Proyecto Fin de Grado
Grado en Ingeniería Aeroespacial

Sistema de control de temperatura y gestión de datos vía Internet con aplicación a sistemas embarcados en aeronaves

Autor: Fernando León Labella

Tutor: Marta Laguna García

Dpto. Teoría de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021

Proyecto Fin de Grado
Ingeniería Aeroespacial

Sistema de control de temperatura y gestión de datos vía Internet con aplicación a sistemas embarcados en aeronaves

Autor:
Fernando León Labella

Tutor:
Marta Laguna García
Profesora titular

Dpto. de Teoría de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2021

Proyecto Fin de Carrera: Sistema de control de temperatura y gestión de datos vía Internet con aplicación a sistemas embarcados en aeronaves

Autor: Fernando León Labella

Tutor: Marta Laguna García

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

A mis padres

Agradecimientos

En primer lugar, agradecer a la profesora Marta Laguna García por su ayuda y disposición prestadas de cara a la realización de este proyecto.

Agradecer también a mis compañeros de clase, por su compañía durante estos años.

A mi familia y amigos de mi pueblo, Fuentes de Andalucía, por escucharme cuando no todo iba bien, ayudarme cuando lo necesitaba y divertirnos cuando tocaba.

Por último, agradecer a mis padres, porque sin ellos nada hubiera sido posible.

Fernando León Labella

Sevilla, 2021

Resumen

En este trabajo se realiza el diseño tanto a nivel de hardware como de software de un sistema de control de temperatura, así como un monitoreo de los datos y un control de la respuesta.

Para ello, en primer lugar se hará una descripción de los elementos hardware y software utilizados para el diseño. A continuación, se comentarán detalladamente las dos estrategias seguidas para la realización de este sistema, las cuales son:

- Control y monitoreo sin conexión a red, es decir, presentación de los datos en una pantalla OLED y un control de la respuesta mediante pulsadores.
- Control tanto de los datos como la respuesta del sistema y gestión del historial de muestras a través de una página web o una aplicación Android.

Finalmente, se comentarán posibles mejoras, así como otras posibles aplicaciones del sistema diseñado.

Abstract

This work focusses on desing of a temperature control at hardware and software level, as well as data monitoring and response control.

To this aim, first, a description of the hardware and software elements using for desing will be made. Next, the two strategies followed for the realization of this system will be discussed in detail. The two strategies are:

- Control and monitoring without network connection, that is, presentation of the data on an OLED screen and control of the response by means of pushbutton.
- Data and response control of the system, and management of the historical data through a website or an Android app.

Finally, a discussion about posible improvements and other application of the designe system.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1 INTRODUCCIÓN

La electrónica es una rama de la física aplicada que comprende la física, la ingeniería, la tecnología y las aplicaciones que tratan con la emisión, el flujo y el control de los electrones (u otras partículas cargadas eléctricamente) en el vacío y la materia [1].

Tanto en el mundo industrial como en la vida cotidiana el uso de sistemas y componentes electrónicos es cada vez más frecuente debido a sus numerosas aplicaciones, por tanto, el desarrollo de sistemas electrónicos es una de las ramas de la ingeniería con mayor proyección de futuro.

Podemos destacar los siguientes campos en los que la electrónica tiene un papel fundamental [2]:

- Control y automatización en industrias
- Domótica
- Vehículos
- Medicina
- Energías alternativas
- Informática y enseñanzas
- Juegos
- Limpieza y bienestar
- Seguridad y protección
- Comunicaciones

Un esquema sencillo de un sistema electrónico podría se el siguiente:

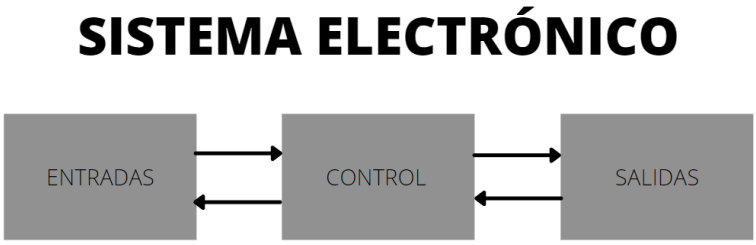

Figura 1-1. Esquema sistema electrónico genérico

El sistema toma medidas de sensores o de indicaciones, se procesan tomándose decisiones y se aplican unas actuaciones, convirtiendo voltajes en señales útiles desde el punto de vista físico.

En este trabajo se realizará el diseño tanto a nivel de hardware como de software de un sistema de control de temperatura, así como un monitoreo de los datos y un control de la respuesta. Este sistema de control es aplicable al mundo aeroespacial, en especial a los sistemas embarcados que son propensos a sufrir altas temperaturas y pueden dañarse, dando tanto una respuesta automática, como la posibilidad de respuesta manual por un usuario.

La respuesta podría ser la activación un sistema de refrigeración para corregir excesos de temperatura en diversos sistemas embarcados, aunque en este proyecto no se tratará dicho sistema de refrigeración y solo se mostrará mediante un diodo LED la activación del mismo.

Se destaca el uso de la electrónica en el mundo aeroespacial, donde la electrónica toma el nombre de aviónica. En este sector es importante destacar la tendencia "*More Electric Aircraft*", donde se pretende fomentar el uso de sistemas eléctricos y electrónicos. Esto se hace con el objetivo de disminuir los costos de operación y mantenimiento, aumentar la fiabilidad y reducir emisiones de gases. Los avances tecnológicos en el campo de la electrónica de potencia, actuadores eléctricos, sistemas de control de vuelo, motores eléctricos, generación de energía y sistemas de conversión han definido la era "*More Electric Aircraft*" [3].

En este sentido, el proyecto cumple con la tendencia actual de la aeronáutica, ya que se diseña un sistema electrónico, cada vez más importante en el sector.

Imporante destacar también la comunicación del sistema con el usuario mediante la creación de una página web y de una aplicación Android, permitiendo el control a través de Internet. Se considera esto primordial en el desarrollo del trabajo debido a la cada vez más importancia de las comunicaciones, ya que ofrecen un sinfín de posibilidades tanto en la vida cotidiana como en el mundo industrial, permitiendo una globalización a gran escala cada vez más importante y notable.

Por último, destacar el gran número de sensores presentes en aeronaves, desde altitud, presión, temperatura, posición, detección de hielo, etc.

2 DESCRIPCIÓN DE ELEMENTOS USADOS

En este capítulo se expondrán las distintas herramientas y elementos hardware y software que se han utilizado para el diseño e implementación del sistema.

2.1. Hardware

2.1.1 Arduino UNO

Arduino UNO es una placa de microcontrolador basada en el microcontrolador ATmega328P. Cuenta con 14 pines de entrada/salida digital, 6 entradas analógicas, una conexión USB, un conector de alimentación, un encabeado ICSP y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador [4].

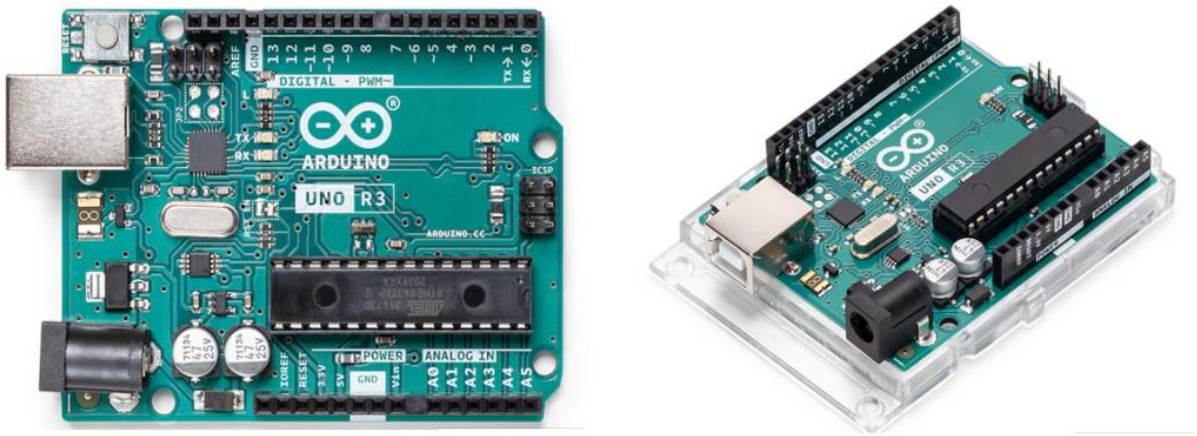

Figura 2-1.Arduino UNO

A continuación se muestra una tabla con las especificaciones del Arduino UNO (si la especificación pertenece al microcontrolador se indicará en la tabla [5].

Microcontrolador	ATmega328P
Tensión de funcionamiento	5 V
Voltaje de entrada (recomendado)	7-12 V
Voltaje de entrada (límite)	6-20 V
Pines de E/S digitales	14
Pines de E/S digitales pwm	6
Pines de entrada analógica	6
Corriente CC por pin de E/S	20 mA
Corriente CC por pin de 3.3V	50 mA
Memoria flash	32 KB (ATmega328P) de los cuales 0.5 KB utiliza el gestor de arranque
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
Led_Builtin	13
Largo	68.6 mm
Ancho	53.4 mm
Peso	25 g

Tabla 2–1. Especificaciones Arduino UNO.

2.1.2 Ethernet Shield W5100

Se trata de un shield compatible con Arduino UNO, que permite a este conectarse a Internet mediante un cable de Ethernet, pudiéndose utilizar como servidor o cliente.

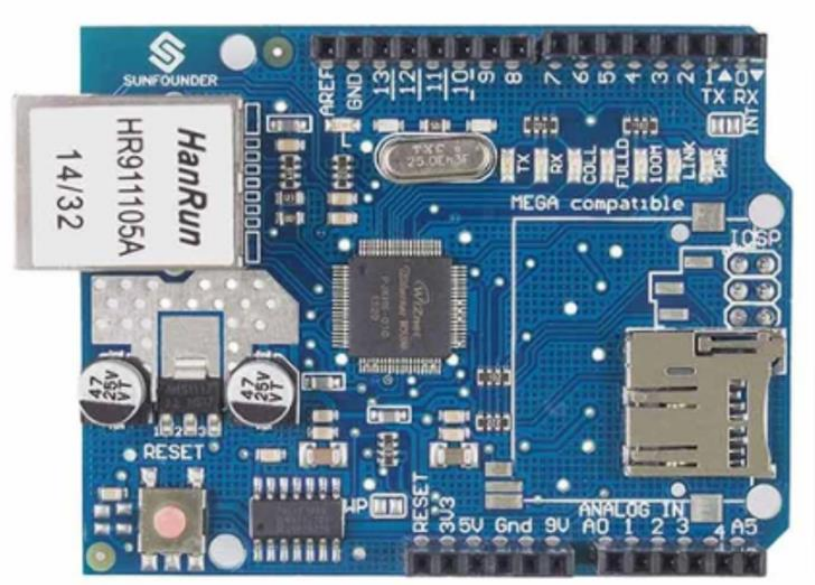

Figura 2-2. Ethernet Shield W5100

Este shield es compatible directamente con Arduino UNO mediante la librería Ethernet oficial de Arduino y se basa en el chip Ethernet Wiznet [6].

Se muestra la tabla de especificaciones del Ethernet Shield W5100 (Tabla 2.2) que se ha utilizado en este proyecto [7].

Marca	SUNFOUDER
Fabricante	SUNFOUDER
Dimensiones del producto	17.78 x 14.73 x 3.81
Número del producto	FBA_DE W5100
Capacidad de la memoria RAM	8 KB
Tipo de memoria	EEPROM
Interfaz del hardware	MicroSD, Ethernet, Tarjeta Secure Digital
Peso	40 g

Tabla 2–2. Especificaciones Ethernet Shield.

2.1.3 Pantalla OLED Diplay 0.96”

Pantalla OLED de 0.96 pulgadas de 128 x 64 píxeles de color blanco y compatible con Arduino mediante las

librerías Adafruit_SSD1306 y Adafruit_GFX. Se conecta a la placa mediante 4 pines.

Figura 2-3. Pantalla OLED 0.96''

Se muestran a continuación las especificaciones de la pantalla OLED Display 0.96'' [8].

Voltaje de alimentación (VCC)	5 V
Voltaje de alimentación máximo (VCC)	3.3 V
Voltaje de alimentación mínimo (VCC)	5.5 V
Controlador IC	SSD108Z
Color de pantalla	Blanco
Matriz de puntos	128 x 64
Tamaño del panel	26.7 x 19.26 mm
Área activa	21.74 x 11.175 mm
Paso de puntos	0.17 x 0.175 mm
Tamaño de punto	0.15 x 0.15 mm
Temperatura de funcionamiento	-20 – 70 ºC

Tabla 2–3. Especificaciones pantalla OLED

2.1.4 Sensores

Pese a que el sistema está pensado y orientado hacia el control de temperatura, se ha utilizado en el desarrollo un sensor de luz por viabilidad de operación, siendo más sencillo aumentar y disminuir la luz que hacerlo con la temperatura, por lo que se describirá tanto el sensor de luz utilzado como un posible sensor de temperatura a

utilizar.

En cuanto al sensor de luz se ha utilizado una fotoresistencia LDR GM5539 5mm, la cual nos da un valor entre 0 y 1023 en función de la luz que recoge.

