

ARQ ARQUITECTURA PARA MONITOREO E INSPECCIÓN DE SISTEMAS ADHERENTES A TECNOLOGÍAS DIGITALES

ARQ ARCHITECTURE FOR MONITORING AND INSPECTION OF SYSTEMS ADHERING TO DIGITAL TECHNOLOGIES

Sergio Andrés Castillo  ORCID, Cristhian Iván Riaño Jaimes 

Universidad de Pamplona, Bogotá, Colombia

Recibido: 10/10/2022

Revisado: 12/11/2022

Aceptado: 27/12/2022

Publicado: 31/01/2023

RESUMEN

En el presente proyecto se creó una arquitectura abierta y flexible que permite realizar medición de variables en tiempo real y enviarlas a un entorno digital cumpliendo con las demandas o exigencias de integración e interoperabilidad. En el presente proyecto se implementó una arquitectura digital con la cual se puede obtener datos de variables físicas para posteriormente, estos datos puedan ser transformados en información y conocimiento que alimenten sistemas expertos o procesos basados en inteligencia artificial. Como requerimientos se tuvo en cuenta que fuera modular, escalable con tecnologías libres y fácil operabilidad por personas de distintas especialidades.

Palabras Clave Arquitectura; Digital; Marco; RaspBerry; Sensores; Supervisión

ABSTRACT

In this project, an open and flexible architecture was created to measure variables in real time and send them to a digital environment complying with the demands or requirements of integration and interoperability. In this project, a digital architecture was implemented to obtain data of physical variables so that these data can be transformed into information and knowledge to feed expert systems or processes based on artificial intelligence. As requirements, it was taken into account that it should be modular, scalable with free technologies and easy to operate by people of different specialties.

Keywords: Architecture; Digital; Framework; RaspBerry; Sensors; Supervision.

INTRODUCCIÓN

Las exigencias digitales actuales interpuestas principalmente por la cuarta revolución industrial buscan que todos los objetos estén integrados y trabajen en un ambiente interoperable. La integración de los sistemas debe ser en todas las capas de una

arquitectura, capa física teniendo contacto con el proceso, capa lógica que define una estructura de intercambio de datos, preferiblemente neutra e independiente tecnológicamente y una capa de presentación que habilita persistencia, acceso, escritura, actualización de los datos adquiridos. Los datos se constituyen en el contexto actual como el átomo o elemento más indivisible de cualquier sistema inteligente; el procesamiento de datos produce información y el procesamiento de información conduce al conocimiento. En este trabajo se proyecta una arquitectura que sea el puente entre el mundo real y el entorno virtual, habilitando conceptos de digitalización que deje disponibles datos tanto históricos como actuales en un entorno virtual y de fácil acceso por cualquier tecnología digital.

La principal contribución radica en la construcción de una arquitectura abierta con tecnologías de libre acceso completamente adaptable a diferentes procesos y útiles para aplicación de conceptos como Digital Twin que se basa en la adquisición de datos en tiempo real para reflejar el comportamiento actual de un sistema. Esta arquitectura es adaptable a diferentes variables físicas como lo son, medición de temperatura, flujo, humedad, detector de lluvia y será validada en un caso de estudio que ejemplifique la conexión del mundo real con el mundo virtual. Varias cuestiones en el desarrollo de la arquitectura necesitan ser solucionadas, cuales dispositivos y componentes emplear en la capa física, cuales entornos de desarrollo brindan flexibilidad en distintas aplicaciones, como dar persistencia a los datos, que estructuras de datos neutros pueden ser empleadas para el control de procesos, proyectar una capa de presentación de datos útil para procesamiento. Los resultados esperados es conseguir un mecanismo con-fiable que valide conceptos como *Internet of Things (IoT)*, *Digital Twin* entre otros emergentes dentro de la cuarta revolución industrial.

Esta solución será útil para resolver un problema de medición digital, y promover el trabajo conjunto con estudiantes y docentes del programa de ingeniería agrónoma, ingeniería ambiental, ingeniería electrónica. Un caso de estudio para este proyecto consiste en la medición y digitalización de una variable de temperatura y humedad proyectada para un invernadero y se puso a prueba su correcto funcionamiento, para la validación del sensor de Lluvia se acopló a una estación meteorológica; los datos adquiridos fueron la base para establecer políticas públicas en pro de optimizar el servicio.

Para llevar a cabo el desarrollo escrito y práctico de este proyecto, se investigaron diferentes artículos para proporcionar información relevante y eficaz en el proceso. Para iniciar, uno de ellos detalló el diseño de un dispositivo para la calibración de flujo de aire, así mismo, el uso de la Raspberry Pi para el diseño de la interfaz junto con el algoritmo que permitiera la calibración inteligente al presionar un botón. Además, se mostraba como evidencia las pruebas que se habían realizado con tres tipos de sensores de flujo de aire, y que les permitía tener como conclusión el efecto negativo que causa la distancia entre los rectificadores de flujo y la unidad terminal VAV [1]. Por otro lado, se encuentra información en la cual se da a conocer un instrumento para realizar análisis colorimétricos automatizados; para la creación de este instrumento se utilizó una

impresora 3D y como controlador una RaspBerry Pi4, este instrumento tiene dos módulos diferenciados, uno es el muestreador de carrusel, mientras que el otro es el módulo de medida de señal, el sensor del módulo de medición es un convertidor de luz de color a digital permitiendo recopilar datos de color RGB de la luz, este instrumento requiere menor cantidad de volumen de muestras y su procedimiento de análisis es menor en comparación con el método de referencia con resultados analíticos similares [2].

Ahora bien, también se obtiene información de uno de ellos, en los que se habla de un sistema que se desarrolló para medir el flujo óptico con un microprocesador el cual detecta obstáculos en tiempo real y también y detecta con precisión su propia velocidad de movimiento en un robot [3]. Posteriormente, se lee un documento en el que se propone mejorar un medidor de volumen de gas basado en el conteo de burbujas, para esto se cambió el cuerpo de vidrio por material plástico, en el que también se agregó un sistema de toma de datos de placa única y una interfaz gráfica, destacando la reducción de costos y haciendo más fácil el método de fabricación de este sensor [4]. Además, se encuentra una idea en la que se propone un ventilador mecánico y portátil en las cuales se puedan visualizar algunas variables fisiológicas importantes. Para el desarrollo de esto se incluye una RaspBerry Pi4, una pantalla táctil con interfaz gráfica simple, ya que es una propuesta rentable, segura, fácil de usar, y no es necesaria la presencia de un profesional, y, espacio adecuado [5].

Otra de las técnicas que se utilizó para la elaboración de este proyecto, además de conocer estructuras similares ya establecidas, fue también la de buscar las falencias que se han presentado en dichos proyectos, pues en uno de los archivos informativos, daba a conocer la problemática que tienen los dispositivos inteligentes de monitoreo de agua ya que son propensos a robos, daños, y tienen costos altos, por lo tanto, se propone un medidor de agua inteligente, monitoreado a distancia y de bajo costo para medir el uso de agua en residencias privadas urbanas, usando sensores de flujo de agua y un microcontrolador como lo es la RaspBerry Pi para monitorear y visualizar continuamente el uso de agua [6]. Incluyendo también un artículo en el que se habla de la detección del tamaño de objetos en tiempo real, para esto se utilizó una RaspBerry Pi en la que el enfoque incluye un objeto que se identifique usando técnicas mejoradas; para detectar bordes se tomará un objeto de referencia donde sus características puedan ser identificadas fácilmente; también se utilizó Python Open CV en RaspBerry Pi donde se termina el dimensionamiento del objeto mediante una referencia fija [7].

