.ValueChangedFcn = createCallbackFcn(app,
@EFFotOut3ValueChanged, true);
app.EFFotOut3.HorizontalAlignment = 'left';
app.EFFotOut3.FontName = 'Times New Roman';
app.EFFotOut3.FontSize = 20;
app.EFFotOut3.Position = [1066 476 115 27];

% Create CoeficienteatenuacinLabel_12
app.CoficienteatenuacinLabel_12 = uilabel(app.UIFigure);
app.CoficienteatenuacinLabel_12.HorizontalAlignment = 'right';
app.CoficienteatenuacinLabel_12.FontName = 'Times New Roman';
app.CoficienteatenuacinLabel_12.FontSize = 20;
app.CoficienteatenuacinLabel_12.FontWeight = 'bold';
app.CoficienteatenuacinLabel_12.FontColor = [1 1 1];
app.CoficienteatenuacinLabel_12.Position = [1035 508 25 26];
app.CoficienteatenuacinLabel_12.Text = 'Io';

% Create EFFotIn3
app.EFFotIn3 = uieditfield(app.UIFigure, 'numeric');
app.EFFotIn3.ValueDisplayFormat = '%.3f';
app.EFFotIn3.ValueChangedFcn = createCallbackFcn(app,
@EFFotIn3ValueChanged, true);
app.EFFotIn3.HorizontalAlignment = 'left';
app.EFFotIn3.FontName = 'Times New Roman';
app.EFFotIn3.FontSize = 20;
app.EFFotIn3.FontColor = [0.149 0.149 0.149];
app.EFFotIn3.Position = [1066 508 115 27];

% Create Titulo
app.Titulo = uilabel(app.UIFigure);
app.Titulo.HorizontalAlignment = 'center';
app.Titulo.FontName = 'Times New Roman';
app.Titulo.FontSize = 20;
app.Titulo.FontWeight = 'bold';
app.Titulo.Position = [388 663 750 65];
app.Titulo.Text = {'SISTEMA DE ADQUISICIÓN Y PROCESAMIENTO DE DATOS
PARA MEDICIÓN '; 'DE ESPECTROS DE LUZ Y CÁLCULO DE COEFICIENTE DE ATENUACIÓN'};

% Create RealizadorporJohaoSnchezMarcoSalazarLabel
app.RealizadorporJohaoSnchezMarcoSalazarLabel =
uilabel(app.UIFigure);
app.RealizadorporJohaoSnchezMarcoSalazarLabel.FontName = 'Times New
Roman';
app.RealizadorporJohaoSnchezMarcoSalazarLabel.FontSize = 16;
app.RealizadorporJohaoSnchezMarcoSalazarLabel.Position = [1176 591
202 39];
app.RealizadorporJohaoSnchezMarcoSalazarLabel.Text = {'Realizado
por: Johao Sánchez'; 'Marco Salazar'};

% Create LEDSwitchLabel
app.LEDSwitchLabel = uilabel(app.UIFigure);
app.LEDSwitchLabel.HorizontalAlignment = 'center';
app.LEDSwitchLabel.FontName = 'Times New Roman';
app.LEDSwitchLabel.Position = [474 584 28 22];
app.LEDSwitchLabel.Text = 'LED';

% Create LEDSwitch
app.LEDSwitch = uiswitch(app.UIFigure, 'slider');

```

```

    app.LEDSwitch.Items = {'On', 'Off'};
    app.LEDSwitch.ValueChangedFcn = createCallbackFcn(app,
@LEDSwitchValueChanged, true);
    app.LEDSwitch.FontName = 'Times New Roman';
    app.LEDSwitch.Position = [455 602 67 30];
    app.LEDSwitch.Value = 'On';

    % Create CONECTARButton
    app.CONECTARButton = uibutton(app.UIFigure, 'push');
    app.CONECTARButton.ButtonPushedFcn = createCallbackFcn(app,
@CONECTARButtonPushed, true);
    app.CONECTARButton.FontName = 'Times New Roman';
    app.CONECTARButton.FontSize = 17;
    app.CONECTARButton.FontWeight = 'bold';
    app.CONECTARButton.Position = [687 588 144 48];
    app.CONECTARButton.Text = 'CONECTAR';

    % Create DESCONECTARButton
    app.DESCONECTARButton = uibutton(app.UIFigure, 'push');
    app.DESCONECTARButton.ButtonPushedFcn = createCallbackFcn(app,
@DESCONECTARButtonPushed, true);
    app.DESCONECTARButton.FontName = 'Times New Roman';
    app.DESCONECTARButton.FontSize = 17;
    app.DESCONECTARButton.FontWeight = 'bold';
    app.DESCONECTARButton.Position = [841 588 144 48];
    app.DESCONECTARButton.Text = 'DESCONECTAR';

    % Create Label
    app.Label = uilabel(app.UIFigure);
    app.Label.HorizontalAlignment = 'center';
    app.Label.Position = [602 551 25 22];
    app.Label.Text = '%';

    % Create Knob
    app.Knob = uiknob(app.UIFigure, 'continuous');
    app.Knob.ValueChangedFcn = createCallbackFcn(app,
@KnobValueChanged, true);
    app.Knob.Position = [585 585 60 60];

    % Create Titulo1
    app.Titulo1 = uieditfield(app.UIFigure, 'text');
    app.Titulo1.ValueChangingFcn = createCallbackFcn(app,
@Titulo1ValueChanging, true);
    app.Titulo1.HorizontalAlignment = 'center';
    app.Titulo1.FontName = 'Times New Roman';
    app.Titulo1.FontSize = 20;
    app.Titulo1.Position = [160 387 175 30];

    % Create Titulo2
    app.Titulo2 = uieditfield(app.UIFigure, 'text');
    app.Titulo2.ValueChangingFcn = createCallbackFcn(app,
@Titulo2ValueChanging, true);
    app.Titulo2.HorizontalAlignment = 'center';
    app.Titulo2.FontName = 'Times New Roman';
    app.Titulo2.FontSize = 20;
    app.Titulo2.Position = [630 387 175 30];

    % Create Titulo3

```

```

        app.Titulo3 = uicontrol(app.UIFigure, 'text');
        app.Titulo3.ValueChangingFcn = createCallbackFcn(app,
@Titulo3ValueChanging, true);
        app.Titulo3.HorizontalAlignment = 'center';
        app.Titulo3.FontName = 'Times New Roman';
        app.Titulo3.FontSize = 20;
        app.Titulo3.Position = [1105 388 175 30];

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = APP_Espectrometro

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        % Execute the startup function
        runStartupFcn(app, @startupFcn)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
end
end

```