Monitoreo y control de cultivo hidropónico en entorno de cúpula geodésica

Autores
Sergio Andrés Arenas Santamaría
Cod: 20191573040
saarenass@udistrital.edu.co

Santiago Ballen Maldonado
Cod: 20191573026
sballenm@udistrital.edu.co

Modalidad
Pasantía

DIRECTOR
Ing. Cesar Augusto Coronel Segrera
Ing. Frank Nixon Giraldo Ramos

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS FACULTAD TECNOLÓGICA
TECNOLOGÍA EN ELECTRÓNICA BOGOTÁ D.C
Octubre 2023

HOJA DE ACEPTACIÓN

“Monitoreo y control de cultivo hidropónico en entorno de cúpula geodésica”

Observaciones.



Firma del director del proyecto

Ing. Frank Nixon Giraldo Ramos

Cesar Augusto Coronel S

Firma del director del proyecto

Ing. Cesar Augusto Coronel Segrera

Firma del jurado

Fecha de presentación: Octubre de 2023

TABLA DE CONTENIDO

1. Descripción de la Propuesta....7
    1.1. Resumen....7
    1.2. Planteamiento del problema....8
2. Objetivos....9
    2.1. Objetivo general....9
    2.2. Objetivos específicos....9
3. Marcos de referencia....10
    3.1. Marco teórico....10
        3.1.1. Cultivo Hidropónico....10
        3.1.2. Sistema de producción en sustrato....10
        3.1.3. Raspberry Pi Zero W....11
        3.1.4. Sensor temperatura DS18B20....12
        3.1.5. Sensor Temperatura y humedad ambiente DHT22....12
        3.1.6. Sensor SEN0244 TDS....13
        3.1.7. Sensor SEN0189 TSS....13
        3.1.8. Módulo ADC ADS1115....14
        3.1.9. Sensor calidad de aire SDS011....14
        3.1.10. Electroválvula....15
        3.1.11. Pantalla 5 inch HDMI LCD de raspberry....15
        3.1.12. Qt Designer....16
        3.1.13. Lenguaje de programación Python....17
    3.2. Estado del arte....19
        3.2.1. Sistema de adquisición de datos climáticos para la docencia....19
        3.2.2. Sistema de Adquisición de datos para un cultivo en Sutamarchán....19
        3.2.3. Sistemas de Riego Inteligente para Agricultura de Precisión - AREThoU5A..19
4. Desarrollo del sistema de adquisición de datos....20
    4.1. Configuración y adquisición de datos del sensor DS18B20....20
    4.2. Configuración y adquisición de datos del sensor DHT22....23
    4.3. Configuración y adquisición de datos del sensor SDS011....23
    4.4. Configuración y adquisición de datos sensores SEN0244 y SEN0189....24
5. Desarrollo de interfaz gráfica....24
    5.1. Planteamiento de visualización....25
    5.2. Desarrollo de visualización....26
    5.3. Desarrollo de código....28
        5.3.1. Programa Principal 'Interfaz.py'....29
        5.3.2. Interfaz de Usuario 'Interfaz.ui'....30
        5.3.3. Gestión de Datos ‘SensoresData.py’ y ‘SensoresPrint.py’....31

5.3.4. Visualización Gráfica ‘Graficos.py’ 33
5.3.5. Comunicación Externa ‘Iconos’ 34
5.4. Visualización de interfaz gráfica 35
5.4.1. Página 'Interfaz’ 36
5.4.2. Página ‘Control’ 37
5.4.3. Página ‘Registro’ 37
5.4.4. Página ‘Creadores’ 38
6. Desarrollo de cúpula geodésica 39
6.1. Calculo de construcción de cúpula geodésica 39
6.2. Uniones de construcción de cúpula geodésica 41
6.3. Cortes de construcción de cúpula geodésica 42
6.4. Ensamble de construcción de cúpula geodésica 44
7. Ensamble 47
7.1. Diseño PCB 47
7.2. Revelado y montaje de PCB 49
8. Conclusiones 56
9. Bibliografía 57

LISTA DE FIGURAS

Figura 1: Cultivo Hidroponico de Lechuga..... 10
Figura 2: Cultivo Hidropónico en sustrato..... 11
Figura 3: Raspberry Pi Zero W..... 11
Figura 4: DS18B20..... 12
Figura 5: DHT22..... 12
Figura 6: SEN0244..... 13
Figura 7: SEN0189..... 13
Figura 8: ADS1115..... 14
Figura 9: SDS011..... 15
Figura 10: Electroválvula..... 15
Figura 11 : Pantalla 5 inch HDMI LCD de raspberry..... 16
Figura 12: Entorno de desarrollo Qt Designer..... 17
Figura 13 : Entorno de desarrollo VSC - Python..... 18
Figura 14: Instalación sistema AREThoU5A 20
Figura 15: Configuración de interfaces de Raspberry Pi 20
Figura 16: Comandos ejecutados para comprobar sensores DS18B20..... 21
Figura 17: Comando para 28-3c80f649bc0a..... 21
Figura 18: datos de 28-3c80f649bc0a..... 21
Figura 19: Prueba de lectura de todos los sensores..... 22
Figura 20: Asignación de las direcciones a variables locales..... 22
Figura 21: Lectura de sensores DS18B20..... 22
Figura 22: Definición sensor y pin usado para DHT22 23
Figura 23: Lectura DHT22 23
Figura 24: Asignación del puerto 23
Figura 25: Lectura SDS011 23
Figura 26: Lectura ADS1115 24
Figura 27: Dashboard para visualización de datos..... 25
Figura 28: Estructura base de Interfaz gráfica de control de cultivo hidropónico..... 26
Figura 29: Interfaz gráfica de control de cultivo hidropónico versión 1..... 27
Figura 30: Interfaz gráfica de control de cultivo hidropónico versión 2..... 28
Figura 31: Estructura de control visual del proyecto de cultivo hidropónico..... 29
Figura 32:Funciones y clases del programa principal 'Interfaz.py'..... 30
Figura 33: Fragmento de Interfaz de Usuario 'Interfaz.ui'..... 31
Figura 34: Fragmento de Gestión de Datos 'SensoresData.py'..... 33
Figura 35: Fragmento de Gestión de Datos 'SensoresPrint.py'..... 33
Figura 36: Fragmento de Visualización Gráfica ‘Graficos.py’..... 34
Figura 37: Interfaz gráfica en la página Registro..... 35
Figura 38: Página de Interfaz de visualización de control del cultivo hidropónico..... 36
Figura 39: Página de Control de visualización de control del cultivo hidropónico..... 37
Figura 40: Página de Registro de visualización de control del cultivo hidropónico..... 38
Figura 41: Página de Creadores de visualización de control del cultivo hidropónico..... 39
Figura 42: Depósito de madera a base de estibas..... 40
Figura 43: Tipo de conexión “Good Karma" o "Wood Karma" en cúpula geodésica..... 41

Figura 44: Resultados de cúpula geodésica con características mencionadas..... 42
Figura 45: Cortes para tablas tipo A de vista lateral..... 43
Figura 46: Cortes para tablas tipo A de vista frontal..... 43
Figura 47: Construcción de triángulos 1 base de estructura..... 45
Figura 48: Construcción de triángulos base 2 de estructura..... 45
Figura 49: Unión de triángulos base para construcción de cúpula geodésica..... 46
Figura 50: Cúpula geodésica de vista aérea..... 47
Figura 51: Cúpula geodésica de vista lateral..... 47
Figura 52: Diseño de PCB en KiCad..... 48
Figura 53: Impresión capa Bottom de la PCB..... 48
Figura 54: Impresión capa Top de la PCB..... 49
Figura 55: Impresión capa Bottom de la PCB..... 49
Figura 56: Proceso de lavado en ácido..... 50
Figura 57: PCB durante el proceso de lavado en ácido..... 50
Figura 58: PCB después del proceso de lavado en ácido..... 51
Figura 59: PCB con máscara de soldadura..... 51
Figura 60: Laminación de máscara de soldadura..... 52
Figura 62: Unidad de exposición a vacío..... 53
Figura 53: Montaje con PCB vista de Buttom..... 54
Figura 64: Montaje con PCB vista Top..... 55

1. Descripción de la Propuesta

1.1. Resumen

La producción hidropónica es un método de cultivo que utiliza soluciones de agua y nutrientes en lugar de suelo para cultivar plantas, lo que permite una mayor eficiencia y sostenibilidad en la producción agrícola. Uno de los principales desafíos de la producción hidropónica es garantizar un suministro adecuado de agua y nutrientes a las plantas, lo que requiere un monitoreo y control preciso del clima, la iluminación y otros factores ambientales.

Mediante sensores transductores se tomarán los datos de las variables pH, temperatura y humedad en suelo y ambiente. Estos datos serán adquiridos y procesados por medio de una Raspberry Pi Zero W y posteriormente serán visualizados en una interfaz de Python. En Colombia, la implementación de estas tecnologías puede tener un impacto significativo en la producción de alimentos, al aumentar la eficiencia y la sostenibilidad de la agricultura, y al fomentar la creación de nuevas empresas y oportunidades de trabajo en el sector tecnológico. La investigación y el desarrollo de tecnologías avanzadas para la producción hidropónica pueden llevar a una mayor producción de alimentos de alta calidad y a precios asequibles para los consumidores, y también pueden tener un impacto positivo en el medio ambiente al reducir la necesidad de tierras y recursos naturales para la producción agrícola. En conclusión, el desarrollo de tecnologías avanzadas para la producción hidropónica tiene un gran potencial para mejorar la producción de alimentos y fomentar el crecimiento económico y la innovación en Colombia.

Este proyecto se realizó en las instalaciones del Servicio Nacional de Aprendizaje - SENA en la sede TecnoParque SENA, con instrumentos y dispositivos electrónicos que proporciona la institución, así como asesorías y consejos por parte de los docentes directores del proyecto.

Palabras clave: Hidroponia, Raspberry Pi, adquisición de datos.

1.2. Planteamiento del problema

Los cultivos hidropónicos son una alternativa cada vez más popular para la producción de alimentos en áreas urbanas y periurbanas, ya que ofrecen un método eficiente y sostenible para cultivar plantas sin la necesidad de suelo y con un menor consumo de agua. Sin embargo, el éxito de los cultivos hidropónicos depende de una serie de factores críticos, como la calidad del agua y de los nutrientes, el pH y la temperatura, que deben ser monitoreados y controlados cuidadosamente para garantizar un crecimiento óptimo de las plantas[1]. La automatización y control de los sistemas de cultivo hidropónico puede ser una solución eficaz para garantizar que estas variables críticas se mantengan en los niveles óptimos para el crecimiento de las plantas, por lo cual el docente Ing. Cesar Augusto Coronel Segrera tomó la decisión de involucrar estudiantes en la modalidad de pasantía (cumpliendo con un mínimo de 192 horas según lo establecido en el acuerdo 038 de 2015 de consejo Académico de la Universidad Distrital) dentro del Servicio Nacional de Aprendizaje - SENA, para desarrollar esta idea.

2. Objetivos

2.1. Objetivo general

- Desarrollar sistema de monitoreo y control de un cultivo hidropónico en cúpula geodésica

2.2. Objetivos específicos

- Diseñar un sistema de adquisición de datos para las variables de temperatura de ambiente y sustrato, humedad de ambiente y en sustrato.
- Procesar las variables obtenidas por los sensores.
- Diseñar una interfaz en el entorno de Python que permita visualizar los datos de las variables obtenidas y permita el control de un sistema de goteo.
- Diseñar un sistema de control para goteo de solución nutritiva.
- Desarrollar y construir cúpula geodésica para el entorno del cultivo hidropónico.

3. Marcos de referencia

3.1. Marco teórico

3.1.1. Cultivo Hidropónico

Los cultivos hidropónicos se diferencian de los cultivos tradicionales en que se elimina completamente el uso de tierra y se sustituye por un sustrato esteril o por una solución nutritiva para que la planta adquiera los minerales necesarios para su crecimiento. Consiste en que las raíces están suspendidas sobre la solución, de esta manera la raíz de la planta absorberá la solución con nutrientes, evitando las posibles enfermedades o plagas provenientes de la tierra.[2][3]



Figura 1: Cultivo Hidroponico de Lechuga.

Fuente: [3]

Esto se convierte en una manera tecnificada de hacer agricultura, creando incluso la posibilidad de tener un huerto hidropónico en casa. Se pueden realizar de manera vertical o horizontal; en caso de usar solución de nutrientes, este se puede implementar de manera estática o circulante; en caso de usar sustrato éste debe ser puro, la más común es fibra de coco o arena.[2]

3.1.2. Sistema de producción en sustrato

Este sistema utiliza un material sólido distinto de la tierra, este permite el desarrollo del sistema radicular de la planta y puede intervenir en la nutrición de la planta. El sustrato brinda soporte a la planta, mantiene la humedad, drenaje, ventilación y una facilidad en la

absorción de nutrientes. Para la elección del sustrato es importante tener en cuenta su PH, su densidad, espacio poroso total, capacidad de aireación, entre otros.[4]



Figura 2: Cultivo Hidropónico en sustrato.

Fuente: [4]

3.1.3. Raspberry Pi Zero W

Raspberry Pi es un sistema embebido diseñado en primer lugar para usos académicos y de aprendizaje. La Raspberry Pi Zero W cuenta con un procesador single-core de 1, 512 MB RAM, 40 pines GPIO, 802.11 b/g/n wireless LAN, Bluetooth 4.1 y adaptador Bluetooth Low Energy (BLE), puerto micro USB On-The-Go (OTG), puerto mini HDMI, puerto CSI para camera, puerto DSI de display para la conexión de una pantalla táctil, puerto microUSB para cargar el sistema operativo y almacenamiento de datos, y, una entrada micro USB de alimentación con fuente mayor a 250 mA [5].

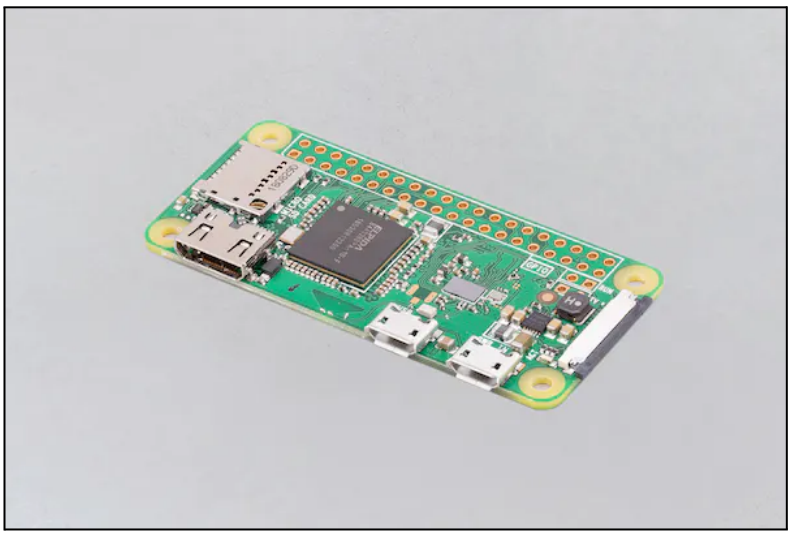


Figura 3: Raspberry Pi Zero W.
Fuente: [5]

3.1.4. Sensor temperatura DS18B20

Sensor de temperatura para líquidos, su encapsulado puede ser TO-92, SO y µSOP, tiene tres pines, VDD, GND y DATA. El protocolo de comunicación es 1-Wire y se hace por medio del pin DATA. Puede medir temperatura entre -55°C y 125°C. La resolución de los datos va desde 9 bits hasta 12 bits, por defecto se utilizan 12 bits.[6]

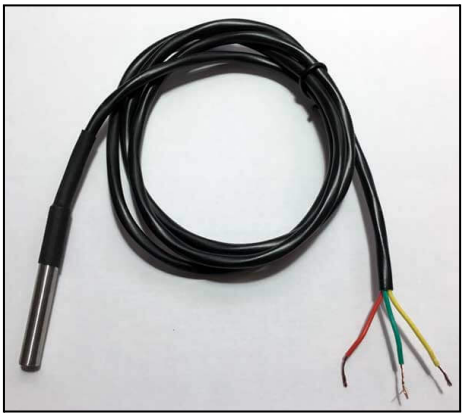


Figura 4: DS18B20.
Fuente: [6]

3.1.5. Sensor Temperatura y humedad ambiente DHT22

El DHT22 es un sensor de temperatura y humedad de ambiente, tiene 3 pines: GND, VCC y DATOS. Su alimentación es de 3.3V a 6V, consumo de 25mA, la señal de salida es digital, tiene un rango de temperatura de -40°C a 125°C, resolución de 8 bits, variación de 0.5°C. La humedad que puede medir va desde 0%RH a 100%RH, con una presión de 2-5%RH. Frecuencia de muestreo de 2 muestras por segundo.[7]

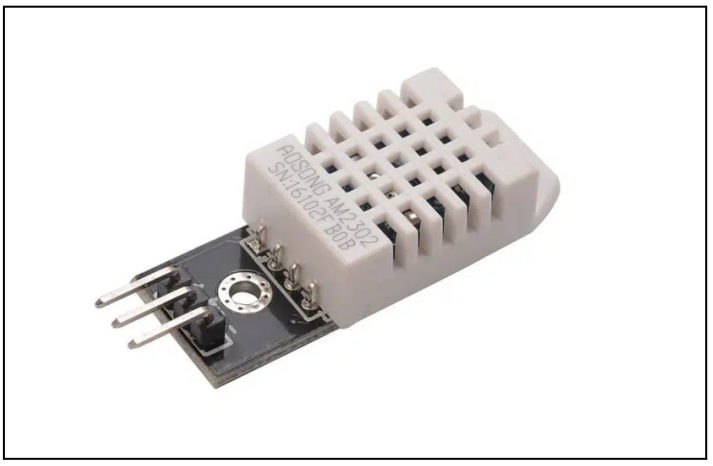


Figura 5: DHT22.
Fuente: [7]