Además, se usó información eficaz y pertinente sobre este tema, en la que se busca crear la integración de un sistema de adquisición de datos mediante el uso de Arduino Mega, Modulo Wi-fi ESP8266 y RaspBerry Pi3 como servidor web y base de datos, vista como una alternativa económica y que facilitara el acceso a la información recolectada [8]. No obstante, al recolectar esta información de diferentes espacios y situaciones, se refleja la ayuda que ha podido brindar, incluso en molinos de viento que se encuentran situados en montañas y disminuyendo el margen de error, en tanto se encuentra monitoreado por esta base de datos; pues en la tarea "IoT" basada en el marco de observación del molino de viento", se monta un sensor de temperatura y humedad y un

ADC en un molino de viento [9]. Esto evidencia que, el proceso de monitoreo por medio del Raspberry Pi y el Internet de las Cosas (IOT), han facilitado el procedimiento en los invernaderos agropecuarios, actualizando sus datos, sin importar las condiciones climáticas para mantener al tanto a los usuarios interesados en dicha información [10]. Posteriormente, se hace referencia a un documento en el que habla de la importancia de los sensores para las incubadoras, puesto que permiten mantener una temperatura adecuada, e incluso un reporte de error que permita tomar precauciones y evitar daños graves en el sistema [11]. Además, pretende crear una alternativa de menor costo y con una interfaz de usuario más completa y gráfica que los PIDs industriales convencionales que dejaría configurar el controlador al igual que los sensores utilizados. Este trabajo consistió en la programación e implementación de un controlador industrial de tipo PD, se utilizó una RaspBerry Pi y tarjetas de adquisición de datos como plataforma hardware; incluyendo la interfaz web que permite una interacción controlada con el usuario [12].

Ahora bien, para desarrollar la parte práctica, se tuvo en cuenta herramientas como ‘API’, que es un desarrollador entre dos aplicaciones a través de reglas [13]. También se usó ‘Flask’ que permite la creación de forma sencilla de aplicaciones web con Python, con Flask se tienen las herramientas necesarias para crear una aplicación web y que sea funcional, la estructura siempre tendrá los mismos elementos y los mismos ficheros [14]. Además, se reforzó este trabajo con el ‘Framework’ ya que es una plantilla que permite la construcción de proyectos haciendo más corto el proceso de construcción y en caso de la programación con un código mejor estructurado [15]. Asimismo, entre ‘HTML’ aporta el lenguaje con el que se crea la estructura de la página web [16]. Reforzándose así, con el uso de ‘Python’ que es un lenguaje utilizado para aplicaciones de todo tipo [17]. No obstante, la importancia del ‘Internet de las cosas (IoT)’ al ser de las cosas que va de la mano con la modernidad ya que a futuro se plantea que todo tenga Internet para facilitar el trabajo humano, la red tiene que ser rápida y la comunicación debe ser precisa y veloz [18], en conjunto con la ‘Digitalización’ que se enfoca en transformar las actividades físicas a un entorno digital [19].

Por otro lado, se usaron herramientas que permitieran desarrollar la elaboración de la parte física del proyecto, iniciando con la ‘Raspberry’ que es un ordenador de bajo costo el cual permite que todo usuario tenga acceso a la informática, su diseño es basado en hardware libres, este microcontrolador es utilizado en prototipos y en formación académica [20], también el ‘Arduino MEGA’ ya que es una placa de desarrollo basada en el microcontrolador ATmega2560 [21], se apoya con el ‘Sensor de temperatura y humedad AM2301’ como dispositivo capaz de medir variables físicas como la temperatura y la humedad [22]. Por último, se encuentra la ‘Adquisición de datos’ que se basa en tomar los datos del mundo real para generar datos que puedan ser manipulados por un ordenador o dispositivo electrónico [23].

METODOLOGÍA

Para iniciar con este proyecto (Figura 1) se llevó a cabo primeramente la identificación de las variables según su necesidad e importancia. Las variables que fueron utilizadas y estudiadas son: Temperatura, humedad y lluvia; posteriormente, se adquirieron los datos de estos sensores por medio de un Arduino MEGA en el cual se diseñó la programación para obtener dichos datos.

Se diseñó y creó la arquitectura donde fueron procesados en la Raspberry pi3 donde se realizó una interfaz gráfica con distintas funciones para así poder monitorear e inspeccionar los datos obtenidos por estos sensores y poder verificar su correcto funcionamiento.

Para tener la seguridad de que los datos obtenidos en tiempo real por los sensores fueran correctos, se tuvo que evidenciar y comprobar que el desarrollo de estos procedimientos fuera óptimo.

Después de adquirir los datos proporcionados por los sensores, se diseñó y creó una página web donde se pudiera tener acceso a los datos de los sensores en tiempo real.

Como método de privacidad de los datos obtenidos, se categorizó por medio de roles que permitiera llevar un control de la información adquirida. Es decir, que el rol docente tiene como privilegio observar y eliminar datos del historial mientras que, el rol estudiante solo puede observar.

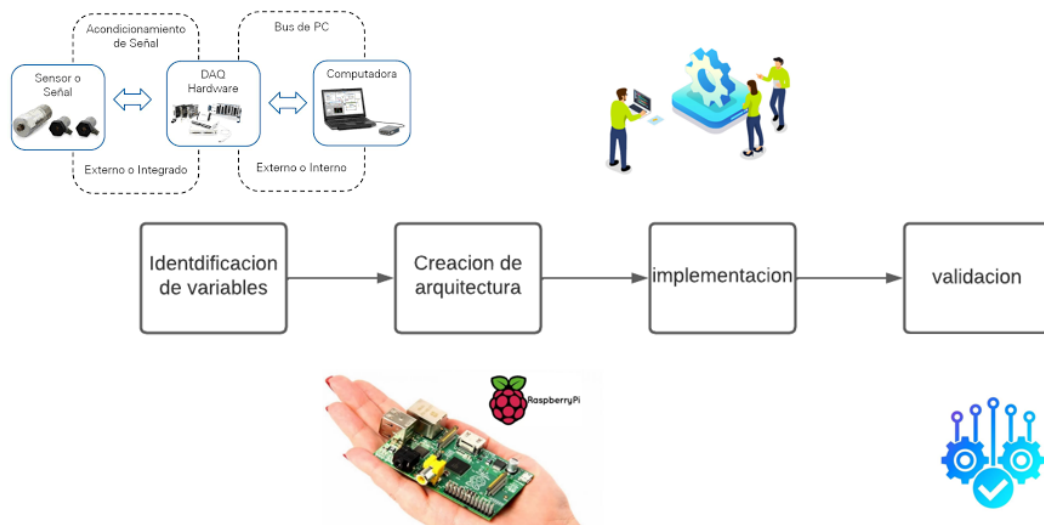


Figura 1. Metodología.