Figura 2-4. Fotoresistencia LDR-GM5539

Las especificaciones son las siguientes [9]:

Voltaje máximo	150 VDC
Potencia máxima	100 mW
Temperatura de funcionamiento	-30 – 70 ºC
Pico espectral	540 nm
Resistencia a la luz	50 -100 KΩ
Impedancia oscura	5 MΩ
Tiempo de respuesta	Up – 20 ms, Down – 30 ms

Tabla 2–4. Especificaciones fotoresistencia LDR-GM5539

En cuanto a un sensor de temperatura sería ideal que dicho sensor sea operable en un amplio rango de temperaturas para hacer operable el sistema en más sistemas a controlar, sin que la temperatura de este último sea limitante para el diseñado en este proyecto.

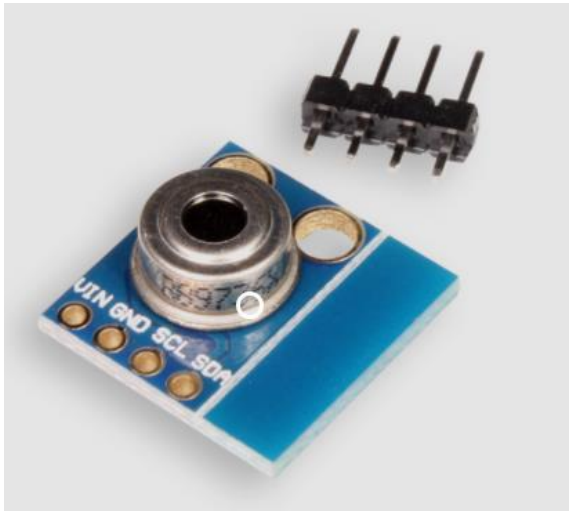

Figura 2-5. Ejemplo de sensor de temperatura

2.1.5 Resistencias, diodos LED, cables y pulsadores

Además de todo lo anteriormente mencionado, también se han utilizado algunos elementos mas comunes en el montaje de un sistema electrónico. En este caso, no se considera necesario el desarrollo de las especificaciones de cada uno de estos elementos debido a que hay un gran conocimiento general acerca de dichos elementos.

Figura 2-6. Diodo LED

Figura 2-7. Cables

Figura 2-8. Resistencias

Figura 2-9. Pulsador

2.2 Software

2.2.1 Arduino IDE

Mediante este software de código abierto, se facilita la escritura de código así como la carga de dicho código en la placa Arduino (este software es válido para cualquier placa de Arduino). Se utiliza para la escritura del código el lenguaje C++ [10].

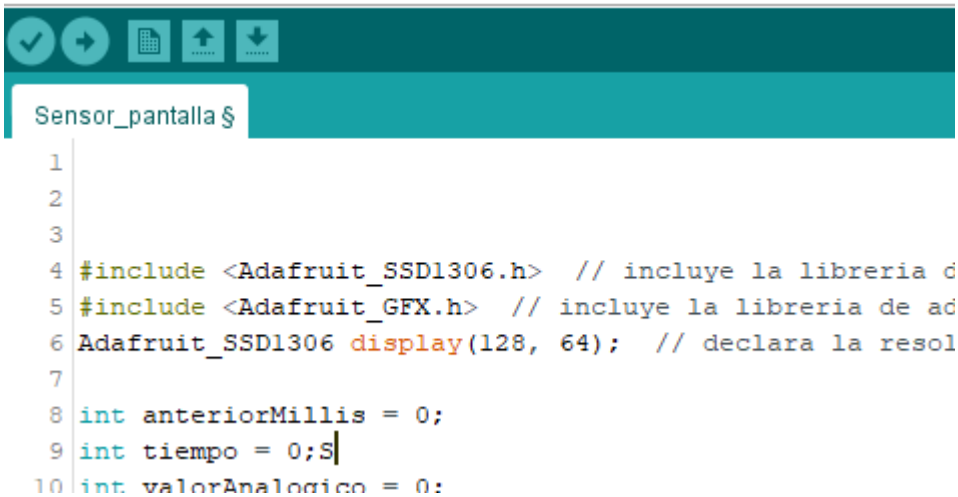

Figura 2-10. Interfaz Arduino IDE

La figura 2-10 es un ejemplo de como se ve la interaz del software. Las funciones que se ven en la parte superior son las más básicas y son respectivamente (y en orden de aparición) las de compilar, subir, nuevo proyecto, cargar proyecto y guardar proyecto. Además de todo esto tiene opciones para editar la interfaz, editar la fuente, incluir librerías, serial plotter (representa los datos del puerto serie)…

2.2.2 Microsoft Excel

Microsoft Excel es una hoja de cálculo desarrollada por Microsoft. Cuenta con cálculo, gráficas, tablas calulares, así como con un leguaje de programación macro llamado Visual Basic.

En las últimas versiones se marcó como límite de columnas por hoja de cálculo en 16384 y el límite de filas en 1048576 dando un total de 17179869184 celdas. El máximo de hojas por libro es de 1024 [11].

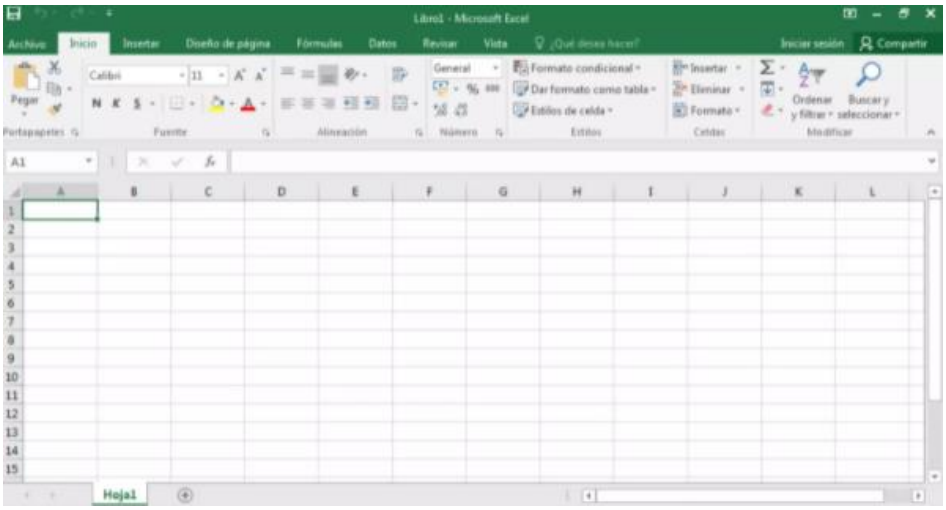

Figura 2-11. Interfaz Microsoft Excel

Se muestra en la figura 2-11 la interfaz de Excel.

Es importante destacar que la gran cantidad de funciones que Excel ofrece permite dar mucha información de los datos almacenados, y se verá que esto será de gran utilidad en la realización de este proyecto. Este tratamiento de los datos que permite realizar Excel hace posible sacar conclusiones más precisas, evitar dar por válidas suposiciones iniciales, detectar problemas y clasificar de forma óptima los problemas.

La decisión de utilizar Excel frente a otros gestores de datos se debe a las ventajas que ofrece, como por ejemplo:

- Gran cantidad de funciones disponibles
- Cuadrícula versátil
- Gran capacidad de almacenamiento
- Posibilidad de insertar múltiples tipos de gráficos

Esta herramienta también permite crear históricos de datos, lo cual en el caso de la temperatura tiene bastante importancia, ya que esta degrada el funcionamiento de los dispositivos. Con el histórico y un adecuado proceso de estudio y análisis se puede alargar la vida útil de dispositivos, así como prevenir roturas y facilitar tareas de mantenimiento.

2.2.3 PLX-DAQ_R2

Este software realmente es un complemento para Microsoft Excel y consiste en una herramienta de datos desarrollada por Parallax que adquiere hasta 26 canales de datos de cualquier microcontrolador y los almacena en una columna de una hoja de Excel, por lo que realiza una función de almacenamiento de datos que pueden ser tomados de distintas formas, por ejemplo sensores, en tiempo real [12].

Este complemento puede conseguir desde la página de la desarrolladora Paralax (https://www.parallax.com/package/plx-daq/).

También tiene la ventaja de que marca los datos tomados con la hora (con formato hh: mm: ss), por lo que se puede hacer un seguimiento exhaustivo de la toma de datos.

Se muestra en la Fig. 2-12 la interfaz del software. Se ve la forma del almacenaje de datos, así como la ventana de control de los mismos. Se puede seleccionar el puerto de entrada, la velocidad en baudios, limpiar las columnas…

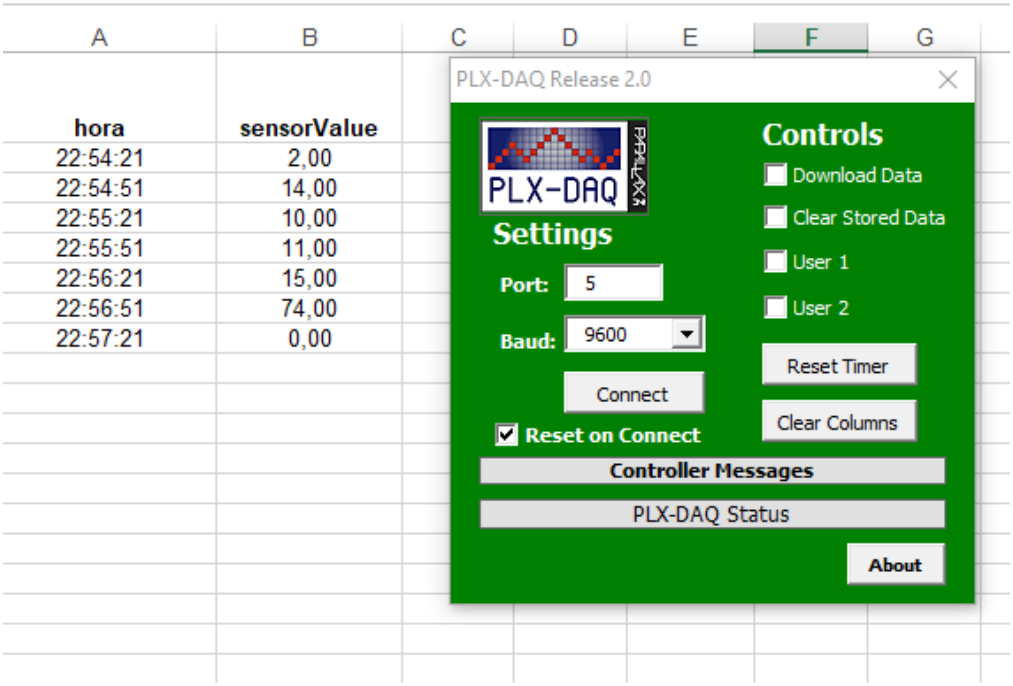

Figura 2-12. Interfaz PLX-DAQ y ejemplo de la toma de datos

2.2.4 Notepad ++

Notepad++ es un editor de texto y código fuente escrito en C++, que utiliza Win32 API y STL [13].

En este proyecto, se ha utilizado como editor de texto para la escrtura de un código en html para la creación de una página web.

Se ha seleccionado por encima de otros editores por la comodidad de su interfaz.

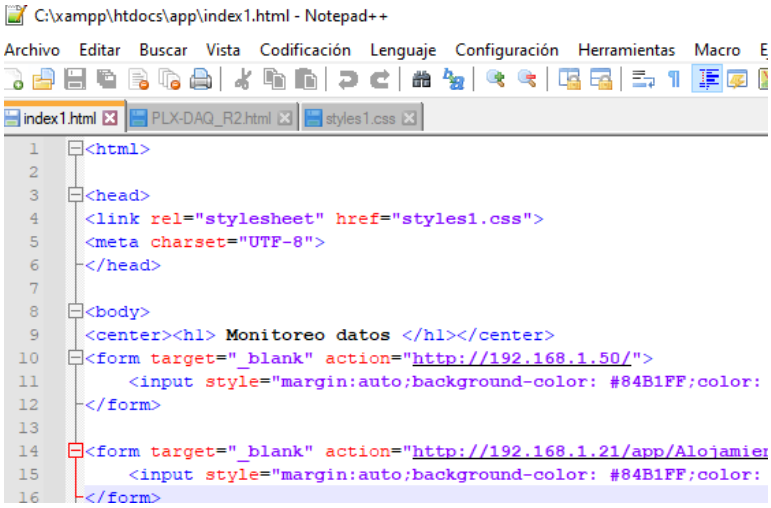

Figura 2-13. Interfaz Notepad++

2.2.5 Xampp

Xampp es un paquete de software libre, que consiste en el sistema de gestión de base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El programa se distribuye con la licencia GNU y actúa como un servidor web libre [14].

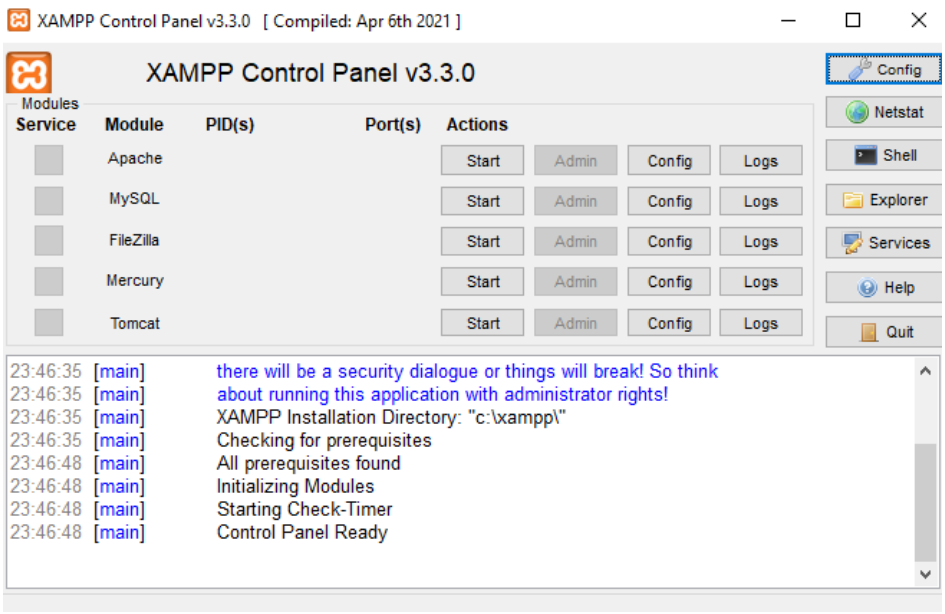

Figura 2-14. Interfaz Xampp

2.2.6 AppsGeyser

AppsGeyser consiste en una página web (https://appsgeyser.com/) que es capaz de generar aplicaciones Android sin necesidad de programar. En el caso de este proyecto, AppsGeyser crea una app a partir de una página web (introduciendo la URL de esta).