3.1.6. Sensor SEN0244 TDS

El sensor análogo SEN0244 es un sensor de Total de Sólidos Disueltos, TDS por sus siglas en inglés. Funciona con un voltaje de 3.3V a 5.5V, con un voltaje de salida de 0 a 2.3V. Tiene un rango de medida de TDS de 0 a 1000 ppm (partes por millón).[8]

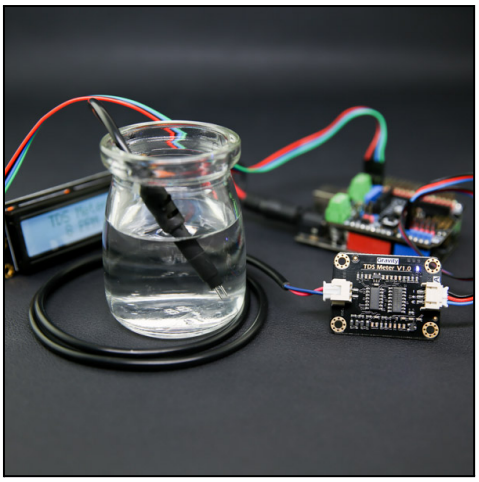


Figura 6: SEN0244.
Fuente: [8]

3.1.7. Sensor SEN0189 TSS

El sensor SEN0189 es un sensor de Total de Sólidos Suspendidos, TSS por sus siglas en inglés. Funciona con un voltaje de 5V, capaz de entregar datos análogos y digitales; la lectura análoga va de 0 a 4.5V mientras a la lectura digital solo High/Low de acuerdo al parámetro que se le ajuste.[9]

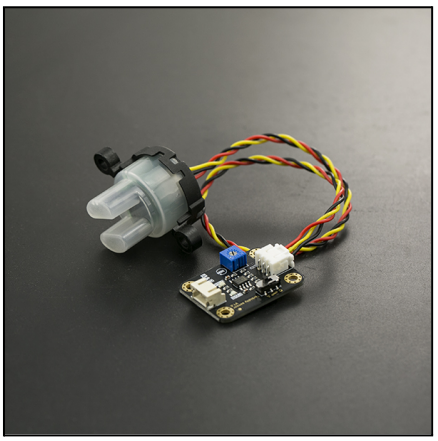


Figura 7: SEN0189.
Fuente: [9]

3.1.8. Módulo ADC ADS1115

El ADS1115 es un módulo conversor analógico a digital (ADC) que opera en un rango de voltaje de 2V a 5.5V, con un bajo consumo de 150uA. Ofrece una alta resolución de 16 bits y admite hasta 4 canales de entrada. Utiliza comunicación I2C, siendo ideal para aplicaciones que requieren precisión en la conversión de señales analógicas y monitoreo de múltiples fuentes de datos. [10]

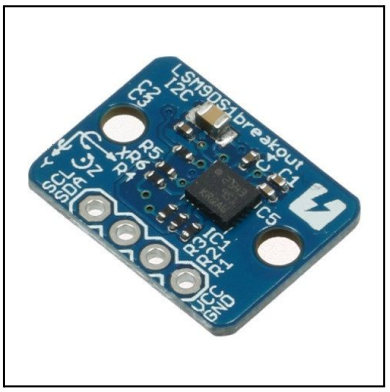


Figura 8: ADS1115.
Fuente: [10]

3.1.9. Sensor calidad de aire SDS011

El SDS011 es un sensor especializado en la medición de partículas de tamaño PM10 y PM2.5 presentes en el aire ambiente. Su funcionamiento se basa en una fuente de alimentación eléctrica de 5 voltios (V), que proporciona la energía necesaria para su operación. Además, este sensor está diseñado con una interfaz de comunicación UART (Universal Asynchronous Receiver-Transmitter), un estándar de comunicación serial ampliamente utilizado en aplicaciones electrónicas, lo que facilita la integración del SDS011 en sistemas y dispositivos que requieran medir y monitorear la concentración de partículas suspendidas en el aire con gran precisión y sensibilidad. [11]



Figura 9: SDS011.
Fuente: [11]

3.1.10. Electroválvula

Una electroválvula solenoide es un dispositivo electromecánico utilizado para controlar el flujo de fluidos, como líquidos o gases, en sistemas automatizados. Consiste en una bobina de alambre enrollada alrededor de un núcleo magnético y un conjunto de componentes móviles, como un émbolo o una válvula, que se activa cuando se aplica una corriente eléctrica a la bobina, opera con una fuente de alimentación eléctrica de 12 voltios (V). []

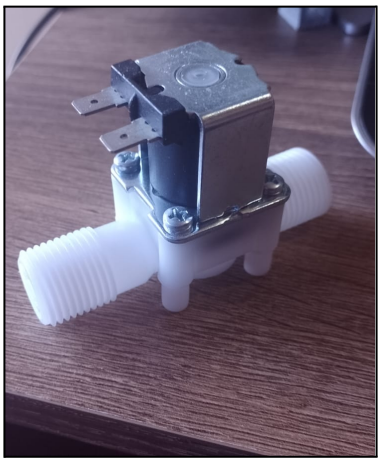


Figura 10: Electroválvula.
Fuente: Propia

3.1.11. Pantalla 5 inch HDMI LCD de raspberry.

La pantalla en cuestión presenta una resolución de 800 x 480 píxeles, lo que la convierte en una elección óptima para la visualización en tiempo real de variables relacionadas con un domo geodésico. Estratégicamente ubicada a un lado del área de cultivo, esta disposición permite al usuario del domo mantener acceso constante a los valores mientras manipula los cultivos.

En condiciones normales de funcionamiento, la pantalla LCD HDMI de 5 pulgadas requiere una alimentación de 5V, con un consumo de corriente de aproximadamente 235 mA cuando la retroiluminación está encendida y 45 mA cuando está apagada. Además, es capaz de operar en un rango de temperatura de 0°C a 70°C, lo que la hace adecuada para una variedad de entornos.

Una característica destacada de esta pantalla es su compatibilidad con Raspberry Pi, ya que cuenta con los conectores necesarios en su parte posterior para la instalación de la tarjeta de desarrollo, sin requerir cables adicionales para la transmisión de datos. Además, ofrece una iluminación de calidad que no causa fatiga visual, y su pantalla táctil brinda al usuario la capacidad de navegar por las pestañas de manera intuitiva y sin complicaciones.

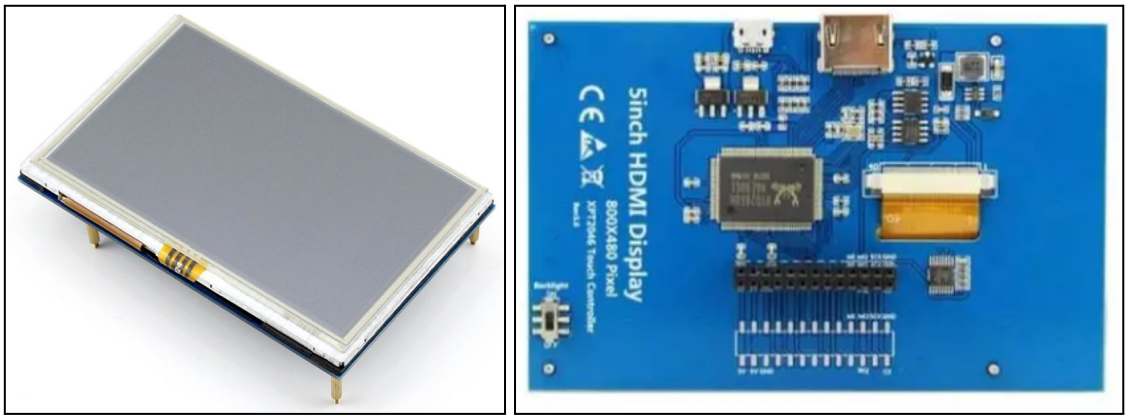


Figura 11 : Pantalla 5 inch HDMI LCD de raspberry
Fuente: [14]

3.1.12. Qt Designer.

Qt Creator representa un entorno de desarrollo integrado (IDE) altamente versátil, diseñado para ser compatible con múltiples plataformas de software. Este potente IDE se ha construido mediante el uso de los lenguajes de programación C++, JavaScript y QML. Fue concebido originalmente por Trolltech y se incorporó como una parte integral del conjunto de herramientas de desarrollo denominado SDK. Su propósito principal es simplificar el proceso de creación de aplicaciones con interfaces gráficas de usuario (GUI) aprovechando las potentes bibliotecas de Qt. Es importante destacar que Qt Creator goza de una sólida compatibilidad con una amplia variedad de sistemas operativos, lo que lo convierte en una elección eficaz y versátil para desarrolladores de diferentes entornos.

Por otro lado, Qt Designer, una herramienta complementaria, desempeña un papel fundamental en la creación de interfaces gráficas de usuario (GUI) con Qt. Esta herramienta permite a los desarrolladores componer y personalizar ventanas o cuadros

de diálogo de manera intuitiva, siguiendo el enfoque de 'lo que ve es lo que obtiene' (WYSIWYG). Asimismo, posibilita la prueba de estas interfaces en distintos estilos y resoluciones de pantalla.

Los widgets y formularios generados con Qt Designer se integran de manera perfecta con el código subyacente. Esto se logra a través del mecanismo de señales y ranuras de Qt, que permite asignar fácilmente comportamientos a los elementos gráficos. Cabe destacar que todas las propiedades configuradas en Qt Designer pueden modificarse dinámicamente desde el código, lo que brinda una gran flexibilidad. Además, Qt Designer ofrece características avanzadas como la promoción de widgets y la capacidad de incorporar componentes personalizados, lo que permite a los desarrolladores utilizar sus propios elementos en conjunción con esta potente herramienta de diseño.

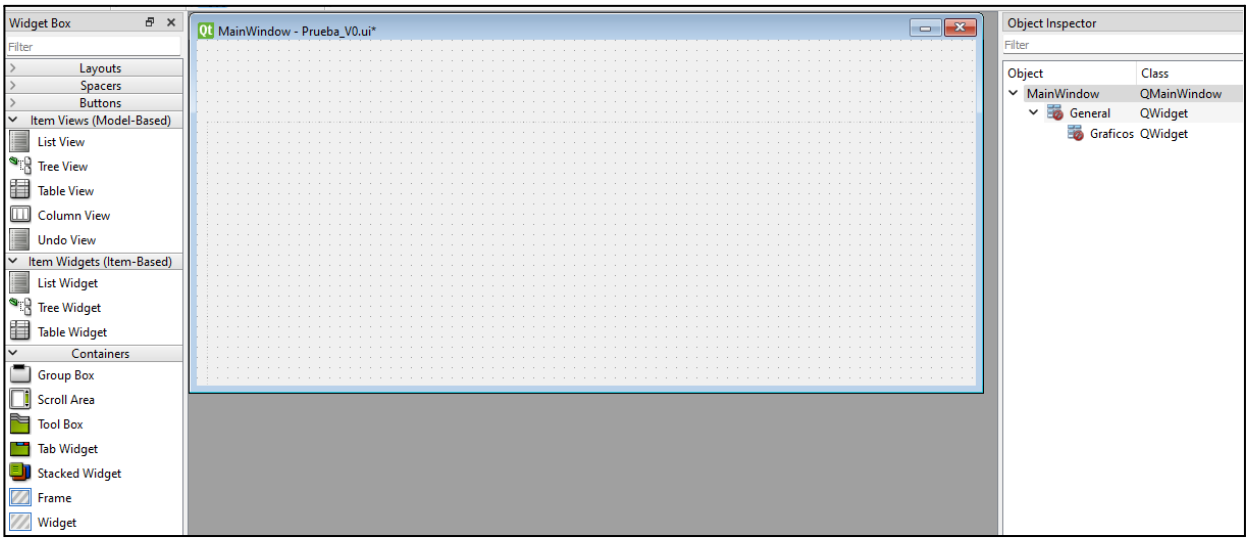


Figura 12: Entorno de desarrollo Qt Designer

Fuente: Qt Designer [16]

3.1.13. Lenguaje de programación Python.

El lenguaje de programación Python, ampliamente aclamado por su elegancia y legibilidad, es una herramienta excepcional en el campo de la programación. Su sintaxis clara y concisa lo hace atractivo tanto para principiantes como para expertos en el desarrollo de software.

Una de las cualidades sobresalientes de Python es su compromiso con la legibilidad del

código, lo que simplifica la colaboración y el mantenimiento a largo plazo. La filosofía de diseño del lenguaje, encapsulada en el Zen de Python, enfatiza la importancia de la claridad y la simplicidad en la programación.

En el ámbito de la visualización de datos, la biblioteca Matplotlib se alza como un pilar esencial. Matplotlib permite crear gráficos y visualizaciones de datos de manera eficaz y personalizable. Con una sintaxis intuitiva, brinda un control preciso sobre la apariencia de los gráficos, lo que facilita la creación de representaciones visuales efectivas de datos complejos.

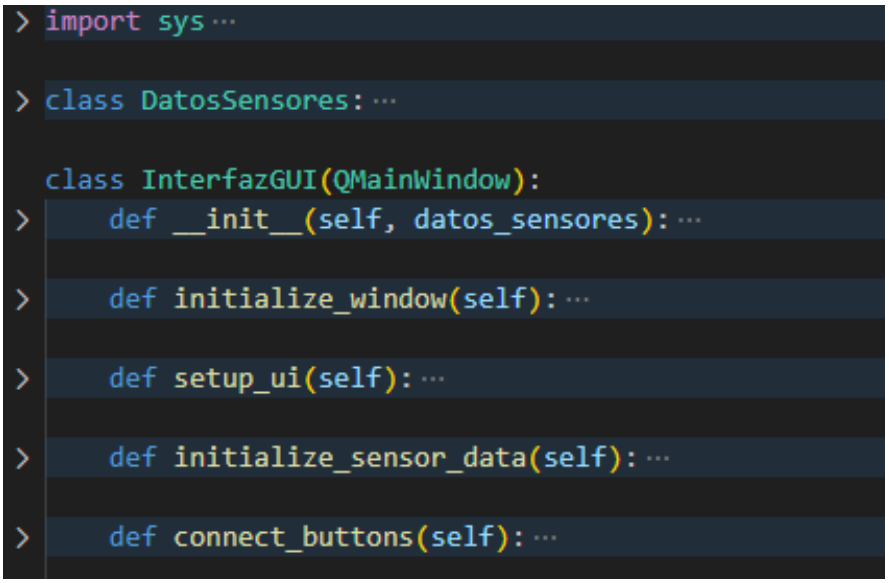


Figura 13 : Entorno de desarrollo VSC - Python

Fuente: Propio

3.2. Estado del arte.

3.2.1. Sistema de adquisición de datos climáticos para la docencia

Este es un sistema docente de supervisión del clima en un invernadero realizado en la Universidad Nacional en el año 2010, utiliza una DAQ para la adquisición y procesamiento de datos que posteriormente se visualiza en LabVIEW. Se utilizó un sensor de temperatura y humedad relativa HMP45C, también un anemómetro A100R para la velocidad del aire, Radiación Neta en el interior por medio de un pirradiómetro, por último para la radiación global solar del interior se utilizó una fotopila. [12]

3.2.2. Sistema de Adquisición de datos para un cultivo en Sutamarchán

Este sistema fue desarrollado en la Universidad Santo Tomás para adquirir información sobre un cultivo de tomate en el municipio de Sutamarchan. Utilizando los siguientes sensores: sensor de humedad 808H5V5, sensor de temperatura MCP9700A, sensor de humedad en suelo WATERMARK, sensor de presión y altura MPX4115A, sensor de radiación solar SQ-110. El envío de los datos se realiza por medio de tecnología ZigBee y el almacenamiento de los datos por medio de MySQL. [**13**]

3.2.3. Sistemas de Riego Inteligente para Agricultura de Precisión - AREThoU5A

El objetivo de este proyecto es desarrollar sistemas de riego inteligente que utilicen tecnologías avanzadas para mejorar la eficiencia del riego en la agricultura de precisión. Estos sistemas buscan proporcionar la cantidad óptima de agua a los cultivos en función de las necesidades reales y las condiciones ambientales, reduciendo así el consumo de agua y los costos de producción agrícola. Se utilizan sensores de humedad y temperatura del suelo para obtener datos que son fundamentales para que el algoritmo usado pueda tomar decisiones sobre cuándo y cuánto regar.[14]



Figura 14: Instalación sistema AREThoU5A .
Fuente: [14]

4. Desarrollo del sistema de adquisición de datos

Para cumplir con el sistema de adquisición era necesario realizar un código de Python de manera que se obtuvieran los datos de los sensores y fuera capaz de enviarlos hacia la interfaz, para esto fueron necesarias varias configuraciones.