I. AQUISICIÓN DE DATOS

Ahora bien, para realizar la parte práctica del proyecto que permitiera la adquisición de los datos en tiempo real (Figura 2), se inició usando el Arduino MEGA, el cual se usó como medio para adquirir los datos del Sensor de lluvia que funciona con una placa que tiene expuestas unas líneas de cobre sirviendo como una referencia variable, esta resistencia cambia según la cantidad de agua que se repose en ella y tiene como objetivo determinar el momento exacto en el que llueve, este tiene las siguientes conexiones: colector de voltaje (VCC), ground o tierra (GND) y por último, una salida analógica que van conectadas al Arduino MEGA (Tabla 1). Por otro lado, para obtener los datos del sensor de temperatura y humedad se utilizó el (AM2301) que es un sensor que permite recibir información sobre la temperatura y la humedad, tiene una salida digital básica y es de bajo costo, este sensor tiene como conexión: voltaje de operación (5v), data (salida digital) y una tierra (GND) (Tabla 2), Después de adquirir estos datos se procede a implementar el uso de la Raspberry pi3 como herramienta que permite darle seguimiento al historial por medio de la interfaz creada.

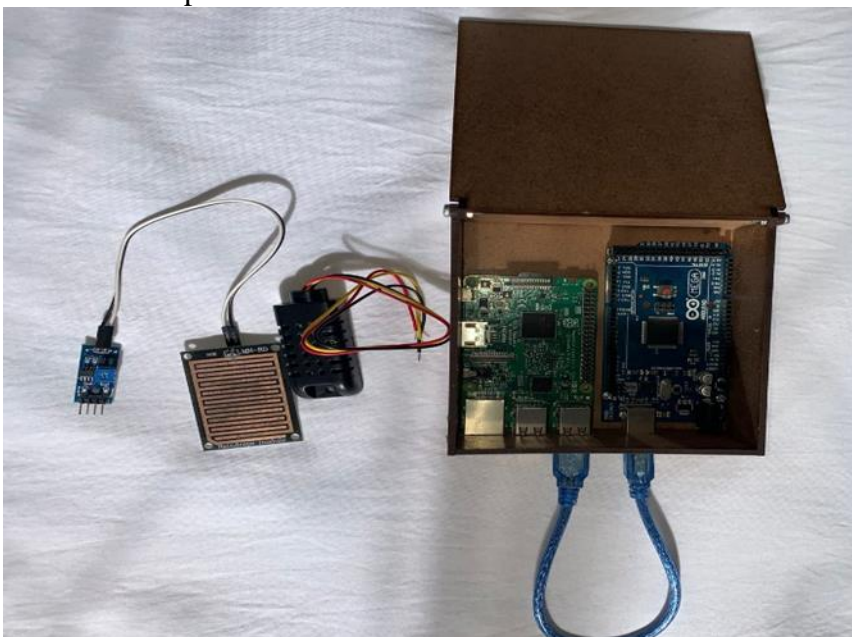


Figura 2. Componentes electrónicos y sistemas embebidos.

➤ COMPONENTE ELECTRÓNICA

En la componente electrónica, se usaron los sensores de temperatura y humedad (AM2301) y de lluvia (YL-83), el sensor AM2301 integra un sensor capacitivo de humedad, un termistor y un microcontrolador encargado de realizar la conversión analógica a digital A/D, la integración del sistema es rápida y fácil, su tamaño es pequeño, tiene un bajo consumo de energía y la distancia de transmisión de la señal es de hasta veinte (20) metros. Este producto tiene una excelente calidad, una respuesta rápida y una gran capacidad anti-interferencias, por lo tanto se considera como la mejor

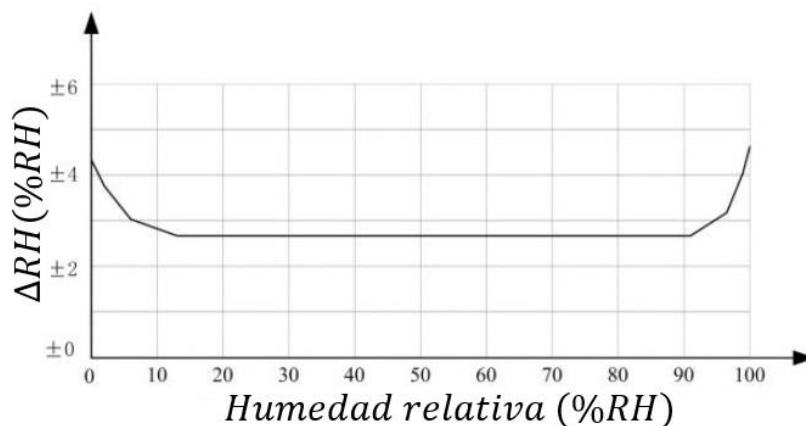
opción de todo tipo de aplicaciones e incluso las más exigentes, las variables a tener en cuenta en la elección de este sensor fueron las siguientes: Ultra-baja potencia, la distancia de transmisión, calibración totalmente automatizada, el uso de sensor de humedad capacitivo, es completamente intercambiable, tiene salida digital estándar, excelente estabilidad a largo plazo y alta precisión en los dispositivos de medición de la temperatura. El AM2301 tiene un rango de alimentación de 3,3V – 5,2V, la tensión de alimentación recomendada es de 5V, para que este sensor funcione correctamente es necesaria la instalación de librerías en Arduino (DHT21).

En la (Tabla 1) se tienen las conexiones de los pines del sensor con el Arduino MEGA, el pin 1 del sensor que es de color rojo va conectado a los 5V del Arduino, el pin 2 que es la salida digital, es de color amarillo, estará conectada al pin digital 2 y el pin 3 que es el GND (Tierra) del sensor, de color negro, irá conectado al GND del Arduino.

Arduino	Modulo
5v	Pin 1 (+)
Pin 2	Pin 2 (out)
GND	Pin 3 (-)

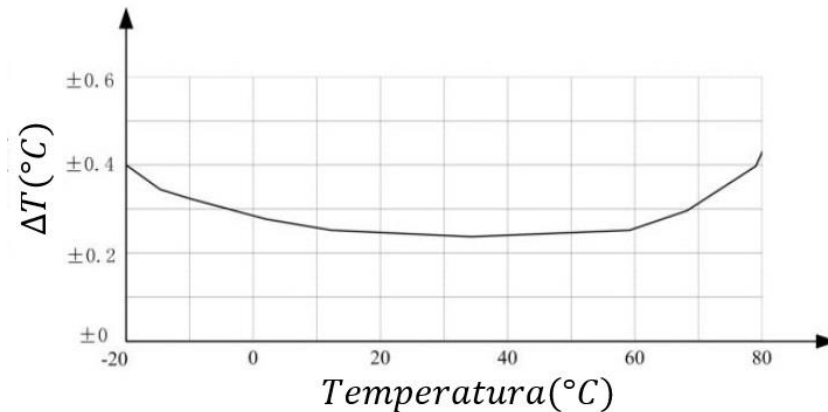
Tabla 1. Conexiones AM2301.