Se trata de una herramienta muy útil al permitir la creación de aplicaciones Android sin la necesidad de que estas sean programadas. Es un servicio gratuito y que permite la posibilidad de publicar las apps creadas con esta herramienta en Google Play.

Cuenta con diversas plantillas desde las que iniciar el trabajo.

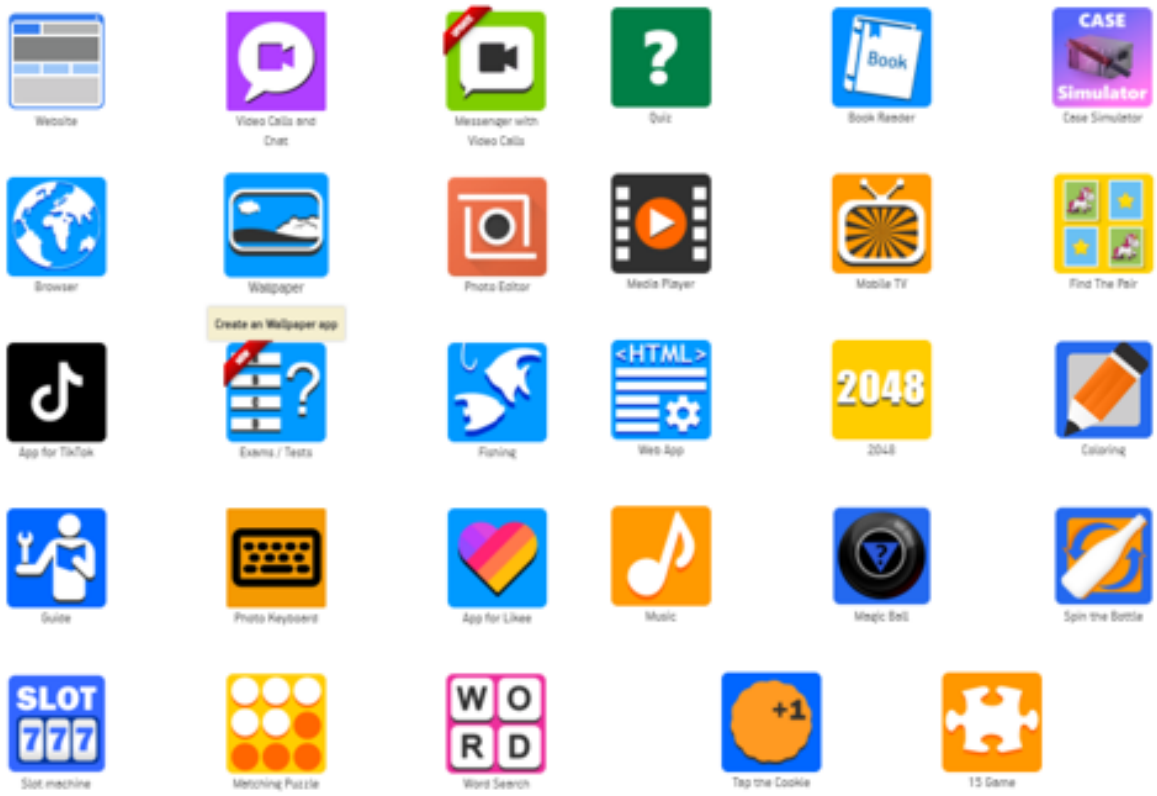

Figura 2-15. Plantillas AppsGeyser

Como se observa en la figura 2-15, AppsGeyser ofrece multitud de plantillas para crear las aplicaciones. Su principal limitación es que solo se puede elegir uno de estos contenidos para la app, no siendo combinables.

3 MONTAJE Y FUNCIONAMIENTO GENERAL

El objetivo de este proyecto es el diseño tanto a nivel de hardware como de software de un sistema de control de temperatura, así como un monitoreo de los datos y un control de la respuesta. Este sistema de control seria aplicable al mundo aeroespacial, en especial a los sistemas embarcados que son propensos a sufrir altas temperaturas y pueden dañarse, dando tanto una respuesta automática, como la posibilidad de respuesta manual por un usuario.

Los elementos usados en el desarrollo representan en ocasiones a otros elementos que por diversas razones no se han utilizado en este proyecto. En cuanto al sistema de respuesta (sistema de ventilación), se representa como un diodo LED, que se enciende o se apaga en función de lo que haría el sistema de ventilación. El sensor de temperatura se emula mediante una fotoresistencia (por facilidad de manejo y sencillez a la hora de hacer pruebas).

Se pretende que el sistema tenga respuesta automática y respuesta manual. Esto se hace para que el sistema sea autónomo y no necesite ser controlado por ningún usuario, pero también se pretende que si se desea, el usuario tenga poder de decisión sobre la respuesta. Se fuerza a que la decisión del usuario tenga prioridad ante la automática.

En este capítulo se mostrará y explicará el montaje del hardware del sistema. El sistema diseñado se tratará desde los dos siguientes puntos de vista para facilitar así el entendimiento del funcionamiento del propio sistema:

- Funcionamiento del sistema sin conexión a Internet
- Funcionamiento del sistema con conexión a Internet

Hay que destacar que en ambos casos se realiza prácticamente el mismo tratamiento de los datos, pero se ha diseñado de esta manera para hacer al sistema inmune a posibles fallos de red, tejiendo así una red de seguridad que lo proteja.

Pese a esto, el montaje es el mismo para ambos casos, aunque no se usen varios elementos en alguno de ellos.

Como objetivo final, se pretende que el sistema sea operable por un usuario que no necesariamente tenga que estar formado en programación o electrónica, ya que el sistema se diseña para que se maneje desde una interfaz web con funciones sencillas y entendibles.

Se destaca la posible inclusión del sistema en una red de tratamiento de distintos datos, gestionando así respuestas mas precisas al tener una mayor cantidad de información, teniéndose una visión más general del estado total del sistema controlado.

3.1 Montaje

Para el diseño del montaje se han utilizado placas protoboard para la conexión de los pines y el sistema se alimenta mediante la placa de Arduino (conectada a una fuente de alimenación externa). A su vez, el Shield Ethernet (anclado al Arduino) está conectado mediante un cable Ethernet a un router wifi.

El esquema de montaje es el siguiente:

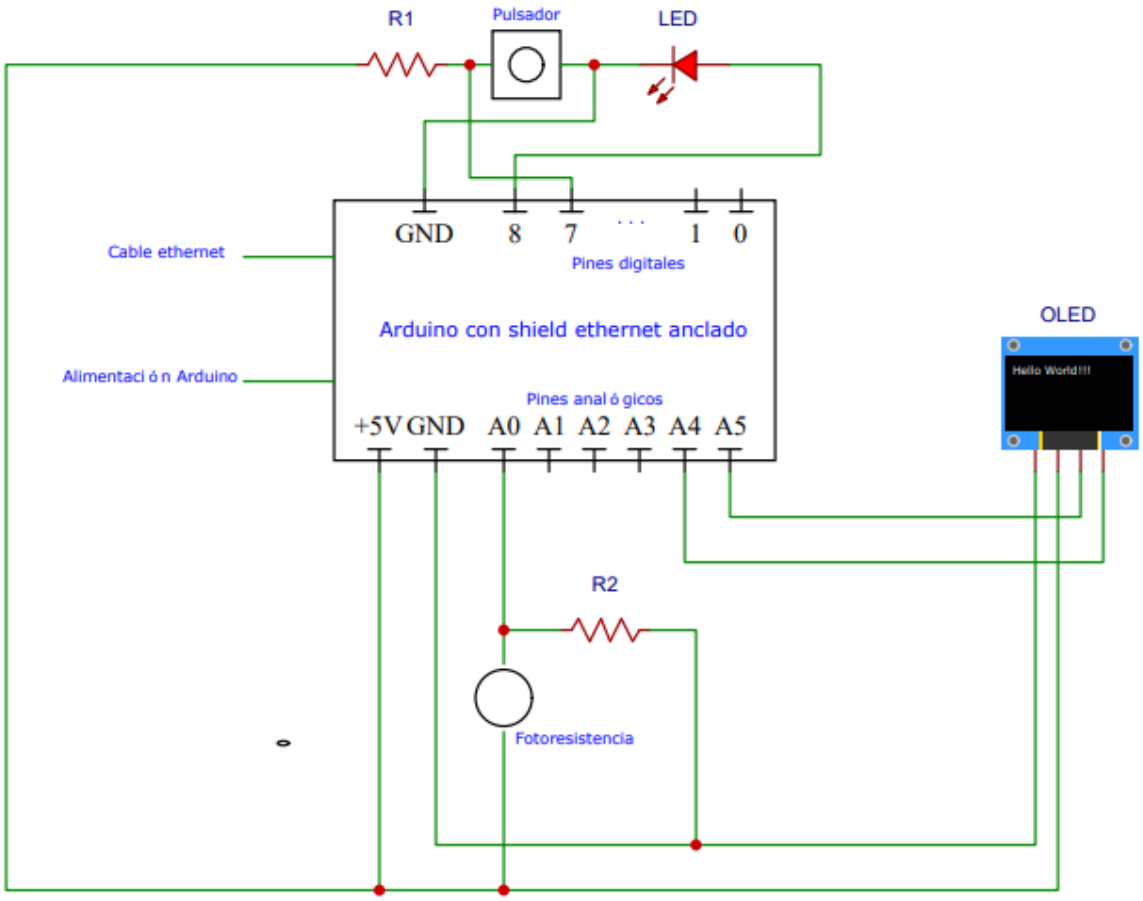

Figura 3-1. Esquema del montaje

Se comentan las conexiones de cada uno de los elementos:

- Placa de Arduino con Shield Ethernet anclado. Como se ha comentado anteriormente, están conectados a una fuente de alimentación externa y a un router wifi.
- El pin de alimentación (+5V en el esquema) de la placa (en adelante nos referiremos como placa en general al Arduino + Shield Ethernet) se conecta a la fotoresistencia, a la pantalla OLED y a la resistencia R1.
- Fotoresistencia. Conectada al pin de alimentación, así como a la resistencia R2 y al pin A0.
- Resistencia R2. Conectada a la fotoresistencia, al pin A0 y a tierra (GND) en el esquema.
- Pantalla OLED. Conectada al pin de alimentación (+5V), a tierra (GND) y a los pines A4 y A5.
- Resistencia R1. Conectada al pin de alimentación (+5V), al pin 7 y al pulsador.

- Pulsador. Conectado a la resistencia R1, al diodo LED y a los pines 7 y 8.
- Diodo LED. Conectado al pin 8, al pulsador y a tierra (GND).

El uso de los pines A0, A4, A5, 7 y 8 es debido a que son los que se han utilizado en programación, por lo que se podrían cambiar en el montaje si se cambian también adecuadamente en el código que se ha usado.

Se adjuntan algunas fotos del montaje.