4.1. Configuración y adquisición de datos del sensor DS18B20

Se comienza activando la interfaz 1-WIRE que incorpora la Raspberry, para eso nos vamos a Preferencias -> Configuración de Raspberry Pi -> Interfaces. Y se activa la opción 1-WIRE. Posteriormente se reinicia la rasp:

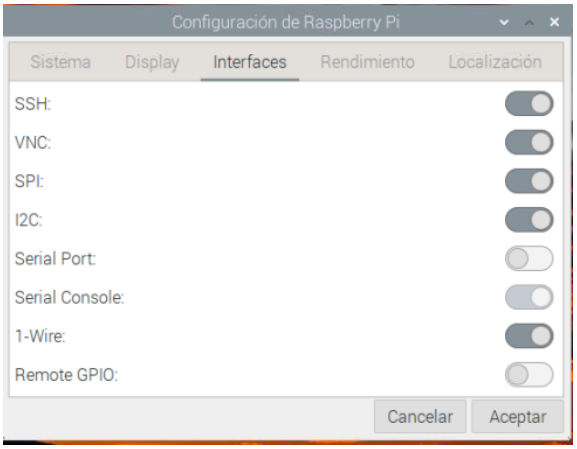


Figura 15: Configuración de interfaces de Raspberry Pi .
Fuente: Propia

Lo segundo que se debe de hacer es probar que la rasp sea capaz de detectar los sensores, para esto primero se conectan los sensor al pin GPIO4 al cual

corresponde este tipo de comunicación 1-Wire, luego se ejecutan los siguiente comandos en la terminal:

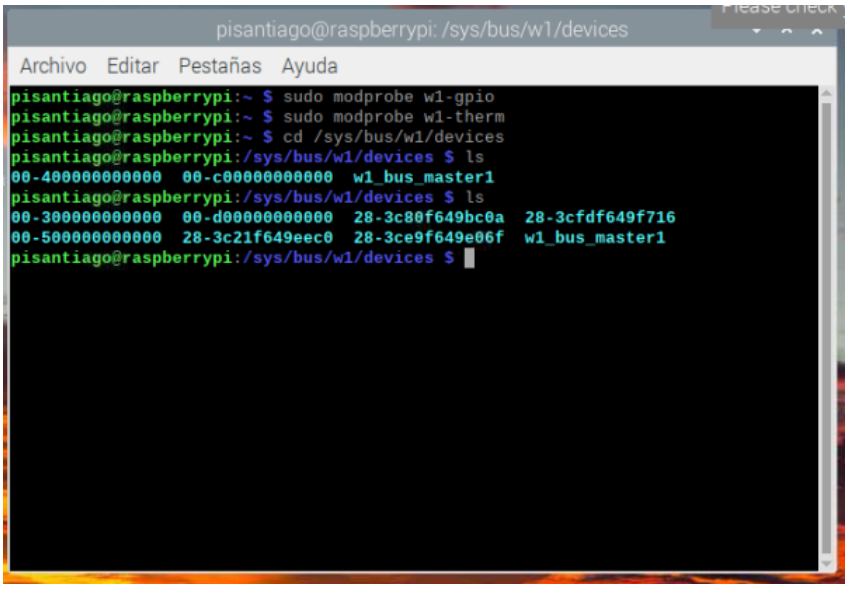


Figura 16: Comandos ejecutados para comprobar sensores DS18B20.
Fuente: Propia

Se puede comprobar que la Raspberry está detectando los sensores al obtener las direcciones de las ROM de cada sensor, estas direcciones son los números que están de forma 28-XXXXXXXXXXX. Para ver la temperatura que esta tomando cada sensor, se entra al directorio del sensor:



Figura 17: Comando para 28-3c80f649bc0a.
Fuente: Propia

Y se dijita el comando: “cat w1_slave”, que se mostrará la siguiente información:

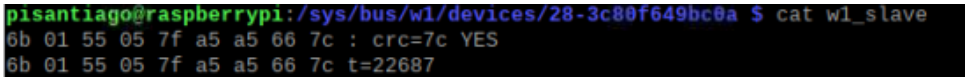


Figura 18: datos de 28-3c80f649bc0a.
Fuente: Propia

El YES en la primera linea indica que fue correcta la validación CRC (datos validos). Y el numero seguido de “t=” es la temperatura que está leyendo, en este caso la temperatura sería de 22.687 °C. Se comprueban todos los sensores:

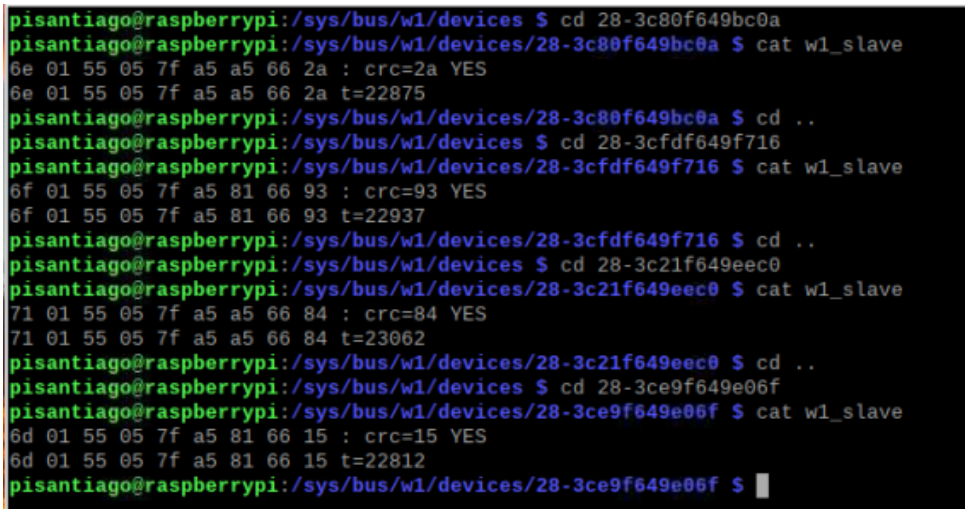


Figura 19: Prueba de lectura de todos los sensores.
Fuente: Propia

Posteriormente se desarrolla un código que emule este mismo proceso, primero es necesario asignar las direcciones a variables que luego se usarán para leer las temperaturas.

```
dir_temp1 = '28-3c21f649eec0'
dir_temp2 = '28-3c80f649bc0a'
dir_temp3 = '28-3ce9f649e06f'
dir_temp4 = '28-3cfdf649f716'
direcciones = [dir_temp1,dir_temp2,dir_temp3,dir_temp4]
```

Figura 20: Asignación de las direcciones a variables locales.
Fuente: Propia

Luego se emulá el proceso anterior dentro del código, utilizando sentencias Try - Except, esto para el caso de que hayan errores en la lectura se retorna un 0.

```
def leer_ds18b20(self):
    temp = ""
    for direccion_sensor in direcciones:
        try:
            with open('/sys/bus/w1/devices/{}/w1_slave'.format(direccion_sensor), 'r') as f:
                lines = f.readlines()
                if lines[0].strip()[-3:] == 'YES':
                    equals_pos = lines[1].find('t=')
                    if equals_pos != -1:
                        temp_string = lines[1][equals_pos+2:]
                        temperatura = float(temp_string) / 1000.0
                # Si ocurre algún error en la lectura del sensor, devolvemos 0
                else:
                    print("Problema")
                    temperatura = 0
        except Exception as e:
            print("Error al leer el sensor {}: {}".format(direccion_sensor, e))
            return 0
        temp = temp + "T" + str(temperatura)
    return temp
```

Figura 21: Lectura de sensores DS18B20.
Fuente: Propia

4.2. Configuración y adquisición de datos del sensor DHT22

Para la lectura desde al raspberry se necesitó descargar una librería llamada "Adafruit DHT", para esto se ejecutaba el comando "pip install Adafruit DHT" en la terminal. Primero se asigna a una variable el objeto de la librería "Adafruit DHT" que define el tipo de sensor, ya que puede ser el DHT22 O DHT11, luego se asigna el pin al cual se conectó que fue el GPIO17.

```
sensor = Adafruit_DHT.DHT22 #Cambia por DHT22 y si usas dicho sensor
pin = 17 #Pin en la raspberry donde conectamos el sensor
```

Figura 22: Definición sensor y pin usado para DHT22 .
Fuente: Propia

Luego se lee el DHT22, utilizando sentencias Try - Except, esto para el caso de que hayan errores en la lectura se retorna un 0.

```
def leer_dht22(self):
    humedad, temperatura = Adafruit_DHT.read_retry(Adafruit_DHT.DHT22, self.dht_pin)
    return "H" + str(round(humedad,2)) + "T" + str(round(temperatura,2))
```

Figura 23: Lectura DHT22 .
Fuente: Propia

4.3. Configuración y adquisición de datos del sensor SDS011

Para la lectura de este sensor se utiliza el puerto serial de los pines GPIO14(Tx) y GPIO15(Rx). Se utiliza la librería "**serial**" y se asigna el puerto a una variable:

```
ser = serial.Serial('/dev/ttyS0', 9600)
```

Figura 24: Asignación del puerto .
Fuente: Propia

Posteriormente se hace una análisis de la estructura de la trama que llega por el puerto y así se identifica los valores de las variables Pm10 y Pm25.

```
def leer_sds011(self):
    ser = serial.Serial(self.sds_serial_port,9600)
    data = ser.read(10)
    if data[0] == 0xAA and data[1] == 0xC0:
        pm25 = data[2] + data[3] / 10.0
        pm10 = data[4] + data[5] / 10.0
    return "A" + str(pm25) + "A" + str(pm10)
```

Figura 25: Lectura SDS011 .
Fuente: Propia

4.4. Configuración y adquisición de datos sensores SEN0244 y SEN0189.

Los sensores SEN0244 y SEN0189 entregan señales análogas, como la Raspberry no tiene ADC's incorporados fue necesario hacer uso del módulo ADS1115, en donde los pines de datos de los sensores se conectan a los pines A0 y A1 del módulo. Luego se conecta la Raspberry al módulo donde se comunican mediante protocolo I2C.

Para la lectura desde la Raspberry fue necesario descargar la librería "Adafruit_ADS1x15", para esto se ejectura el comando **"pip install adafruit-circuitpython-ads1x15"** en la terminal. Luego dentro del código de python se importan las librerías y se configura como se ve a continuación:

```
import time
import board
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn


i2c = board.I2C()
ads = ADS.ADS1115(i2c)
sen0189 = AnalogIn(ads, ADS.P0)
sen0244 = AnalogIn(ads, ADS.P1)
```

Figura 26: Lectura ADS1115 .
Fuente: Propia

De esta manera solo se llama a la variable asignada para obtener el valor de ADC.

5. Desarrollo de interfaz gráfica.

Tras la recopilación continua de datos por parte de los sensores encargados de monitorear las variables pertinentes en la cúpula geodésica, se procede a transmitir esta información en forma de una cadena de caracteres, empleando un enfoque de 'paquetización'. Posteriormente, en el código, se realiza la operación de 'desempaquetamiento' con el fin de acceder a los datos contenidos en el paquete. La representación visual de estos datos se lleva a cabo mediante el empleo de una pantalla conocida como '5 inch HDMI LCD de Raspberry'. Este proceso de visualización se apoya en diversas herramientas, entre las que destacan la mencionada pantalla 5 inch HDMI LCD de Raspberry, el lenguaje de programación Python, la plataforma Qt Designer y la librería matplotlib.

5.1. Planteamiento de visualización.

En nuestro enfoque para la visualización de datos provenientes de los sensores, exploramos diversas estrategias. Una de las opciones más destacadas fue la consideración de un canal de YouTube que desarrollaba una 'Dashboard' en la cual toda la información se actualiza en tiempo real en la página principal, sin la necesidad de pestañas adicionales. Sin embargo, al implementar un prototipo con la pantalla 5 inch HDMI LCD de Raspberry, se evidenció que esta modalidad resultaba insuficiente, ya que la visualización de los datos a distancia era problemática para los usuarios.

Por lo tanto, se tomó la decisión de optar por una interfaz alternativa que hiciera un uso óptimo de la funcionalidad táctil de la pantalla. De esta manera, los usuarios pueden navegar cómodamente entre las distintas pestañas, que incluyen 'Interfaz', 'Control', 'Registro' y 'Creadores'. Esta nueva aproximación garantiza que los datos, tanto en formato numérico como gráfico, sean fácilmente legibles por los usuarios, incluso a una distancia máxima de 2.5 metros, mejorando significativamente la experiencia de visualización.

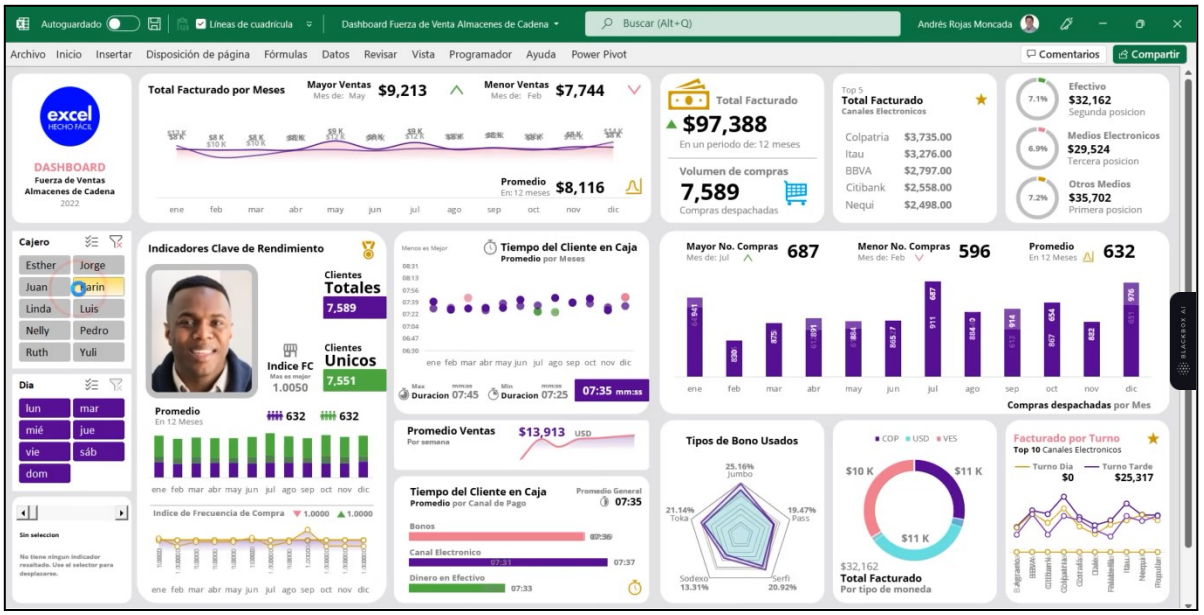


Figura 27: Dashboard para visualización de datos.
Fuente: Cuenta de YouTube “Excel Hecho Fácil” [17]

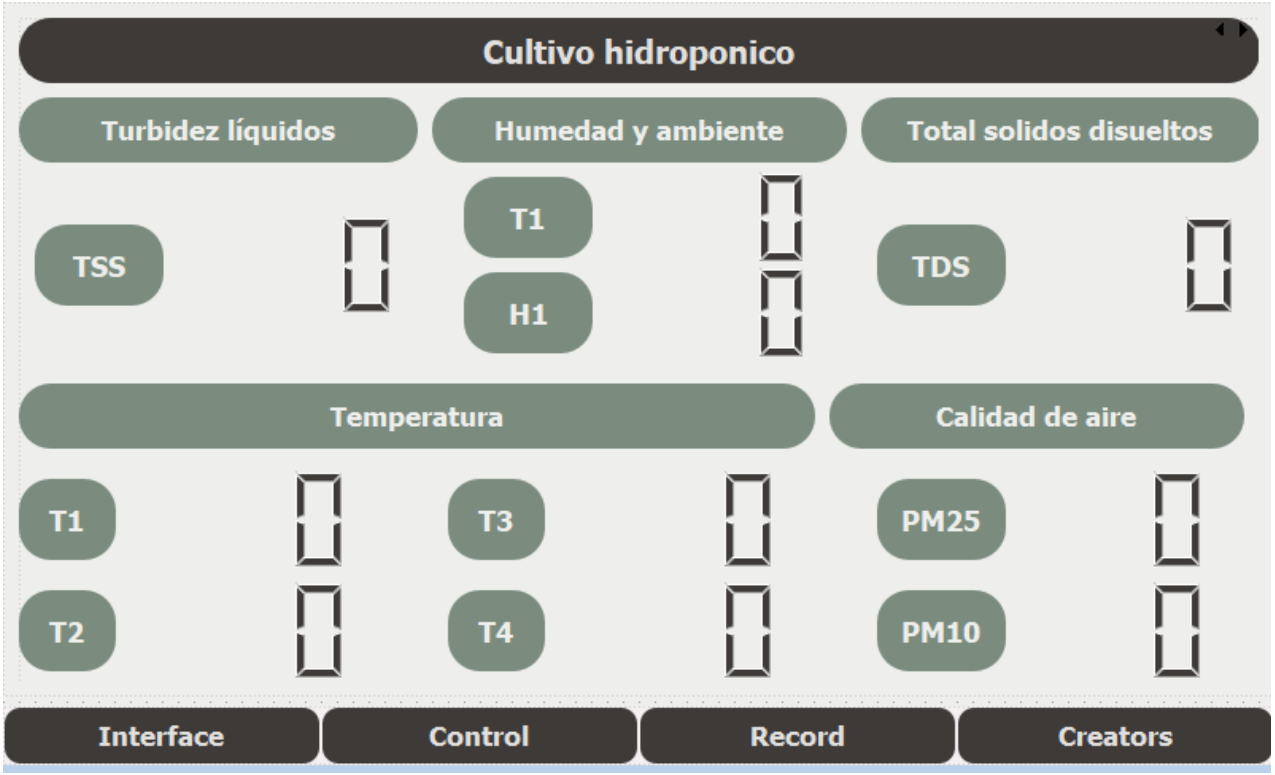


Figura 28: Estructura base de Interfaz gráfica de control de cultivo hidropónico.
Fuente: Propia

5.2. Desarrollo de visualización.

En el contexto del desarrollo de la interfaz inicial, ilustrada en la Figura 28, se identificaron varios desafíos en términos de diseño y optimización del espacio en pantalla. En primer lugar, se notó que los números y datos presentados en la interfaz no estaban utilizando eficientemente todo el espacio disponible en la pantalla. Además, se observó que la barra de interfaz, que desempeñaría un papel crucial para la navegación entre las diversas páginas, no era visible en la interfaz.

La razón detrás de esta falta de visibilidad radicaba en que la interfaz en uso mantenía elementos de decoración estándar, como bordes, botones de cierre, minimizar y maximizar, y una barra de título. Esto contribuyó a la reducción del espacio utilizable en la pantalla.

Adicionalmente, la barra de herramientas del sistema operativo estaba ocupando una cantidad significativa de píxeles, lo que a su vez desplazaba todo el contenido de la

interfaz del sistema de control de cultivo hidropónico hacia abajo. Para resolver este problema, se procedió a realizar una configuración en el sistema operativo, de manera que la barra de herramientas se mostrará únicamente cuando el cursor se ubicara en la parte superior de la pantalla, minimizando así su impacto en la interfaz.