En la (Gráfica 1) se observa el porcentaje de error de la humedad relativa, en el eje Y se tiene a delta de ΔRH que hace referencia a la variación con respecto al porcentaje de RH, en el eje X se tiene el porcentaje real de la humedad relativa, se observa que el porcentaje de erros varía entre más o menos 2.7 y 4.3.



Gráfica 1. Error de la humedad relativa.

En la (Gráfica 2) se observa el máximo error en la temperatura, en el eje Y se tiene a delta de ΔT que hace referencia a la variación con respecto a la temperatura en grados centígrados, en el eje X se tiene los grados centígrados en tiempo real de la temperatura, se observa que el valor de ΔT varía entre más o menos 2.6 y 4.



Gráfica 2. El máximo error en la temperatura.

A continuación, se demostrará las ecuaciones y conversiones internas que hace la librería DHT21 para obtener el porcentaje de humedad relativa y la temperatura en grados centígrados. Primero se obtienen los datos digitales del sensor (ceros y unos) sobre la humedad y la temperatura; segundo se suman todos los datos digitales adquiridos para poder encontrar el bit de paridad; tercero para saber el porcentaje de humedad se convierte el dato digital de alta humedad y baja humedad en hexadecimal para luego, realizar una multiplicación y una suma que permita conocer el porcentaje de humedad relativa en tiempo real, se realiza nuevamente este procedimiento para conocer la temperatura en grados centígrados.

0000 0010
Alta humedad

1001 0010
Baja humedad

0000 0001
Alta temperatura

0000 1101
Baja temperatura

1010 0010
Bits de paridad

$$0000\ 0010 + 1001\ 0010 + 0000\ 0001 + 0000\ 1101 \\ = 1010\ 0010 \text{ (Bits de paridad)}$$

$$0000\ 0010\ 1001\ 0010 = 0292H \text{ (Hexadecimal)} = 2 \times 256 + 9 \times 16 + 2 = 658$$

$$\Rightarrow \text{Humedad} = 65.8\%$$

$$0000\ 0001\ 0000\ 1101 = 10DH \text{ (Hexadecimal)} = 1 \times 256 + 0 \times 16 + 13 = 269$$

$$\Rightarrow \text{Temperatura} = 26.9\ ^\circ\text{C}$$

Por otro lado, se usó el sensor de lluvia (YL-83) con el objetivo de discernir cuando esta lloviendo y el nivel de intensidad de la lluvia, puesto que este sensor cuenta con una placa y una superficie niquelada para resistir mejor la oxidación y alargar su vida útil, además, cuenta con sensibilidad ajustable por medio del potenciómetro azul. Este sensor permite detectar gotas de lluvia, este modulo genera un corto circuito cada vez que las pistas se mojan, pues el agua hace que se cree un camino de baja resistencia entre las pistas conectadas al GND, la corriente que corre a través de estos caminos se ve limitada por resistencias de $10K\Omega$ en cada conductor; es decir, que impide que el corto circuito que se genera cuando se moja la placa vaya a ocasionar daños temporales o permanentes en el microcontrolador.

En la (tabla 2) se puede observar que el pin 1 del modulo (VCC) irá conectado a 5V del Arduino, el pin 2 conocido como salida analoga, convierte en números la señal de referencia, este pin estará conectado al pin A0 del Arduino, por último, el pin 3 que es GND (Tierra) se conectara al GND el Arduino.

Arduino	Modulo
5v	Pin 1 (+)
Pin A0	Pin 2 (out)
GND	Pin 3 (-)

Tabla 2. Conexiones sensor de lluvia.

➤ COMPONENTE DE PROGRAMACIÓN

Para el desarrollo de la componente de programación, se inició descargando las librerías de DHT21 para Arduino, se procedió a crear un código en el cual se implementó la lectura del pin digital que salía del sensor de temperatura y humedad (AM2301), incluyendo también la lectura del pin analógico del sensor de lluvia (YL-83) para adquirir los datos de los sensores y ser procesados.

En el (Diagrama de flujo 1) se representa el orden en el que se realizó la programación en Arduino, se da inicio al programa, luego, se declaran las variables de tipo float o int, ahora bien, se procede a iniciar la función setup, se configura la velocidad serial en la

que va a estar el programa y con esto se da fin a la función setup. Aun así, se comienza a desarrollar la función loop donde se pregunta si existe un (1) byte disponible en el puerto, si existe se activa la lectura análoga y digital de los pines y se dará un retraso de censado de dos (2) segundos entre cada dato, luego se preguntara si existe algún criterio que cierre el bucle, si esto llega a ser positivo se llevará el programa a una terminación, pero si el criterio es negativo seguirá dentro de la función loop hasta que exista un criterio de cierre, si no existe un bit disponible en el puerto serial, entonces se genera una advertencia que estipula que el sensor no es activado y luego se llevara a un final del proceso, para dar paso a la programación en Python.