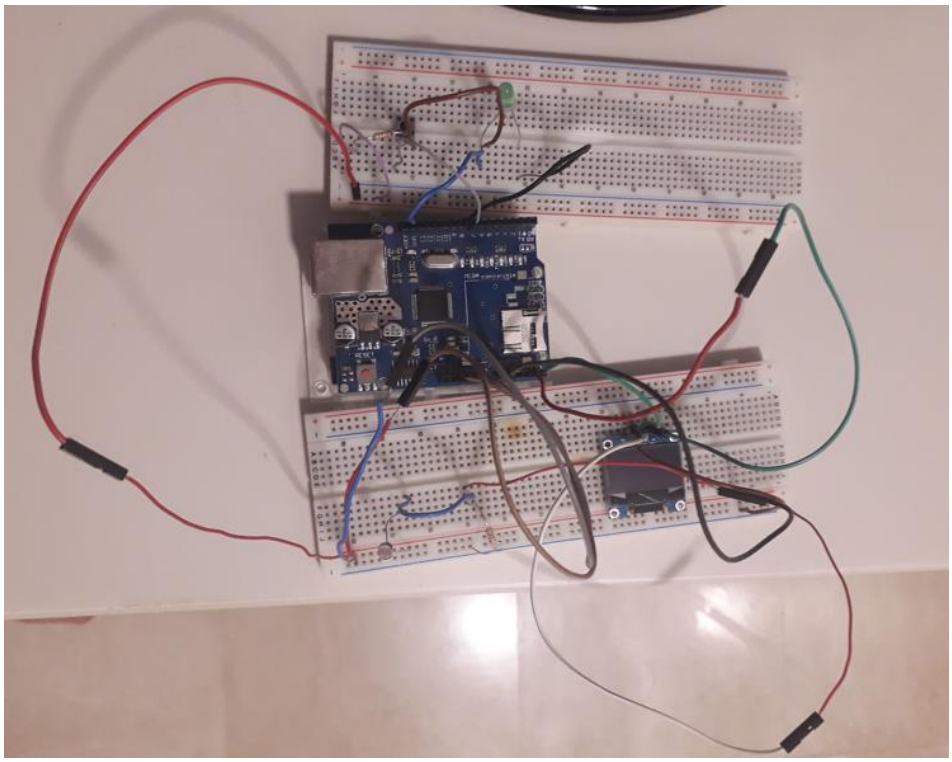

Figura 3-2. Foto 1 del montaje

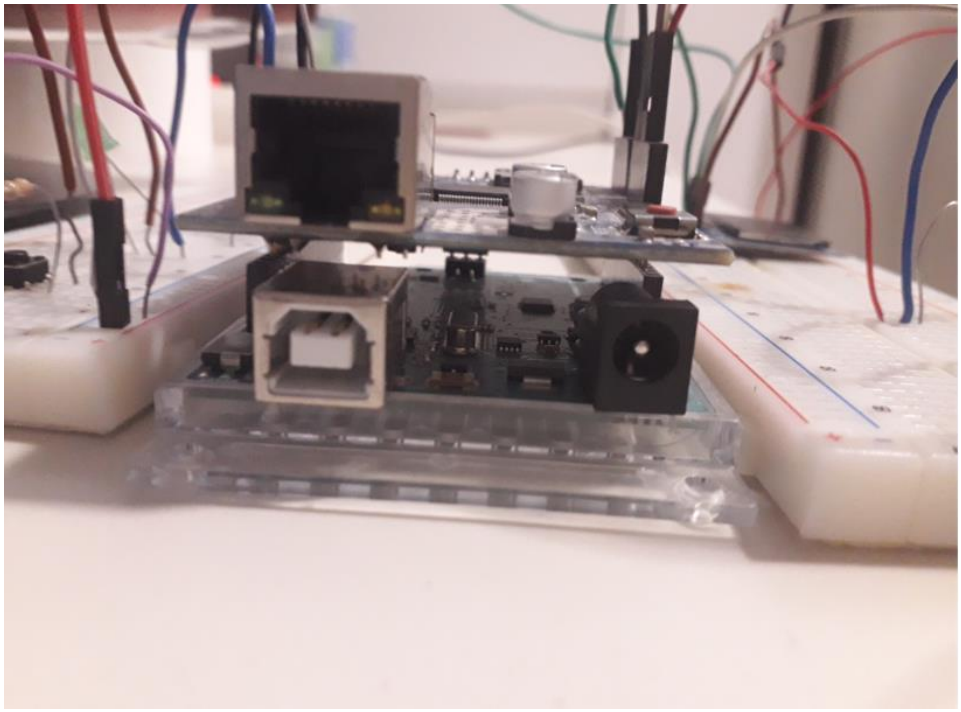

Figura 3-3. Foto 2 del montaje

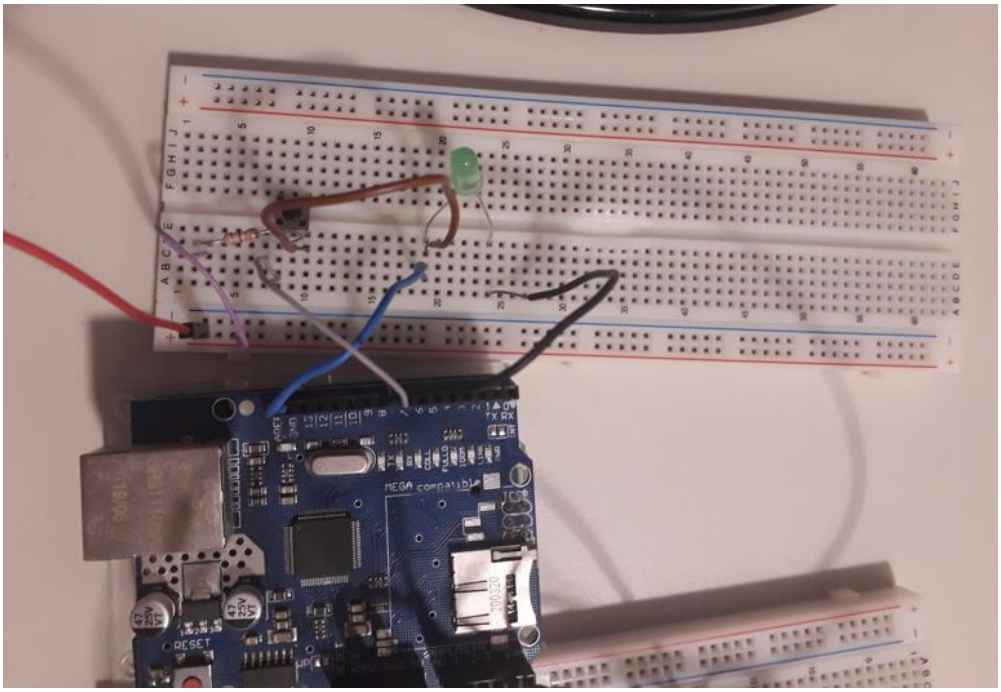

Figura 3-4. Foto 3 del montaje

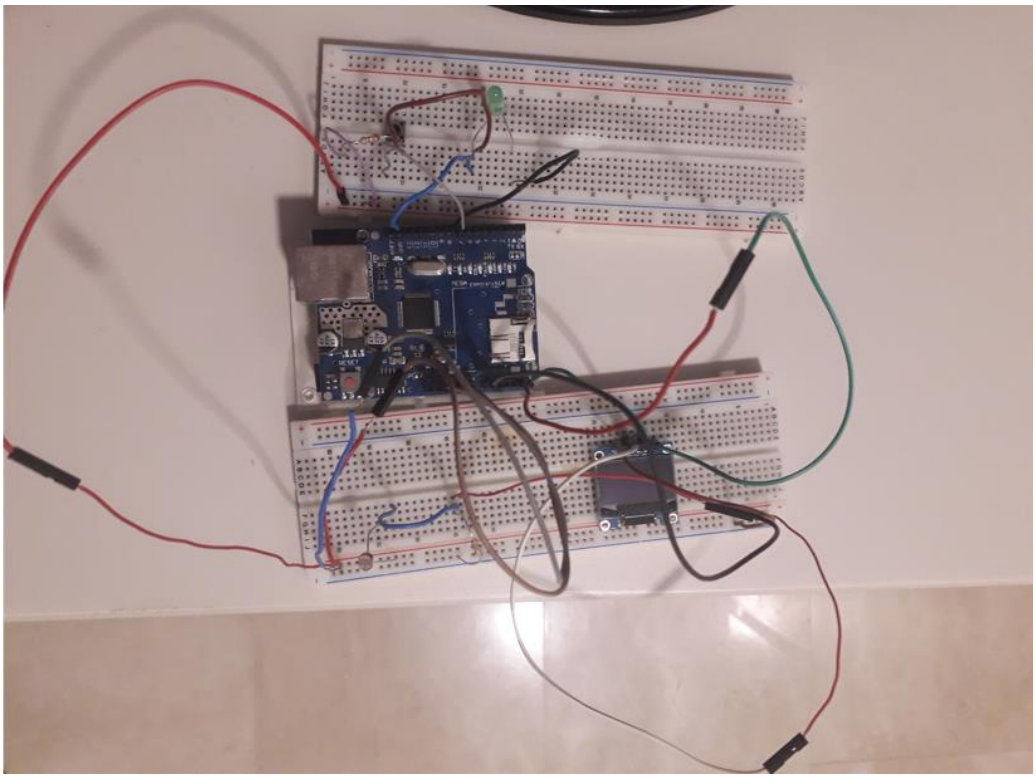

Figura 3-5. Foto 4 del montaje

3.2 Funcionamiento general

Como se comentó al principio del capítulo, se comentará el funcionamiento del sistema por separado para una mayor facilidad de entendimiento y comprensión.

Es importante añadir que en este capítulo se hará una descripción breve y esquemática y será en el siguiente capítulo donde se comentará detalladamente el funcionamiento, comentando resultados y analizando partes de los códigos de programación untilizados para un conocimiento más amplio.

3.2.1 Funcionamiento del Sistema sin conexión a Internet

Consiste en el monitoreo en las pantalla OLED de los datos tomados por el sensor, así como el control tanto automático como manual del sistema de respuesta. La respuesta se mostrará como un diodo LED pero podría indicar la activación de un sistema de refrigeración (que no se va a tratar en este proyecto) para proteger de temperaturas altas a los sistemas embarcados.

El usuario podrá ver la representación de los datos en la pantalla y podrá activar y desactivar mediante el pulsador el sistema de refrigeración cuando desee. Además, está programada la respuesta automática, que activará la refrigeración cuando la temperatura medida por el sensor supere cierto umbral elegible por el usuario.

Para esta parte del funcionamiento no haría falta el Shield Ethernet.

Se facilita un esquema del funcionamiento:

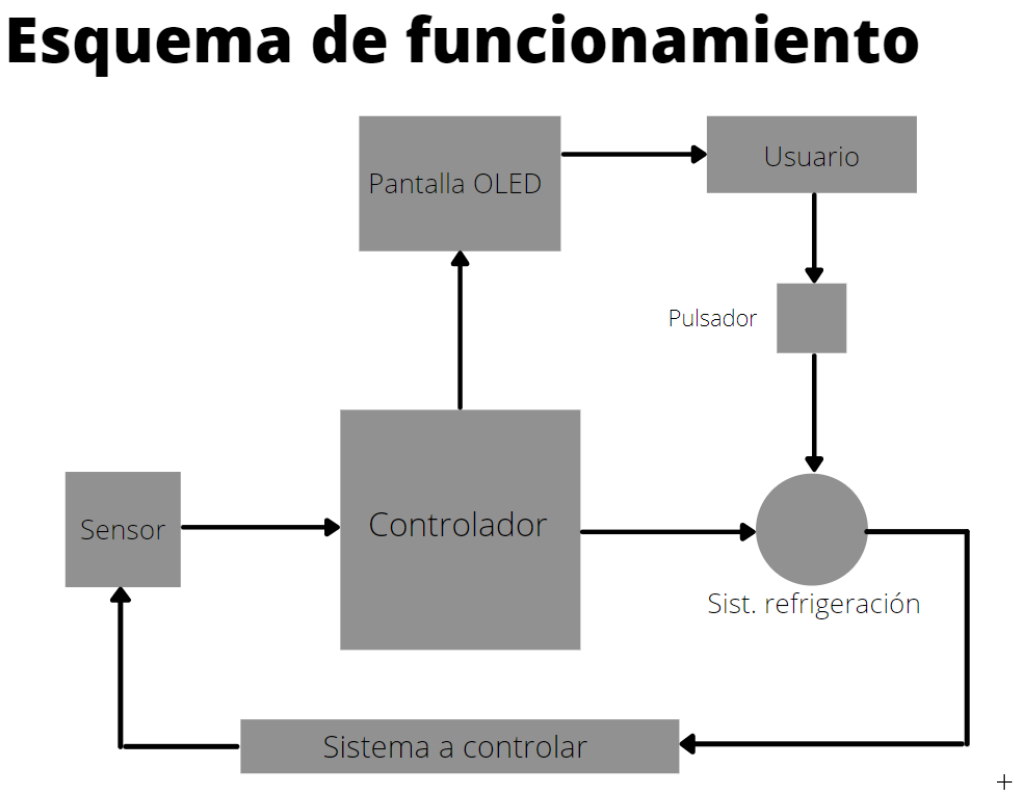

Figura 3-6. Esquema de funcionamiento sin conexión a Internet

Como se aprecia en la figura 3-6, el sistema es retroalimentado, ya que el sensor recoge los datos del sistema a controlar, y los cambios que se provoquen (de forma manual o de forma automática) por el sistema de refrigeración son de nuevo recogidos por el sensor.

3.2.2 Funcionamiento del sistema con conexión a Internet

En este caso, los datos del sensor van siendo almacenados (con el tiempo de muestreo que se elija) en una tabla de Excel donde también se indica la hora (hh mm ss) de la toma de la muestra. Esta operación se lleva a cabo mediante el software PAQ- DAQ_R2, ya comentado en la sección 2.2.3. Los datos se introducen en la tabla Excel mediante puerto serie, procedentes de la placa de Arduino (donde se aloja el sensor).

A	B
hora	**sensorValue**
22:54:21	2,00
22:54:51	14,00
22:55:21	10,00
22:55:51	11,00
22:56:21	15,00
22:56:51	74,00
22:57:21	0,00

Figura 3-7. Ejemplo toma de datos en Excel

Mediante la herramienta de Excel podemos obtener mucha información de los datos, ya que esta nos ofrece muchas funciones estadísticas (media, moda, mediana…) y de representacón que pueden ser útiles y que se se van actualizando a media que van llegando datos nuevos.

Con los datos ya en la hoja de Excel, se pretende que estos puedan ser alojados en una página web. Para ello se ha programado una macro en lenguaje VisualBasic que hace que la hoja con los datos (y las representaciones, datos estadísticos, etc) se guarden en formato .html periódicamente.

Una vez tenemos este archivo .html (el cual se va actualizando periódicamente gracias a la macro mencionada) se incrusta mediante lenguaje html en otro archivo .html, que será el que se suba a un servidor web mediante el software Xampp (mencionado en la sección 2.2.5). Se fuerza desde la programación que la página web (llamaremos a esta página web primaria) se actualice periódicamente para captar los cambios en el archivo .html generado por Excel (el que contiene los datos).

De esta forma, ya tendríamos los datos alojados en una página web, pero habría que realizar el camino inverso, es decir, que el usuario pueda activar a través de Internet el sistema de ventilación. Esto se hace desde la progrmación de Arduino, creandose otra página web (la llamaremos secundaria) desde donde el usuario puede activar y desactivar el sistema de ventilación. En el capítulo siguiente se comentará más detalladamente como se consigue esto.