Para abordar la cuestión de los bordes de la interfaz, que incluían botones de cierre, minimizar y maximizar, se implementó la línea de código

- self.setWindowFlag(Qt.FramelessWindowHint)`.

Esta instrucción específica proporciona una ventana sin bordes, comúnmente referida como "frameless window", eliminando de esta manera los elementos estándar de decoración de ventanas. Esto permitió una apariencia personalizada y, más importante aún, un uso eficiente del espacio en pantalla en la aplicación de control de cultivo hidropónico.

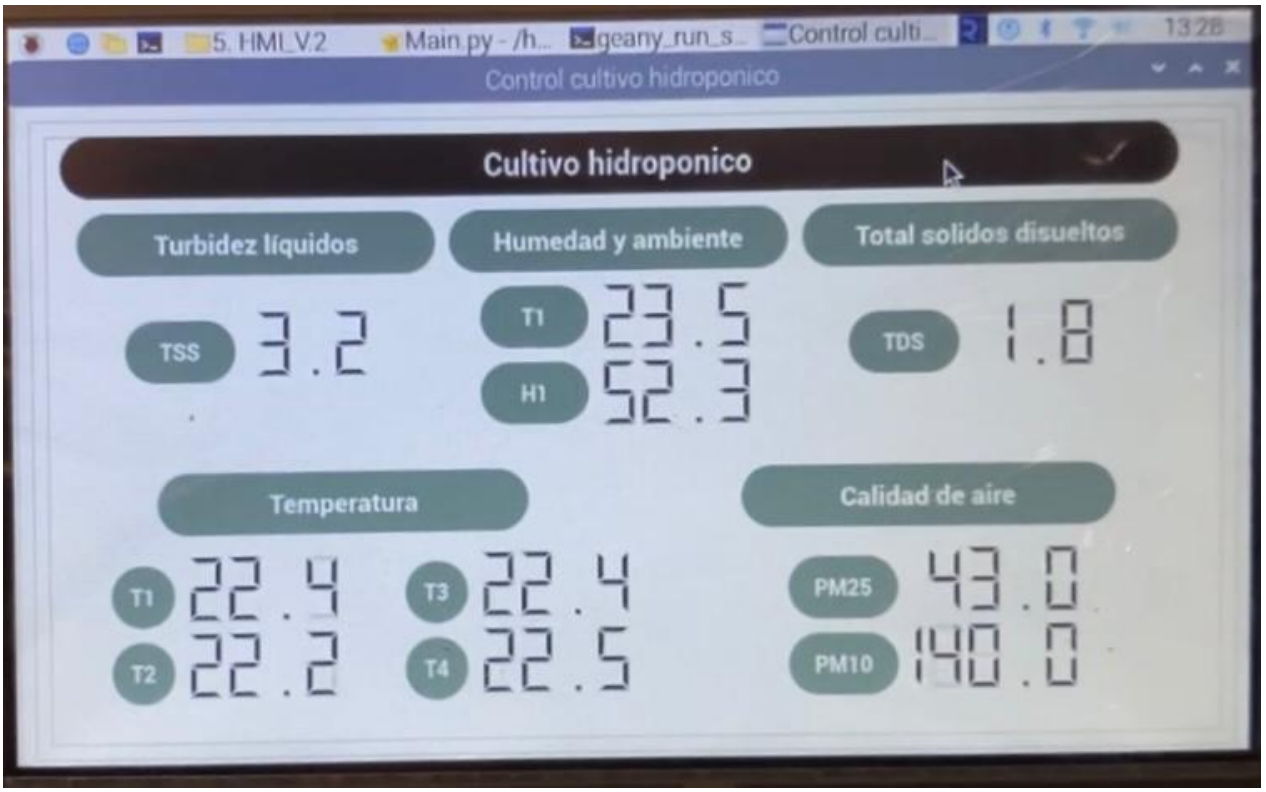


Figura 29: Interfaz gráfica de control de cultivo hidropónico versión 1.
Fuente: Propia

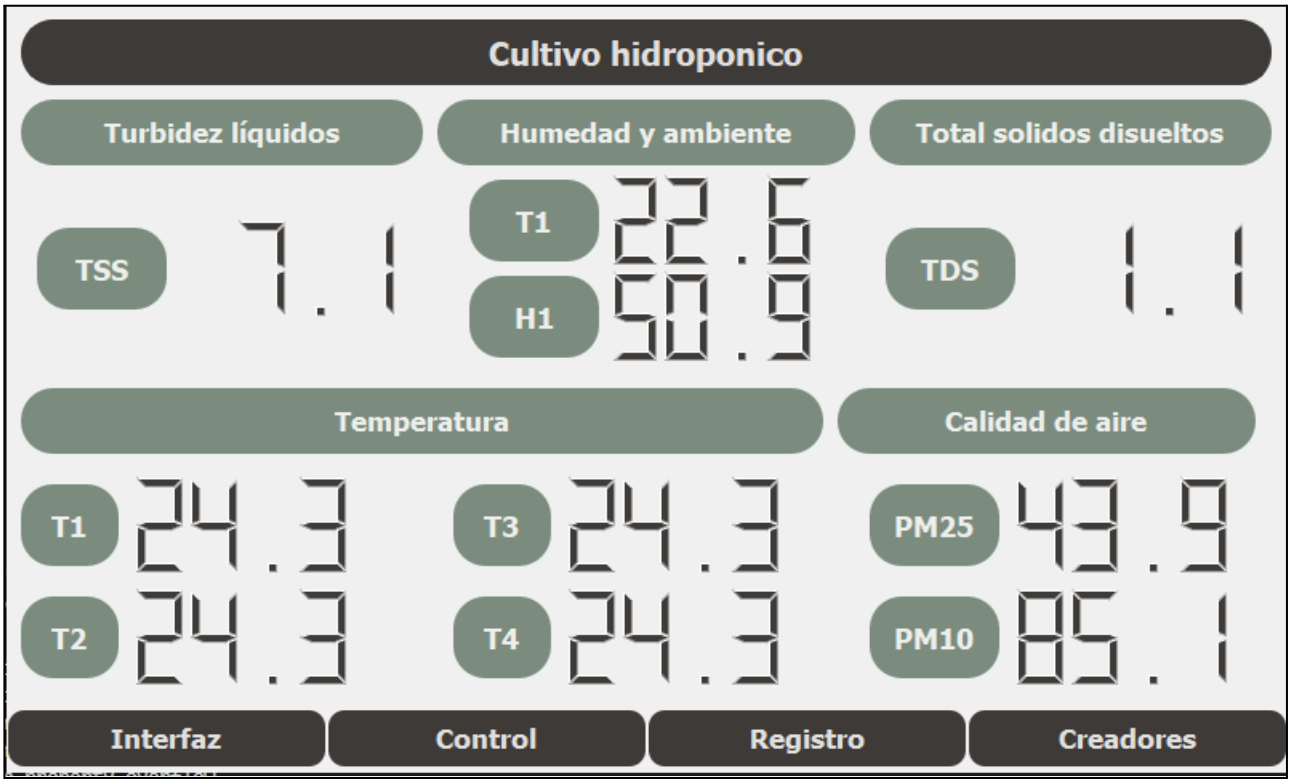


Figura 30: Interfaz gráfica de control de cultivo hidropónico versión 2.
Fuente: Propia

5.3. Desarrollo de código.

En el proceso de desarrollo del código, se adoptó una estructura modular y eficiente para garantizar la claridad y limpieza del código. Esto implicó la creación de varios programas que se interconectan, lo que resulta en una arquitectura de código modular. A continuación, se ofrece una breve descripción de la función de cada uno de estos programas que componen la carpeta de desarrollo de software dedicada al control visual del proyecto de cultivo hidropónico.

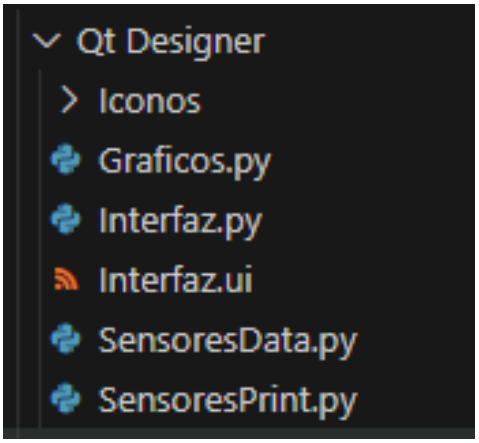


Figura 31: Estructura de control visual del proyecto de cultivo hidropónico.
Fuente: Propia

5.3.1. Programa Principal 'Interfaz.py'.

El programa central, conocido como 'Interfaz.py', asume la función primordial de ser el núcleo del sistema, encargándose de coordinar la ejecución de las diversas partes del software. Su tarea principal consiste en gestionar de manera eficiente la interacción entre los diversos módulos, lo que garantiza un flujo de control coherente en toda la aplicación. 'Interfaz.py' se integra de múltiples clases y funciones que establecen comunicación con los distintos elementos, como widgets y layouts, de la interfaz gráfica previamente diseñada en Qt Designer. Estas interacciones posibilitan la vinculación de páginas con botones, la organización de gráficos mediante 'StackedWidget' y la definición de dimensiones y atributos de la ventana, elementos esenciales para la experiencia de usuario en la aplicación de control de cultivo hidropónico. Este programa centraliza operaciones cruciales que aseguran una comunicación fluida y una interacción sin inconvenientes entre el usuario y la interfaz, optimizando la eficiencia y la funcionalidad del sistema en su conjunto.

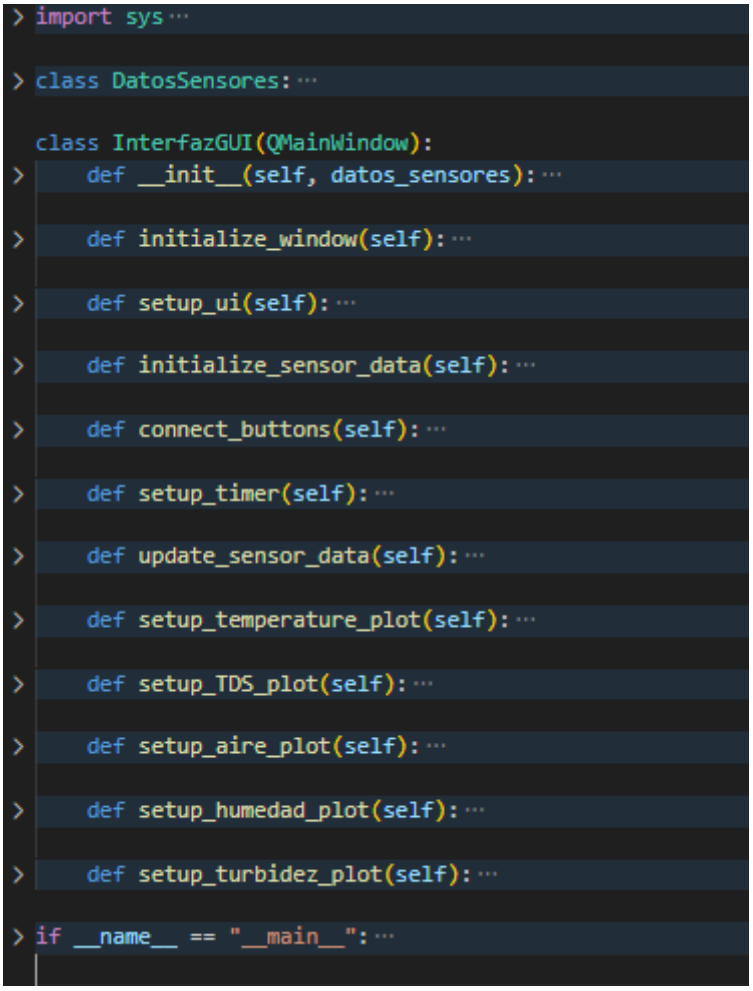


Figura 32: Funciones y clases del programa principal 'Interfaz.py'.
Fuente: Propia

5.3.2. Interfaz de Usuario 'Interfaz.ui'.

El programa denominado 'Interfaz.ui' alberga la totalidad de las características de la interfaz de usuario para el control visual del proyecto de cultivo hidropónico. Este programa desempeña un papel fundamental, ya que contiene toda la infraestructura generada en formato de código XML a partir de la aplicación de desarrollo Qt Designer.

Es crucial destacar que cualquier modificación en este programa podría resultar en errores en la ejecución del software. Por lo tanto, se debe abordar con extrema precaución para garantizar la integridad y la funcionalidad adecuada del sistema en su conjunto. 'Interfaz.ui' actúa como un componente esencial que define la experiencia del usuario en la aplicación y, como tal, su preservación en su estado original es de suma

importancia.

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
 <class>MainWindow</class>
 <widget class="QMainWindow" name="MainWindow">
  <property name="geometry">
   <rect>
    <x>0</x>
    <y>0</y>
    <width>800</width>
    <height>480</height>
   </rect>
  </property>
  <property name="windowTitle">
   <string>MainWindow</string>
  </property>
  <widget class="QWidget" name="centralwidget">
   <widget class="QFrame" name="FrameMenu">
    <property name="enabled">
```

Figura 33: Fragmento de Interfaz de Usuario 'Interfaz.ui'.
Fuente: Propia

5.3.3. Gestión de Datos ‘SensoresData.py’ y ‘SensoresPrint.py’

Los programas 'SensoresData.py' y 'SensoresPrint.py' cumplen un papel fundamental en el control y gestión de los datos adquiridos por los sensores en el sistema. 'SensoresData.py' se desempeña como una biblioteca de sensores, abarcando un conjunto de sensores que incluyen el DS18B20 termocuplas para medir la temperatura, el DHT22 para medir humedad y temperatura, el SDS011 para medir la calidad del aire, el sensor de turbidez de líquidos SEN0189 y el Total Dissolved Solid (TDS) SEN0244.

Este programa no solo se encarga de la recopilación de datos, sino también de la detección y manejo de posibles errores para prevenir fallos en la interfaz. Se implementa un proceso de depuración de errores para abordar desconexiones de sensores o mal funcionamiento, lo que contribuye a garantizar la integridad de los datos recopilados y su

envío adecuado sin interrupciones en el flujo de información.

Por otro lado, el programa 'SensoresPrint.py' asume la responsabilidad de supervisar y gestionar los datos que se envían, teniendo en cuenta tanto la tasa de actualización como los intervalos de retardo apropiados. Esta fase es esencial para evitar la transmisión de datos que la interfaz no pueda procesar, lo que podría resultar en errores en la visualización. En resumen, ambos programas desempeñan un papel crucial en el control y la gestión eficiente de los datos adquiridos por los sensores, contribuyendo a la robustez y el rendimiento del sistema en su conjunto.

```python
import os                   #Para DS18B20
import time
import serial               #Para SDS011
import Adafruit_DHT         #Para DHT22
from smbus import SMBus     #PARA SEN0244 y SEN0189

dir_temp1 = '28-3c21f649eec0'
dir_temp2 = '28-3c80f649bc0a'
dir_temp3 = '28-3ce9f649e06f'
dir_temp4 = '28-3cfdf649f716'
direcciones = [dir_temp1,dir_temp2,dir_temp3,dir_temp4]

class lecturaSensores:

    def leer_ds18b20(self):
        temp = ""
        for direccion_sensor in direcciones:
            try:
                with open('/sys/bus/w1/devices/{}/w1_slave'.format(direccion_sensor), 'r') as f:
                    lines = f.readlines()
                    if lines[0].strip()[-3:] == 'YES':
                        equals_pos = lines[1].find('t=')
                        if equals_pos != -1:
                            temp_string = lines[1][equals_pos+2:]
                            temperatura = float(temp_string) / 1000.0
                    # Si ocurre algún error en la lectura del sensor, devolvemos 0
                    else:
                        print("Problema")
                        temperatura = 0
            except Exception as e:
                print("Error al leer el sensor {}: {}".format(direccion_sensor, e))
                return 0
            temp = temp + "T" + str(temperatura)
        return temp
    def __init__(self, dht_pin, sds_serial_port):
        self.dht_pin = dht_pin
        self.sds_serial_port = sds_serial_port
        self.ds18b20_sensor_id = direcciones
        os.system('modprobe w1-gpio')
        os.system('modprobe w1-therm')

    def leer_dht22(self):
        humedad, temperatura = Adafruit_DHT.read_retry(Adafruit_DHT.DHT22, self.dht_pin)
        return "H" + str(round(humedad,2)) + "T" + str(round(temperatura,2))

    def leer_sds011(self):
        ser = serial.Serial(self.sds_serial_port,9600)
        data = ser.read(10)
        if data[0] == 0xAA and data[1] == 0xC0:
            pm25 = data[2] + data[3] / 10.0
            pm10 = data[4] + data[5] / 10.0
        return "A" + str(pm25) + "A" + str(pm10)
```

Figura 34: Fragmento de Gestión de Datos 'SensoresData.py'.
Fuente: Propia

```python
# Inicializar variables para el cálculo del promedio
inicio_minuto = time.time()
contador_datos = 0

while True:
    datos_sensores = SensoresData.generar_datos_sensores()

    # Calcular el tiempo transcurrido desde el último envío de datos
    tiempo_transcurrido = time.time() - inicio_minuto

    # Imprimir cada clave en una línea diferente
    for clave, valor in datos_sensores.items():
        print(f"{clave}: {valor}")

    # Incrementar el contador de datos
    contador_datos += 1

    # Si ha pasado 1 minuto, calcular el promedio y reiniciar el contador
    if tiempo_transcurrido >= 60:
        promedio_por_minuto = contador_datos / (tiempo_transcurrido / 60)
        print(f"Promedio por minuto: {promedio_por_minuto} datos")
        contador_datos = 0
        inicio_minuto = time.time()

    time.sleep(2)  # Esperar 2 segundos antes de llamar a la función nuevamente
S
```

Figura 35: Fragmento de Gestión de Datos 'SensoresPrint.py'.
Fuente: Propia

5.3.4. Visualización Gráfica ‘Graficos.py’

El programa 'Graficos.py' desempeña la generación de representaciones visuales a partir de los datos recopilados en el sistema. Este programa almacena toda la estructura y el estilo necesarios para la creación de gráficos que se proyectarán en la interfaz gráfica de usuario. Es importante destacar que 'Graficos.py' solo se activa cuando el usuario accede a la página de Registro, esto se hace con el propósito de optimizar los recursos del sistema y garantizar un rendimiento fluido de la interfaz.

La generación de gráficos se lleva a cabo utilizando la librería Matplotlib, para obtener los

datos necesarios, 'Graficos.py' se comunica con el programa 'Interfaz.py', el cual a su vez realiza las llamadas pertinentes al programa 'SensoresData.py'. Los datos se procesan y almacenan en una lista que se utiliza para generar la gráfica. El programa 'Graficos.py' se configura para visualizar aproximadamente 15 puntos de datos con un intervalo de 1 segundo entre cada uno. Es importante mencionar que este intervalo puede ajustarse de manera sencilla según las necesidades específicas del operador encargado del cultivo hidropónico.