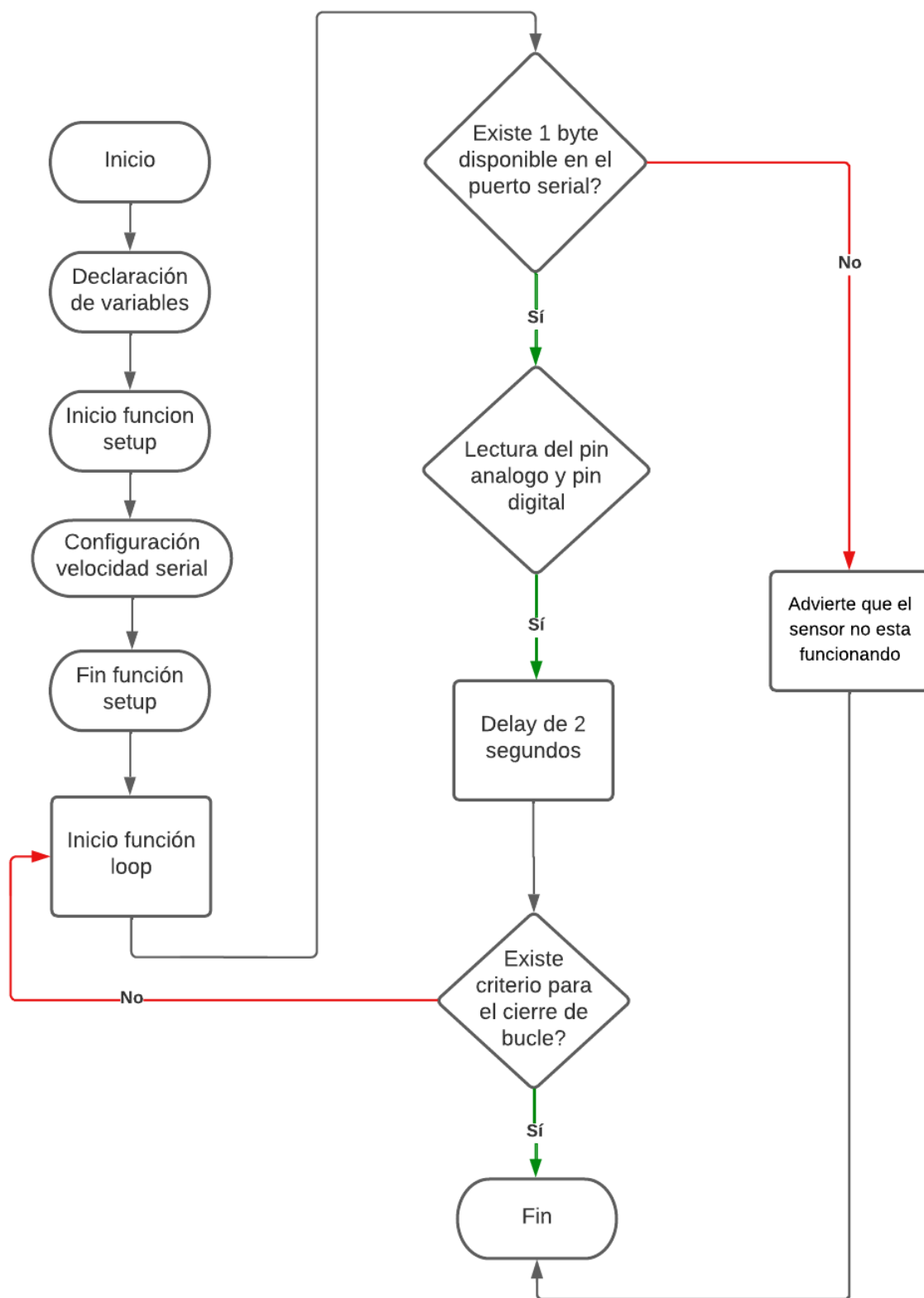


Diagrama de flujo 1. Código Arduino.

II. PROCEDIMIENTO Y RESULTADOS

El procedimiento que se realizó para cumplir con las expectativas propuestas en este proyecto, se identificaron las diferentes variables que se pudieran presentar alrededor de indicadores como la temperatura, humedad y lluvia, para esto se utilizó un Arduino MEGA que tenía como objetivo adquirir los datos de los sensores en tiempo real y enviarlos por comunicación serial al sistema embebido “Raspberry pi 3” (Figura 3), este tiene conexiones como: fuente, ethernet y comunicación serial.

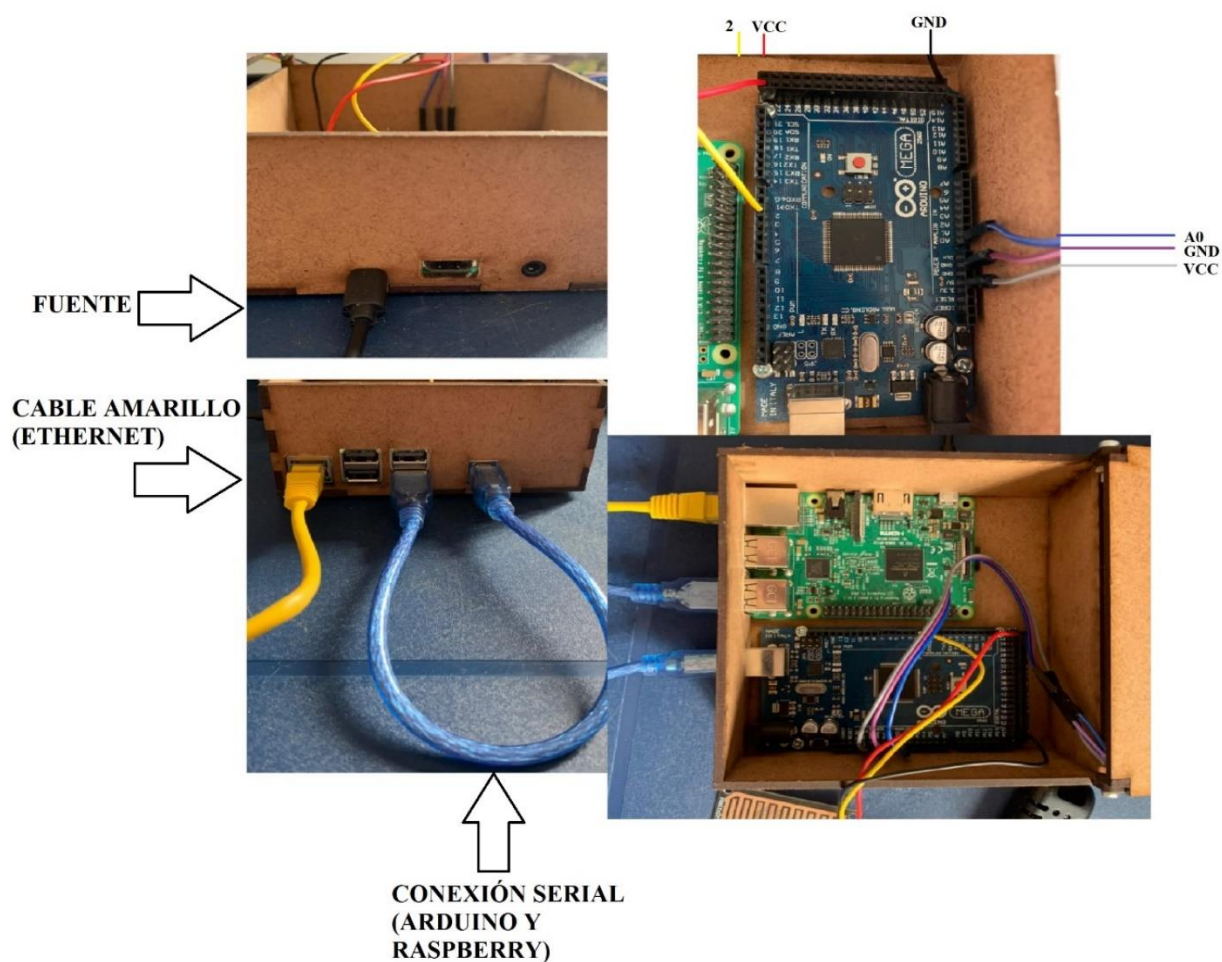


Figura 3. Sistema de adquisición con Arduino MEGA, comunicación serial con la Raspberry pi 3 y conexiones de sensores.

Para el desarrollo se usó la Raspberry pi 3 que facilitó el diseño y creación de la interfaz gráfica por medio de Python y Tkinter para monitorear y visualizar las variables físicas en tiempo real, incluyendo un acceso a un historial de datos controlado a través de un usuario y una contraseña colectiva, estos usuarios se encuentran basados en roles, el usuario del estudiante tiene como rol únicamente visualizar el contenido informático que allí reposa conocido como el historial de datos, mientras que, el rol del docente puede visualizar y eliminar dichos datos, esta información puede ser visualizada por medio de una página web local.