Para enlazar las dos páginas que actualmente se tienen, simplemente se enlaza la secundaria a la primaria mediante un acceso directo desde esta última.

Por último, con el software AppGeyser (sección 2.2.5) se crea una aplicación Android a partir de la página web que ya tenemos alojada en el servidor.

Se muestra un esquema del funcionamiento (figura 3-8).

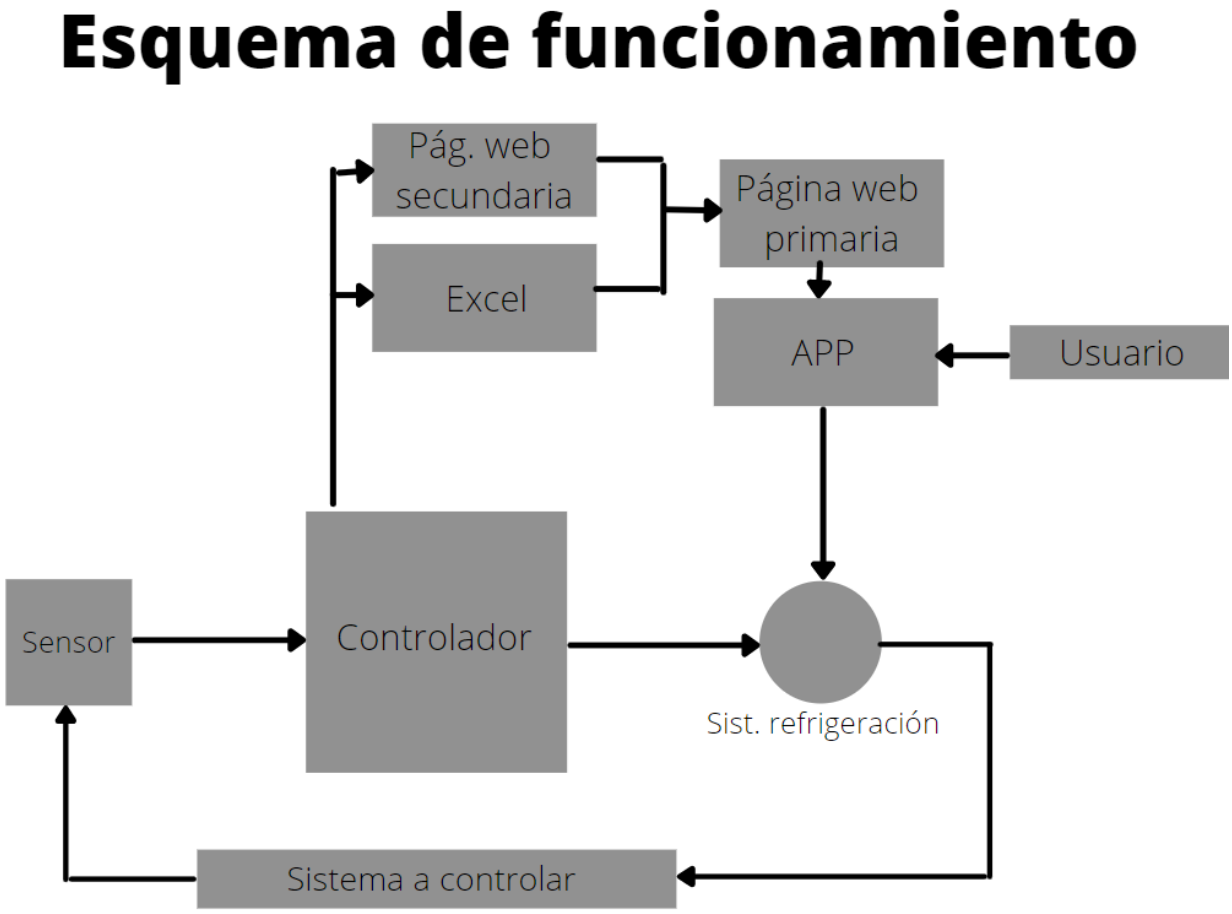

Figura 3-8. Esquema de funcionamiento sin conexión a Internet

Además, se ha creado una base de historial de datos, que se enlaza a la página web primaria y se consigue programando otra macro en Excel que guarde los datos con la fecha y la hora del guardado en el título de una forma periódica. Una opción podría ser hacerlo una vez al día, obteniéndose un historial de muestras diario

4 DESCRIPCIÓN DEL FUNCIONAMIENTO

En este capítulo se realizará una explicación más exhaustiva del funcionamiento, aportando imágenes y comentando partes importantes de los códigos de programación.

Como en el capítulo anterior, se dividirá esta sección en dos, tratando cada una de uno de los modos de funcionamiento ya mencionados anteriormente.

4.1 Descripción del Sistema sin conexión a Internet

Como se comentó en la sección 3.2.1, este modo se basa en la representación dinámica de los datos en una gráfica mediante una pantalla OLED. A esto se le suma la activación manual y automática del sistema de refrigeración.

La parte más importante y compleja de este modo de funcionamiento es la representación en la pantalla OLED. Para ello se conecta la pantalla a la placa de Arduino como se indica en la sección 3.1.

Se muestra una imagen de como se representarían los datos (Fig.4-1)

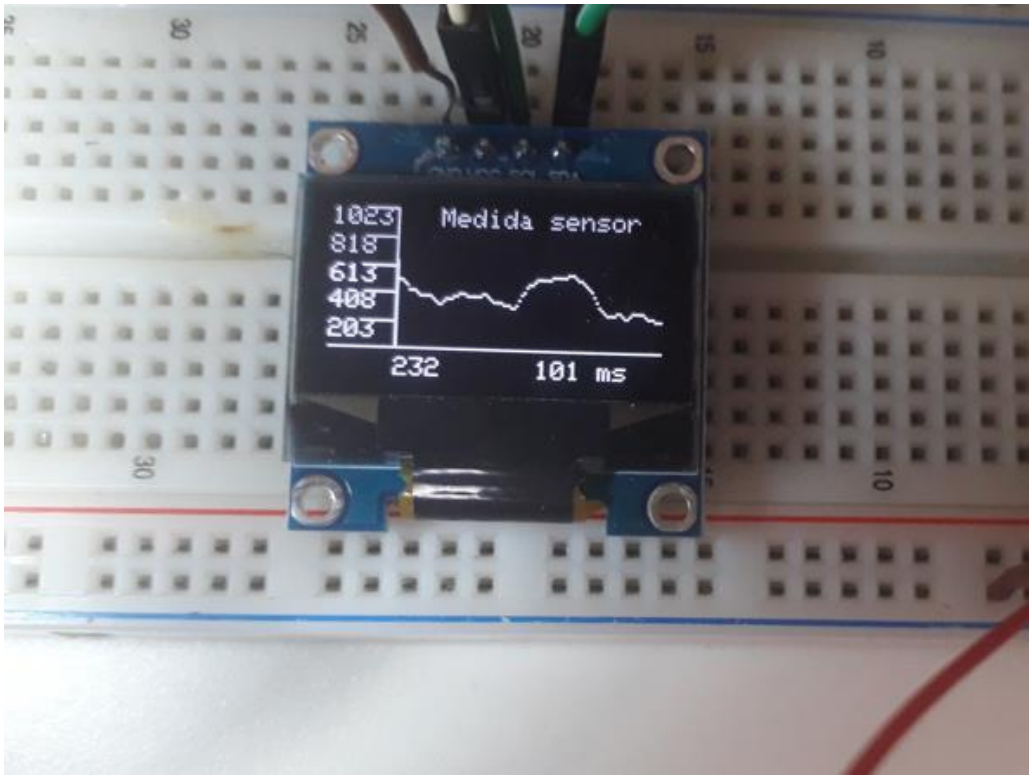

Figura 4-1. Ejemplo representación de datos en pantalla OLED

Como se aprecia en la Fig 4-1 en la pantalla se representan los ejes X (tiempo) e Y (medida del sensor). En el eje Y se han equiespaciado las posibles medidas del sensor (que toma medidas de 0 a 1023). Además se observa en la parte inferior de la pantalla tanto el valor de la última medida (valor de la izquierda) y el tiempo que transurre entre un dato y otro. Este último parámetro es modificable a nivel de código.

Recordar que el sistema se ha diseñado con un sensor de luz (fotoresistencia), pero sería extrapolable a valores de temperatura. Se ha utilizado un sensor de luz por la mayor facilidad de generar cambios, pero no habría diferencia ni con la temperatura ni con cualquier sensor analógico, ya que todos estos generan una tensión analógica medible por el Arduino.

El sistema de respuesta está programado cuando la medida del sensor es 400 (dato configurable en programación), es decir si se sobrepasa este valor umbral se activa automáticamente. Se observa en las figuras Fig 4-2 y Fig. 4-3.

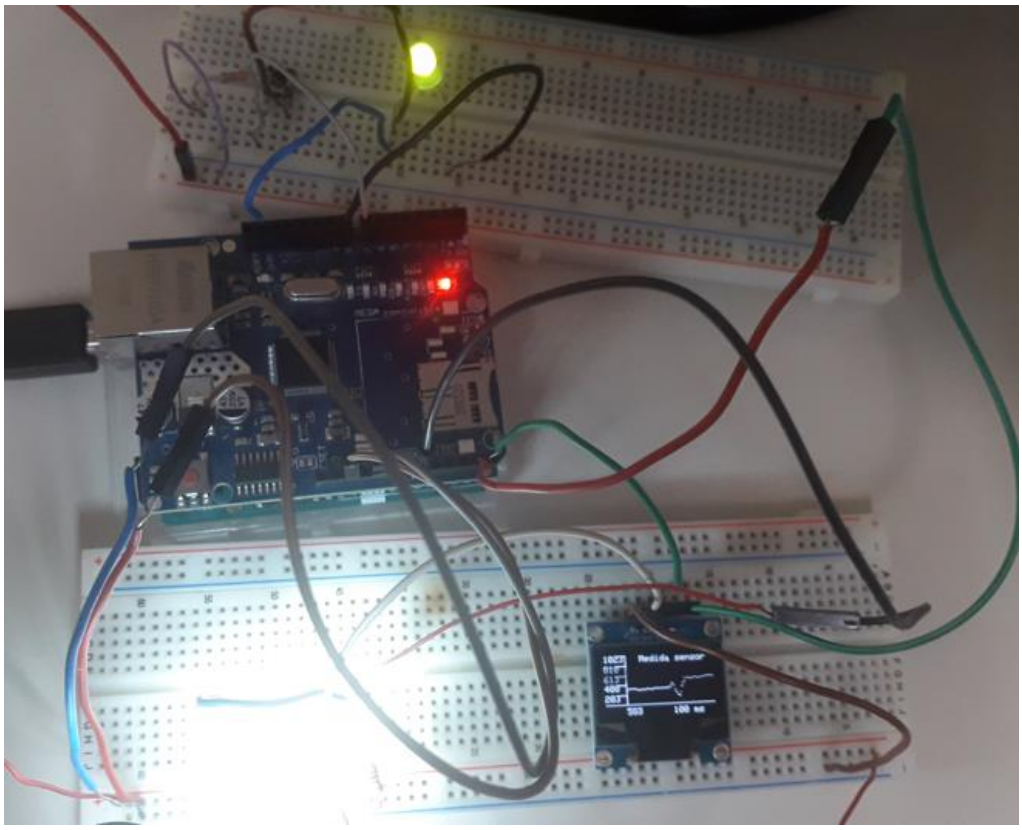

Figura 4-2. Sistema con el valor umbral superado

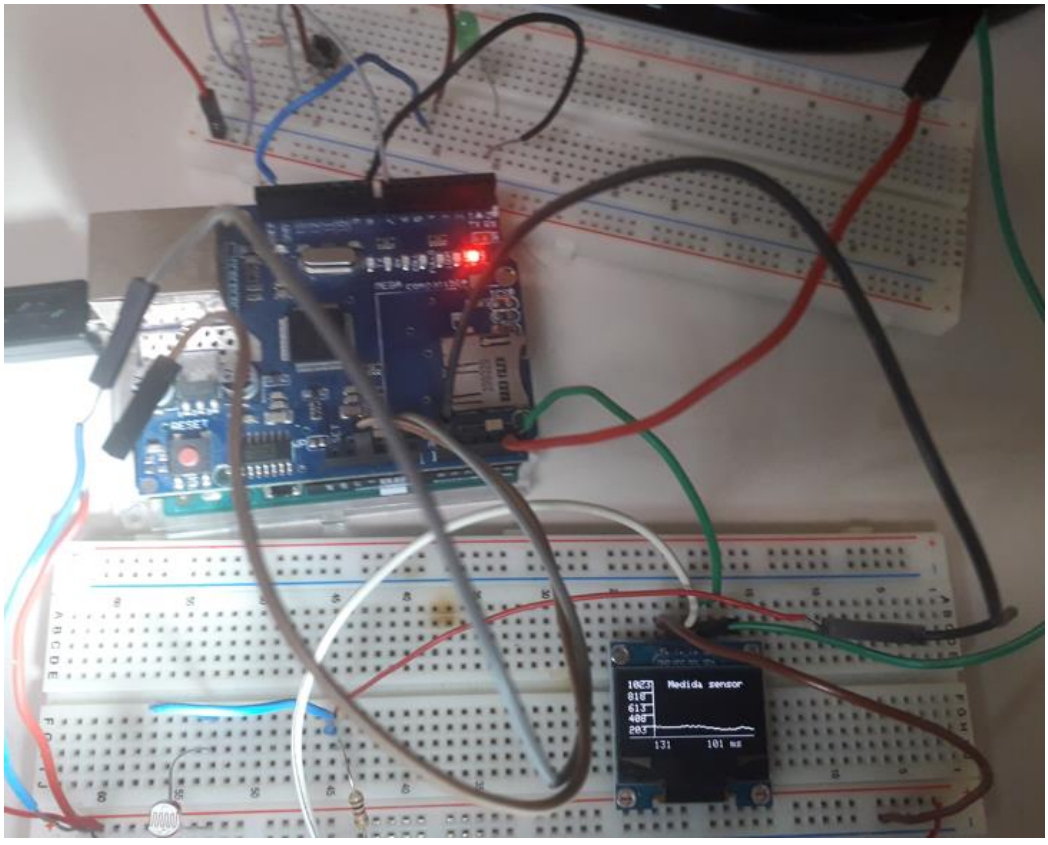

Figura 4-3. Sistema con el valor umbral no superado

Se observa en la Fig.4-2 que la medida del dato supera el umbral de 400 establecido, por lo que automáticamente se activa el sistema de ventilación (un LED en nuestro caso).