```
import time···

class TemperatureGraphWidget(FigureCanvas):
    def __init__(self, parent=None, width=10, height=10, dpi=100, datos_sensores=None):

    def update_temperature_plot(self):···

class TDSGraphWidget(FigureCanvas):
    def __init__(self, parent=None, width=10, height=10, dpi=100, datos_sensores=None):

    def update_TDS_plot(self):···

class AireGraphWidget(FigureCanvas):

    def __init__(self, parent=None, width=10, height=10, dpi=100, datos_sensores=None):

    def update_Aire_plot(self):···

class HumedadGraphWidget(FigureCanvas):
    def __init__(self, parent=None, width=10, height=10, dpi=100, datos_sensores=None):

    def update_Humedad_plot(self):···

class TurbidezGraphWidget(FigureCanvas):
    def __init__(self, parent=None, width=10, height=10, dpi=100, datos_sensores=None):

    def update_Turbidez_plot(self):···
```

Figura 36: Fragmento de Visualización Gráfica ‘Graficos.py’
Fuente: Propia

5.3.5. Comunicación Externa ‘Iconos’

La carpeta 'Iconos' alberga imágenes que se utilizan en la interfaz del programa 'Interfaz.ui'. Estas imágenes se destinan a la inserción en los botones que, a su vez, activarán diferentes tipos de gráficos. La razón detrás de esta elección radica en la

búsqueda de una mayor integración visual y usabilidad en la interfaz de usuario.

En lugar de una presentación excesivamente textual que podría fatigar la vista del usuario, se optó por una representación gráfica. Cada icono dentro de esta carpeta representa un tipo específico de sensor utilizado en el control del cultivo hidropónico. Esta decisión contribuye significativamente a la mejora de la experiencia del usuario, al brindar una interfaz más amigable visualmente y al adherirse a los estándares de diseño actuales.

La inclusión de estos iconos en los botones proporciona una forma intuitiva y eficiente para que el usuario interactúe con el sistema, identificando y seleccionando fácilmente las funcionalidades relacionadas con cada sensor.

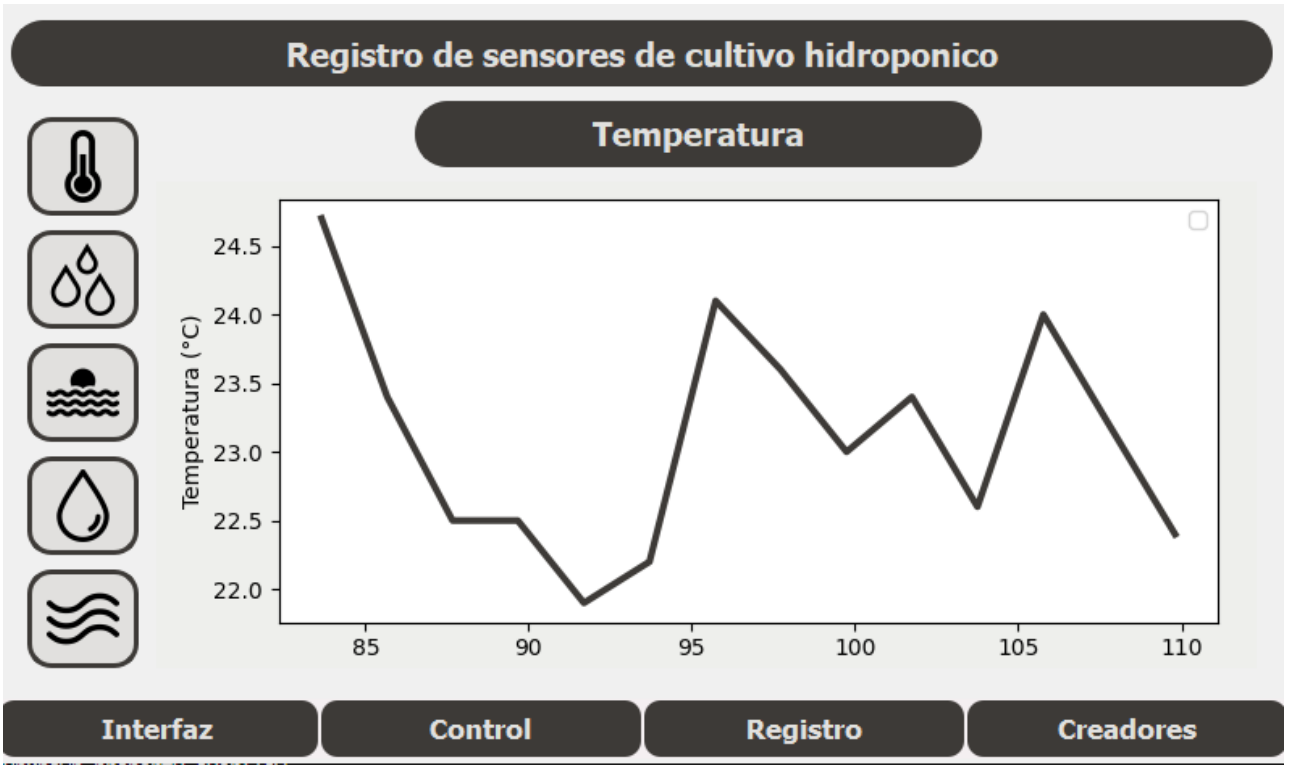


Figura 37: Interfaz gráfica en la página Registro
Fuente: Propia

5.4. Visualización de interfaz gráfica.

La visualización de los datos recopilados por los sensores se organiza en tres páginas principales: 'Interfaz', 'Control' y 'Registro'. Además, se incluye una cuarta página

destinada exclusivamente a los desarrolladores del proyecto, denominada 'Monitoreo y Control de Cultivo Hidropónico en Entorno de Cúpula Geodésica'.

5.4.1. Página 'Interfaz’.

La página 'Interfaz' se caracteriza por su disposición ordenada de 10 variables, cada una acompañada de un título específico que refleja la información capturada por un sensor en particular. Además, se incluye una leyenda en el lado izquierdo de cada variable para proporcionar al usuario una indicación clara de qué dato está observando en ese momento.

Un aspecto distintivo de esta interfaz es la elección de representar los valores numéricos de las variables en un estilo tipo LCD. Esta decisión de diseño se basa en la consideración de brindar una visualización óptima, especialmente cuando el usuario se encuentra a cierta distancia de la pantalla. El uso de caracteres estilo LCD mejora la legibilidad y facilita la comprensión de los datos, incluso en condiciones donde la distancia podría ser un factor limitante.

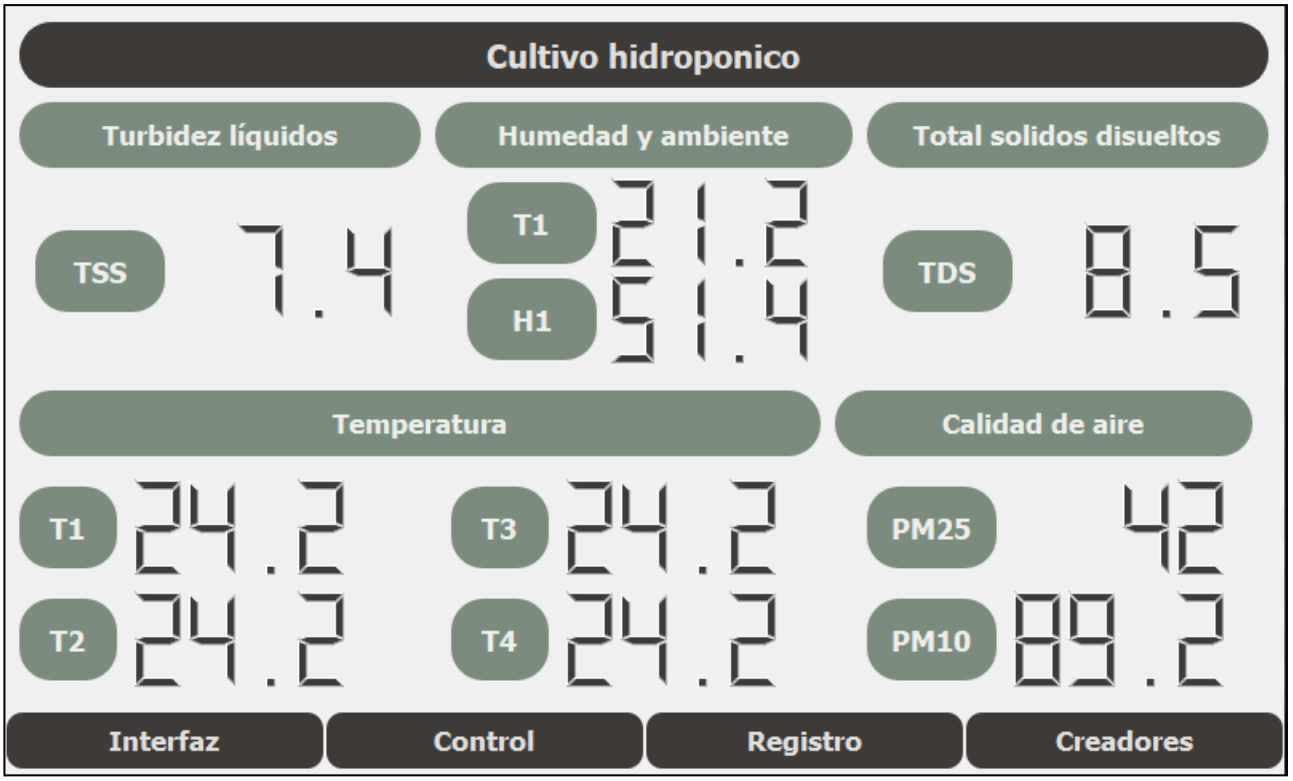


Figura 38: Página de Interfaz de visualización de control del cultivo hidropónico.
Fuente: Propia

5.4.2. Página ‘Control’.

La página 'Control' asume un rol fundamental al brindar al usuario la capacidad de controlar diversas variables críticas. Estas funciones incluyen la apertura o cierre de electroválvulas para gestionar el flujo de líquidos, así como la activación o desactivación del flujo de aire, entre otras acciones de control significativas.

Un aspecto relevante es que estas acciones de control sólo son accesibles y ejecutables desde la página 'Control' específicamente. Esto se ha diseñado con un propósito claro: garantizar un control preciso y seguro de las variables esenciales del sistema.

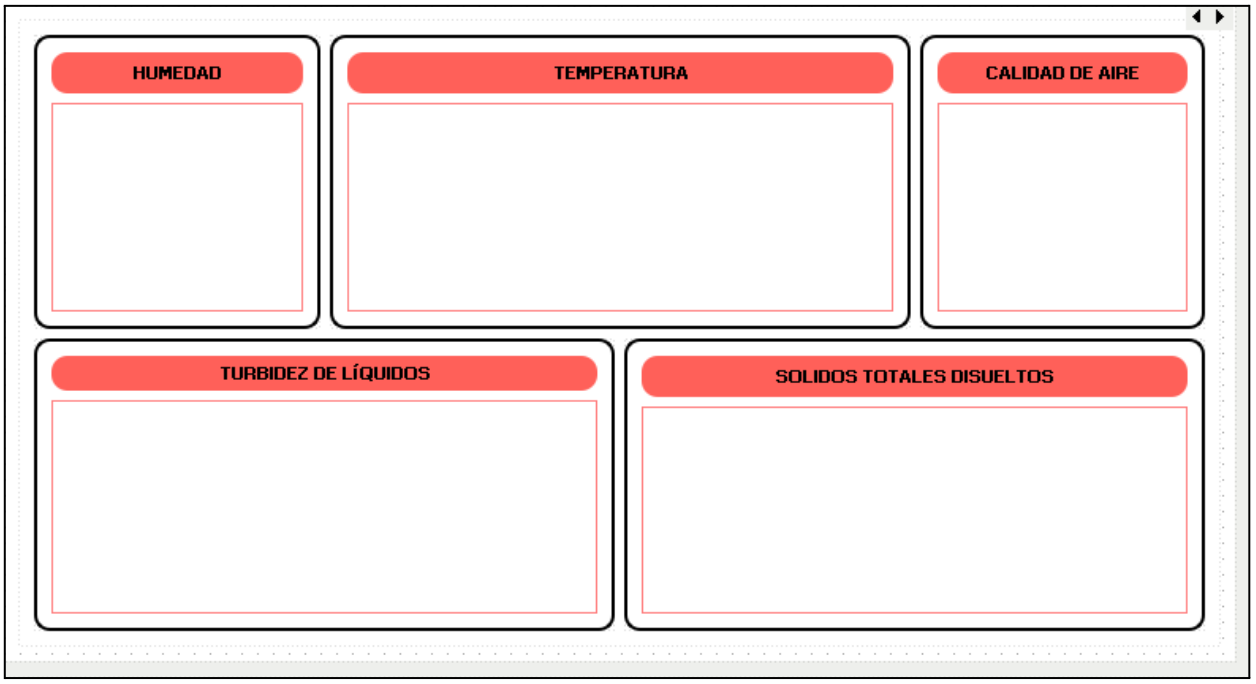


Figura 39: Página de Control de visualización de control del cultivo hidropónico.
Fuente: Propia

5.4.3. Página ‘Registro’.

La página 'Registro' cumple la función de presentar los datos de los sensores a través de gráficos. El acceso a esta página se realiza mediante botones con iconos que se asemejan a los sensores que se desean observar. La característica destacada de estas gráficas es su capacidad de adaptarse a la cantidad de datos que se desean visualizar.

La presencia de estos iconos en los botones agrega una capa de usabilidad intuitiva al sistema. Facilitan la interacción del usuario al permitirle identificar y seleccionar de manera sencilla las funcionalidades asociadas a cada sensor.

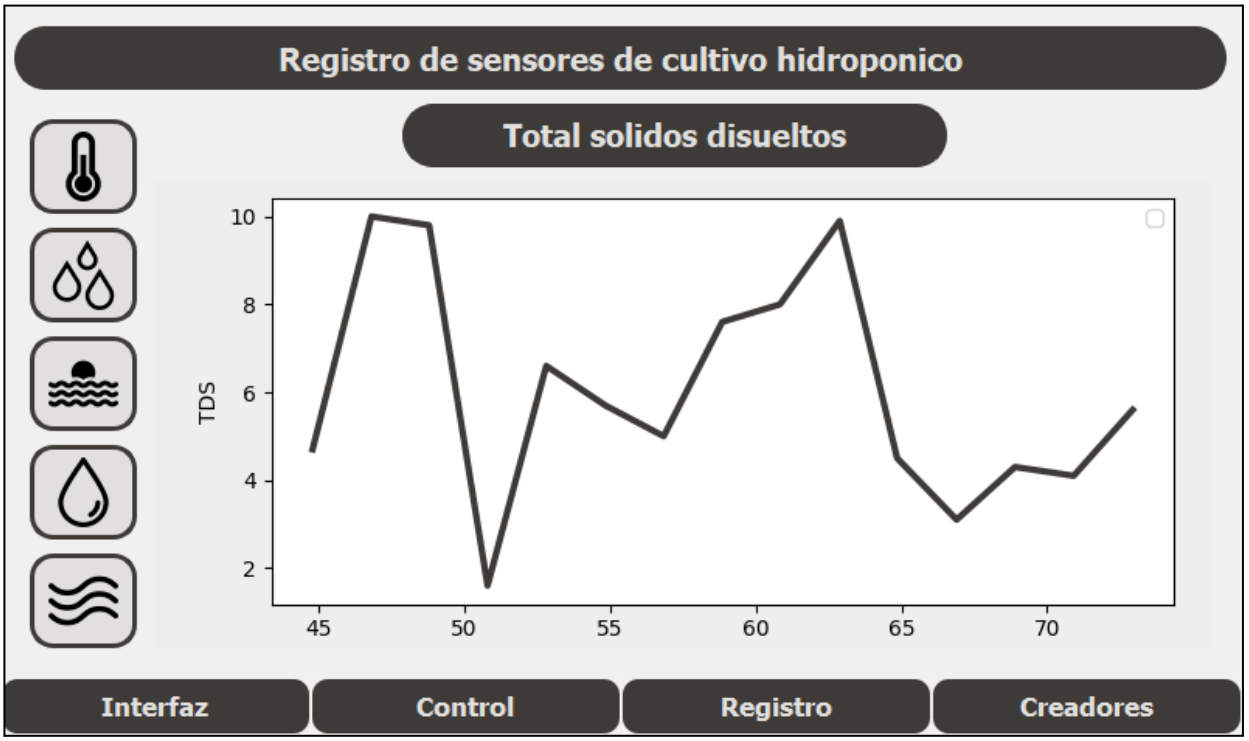


Figura 40: Página de Registro de visualización de control del cultivo hidropónico.
Fuente: Propia

5.4.4. Página ‘Creadores’.

La página 'Creadores' desempeña un papel destacado al brindar un espacio dedicado al reconocimiento de los desarrolladores involucrados en la creación de este proyecto. En esta sección, se incluye una breve descripción de los desarrolladores, destacando sus contribuciones y roles específicos en el desarrollo del proyecto.

Además de proporcionar información sobre los desarrolladores, la página 'Creadores' también ofrece detalles sobre el proceso de desarrollo, la ubicación geográfica donde se llevó a cabo y la investigación integral que respalda el proyecto en su conjunto.

Un elemento importante es la inclusión de datos de contacto que permiten a terceros interesados en contratación o desarrollo relacionado con la línea del proyecto

comunicarse de manera directa con los desarrolladores. Esta característica facilita la colaboración y la expansión del proyecto en un contexto profesional.



Figura 41: Página de Creadores de visualización de control del cultivo hidropónico.
Fuente: Propia

6. Desarrollo de cúpula geodésica