Se necesitó la instalación de las librerías pertinentes que brindaran las herramientas necesarias para el desarrollo de la interfaz gráfica, se procede a elaborar el código en Python, siguiendo la siguiente secuencia de pasos, primero, se importa las librerías, segundo, se crea la ventana de presentación donde se podrá visualizar el nombre del proyecto, nombre de la institución en la que se ejecutó y nombre de la persona que elaboró el mismo, también, se tiene tres botones con diferentes funciones, para el botón de “Datos en tiempo real”, se creó una ventana aparte donde mostraba dos botones con el nombre del sensor que deseaba monitorear e inspeccionar, al escoger un sensor se abrirá una ventana donde graficará los datos que se están obteniendo en tiempo real. Con el botón de “inicio de sesión” se creó una ventana para ingresar los datos de “usuario” y “contraseña”, para poder acceder al historial de datos se tiene una seguridad basada en roles, si se ingresa como estudiante solo tendrá el privilegio de visualizar los datos, pero al ingresar con el usuario de docente se tendrá el privilegio de eliminar y visualizar los datos que han sido obtenidos por los sensores. Además, se tiene un botón de ayuda, al pulsar ese botón se abre una ventana en la que se podrá ver un tutorial de como conectar el Arduino, a que puerto USB de la Raspberry conectarlo y como conectar los sensores, también, se presenta un cuadro de texto donde se encontraran los usuarios con sus respectivas contraseñas (Figura 4).

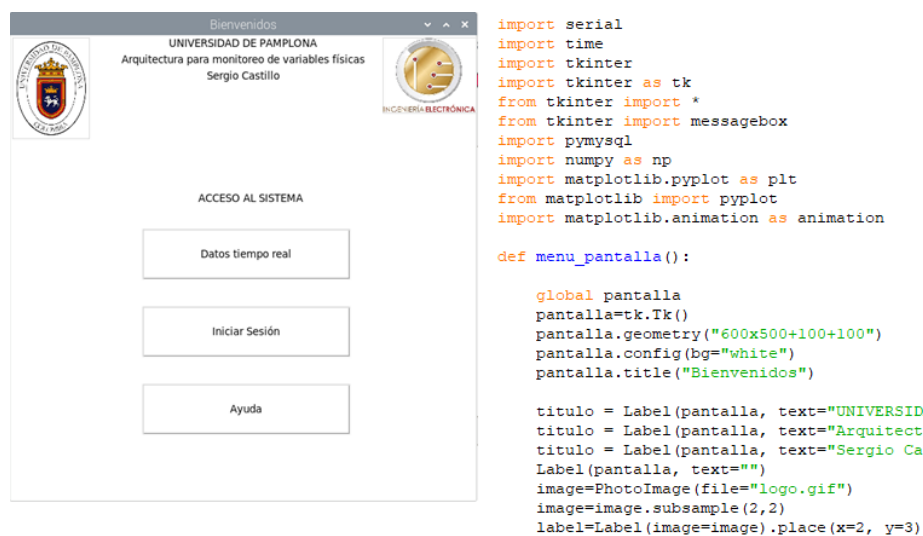


Figura 4. Interfaz gráfica y código en Python.

Al escoger el sensor de lluvia (Figura 5) automáticamente se mostrará la gráfica en tiempo real de dicha variable, esta gráfica fue tomada en una hora la cual empezó con una fuerte lluvia y luego se mantuvo en una lluvia más suave.

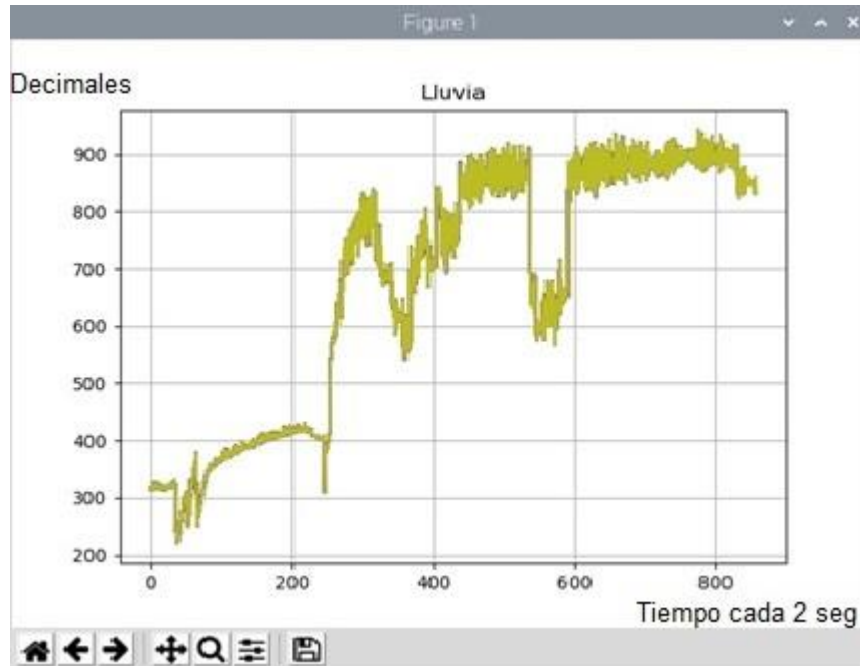


Figura 5. Gráfica sensor de lluvia una hora de prueba.

Por otro lado, al escoger el sensor de Temperatura/Humedad (Figura 6) se evidenciará la gráfica en tiempo real de esta condición, estos datos se tomaron por seis (6) horas donde la temperatura oscilo entre los trece (13°) y quince (15°) grados centígrados y la humedad varía entre el noventa (90%) y noventa y nueve (99%) por ciento (%).

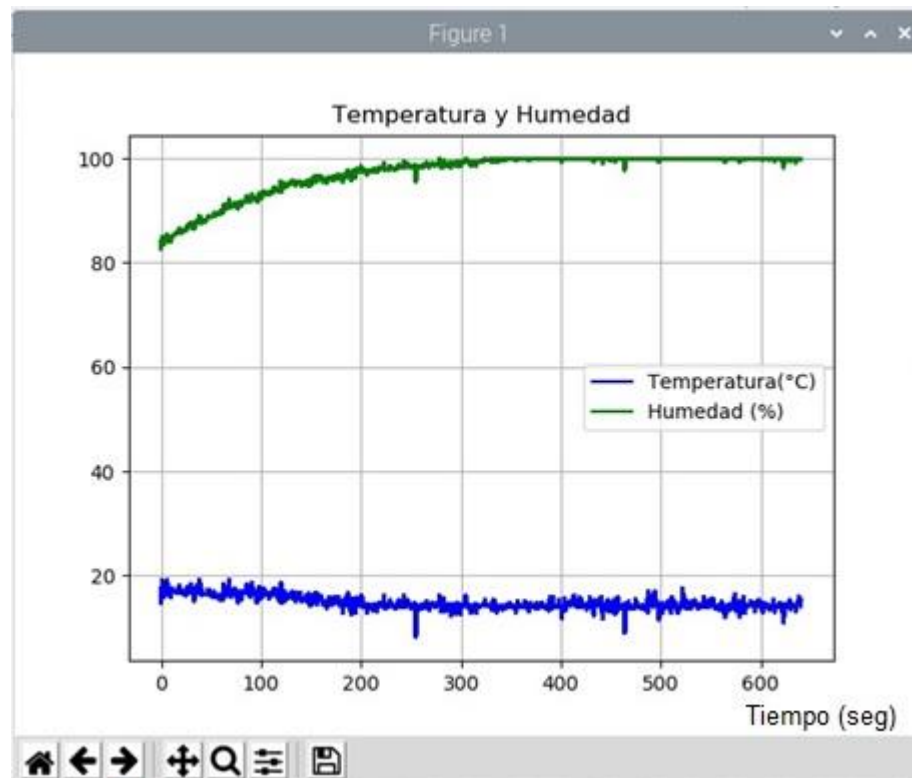


Figura 6. Gráfica sensor de temperatura y humedad.