En cambio, en la Fig.4-3 se observa que no se alcanza el umbral, por lo que el sistema de ventilación no está activado.

En el caso de que se quiera activar el sistema de ventilación de forma manual, solo habría que pulsar el pulsador, y el sistema se activará sea cual sea la medida del sensor. Se aprecia en la Fig. 4-4.

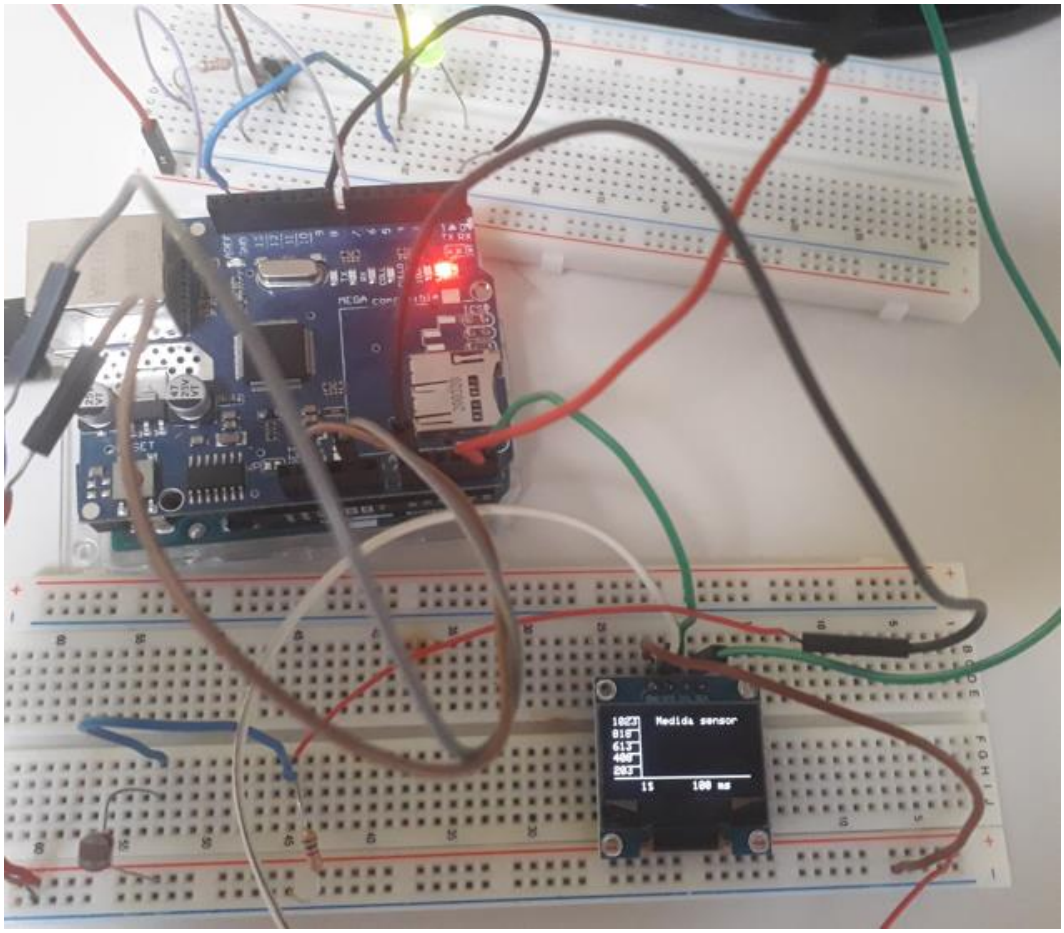

Figura 4-4. Sistema con la activación manual activada

En cuanto al código, se adjuntará en el Anexo de este documento, pero se van a comentar los aspectos más importantes.

El código del sistema automático y manual de respuesta del sistema es el siguiente:

```
val= digitalRead(BOTON); // lee el estado del Boton
if ((val == HIGH) && (old_val == LOW)){
state=1-state;
 delay(10);
 }
 old_val = val; // valor del antiguo estado
 if (state==1){
 digitalWrite(8, HIGH); // enciende el LED de forma manual
 }
 else{
 digitalWrite(8,LOW); // apagar el LED de forma manual


       if (valorAnalogico>400){

         digitalWrite(8 , HIGH);   // enciende el LED de forma automática

       }

       else {
           digitalWrite(8 , LOW);   // apaga el LED de forma automática
       }

     }
```

Figura 4-5.Código respuesta automática y manual (no conexión)

Para elegir el estado del sistema de respuesta (activable mediante el pulsador) se comprueba el estado del botón y se compara con el estado anterior. En función de esto, si el estado anterior era el que permitía tener el sistema de ventilación encendido, al cambiar el estado del botón, apaga el sistema de ventilación. Funciona de la misma manera en el caso contrario.

Para hacerlo de forma automática, simplemente se activa la ventilación en función del valor del dato del sensor comparado con el valor umbral. Esto ocurre en el caso de que no se haya activado manualmente la ventilación (se da prioridad a la decisión manual del usuario).

Para la representación en la pantalla OLED toma gran importancia el uso de las librerías Adafruit_SSD1306 y Adafruit_GFX. Estas nos permiten el uso de la pantalla OLED, además de tener funciones que hacen más asequible la tarea de representar los datos. Se recuerda que la pantalla es de 0.96''.

```
3
4 #include <Adafruit_SSD1306.h>
5 #include <Adafruit_GFX.h>
6 Adafruit_SSD1306 display(128, 64);  // declara la resolucion del display
7
```

Figura 4-6. Inclusión librerías pantalla OLED y elección de la resolución

Como se aprecia en la Fig.4-6, hay que declarar la resolución de la pantalla.

Para la representación tanto de los ejes como de los valores de referencia y el título se han utilizado las siguientes funciones:

- *display.drawLine*. Esta función representa una línea en el display. Los valores de entrada son las coordenadas de los puntos iniciales y finales de la recta (siendo el [0 0] la esquina superior izquierda) y el color de la misma.
- *display.setCursor* y *display.print.* Se describen juntas estas funciones, ya que en este proyecto se utilizan de forma conjunta. La función *display.setCursor* indica la coordenada donde se van a representar los caracteres indicados en la función *display.print.*
- *display.drawPixel.* Esta función representa un punto. Sus entradas son las coordenadas del punto y el color del mismo.

```
valorAnalogico = analogRead(A0); //lee el valor analogico del pin A0

graficaValor=map(valorAnalogico,0,1023,53,0); //escala el valor analogico a un pixel imprimible en pantalla

x[127]=graficaValor; //asigna el valor escalado a el ultimo dato de la matriz

for(int i=127;i>=25;i--){
    display.drawPixel(i, x[i], WHITE); //dibuja punto a punto el contenido de x

    y[i-1]=x[i]; //guarda la informacion desplazada una posicion temporalmente en y
  }
```

Figura 4-7. Código representación valores en pantalla OLED

Para la representación de los datos se reescalan los 1024 posible valores que puede tomar la medida del sensor a los 54, que son los píxeles que nos quedan libres por encima del eje X. Tras esto, se representa el valor del dato en la gráfica, y cuando llegue el siguiente se guarda en una posición anterior. Se puede observar en la Fig. 4-7.

Tasa de muestreo y memoria

El tiempo de muestreo se puede modificar, ya que en el código hay un delay que frena la simulación y marca el tiempo de recogida entre dato y dato. El tiempo de muestreo mínimo posible es de unos 40 milisegundos, lo cual es debido principalmente al tiempo que tarda el sensor en dar el valor que recoge y el tiempo que tarda la pantalla OLED en procesar la representación.

En cuanto a la memoria, en este modo va a tener un valor limitante, que va a ser el número de datos que caben en la pantalla OLED. En nuestro caso, la pantalla tiene 128 píxeles horizontales, a lo que habría que restar los que se han usado para la representación de los valores de referencia y el eje Y. En nuestro caso nos quedarían 103 píxeles.

Debido a que la temperatura no es un parámetro que cambie brucamente no hace falta que se tomen datos de manera continua. Si se tomara un dato cada 30 segundos (lo cual sería un control de la temperatura suficientemente exhaustivo) y se pueden ver 103 medidas a la vez, estaríamos viendo el historial de medida de los últimos 51 minutos.

4.2 Descripción del Sistema con conexion a Internet

En esta sección se analizará la toma de datos y el monitoreo de los mismos a través de una página web (en nuestro caso subida a un servidor local, aunque se podría subir a un servidor externo, lo cual no se ha hecho para no tener que disponer de un dominio) para poder acceder a la recogida de datos y al control del sistema de respuesta desde cualquier dispositivo con conexión a Internet, ya sea mediante la web o mediante la app de Android asociada a la misma.

Como se comentó en la sección 3.2.1, los datos recogidos del sensor son almacenados en una hoja de Excel mediante el software PAQ-DAQ_R2. Esto se hace de forma muy sencilla, ya que desde el código de Arduino, simplemente se leen los datos y se mandan por puerto serie, siendo estos datos recogidos en Excel. Se aprecia en la Fig.4-8 un bucle while que modifica la tasa de muestreo (tiempo entre dato y dato).

```
int sensorValue = analogRead(0);
TiempoAhora = millis();
Serial.print("DATA,TIME,");
Serial.println(sensorValue);
while(millis() < TiempoAhora+30000){
}
```

Figura 4-8. Código recogida de datos por puerto serie

Se marcará la tasa de muestreo de 30 segundos, ya que se considera una tasa aceptable al tratarse de temperatura, la cual no cambiará de manera abrupta en poco tiempo.

Por supuesto, se mantiene la respuesta automática (Fig.4-9).

```
if (sensorValue>400){
        digitalWrite(LED , HIGH);
        }

      else {
          digitalWrite(LED , LOW);
      }
```

Figura 4-9. Código sistema de respuesta automático

Se activa el sistema de ventilación cuando se supera el umbral establecido.

Una vez se tienen los datos en Excel (van llegando periódicamente) se pueden hacer con ellos las operaciones y medidas que se deseen. Destacar la importancia y potencia de esto, ya que todas estas operaciones y medidas estadísticas se verán reflejadas en la página web.

Se muestra como se ven los datos en la hoja Excel (Fig.4-10). Se muestra como ejemplo de tratamiento de los datos una gráfica y la media, pero se reitera que las opciones son muy amplias. Tanto la gráfica como la media son dinámicas, es decir, se van actualizando a medida que van llegando los datos.

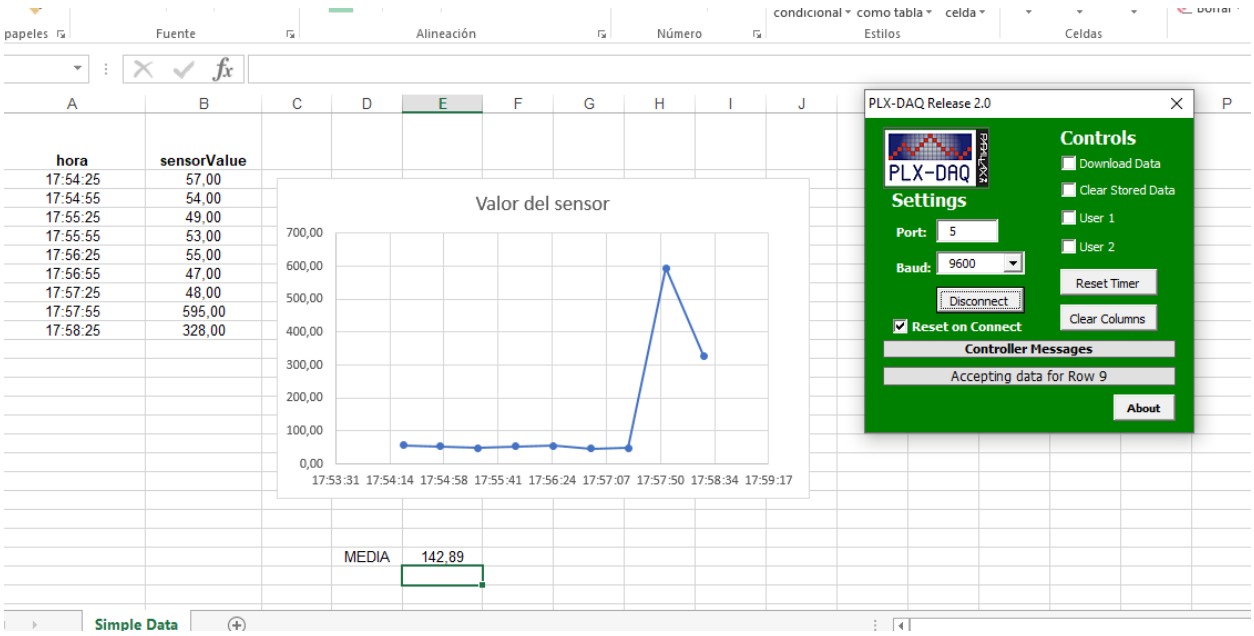

Figura 4-10. Ejemplo datos en Excel con representación gráfica

Una vez que se tienen los datos en la hoja de Excel, hay que conseguir que esta se guarde como .html (para poder tener los datos en la página web) y que lo haga periódicamente, para que actualice los datos que van llegando. Para ello se ha programado una macro en el lenguaje VisualBasic de Excel.