6.1. Calculo de construcción de cúpula geodésica

Para la realización del proyecto de construcción de un domo geodésico, se dedicaron numerosas horas de estudio con el propósito de comprender a fondo su comportamiento y determinar la metodología más efectiva y eficiente para llevar a cabo el montaje. En una primera etapa, se llevaron a cabo múltiples cotizaciones con el fin de evaluar la relación calidad-precio de diversos materiales y recursos disponibles en diferentes ubicaciones dentro de Bogotá, D.C.

Enfatizamos que nuestra principal preocupación fue la optimización de los recursos económicos destinados a la construcción de la cúpula geodésica. Con este objetivo en

mente, priorizamos la búsqueda de madera de estructura similar a la utilizada en estibas, dado que cumplía con aproximadamente el 80% de las especificaciones requeridas para el proyecto. Los restantes detalles, que conforman el 20% restante de las especificaciones, pudieron ser resueltos mediante el uso de maquinaria de corte especializada, proceso que se llevó a cabo en el Centro Industrial y de Desarrollo Empresarial Tecnoparque Nodo Cazuca del SENA.

N°	Ubicación	Tipo de madera	Valor unidad	Dimensiones
1	Socha Auto Sur x Cl 40 Su	DNS*	$ 1.416	100 cm x 12 cm x 2 cm
2	Bogotá D.C. Av. Boyaca x Dg 52 A Sur No. 44	Eucalipto y pino Canadiense	$ 1.130	100 cm x 10 cm x 2 cm
3	Bogotá D.C. Cl 44 C Sur x Tv 68 D	Eucalipto y pino Canadiense	$ 3.083*	300 cm x 25 cm x 4 cm
4	Bogotá D.C. AC. 24 x KR 128	Pino Canadiense	$ 1.700	100 cm x 10 cm x 2 cm

*Se realizo el calculo de cuentas tablas de 100 cm x 10 cm x 2 cm podrían salir de los tablones cotizados de medidas 300 cm x 25 cm x 4 cm, la cantidad máxima aprovechable eran 12 tablas, por lo que se tomó el precio inicial de 37.000 que costaba 1 tablón y se dividió en 12 tablas aprovechables.

*DNS: Dato no suministrado.



Figura 42: Depósito de madera a base de estibas.

Fuente: Propia

Basándonos en los datos previamente mencionados, se consideraron dos características principales para la toma de decisiones: el precio y el tipo de madera. Luego de una reunión de consulta con el profesional a cargo de la pasantía, se llegó a la conclusión de que la inversión más adecuada sería en madera de Pino Canadiense, la cual corresponde a la opción 4 y tiene un valor de 1.700 por unidad.

6.2. Uniones de construcción de cúpula geodésica

En el contexto de la construcción de la cúpula geodésica, se concede una gran relevancia al aspecto de las uniones, dado que la resistencia de la estructura depende en gran medida de estas conexiones. En esta ocasión, se ha optado por emplear la técnica de conexión denominada "Técnica Sin Conectores - Good Karma" o "Wood Karma". Esta elección se fundamenta en la decisión de evitar costos adicionales asociados a armazones o conectores adicionales.

Es importante señalar que esta técnica, aunque efectiva, posee cierta complejidad debido a los cortes y ángulos específicos que deben ser cuidadosamente respetados para lograr una perfecta coincidencia en las uniones. Sin embargo, se ha facilitado el proceso al contar con una sólida base de corte, lo que ha permitido que las piezas se corten de manera más eficiente y precisa.



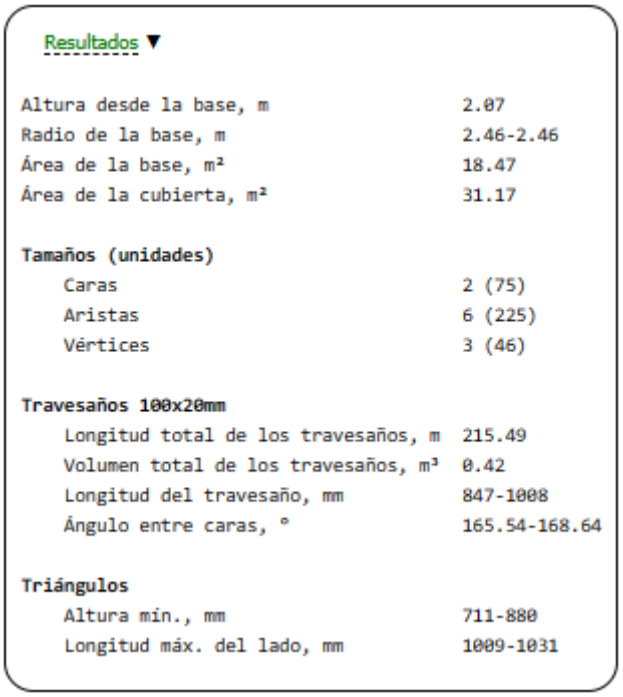
Figura 43: Tipo de conexión “Good Karma" o "Wood Karma" en cúpula geodésica.
Fuente: [18]

6.3. Cortes de construcción de cúpula geodésica

Para llevar a cabo los cortes de madera necesarios en la construcción de la cúpula geodésica, se requieren diversas piezas con distintas longitudes y ángulos de corte. Sin embargo, es esencial establecer previamente las características fundamentales de nuestra cúpula geodésica. En esta ocasión, hemos diseñado una cúpula que cumple con las siguientes especificaciones:

- Poliedro: Icosaedro
- Frecuencia: 3V
- Método de subdivisión: Cuerdas iguales
- Porción de esfera: 5/12
- Diámetro de la esfera: 5 metros
- Tipo de conexión: "Good Karma"
- Tamaño de las piezas: Ancho de 100 mm y grosor de 20 mm

Estos parámetros son cruciales para determinar los cortes precisos que serán requeridos en la madera, y garantizan que las piezas resultantes se ajusten correctamente a la geometría de la cúpula geodésica deseada.



Resultados ▼

Altura desde la base, m	2.07
Radio de la base, m	2.46-2.46
Área de la base, m²	18.47
Área de la cubierta, m²	31.17
Tamaños (unidades)	
Caras	2 (75)
Aristas	6 (225)
Vértices	3 (46)
Travesaños 100x20mm	
Longitud total de los travesaños, m	215.49
Volumen total de los travesaños, m³	0.42
Longitud del travesaño, mm	847-1008
Ángulo entre caras, °	165.54-168.64
Triángulos	
Altura mín., mm	711-880
Longitud máx. del lado, mm	1009-1031

Figura 44: Resultados de cúpula geodésica con características mencionadas.
Fuente: [19]

Tenemos seis tipos distintos de piezas de corte de madera. Para obtener una orientación precisa acerca de los cortes requeridos y los métodos de unión recomendados para una construcción adecuada, el sitio web www.acidome.com nos proporciona esta información. Esta plataforma web proporciona una guía exhaustiva que detalla los procedimientos necesarios para garantizar la correcta ejecución de las conexiones y cortes en la construcción de nuestras estructuras.

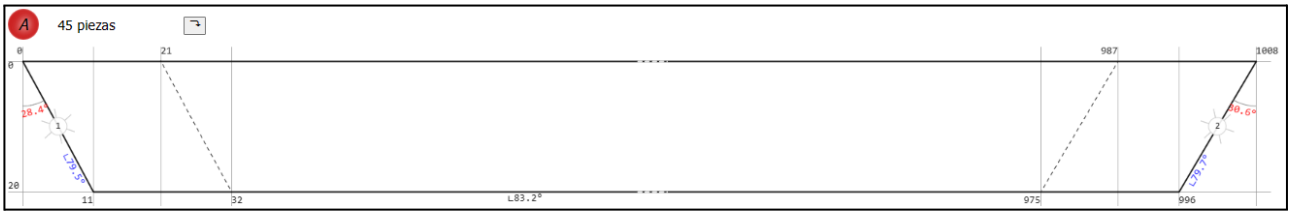


Figura 45: Cortes para tablas tipo A de vista lateral.

Fuente: [19]

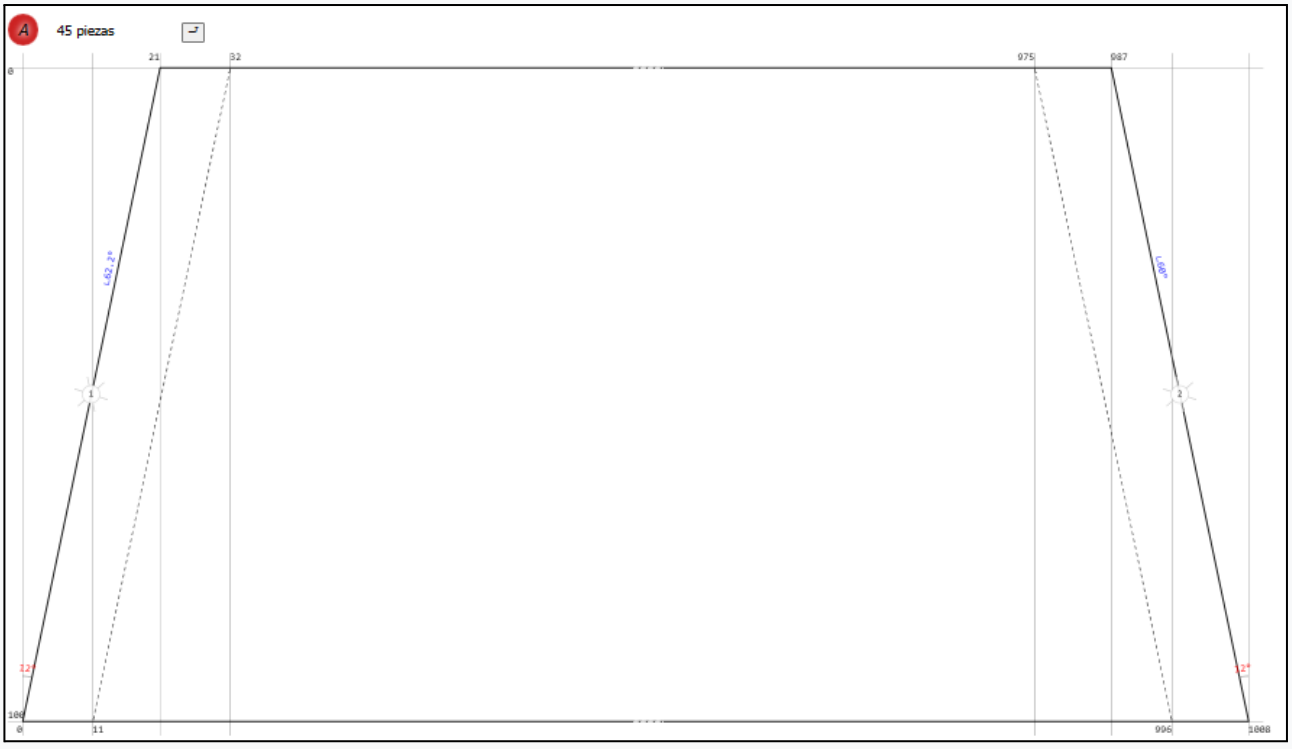


Figura 46: Cortes para tablas tipo A de vista frontal.

Fuente: [19]

La construcción de la cúpula geodésica, de acuerdo con las características previamente mencionadas, demandó un total de 225 tablas de madera. Estas tablas se dividen en seis categorías distintas, con una distribución específica de unidades como sigue:

- Tablas tipo A: 45 unidades.
- Tablas tipo B: 45 unidades.
- Tablas tipo C: 45 unidades.
- Tablas tipo D: 30 unidades.
- Tablas tipo E: 30 unidades.
- Tablas tipo F: 30 unidades.

Esta asignación de tablas según su tipo y cantidad es esencial para asegurar la disponibilidad y organización adecuada de los materiales necesarios en el proceso de construcción de la cúpula geodésica.