Para el desarrollo de la página web se utilizó HTML con el cual se estructuró la página, con CSS se le dieron estilos a la página web, Chart.JS se utilizó para renderizar las gráficas y con Flask se obtuvieron los datos de los sensores y se enviaron a las gráficas, en la interfaz principal (Figura 7) se encuentran dos botones que permiten visualizar las variables físicas en tiempo real del proyecto (Figura 8) (Figura 9).



Figura 7. Inicio página web.

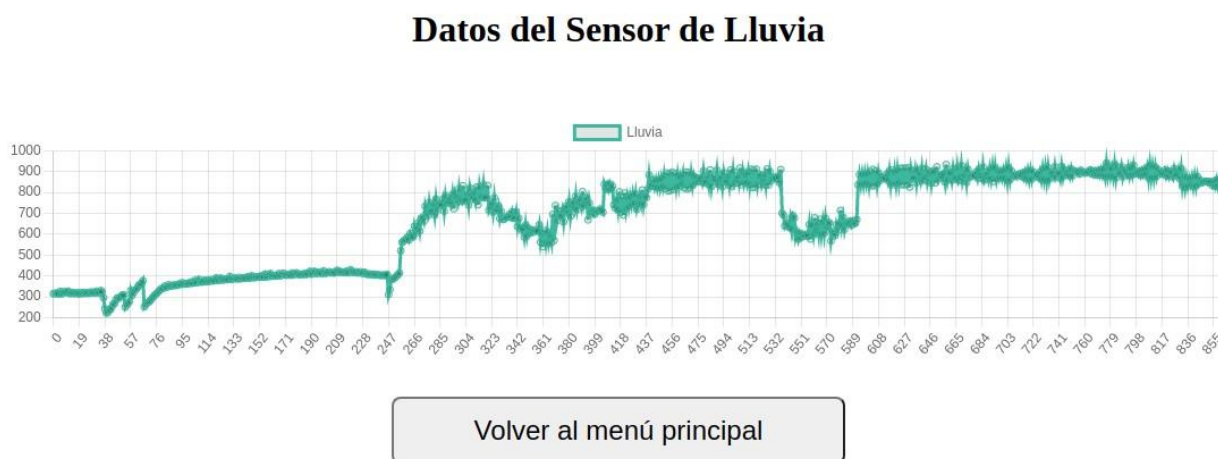


Figura 8. Gráfica Sensor de lluvia visualizada en la página web.

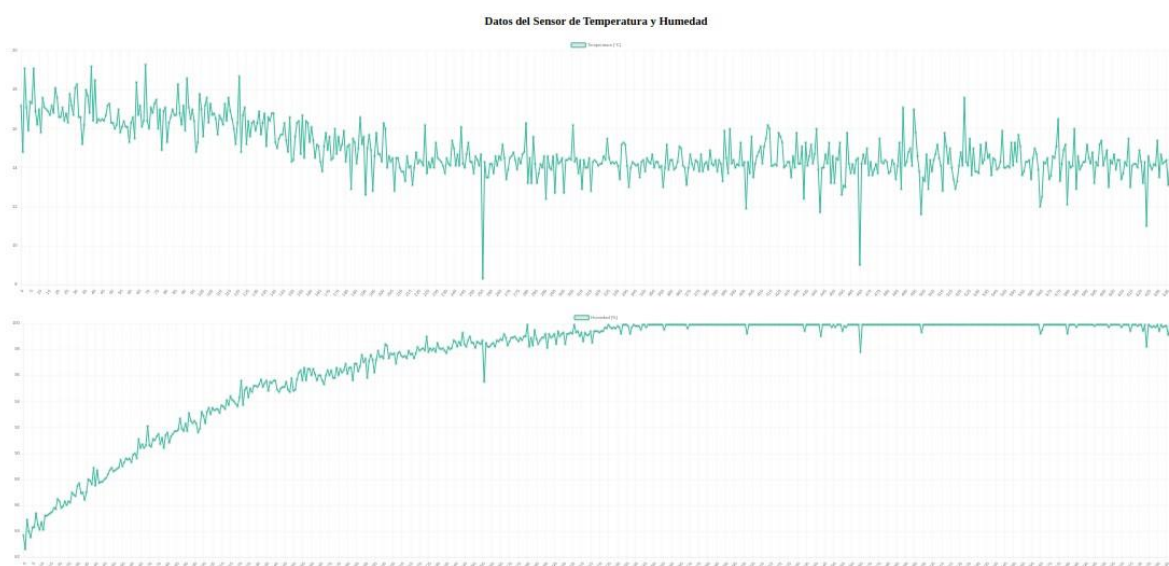


Figura 9. Gráfica sensor de temperatura y humedad visualizada en la página web.

En la (Figura 10) se puede observar el prototipo físico de la arquitectura para monitoreo e inspección de sistemas adherentes a tecnologías digitales.

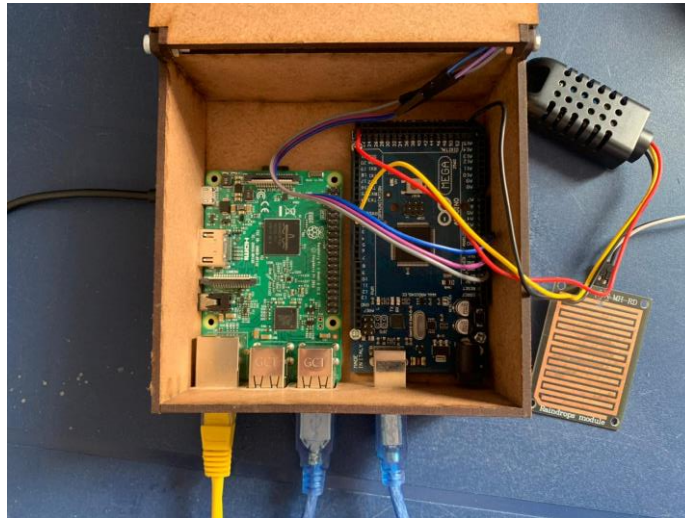


Figura 10. Arquitectura.