PLX-DAQ_R2.xlsm - Módulo1 (Código)

(General) | auto_open

```
Sub auto_open()
tiempo = Now + TimeValue("00:00:25")
tiempo2 = Now + TimeValue("00:03:30")
Application.OnTime tiempo, "Guardar"
Application.OnTime tiempo2, "Guardar2"
End Sub

Sub Guardar()
    Application.DisplayAlerts = False

    ActiveWorkbook.SaveAs Filename:= _
        "C:\xampp\htdocs\app\PLX-DAQ_R2.html", FileFormat:=xlHtml _
        , ReadOnlyRecommended:=False, CreateBackup:=False
    Application.DisplayAlerts = True
Call auto_open
End Sub
```

Figura 4-11. Código VisualBasic guardado automático

La función *auto_open* hace que la macro se active automáticamente al abrir el libro de Excel, y hace que se ejecute la función *Guardar,* que guarda la hoja Excel cada 25 segundos en formato .html en la ruta adecuada para subir la hoja al servidor local. La función *Aplication.DisplayAlerts=False* evita que salga la pantalla emergente que pregunta si sobrescribir el archivo .html (ya que se guarda con el mismo nombre).

Se aprecia que se activa la función *Guardar2* cada 3.5 minutos., la cual será comentada posteriormente, pero se adelanta que es para crear una base de datos del historial de medidas.

Una vez que se tiene un archivo .html que se actualiza automáticamente, es momento de incrustarlo en otro que será el que se alojará en el servidor local.

```
18  <div class="container">
19  <iframe id="iframe1" frameBorder="0" class="responsive-iframe" src="PLX-DAQ_R2.html"></iframe>
20  </div>
21
22  <script>
23          window.setInterval(function() {
24              reloadIFrame()
25          }, 10000);
26
27          function reloadIFrame() {
28              document.getElementById('iframe1').contentWindow.location.reload();
29          }
30      </script>
31  </body>
32
33  </html>
```

Figura 4-12. Parte de código html para la creación de la web

Se observa en la Fig.4-12 como se incrusta el archivo generado mediante Excel (línea 19) y mediante funciones de JavaScript se fuerza a que actualice cada 10 segundos, para reflejar los cambios que se van dando en el archivo PLX-DAQ_R2.html (generado por Excel).

A su vez, mediante el software Xampp, se crea un servidor local en el que se aloja la página web creada mediante este último código. Esto se hace activando el servidor desde Xampp, siendo la URL, la IP del ordenador, seguida del puerto adjudicado por la aplicación Xampp y seguido de la carpeta donde se aloja el código.

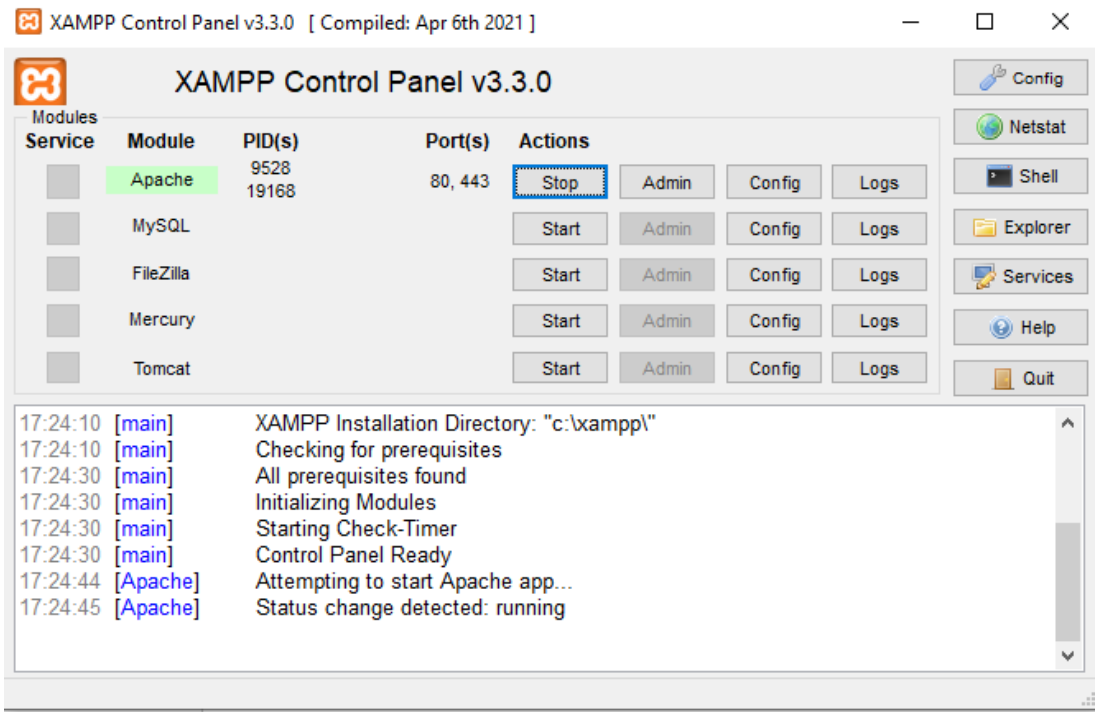

Figura 4-13. Interfaz Xampp con servidor activo

Se aprecia que los puertos en los que se alojan son el 80 y el 443. Si entramos en la URL http://192.168.1.21/app/index1.html, vemos lo siguiente.

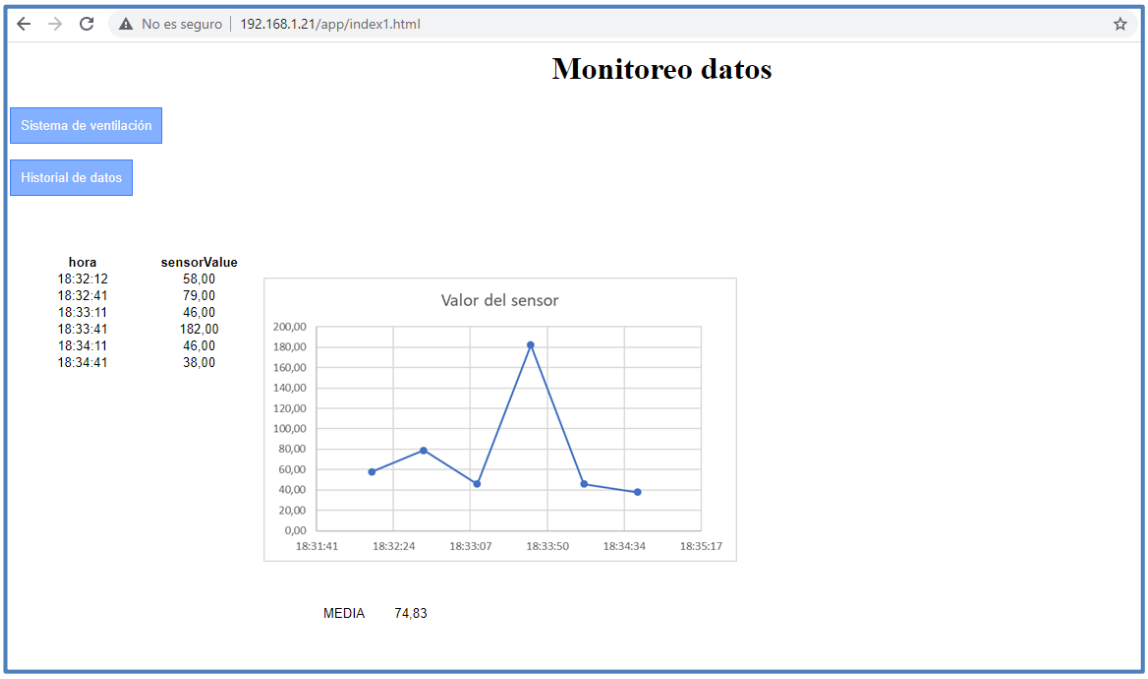

Figura 4-14. Vista general página web

Como se puede observar, ya tenemos una página web que recoge los datos del sensor, los va almacenando y

los va representando.

Se observa en la Fig.4-14 que están las opciones de acceder al historial de datos y al sistema manual de activación del sistema de ventilación.

Para obtener un historial de datos se fuerza a que la hoja de Excel con los datos tomados por el sensor se guarde periódicamente mediante una macro, y en el nombre contenga la fecha y la hora de guardado. Se consigue así tener un registro de todas las medidas tomadas antriormente.

Como se observa en la Fig.4-11 había otra instrucción programada llamada *Guardar2,* que se observa en la Fig.4-15.

```
Sub Guardar2()
    Application.DisplayAlerts = False
    'Update 20141111
    ActiveWorkbook.SaveAs ("C:\xampp\htdocs\app\Alojamiento\Guardado" & Format(Now(), "DD-MMM-YYYY hh mm AMPM")

    Application.DisplayAlerts = True
Call auto_open
End Sub
```

Figura 4-15. Código macro Guardar2

Estos archivos se almacenan en la carpeta que esta alojada en el servidor local, pudiendo acceder a ella desde la página web mediante el enlace “Historial de datos” de la Fig. 4-14. Esto se programa de la siguiente manera (Fig.4-16).

```
13
14  <form target="_blank" action="http://192.168.1.21/app/Alojamiento/">
15      <input style="margin:auto;background-color: #84B1FF;color: snow;padding: 10px;border: 1px solid #3F7CFF;" type="submit" value="Historial de datos" />
16  </form>
```

Figura 4-16. Código creación acceso directo al historial de datos

De esta manera, si se pulsa en el enlace, se llega a lo siguiente (Fig.4-17):

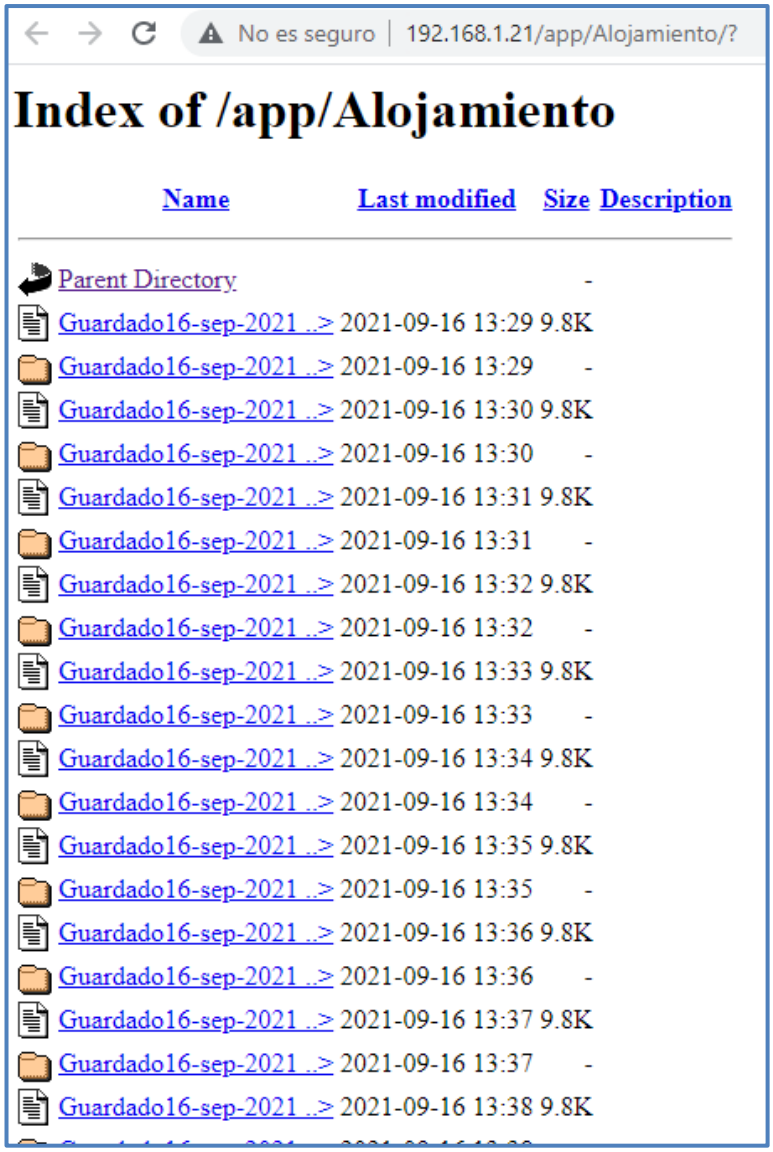

Figura 4-17. Historial de datos

Si se pulsa en cualquiera de los enlaces, muestra los datos tomados en la fecha y hora indicada.

Una posibilidad sería programar la macro *Guardar2* de Excel para que guarde cada 24h para tener un seguimiento diario y un historial estructurado por días.

En cuanto al sistema de activación manual de la ventilación, este permite al usuario activar desde la web la mencionada ventilación. Al clickar en “Sistema de ventilación” (Fig.4-14), se abre la siguiente pestaña (Fig.4-18), desde la que se puede activar y desactivar el sistema a placer.