6.4. Ensamble de construcción de cúpula geodésica

En el proceso de ensamblaje de la cúpula geodésica, se aseguró previamente que todas las piezas de madera estuvieran cortadas con anticipación y se organizaron en grupos según el tipo de tabla asignada. Posteriormente, se procedió a la construcción de los triángulos base de la estructura. El Triángulo 1 se conformó utilizando las tablas de tipo A, B y C, mientras que el Triángulo 2 se construyó utilizando las tablas de tipo D, E y F, como se detalla a continuación:

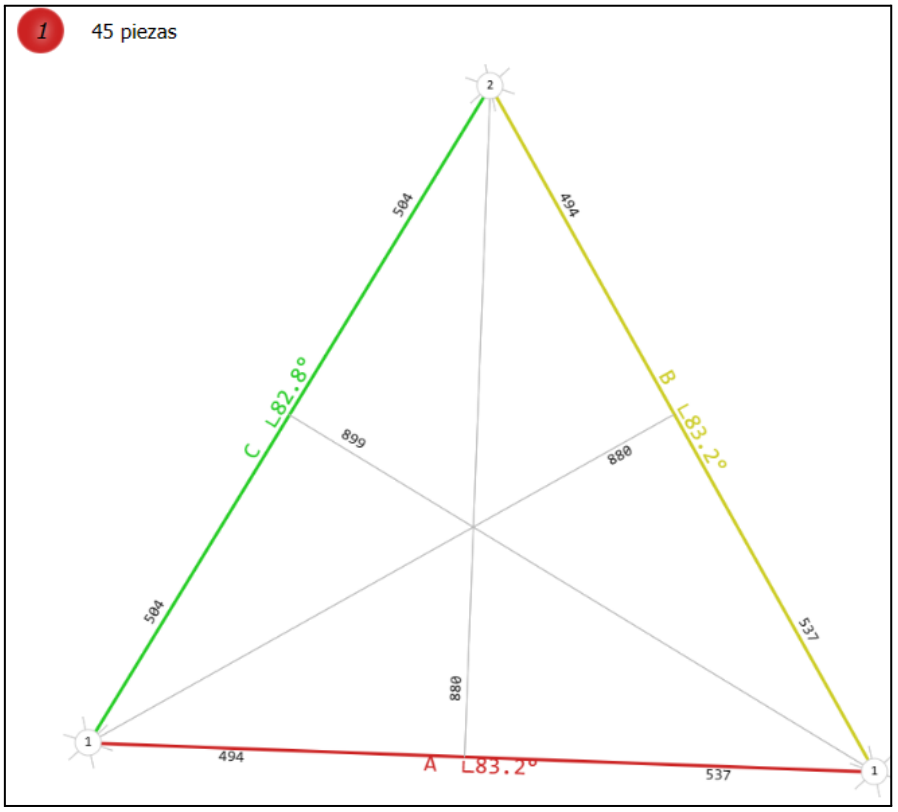


Figura 47: Construcción de triángulos 1 base de estructura.

Fuente: [19]

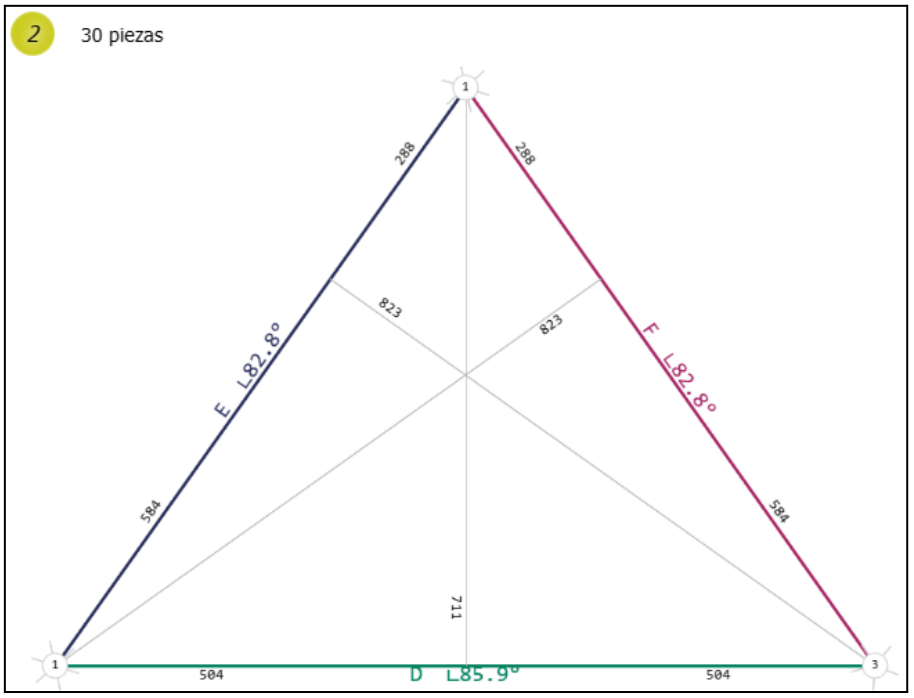


Figura 48: Construcción de triángulos base 2 de estructura.

Fuente: [19]

Después de la unión de los triángulos base, se procede a la construcción continua de la cúpula geodésica, ensamblando cada triángulo tal como se ilustra en la figura [49]. Este proceso se basa en los cortes realizados previamente en la madera, los cuales han sido diseñados de manera que generen la curvatura natural del domo. A medida que se unen más partes, se puede apreciar claramente cómo la cúpula geodésica va tomando forma y cerrándose gradualmente.

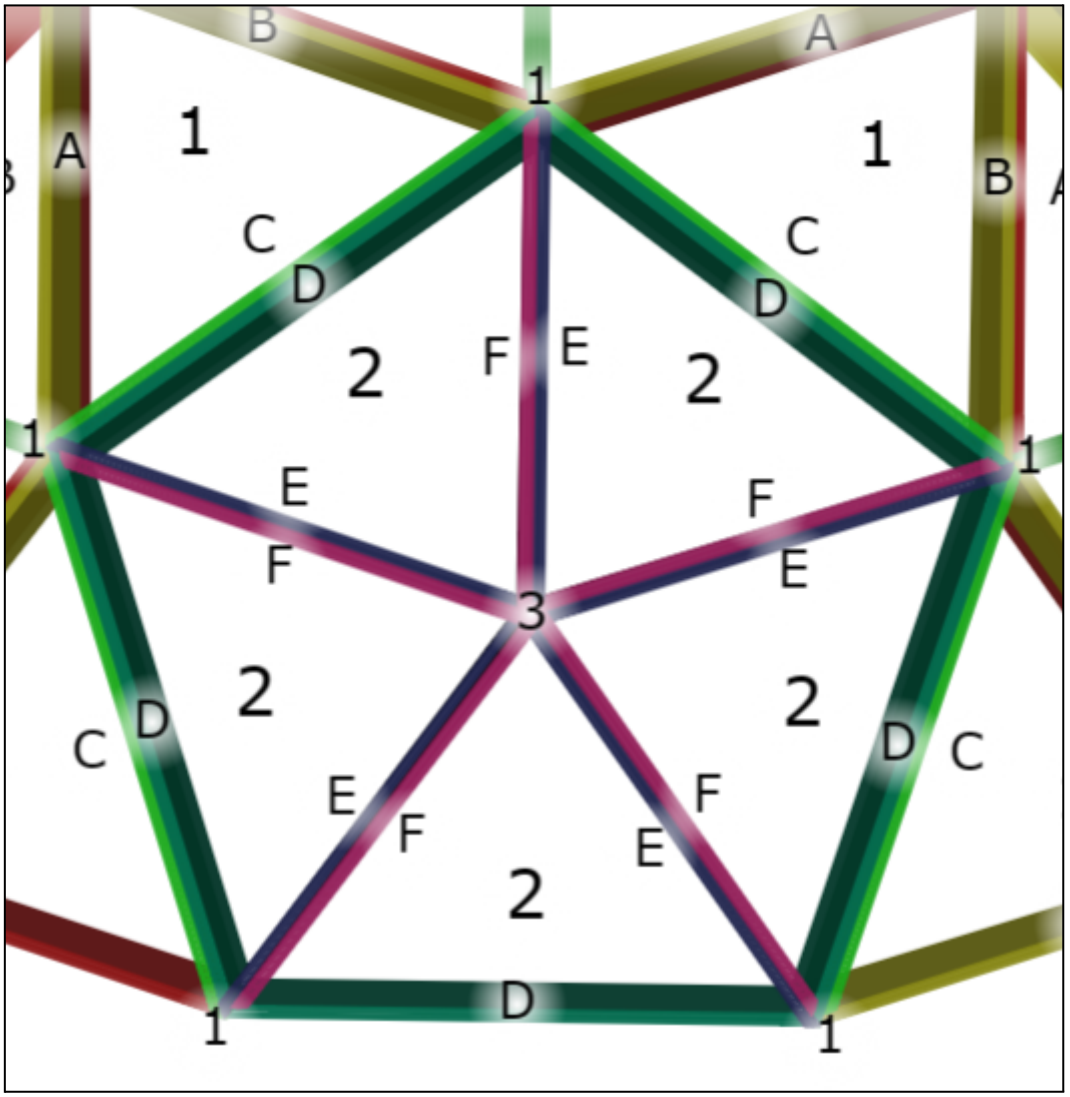


Figura 49: Unión de triángulos base para construcción de cúpula geodésica.

Fuente: [19]

La construcción de la cúpula geodésica se ha llevado a cabo de manera exitosa, culminando en la estructura final como se muestra a continuación:

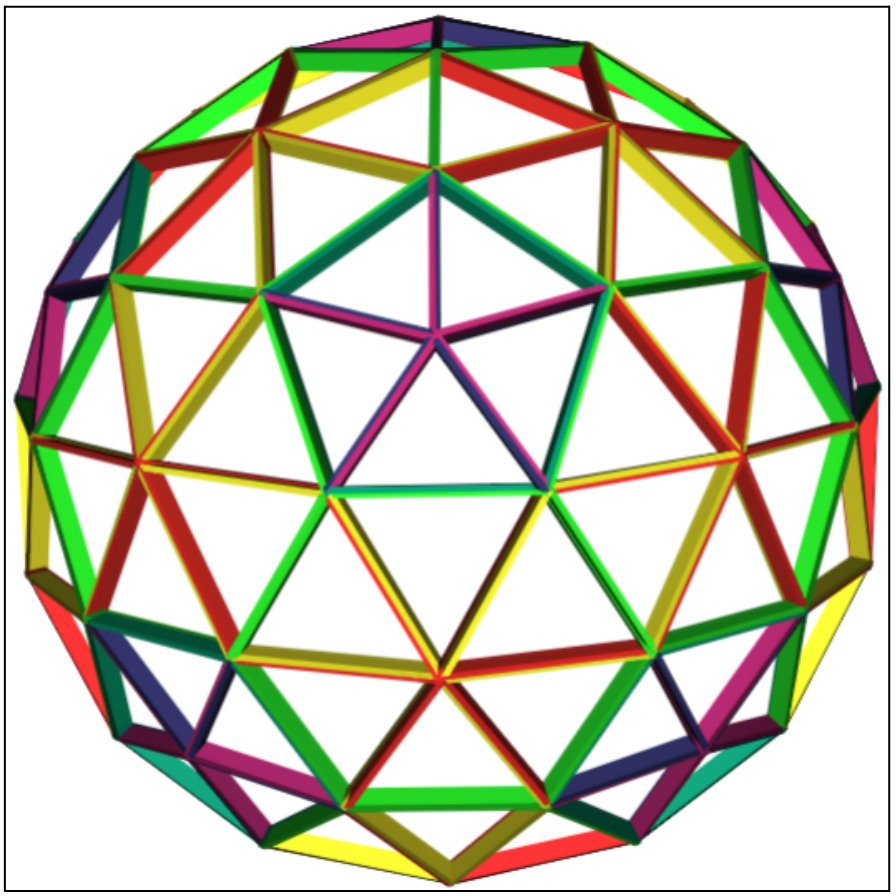


Figura 50: Cúpula geodésica de vista aérea.

Fuente: [19]

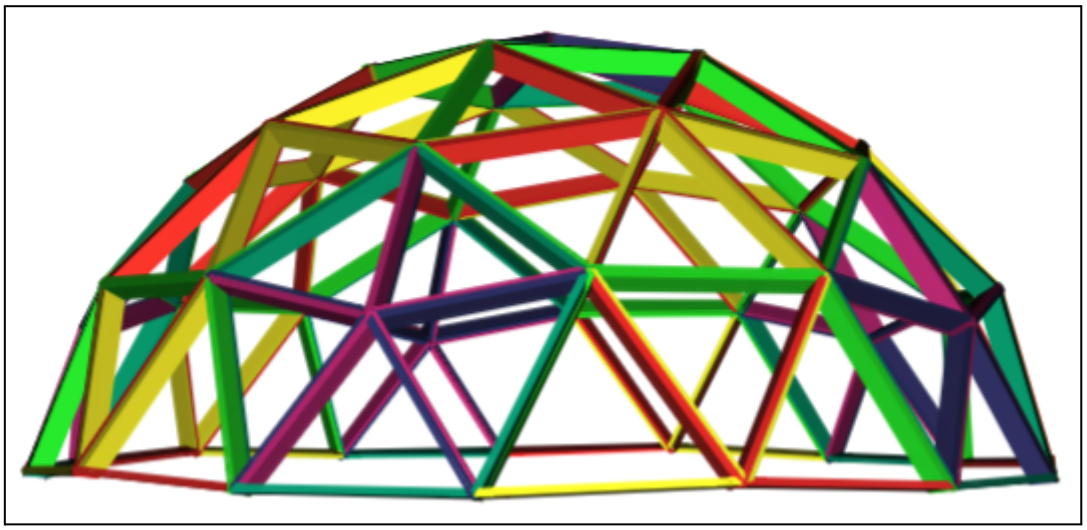


Figura 51: Cúpula geodésica de vista lateral.

Fuente: [19]

7. Ensamble

7.1. Diseño PCB.

La placa de circuito impreso (PCB) fue diseñada utilizando el software KiCad, con la integración de todos los componentes relacionados con la alimentación, la adquisición de datos y el control en una única placa. A continuación, se presenta el diseño resultante y su correspondiente impresión

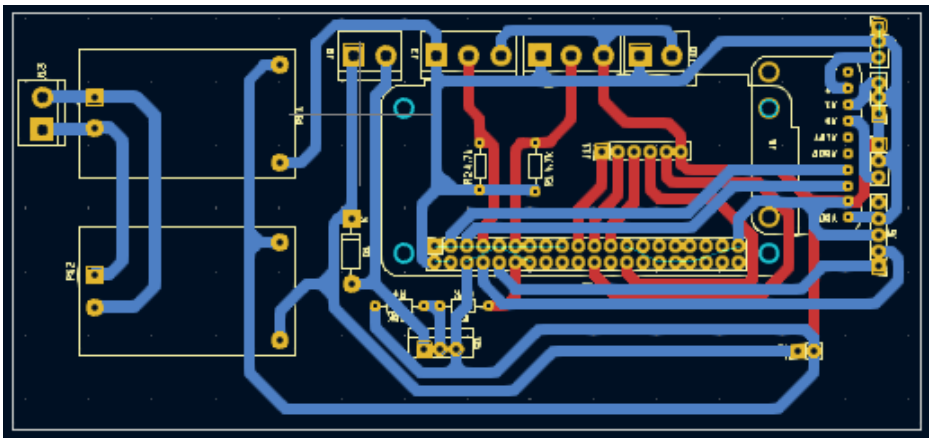


Figura 52: Diseño de PCB en KiCad.

Fuente: Propia

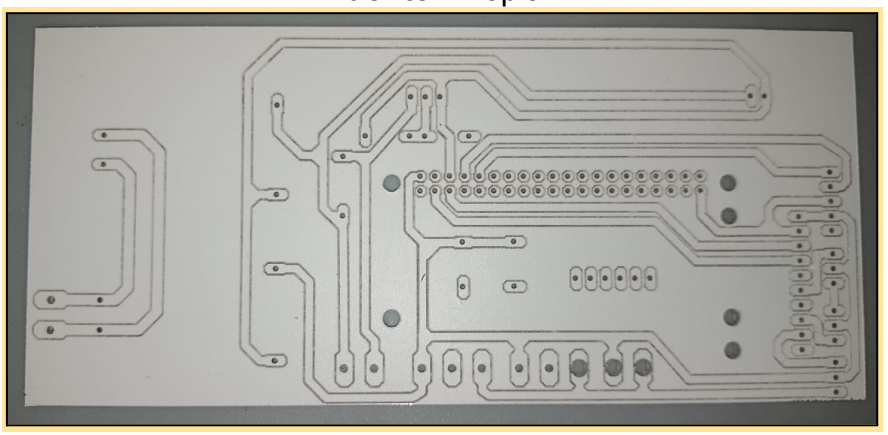


Figura 53: Impresión capa Bottom de la PCB.

Fuente: Propia

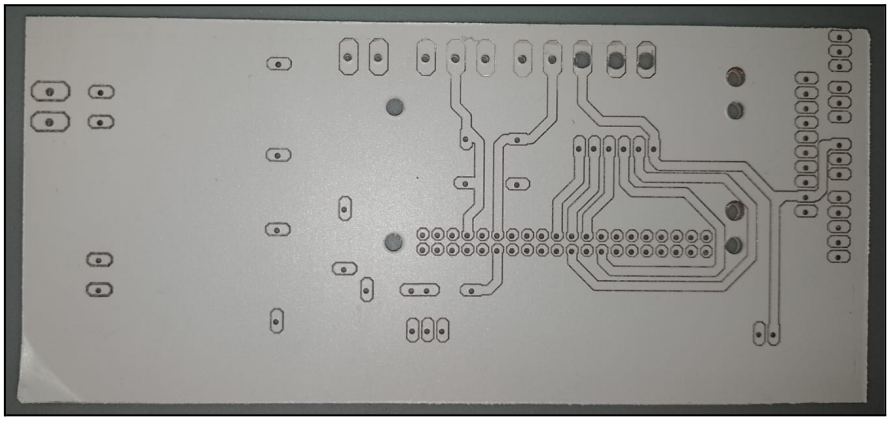


Figura 54: Impresión capa Top de la PCB.
Fuente: Propia

7.2. Revelado y montaje de PCB

Una vez completada la eliminación del exceso de material, dejando únicamente las pistas conductoras y los pads necesarios, la placa de circuito impreso (PCB) adquiere la apariencia que se muestra a continuación:

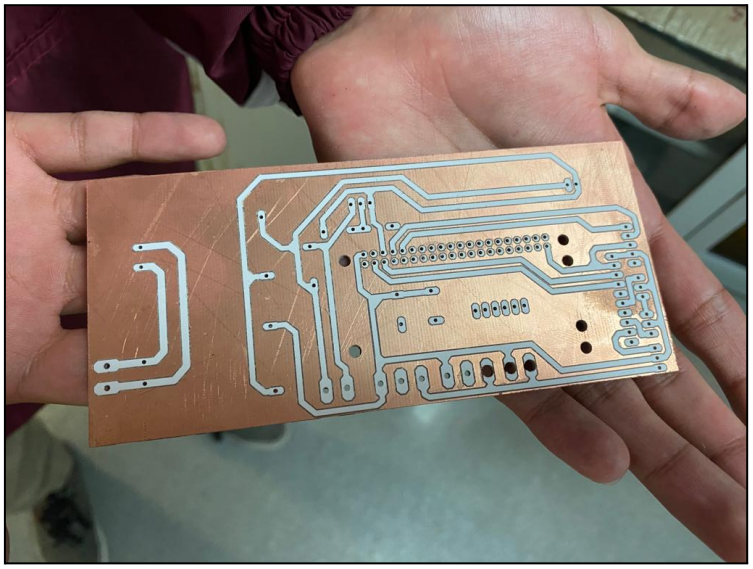


Figura 55: Impresión capa Bottom de la PCB.
Fuente: Propia

El exceso de cobre remanente se elimina mediante un proceso de lavado con ácido en la PCB. Este procedimiento se lleva a cabo utilizando una máquina que calienta el ácido y lo aplica a la placa. En esta ocasión, se sumergió la PCB en el baño ácido durante un período de 4 minutos, permitiendo así la eliminación completa del cobre expuesto.

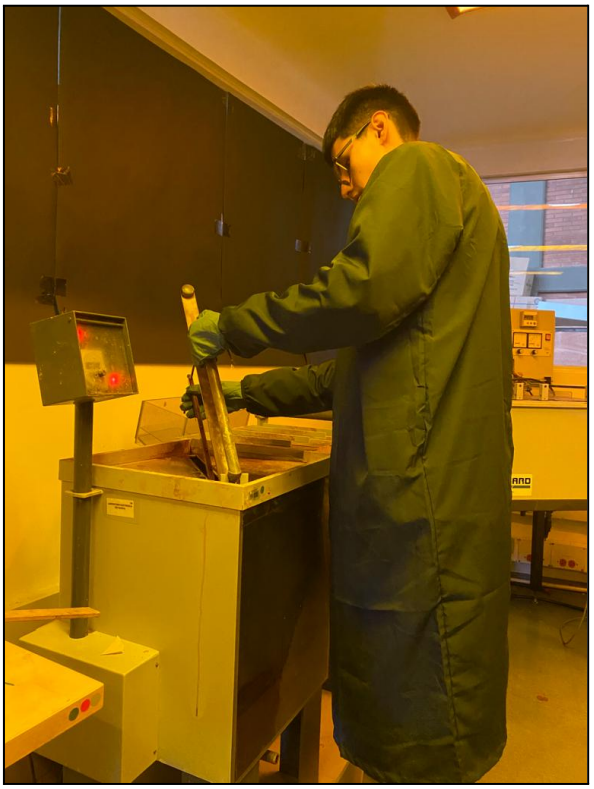


Figura 56: Proceso de lavado en ácido.
Fuente: Propia

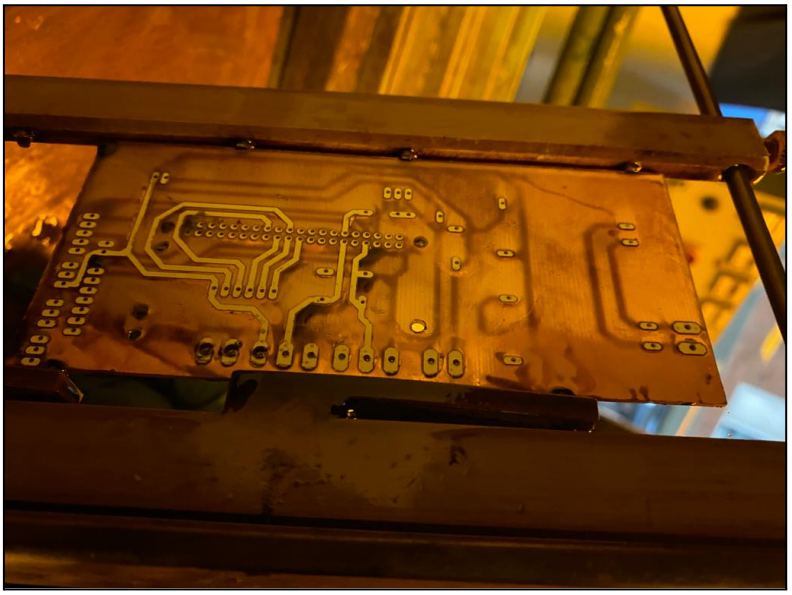


Figura 57: PCB durante el proceso de lavado en ácido.
Fuente: Propia

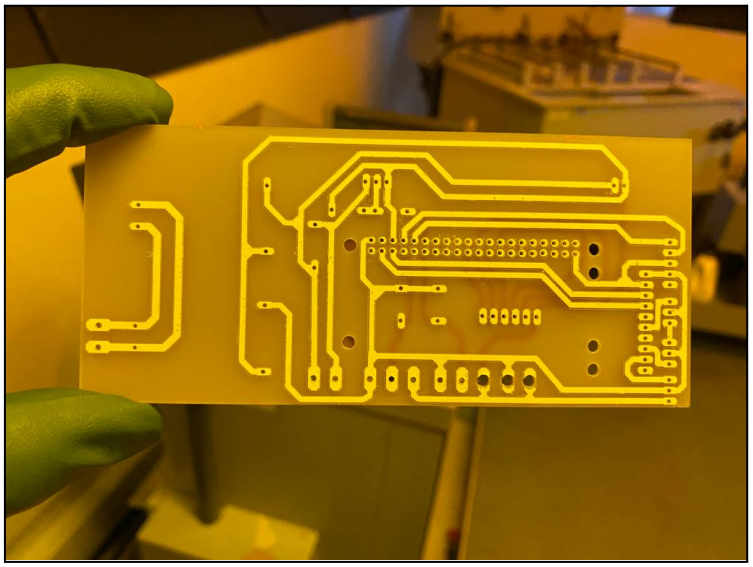


Figura 58: PCB después del proceso de lavado en ácido.
Fuente: Propia

A continuación, se procede a aplicar la máscara de soldadura en ambas caras de la PCB. Este proceso implica el corte a medida y la adherencia cuidadosa de la máscara de soldadura sobre la superficie de la placa. La máscara de soldadura cumple una función crítica al proteger las áreas que no deben ser soldadas durante el proceso de ensamblaje, asegurando así la precisión y confiabilidad de las conexiones eléctricas en la PCB.

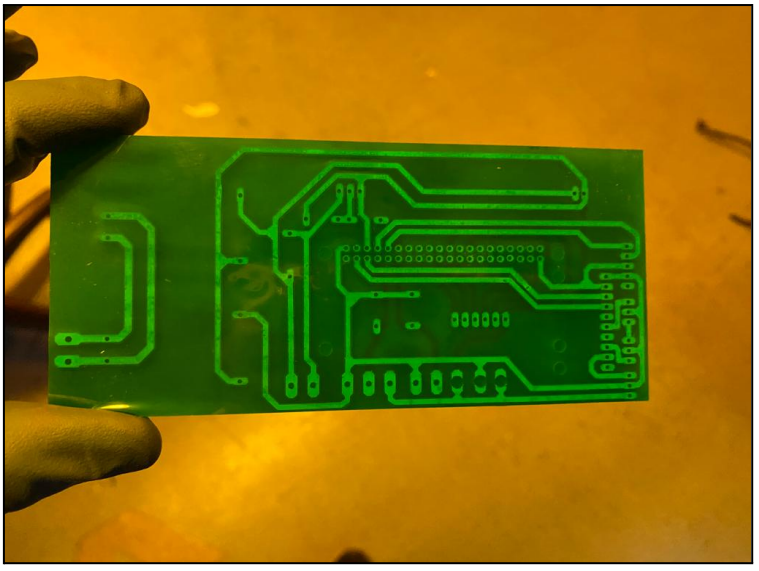


Figura 59: PCB con máscara de soldadura.
Fuente: Propia

Para asegurar una adhesión óptima de la máscara de soldadura a la PCB, se emplea una máquina laminadora especializada. Esta máquina aplica calor de manera controlada, lo

que permite que la película de la máscara de soldadura se funda y se adhiera de manera uniforme y firme a la superficie de la placa. Este proceso de laminación es esencial para garantizar que la máscara de soldadura se mantenga en su lugar de manera segura, protegiendo las áreas designadas de la PCB durante el proceso de soldadura, y contribuyendo así a la integridad y calidad de la soldadura en el montaje final del circuito.



Figura 60: Laminación de máscara de soldadura.
Fuente: Propia

Como último paso en este proceso de preparación, la PCB se introduce con cuidado en una unidad especializada de exposición al vacío. Esta etapa reviste gran importancia, ya que el vacío elimina cualquier posible contaminante o partícula presente en la superficie de la placa, asegurando así un entorno limpio y controlado para las etapas posteriores del proceso de fabricación.

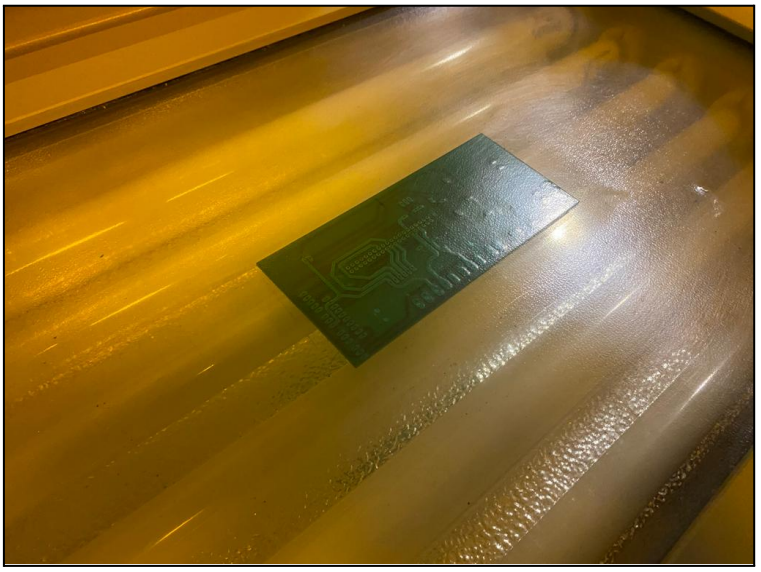


Figura 61: PCB dentro de la unidad de exposición a vacío.
Fuente: Propia



Figura 62: Unidad de exposición a vacío.
Fuente: Propia

Una vez que la PCB ha pasado por todas las etapas previas de preparación y tratamiento, se procede a la fase de montaje final y soldadura de los componentes. Este proceso implica la colocación precisa de los componentes electrónicos en sus ubicaciones designadas en la PCB, asegurando una alineación y orientación correctas. Luego, se utiliza un equipo de soldadura para fundir y conectar eléctricamente estos componentes con las pistas de cobre en la placa.

Es importante destacar que este paso requiere un alto grado de precisión y destreza, ya

que la calidad de las conexiones soldadas tiene un impacto directo en la funcionalidad y durabilidad del dispositivo electrónico final. El uso de técnicas avanzadas de soldadura, como soldadura por reflujo o soldadura selectiva, garantiza una unión sólida y confiable entre los componentes y la PCB.