III. CONCLUSIONES

Se observó las falencias y necesidades que se tenían en ciertas carreras y su población estudiantil para poder identificar dichas variables y asimismo poder crear ciertas soluciones que las pudieran suplir en gran medida. Por lo tanto, puede decir que, se logró identificar las variables para definir las componentes que hicieron parte de la arquitectura, evidenciando y comprobando el desarrollo óptimo de los procedimientos que se debían llevar a cabo por los sensores que serían los adecuados para ciertas condiciones climáticas establecidas. Ahora bien, al tener clara la información que se obtuvo a través de los sensores, se creó una arquitectura implementable compuesta por: Arduino MEGA, Raspberry Pi3, Sensor de temperatura y Humedad (Am2301), Sensor de lluvia (YL-83) y las respectivas conexiones de los sensores. Esta arquitectura soporta la inspección, el monitoreo y el tratamiento de variables físicas por medio de los sensores que se utilizaron en los espacios correspondientes para luego someterlos a una validación de la información recolectada, transmitida y recibida por los canales establecidos. Además, al validar la información recolectada por el Arduino MEGA, se logró configurar los sistemas involucrados a través de la comunicación serial entre el Arduino Mega y la Raspberry pi3, por medio de un código que se creó en Python, se nombra el puerto USB que está siendo usado por Arduino en la Raspberry para poder visualizar los datos que son obtenidos por ellos mismos y así crear una sola arquitectura. Después de tener la arquitectura 100% funcional, se sometieron los sensores a prueba en un entorno real, el sensor de temperatura y humedad se dejó activo por seis (6) horas para observar el comportamiento de estas variables, el sensor de lluvia se dejó encendido por una (1) hora en la cual tuvo un intervalo de alta lluvia y poca lluvia y comprobar su correcto funcionamiento. Ahora, se da paso a evaluar el funcionamiento de la arquitectura se documentan los resultados por medio de este artículo, teniendo en cuenta

que, se facilitó la interacción con la página web y el usuario para la obtención de datos climáticos, los resultados que son observados en la página web pueden ser visualizados por el rol estudiante y visualizados y eliminados por el rol de docente. Y, por último, se le da cumplimiento al objetivo principal, puesto que se desarrolló una arquitectura que permite monitorear e inspeccionar sistemas compatibles con entornos digitales en tiempo real. Cabe resaltar que, por cuestiones de comunicación con otras carreras interesadas en la elaboración de este proyecto, no se pudo disponer del sensor de flujo para poder determinar el flujo de un caudal, ya que hacía parte de las aspiraciones del resultado final.

IV.REFERENCIAS

- [1] H. C. H. Z. X. C. W. Y. Z. Zhang, «A Universal Calibration Device for an Air Flow Sensor of the VAV Terminal Unit,» *Sensors* 2022, Vol. 22, Page 5797, vol. 22, n° 15, p. 5797, 2022.
- [2] K. U. S.-H. R. C. M. L. M.-R. A. L. M. J. Antela, «Development of an Automatized Colorimeter Controlled by Raspberry Pi4,» *SSRN Electronic Journal*, 2022.
- [3] M. Y. T. Ozaki, «High-speed embedded optical flow measurement system for real-time use,» *International Conference on Control, Automation and Systems*, Vols. % 1 de %2021-October, pp. 331 - 335, 2021.
- [4] L. S. F. L. M. G. A. L. M. B. F. S. J. N. C. E. B. M. P. Herculano, «Improvements on time-resolved measurement of gas volume production in laboratories bench systems,» *Flow Measurement and Instrumentation*, vol. 84, p. 102102, 2022.
- [5] A. M. G. A. V. T. F. P. Murciano Hueso, « [Systematic review of digital technology acceptance in older adults. Perspective of TAM models], » *Revista española de geriatría y gerontología*, vol. 57, pp. 105 - 117, 2022.
- [6] N. S. H. N. L. J. M. Sushma, «Smart Water Flow Meter for Improved Measurement of Water Usage in a Smart City,» *2022 2nd International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies, ICAECT 2022*, 2022.
- [7] S. S. S. M. S. S. K. Khadane, «Real-Time Object Size Dimensioning in Raspberry Pi,» *7th International Conference on Communication and Electronics Systems, ICCES 2022 - Proceedings*, pp. 344 - 348, 2022.
- [8] D. D. Agurto Valverde, «Integración de un sistema de adquisición de datos mediante el uso de un Arduino Mega y Raspberry Pi 3 como servidor web y base de datos,» *Universidad Nacional de Piura*, 2020.
- [9] K. G. Chaudhari, «Windmill Monitoring System Using Internet of Things with Raspberry Pi,» *SSRN Electronic Journal*, 2019.
- [10] S. F. A. M. M. I. Y. Purwanti, «Temperature Monitoring System for Egg Incubators Using Raspberry Pi3 Based on Internet of Things (IoT),» *Journal of Robotics and Control (JRC)*, vol. 2, pp. 349 - 352, 2021.
- [11] K. R. S. S. R. K. R. N. S. M. Nataraj, «Greenhouse Service Implemented on Raspberry Pi-3,» *Cognitive Science and Technology*, pp. 579 - 586, 2022.
- [12] E. García Martínez, «Desarrollo de un controlador PID industrial de bajo coste mediante Raspberry Pi para control de temperatura,» 2016.
- [13] D. L. A. L.-F. H. R.-J. M. F.-R. F. Glez-Peña, «Web scraping technologies in an API world,» *Briefings in Bioinformatics*, vol. 15, pp. 788 - 797, 2014.
- [14] A. C. A. S. U. B. S. Choudhury, «HealthSaver: a neural network based hospital recommendation system framework on flask webapplication with realtime database and RFID based attendance system,» *Journal of Ambient Intelligence and Humanized Computing* 2021 13:10, vol. 13, n° 10, pp. 4953 - 4966, 2021.
- [15] F. K. C. Crick, «A framework for consciousness,» *Nature Neuroscience* 2003 6:2, vol. 6, pp. 119 - 126, 2003.
- [16] R. C. M. Levering, «The portrait of a common HTML web page,» *Proceedings of the 2006 ACM Symposium on Document Engineering, DocEng 2006*, vol. 2006, pp. 198 - 204, 2006.
- [17] Challenger-Pérez, Y. Diaz-Ricardo y R. A. Becerra-Garcia, «El lenguaje de programación Python,» *ciencias Holguin*, vol. XX, n° 2, pp. 1 - 13, 2014.
- [18] S. M. A. (. N. I. o. I. E. L. V. M. G. I. R. A. (. N. I. o. I. E. L. V. M. G. I. T. Group, «Internet of Things (IoT): A Literature Review,» *Journal of Computer and Communications*, vol. 03, n° 05, pp. 164 - 173, 2015.
- [19] F. S. G. Vilaplana, «Digitalización y personas,» *Revista Empresa y Humanismo*, vol. 23, n° 1, pp. 113 - 137, 2020.
- [20] C. Severance, «Eben upton: Raspberry Pi,» *Computer*, vol. 46, n° 10, pp. 14 - 16, 2013.
- [21] J. Z. L. N. H. H. Arifin, «PERANCANGAN MUROTTAL OTOMATIS MENGGUNAKAN MIKROKONTROLLER ARDUINO MEGA 2560,» *JURNAL MEDIA INFOTAMA*, vol. 12, n° 1, 2016.

- [22] S. E. M. Z. A. H. M. M. K. M. A. H. M. B. Saha, «IOT BASED SMART HOME AUTOMATION AND MONITORING SYSTEM,» Khulna University Studies, pp. 133 - 143, 2022.
- [23] B. D. P. N. C. L. M. J. A. S. P. C. A. S. S. M. G. A. V. R. J. P. R. T. A. S. Vázquez, «Implementación de una estación de monitoreo y adquisición de datos a distancia con Arduino y Visual Studio.NET,» Jóvenes en la Ciencia: XXVI Verano de la Ciencia. Vol. 10(2021), 2022.