Figura 4-18. Pantalla de activación del sistema de ventilación (desde la web)

Esta página se sube al servidor local, pero no lo hace desde el software Xampp, sino que lo hace el propio código de Arduino a través del Shield Ethernet.

```
#include <SPI.h>
#include <Ethernet.h>

// Declaración de la direcciones MAC,IP,GATEWAY y SUBNET.
byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED};

IPAddress ip(192,168,1,50); // 192.168.1.XX donde XX es una dirección que no esté utilizada
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);
```

Figura 4-19. Trozo de código creación de web desde Arduino

Como se observa en la Fig.4-19 hay que incluir las librerías SPI y Ethernet y hay que elegir una dirección IP que no esté utilizada (en este caso 192.168.1.50).

Destacar la librería Ethernet, la cual se usa para manejar el Ethernet Shield. La función principal es mandar por Ethernet el protocolo programado en Arduino. Se destacan algunas de las funciones más importantes:

- *IPAddress*. Define dirección IP
- *EthernetServer.* Crea un servidor que escucha por las conexiones entrantes del puerto definido
- *Server.begin.* Hace que el servidor comience a escuchar
- *Client.println*. Escribe datos al servidor
- *Client.stop.* Desconecta del servidor

Es importante añadir la existencia de una librería llamada Ethernet2, que es similar, pero compatible con el Ethernet Shield 2 W5500.

El modo de funcionamiento de este código para crear lo que se observa en la Fig.4-18, es introducir por el puerto serie instrucciones html para la creación de la página. Se muestra un trozo de código para que se vea (Fig.4-20):

```
client.println("<html>");
client.println("<head>");
client.println("<meta charset='UTF-8'>");
client.println("</head>");
client.println("<body style='background-color: #FFFFFF;'>");
client.println("<br/><br/>");
client.println("<h1 align='center'>SISTEMA DE VENTILACIÓN</h1>");
// Creamos los botones.
// Para enviar parámetros a través de HTML se utiliza el método URL encode.
// Los parámetros se envían a través del símbolo '?'
client.println("<div style='text-align:center;'>");
client.println("<button onClick=location.href='./?LED=ON\' style='margin:auto;background-color: #
client.println("ON");
client.println("</button>");
client.println("<button onClick=location.href='./?LED=OFF\' style='margin:auto;background-color:
client.println("OFF");
client.println("</button>");
client.println("<br/><br/>");
client.println("<b>ESTADO DEL SISTEMA DE VENTILACIÓN = ");
client.print(estado);
```

Figura 4-20. Ejemplo funciones para creación de web desde Arduino

Se observa claramente que son instrucciones html que se dan a través del puerto serie.

Una vez se tiene la página web operativa lo único que quedaría es crear una app Android para poder acceder fácilmente desde un Smartphone.

Esto se hace desde una web llamada AppGeyser, que genera la aplicación introduciendo simplemente la URL de la página web creada.

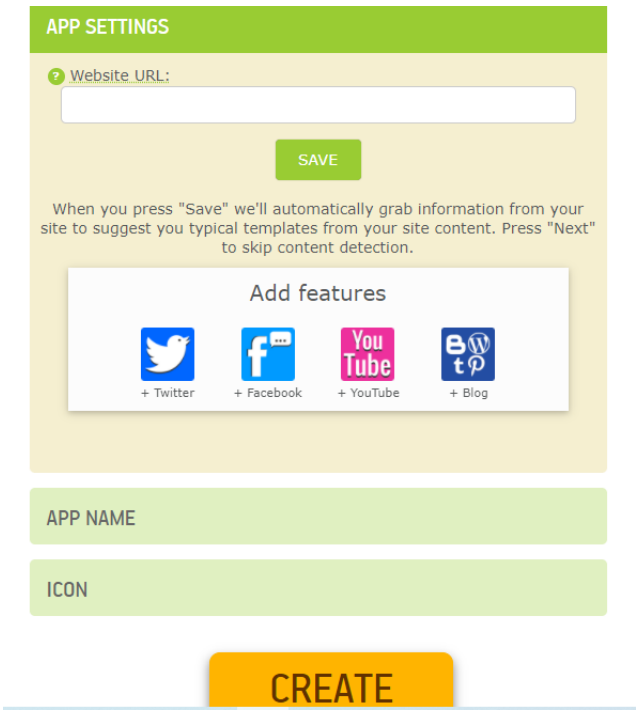

Figura 4-21. Interfaz AppGeyser

Se adjuntan capturas de la app en funcionamiento:

Figura 4-22. Captura 1 app desde Smartphone

Figura 4-23. Captura 2 app desde Smartphone

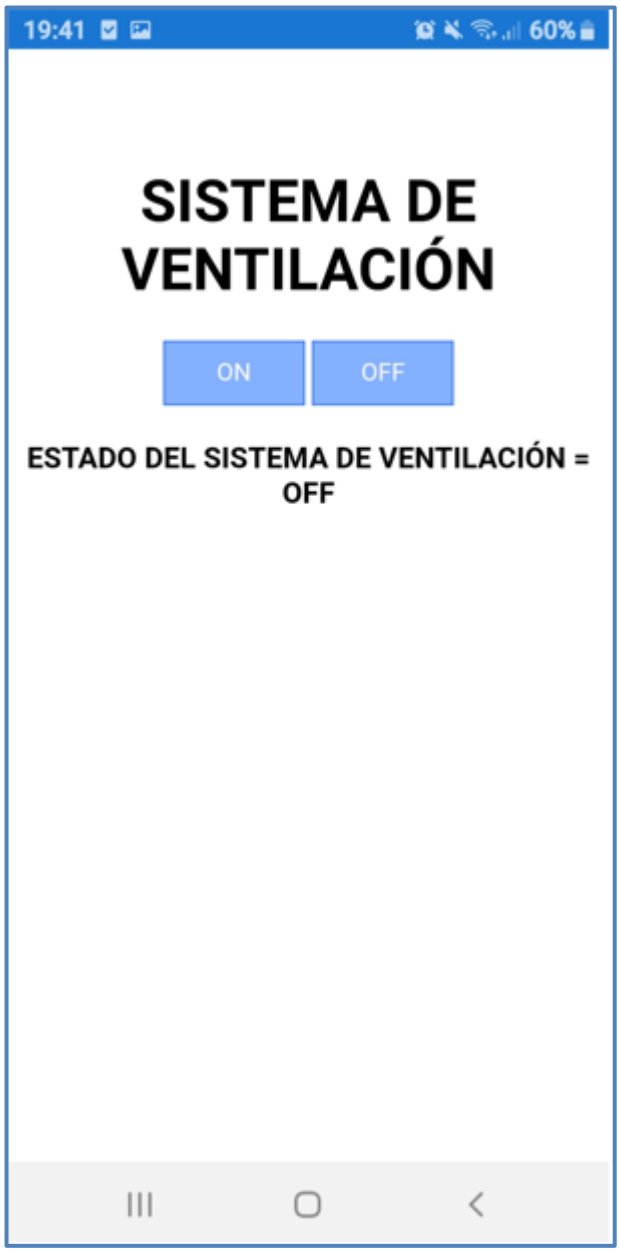

Figura 4-24. Captura 3 app desde Smartphone

Tasa de muestreo y memoria

Mediante este sistema la tasa de muestreo también es elegible, pero como ya se ha comentado anteriormente, al ser temperatura lo que queremos monitorear, no es necesaria una tasa de muestreo excesivamente baja, ya que no es un parámetro que cambie bruscamente. Se ha fijado la tasa de muestreo en 30 segundos, que es un valor con el que se hace un seguimiento bastante exhaustivo.

En cuanto a la memoria, mediante la base de datos que se ha creado la se ha conseguido que el valor de esta sea bastante elevado, siendo el valor limitante la memoria libre que se tenga en el pc en el que se aloje el el Excel y la carpeta que se envía a Internet mediate el servidor local.

Además, se cuenta con la ventaja de que se guardan los datos con la fecha de guardado en el nombre, haciendo muy fácil el acceso a la información desde cualquier dispositivo, ya sea un ordenador desde la web o un Smartphone desde la aplicación o desde la misma página web.

5 CONCLUSIONES

En primer lugar, con la ralización de este proyecto se ha pretendido exponer una forma de poder integrar conocimientos y herramientas distintas para el cumplimiento de un objetivo, utilizándose tanto distintos lenguajes de programación como distintos softwares. También se ha intentado desarrollar y aportar en el aspecto de acceso remoto a sistemas electrónicos, lo que se considera cada vez más importante en el desarrollo de proyectos como se observa en la tendencia IoT (Internet of Things) con la que conecta directamente este proyecto. Esta tendencia pretende la agrupación e interconexión de dispositivos a través de la red, donde se hagan visibles los datos de dichos dispositivos y puedan interaccionar entre ellos. Esto se conoce como una intercción "máquina a máquina", donde los dispositivos interaccionan entre sí sin la necesidad de intervención humana.

El objetivo de este proyecto era el diseño tanto a nivel de hardware como de software de un sistema de control de temperatura, así como un monitoreo de los datos y un control de la respuesta. Este sistema de control debía ser aplicable al mundo aeroespacial, en especial a los sistemas embarcados que son propensos a sufrir altas temperaturas y pueden dañarse, dando tanto una respuesta automática, como la posibilidad de respuesta manual por un usuario.

Este proceso se ha realizado simultáneamente por dos caminos, uno mediante un monitoreo de los datos y control de la respuesta a través de Internet, diseñándose también una base de datos para alacenar el histórico de datos; y otro sin necesidad de conexión ni dispositivos externos al hardware utilizado para el montaje, tejiéndose así una red de seguridad que sería muy útil en caso de cualquier fallo de red o similar.

El proceso realizado para el diseño del sistema ha sido un poco tedioso, ya que la elección de esta forma de disño y montaje no fue la inicial, la cual fue derivando y cambiando a medida que se encontraban problemas o trabas en el proceso.

En un principio se pretendía que la representación fuera en una pantalla LCD, la cual tenía 13 pines de entrada y su proceso de programación era muy complejo, por lo que se optó por buscar una alternativa que diera los mismos resultados con un tratamiento más sencillo, eligiéndose finalmente una pantalla OLED.

También se pretendía en un principio subir la toma de datos a internet directamente desde la programación de Arduino, lo que fue inviable debido, tanto a la poca memoria del mismo, como a la dificultad de crear una base de datos con el historial de muestras a partir del mismo.

En cuanto a los resultados obtenidos, se ha conseguido que el sistema diseñado cumpla con los requisitos inicialmente propuestos, consiguiéndose esto con el uso de distintos software y lenguajes de programación. Se han usado en este proyecto los lenguajes C++, html, Javascript y VisualBasic.

Destacar la forma de desarrollo del trabajo, ya que se ha ido integrando información en función de las necesidades que iba presentando el proyecto, buscando una forma de conseguir coordinar y dar una visión global a lenguajes y herramientas muy diversos.

En cuanto a posibles mejoras a desarrollar en el futuro, se podría hacer un montaje de hardware más robusto y manejable (se podría haber hecho un PCB que haría las conexiones más seguras y el diseño más estético y funcional), así como aumentar el número de sensores, pudiendo así tomar datos de temperatura en distintas zonas. Se podrían utilizar elementos de mayor calidad, con lo que se podría aumentar la sensibilidad y disminuir el tiempo de muestreo límite.

Cambiando el tipo de sensor se podrían monitorear humo, humedad, altitud, posición, etc. Estos datos podrían gestionarse de forma conjunta creando una red de datos que diese una visión muy global del estado de la aeronave, sumando a esto la capacidad de que cualquier usuario con permiso y acceso a Internet pudiera consultar datos y tomar decisiones.

Pese a que el sistema ha sido pensado con el objetivo de utilizarse en el sector aeroespacial puede tener distintas aplicaciones, incluso de uso doméstico, ya que se han utilizado elementos sencillos y al alcance de la mayoría de la población. El sistema podría tener muchas aplicaciones en domótica también.

Una de las mayores ventajas del proyecto realizado es su capacidad de seguir evolucionando y creciendo, ya que aumentando el número de parámetros controlables y desarrollando una interacción compleja y adecuada sobre las respuestas se podría acabar consiguiendo el desarrollo de una herramienta bastante potente. A esto, se le suman los avances de hardware, consiguiéndose cada vez mejores y más compactos elementos que mejorarían el resultado final. En cuanto a las mejoras que se vayan dando en los software utilizados en la realización de este proyecto, también aportarían riqueza de cara al futuro, al poder añadir funcionalidades que actualmente serían difícil de implementar.

REFERENCIAS

[1] [En línea]. Available: https://www.britannica.com/technology/electronics.

[2] [En línea]. Available: http://myelectronic.mipropia.com/aplicaciones.html?i=1.

[3] [En línea]. Available: https://moreelectricaircraft.com/contact.php.

[4] [En línea]. Available: https://store.arduino.cc/products/arduino-uno-rev3/.

[5] [En línea]. Available: https://store.arduino.cc/products/arduino-uno-rev3/.

[6] [En línea]. Available: https://www.sunfounder.com/products/w5100-ethernet-shield.

[7] [En línea]. Available: https://www.amazon.es/gp/product/B00HG82V1A/ref=ppx_yo_dt_b_asin_title_o01_s00?.

[8] [En línea]. Available: https://store.arduino.cc/products/grove-oled-display-0-96?queryID=undefined.

[9] [En línea]. Available: https://www.amazon.es/gp/product/B08BKRPBVL/ref=ppx_yo_dt_b_asin_title_o03_s00?.

[10] [En línea]. Available: https://www.arduino.cc/en/software.

[11] [En línea]. Available: https://es.wikipedia.org/wiki/Microsoft_Excel.

[12] [En línea]. Available: https://www.parallax.com/package/plx-daq/.

[13] [En línea]. Available: https://notepad-plus-plus.org/.

[14] [En línea]. Available: https://www.apachefriends.org/es/index.html.