Una vez completado este proceso de ensamblaje, la PCB se somete a pruebas de funcionamiento y calidad para verificar su correcto funcionamiento y cumplimiento de las especificaciones requeridas.

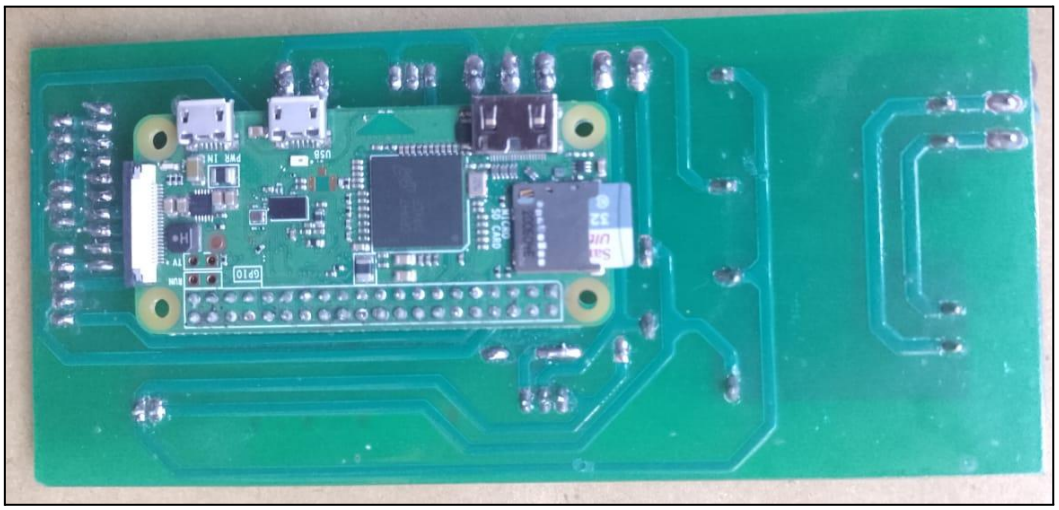


Figura 53: Montaje con PCB vista de Buttom.
Fuente: Propia

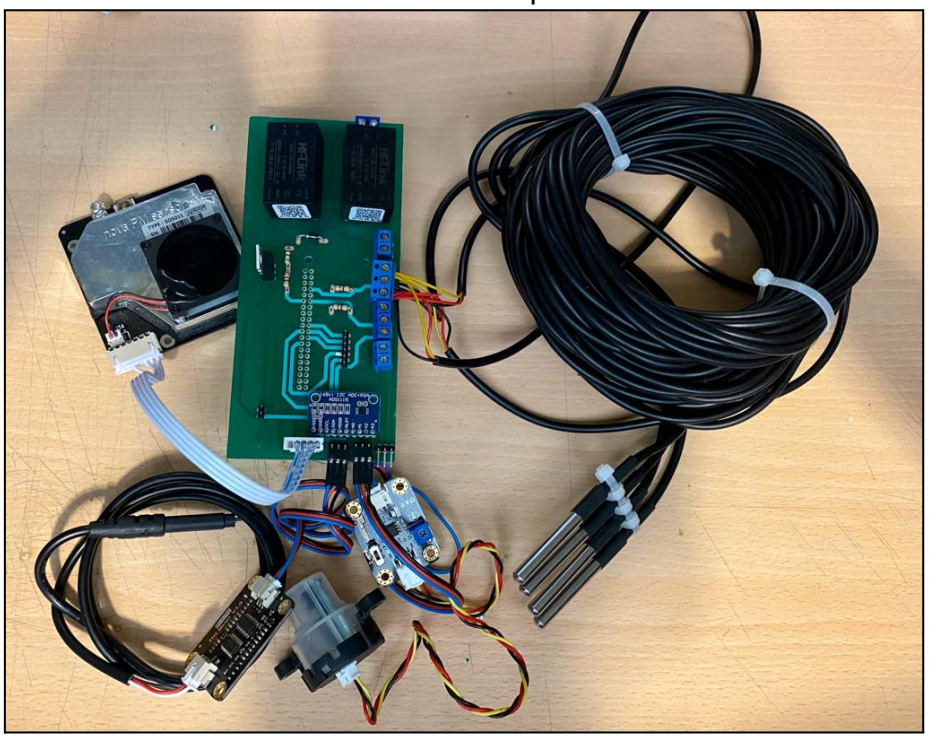


Figura 64: Montaje con PCB vista Top.
Fuente: Propia

8. Conclusiones

- Tras un análisis, se ha comprobado que un fondo de pantalla completamente blanco reduce la vida útil de la Pantalla 5 inch HDMI LCD de Raspberry Pi en un 30% a 45%. Optamos por un tono ligeramente más opaco que mantiene todas las funcionalidades y visibilidad, preservando la durabilidad del componente.

- La incorporación de un sistema de gestión de errores a través del código, que aborda los posibles fallos en los sensores durante la monitorización de variables en el control del cultivo hidropónico, asegura la robustez del sistema ante eventos sensoriales adversos. Esto permite que el sistema siga funcionando sin interrupciones sustanciales y que, al mismo tiempo, emite notificaciones de alerta para facilitar el mantenimiento técnico oportuno.

- La combinación de la tecnología Qt Designer junto con Python simplificó considerablemente el proceso de construcción de la interfaz gráfica y su gestión. La inversión de tiempo requerida para familiarizarse con estas herramientas fue notablemente reducida en comparación con opciones más complejas. Esto resultó en una eficiencia significativa en el desarrollo de la interfaz gráfica y su funcionamiento.

- La elección de construir el domo geodésico con una frecuencia baja y utilizando una porción de esfera del 50% se basó en la estrategia de optimización de costos y simplificación del ensamblaje. Esta decisión se tomó en consideración de las limitaciones de conocimiento especializado y recursos financieros disponibles en el proyecto. Al adoptar esta aproximación, pudimos mitigar desafíos significativos relacionados con la construcción de estructuras geodésicas, al tiempo que garantizamos la viabilidad económica del proyecto.

9. Bibliografía

[1] Lee, J. H., Park, H. J., Lee, J. M., Kim, K. H., & Hwang, Y. H. (2021). Evaluation of growth and nutritional properties of lettuce under hydroponic cultivation with different electrical conductivity and nutrient concentration. Journal of the Korean Society for Applied Biological Chemistry,

[2] "Cultivos sin tierra: la hidroponÃa revoluciona la gastronomÃa". BBVA NOTICIAS. https://www.bbva.com/es/sostenibilidad/cultivos-sin-tierra-la-hidroponia-revolucion a-la-gastronomia/ (accedido el 27 de abril de 2023).

[3] Cómo funciona un sistema hidropónico y TvAgro. Cómo funciona un sistema hidropónico - TvAgro por Juan Gonzalo Angel Restrepo. (8 de febrero de 2023). Accedido el 27 de abril de 2023. [Video en línea]. Disponible: https://www.youtube.com/watch?v=cuMV43ov7zA

[4] "La Hidroponía: Cultivos sin Suelo | Intagri S.C." Intagri - Cursos Agrícolas - Artículos de agronomía. https://www.intagri.com/articulos/horticultura-protegida/la-hidroponia-cultivos-sin-s uelo (accedido el 27 de abril de 2023).

[5] “Buy a Raspberry Pi Zero W â ‘ Raspberry Pi”. Raspberry Pi. Accedido el 14 de septiembre de 2023. [En línea]. Disponible: https://www.raspberrypi.com/products/raspberry-pi-zero-w/

[6] “DS18B20 sensor de temperatura para líquidos con Arduino”. Programar fácil con Arduino. Accedido el 14 de septiembre de 2023. [En línea]. Disponible: https://programarfacil.com/blog/arduino-blog/ds18b20-sensor-temperatura-arduin o/

[7] “Qué es Redux: todo lo que debes saber sobre esta biblioteca”. Hardware libre. Accedido el 14 de septiembre de 2023. [En línea]. Disponible: https://www.hwlibre.com/que-es-redux/

[8] “Analog TDS Sensor Meter for Arduino / ESP32 / Raspberry Pi - DFRobot Wiki”. wiki_EN_content_page-DFRobot. Accedido el 14 de septiembre de 2023. [En línea]. Disponible: https://wiki.dfrobot.com/Gravity__Analog_TDS_Sensor___Meter_For_Arduino_SK U__SEN0244

[9] “Turbidity_sensor_SKU__SEN0189-DFRobot”. wiki_EN_content_page-DFRobot. Accedido el 14 de septiembre de 2023. [En línea]. Disponible: https://wiki.dfrobot.com/Turbidity_sensor_SKU__SEN0189.

[10] “ADS1115 convertidor analógico digital ADC para Arduino y ESP8266”. Programar fácil con Arduino. Accedido el 14 de septiembre de 2023. [En línea]. Disponible: https://programarfacil.com/blog/arduino-blog/ads1115-convertidor-analogico-digital -adc-arduino-esp8266/

[11] “Detector de partículas SDS011 | Tienda y Tutoriales Arduino”. Indice de tutoriales Arduino | Tienda y Tutoriales Arduino. Accedido el 14 de septiembre de 2023. [En línea]. Disponible: https://www.prometec.net/detector-de-particulas-sds011/

[12] S. Vásquez López, «Agro-Tecnología educativa: Sistema de adquisición de datos climáticos para la docencia», ava. sis. inf, vol. 7, n.º 2, pp. 37–46, may 2010.

[13] Ibarra Daza, L. (2012). Diseño e implementación de un sistema de adquisición de datos con sensores: 808h5v5, mcp9700a, watermark, mpx4115a, sq-110, comunicación mediante protocolo zigbee y mysql, para un cultivo de tomate en Sutamarchán, Boyacá (Colombia). Universidad Santo Tomás

[14] 5inch HDMI LCD - Waveshare Wiki. (s.f.). Waveshare Electronics. https://www.waveshare.com/wiki/5inch_HDMI_LCD

[15] A. D. Boursianis et al., "Smart Irrigation System for Precision Agriculture—The ARETHOU5A IoT Platform," in IEEE Sensors Journal, vol. 21, no. 16, pp. 17539-17547, 15 Aug.15, 2021, doi: 10.1109/JSEN.2020.3033526

[16] Qt Designer Manual. (s.f.). Qt Documentation | Home. https://doc.qt.io/qt-6/qtdesigner-manual.html

[17] Excel Hecho Fácil. (2021, 21 de diciembre). Crea este Dashboard desde cero completamente en Excel ¡Impresiona a tu jefe! [Video]. YouTube. https://www.youtube.com/watch?v=03ptcgg9Rs8

[18] Vargas, G. G. (s.f.). Unión sin conectores | Domos geodesicos, Cúpula de casa, Construccion. Pinterest. https://co.pinterest.com/pin/670051250786068719/

[19] Geodesic dome calculator - Acidome.ru. (s.f.). https://acidome.com/lab/calc/#5/12_GoodKarma_3V_R2.5_beams_100x20