

**INSTITUTO TECNOLÓGICO DE COSTA RICA**

**Carrera de Ingeniería en Mantenimiento Industrial**

**TEC** | Tecnológico  
de Costa Rica

**Sistema de monitoreo y registro en tiempo real de la humedad relativa y temperatura en el cuarto húmedo del Centro de Investigaciones en Vivienda y Construcción**

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en Mantenimiento Industrial con el grado académico de Licenciatura**

**Luis Hernesto Pérez Buendía**

**Cartago, Noviembre de 2023**



**Carrera evaluada y acreditada por:**

Canadian Engineering Accreditation Board  
Bureau Canadien d'Accréditation des Programmes d'Ingénierie

**Datos personales**

*Nombre completo: Luis Hernesto Pérez Buendía*

*Número de cédula: 305200713*

*Número de carné: 2017116824*

*Edad: 24 años*

*Números de teléfono: 85881632*

*Correo electrónico: [heperez@estudiantec.cr](mailto:heperez@estudiantec.cr)*

*Provincia de Cartago, Turrialba, Residencial El Coyol Casa 20e*

**Datos de la Empresa**

*Nombre: Centro de investigaciones en vivienda y construcción (CIVCO)*

*Actividad Principal: Investigación y prestación de servicios en el sector construcción.*

*Dirección: V33R+PXG, Provincia de Cartago, Cartago, Costa Rica*

*Contacto: [civco@tec.ac.cr](mailto:civco@tec.ac.cr)*

*Teléfono: 25502309*



Sistema de monitoreo y registro en tiempo real de la humedad relativa y temperatura en el cuarto húmedo del Centro de Investigaciones en Vivienda y Construcción © 2023 by Luis Hernesto Pérez Buendía is licensed under [Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/)

## **Dedicatoria**

A mis papás y hermanos.

A Dios.

# Carta de aceptación de la empresa

## Aceptación de proyecto de graduación

Giannina Ortiz Quesada

Jue 18/05/2023 11:00

Para: Ignacio del Valle Granados

---

### Este mensaje proviene de un remitente externo

Este mensaje vino de fuera de su organización.

Buenos días, estimado Ignacio,

Por este medio le indico que el anteproyecto de graduación titulado: **“Sistema de monitoreo y registro de la humedad y temperatura en el cuarto húmedo del CIVCO basado en tecnología IoT”**, presentado por el estudiante Luis Hernesto Buendía Pérez, número de cédula 305200713 y número de carné del estudiante 2017116824, cuenta con la aceptación del CIVCO.

Para el CIVCO es de gran importancia contar con este apoyo para garantizar las condiciones de la cámara húmeda utilizada tanto en procesos de investigación como de extensión.

Agradezco de antemano el apoyo de su Escuela para el desarrollo de este tipo de proyectos.

Ing. Giannina Ortiz Quesada, MSc. Coordinadora  
Centro de Investigaciones en Vivienda y Construcción - CIVCO  
Escuela de Ingeniería en Construcción  
Tecnológico de Costa Rica  
[gortiz@tec.ac.cr](mailto:gortiz@tec.ac.cr)  
(506) 8921-6315

**TEC** | Tecnológico  
de Costa Rica

Profesor guía

Ing. Suzanne Karina Melara Cruz.

Coordinador de la práctica

Ing. Ignacio Del Valle Granados

Asesor Industrial

Ing. Giannina Ortiz Quesada.

Tribunal Examinador

Ing. Juan José Rojas Hernández.

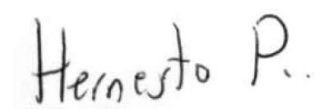
Ing. Luis Chevez Gómez.

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, Septiembre 2023

A handwritten signature in black ink that reads "Hernesto P." with a period at the end.

Luis Hernesto Pérez Buendía

Céd: 305200713

## Resumen

El presente proyecto se centró en la elaboración e implementación de un sistema de monitoreo en tiempo real para temperatura y humedad utilizando Arduino, específicamente diseñado para supervisar las condiciones ambientales del cuarto húmedo del Centro de Investigaciones en Vivienda y Construcción. La motivación principal de este trabajo radicó en asegurar que se cumplan los parámetros establecidos por las normativas ASTM C511 y la ISO/IEC 17025, garantizando la confiabilidad de los procesos que se llevan a cabo en dicho ambiente.

A través de un análisis, se seleccionaron los componentes óptimos para el desarrollo de este prototipo. El sistema no solo monitoriza, sino que también se integra con la nube, lo que permite almacenar datos, generar gráficos históricos y activar alertas ante cualquier anomalía detectada. Además, se realizó un análisis económico comparativo entre el prototipo desarrollado y las soluciones comerciales existentes.

El proyecto culminó con la exitosa elaboración y puesta en marcha de un prototipo de sistema de monitoreo y registro de humedad relativa y temperatura, adaptado a las necesidades específicas del cuarto húmedo del Centro de Investigaciones en Vivienda y Construcción. Este sistema proporciona reportes de monitoreo continuo y alertas en tiempo real, cumpliendo con las demandas operativas y los requisitos de monitoreo alineándose con los estándares estipulados por la norma ISO/IEC 17025, que son esenciales para la acreditación del laboratorio. No obstante, es importante señalar que la confiabilidad de los datos recabados no ha podido ser plenamente confirmada, ya que el certificado de calibración requerido no llegó a tiempo para validar las mediciones del sistema.

Palabras clave: análisis económico, Arduino, cuarto húmedo, CIVCO, integración con la nube, monitoreo, normativas ASTM C511, ISO/IEC 17025, tiempo real.

## **Abstract**

This project focused on the development and implementation of a real-time monitoring system for temperature and humidity using Arduino, specifically designed to supervise the environmental conditions of the moist room at the Centro de Investigaciones en Vivienda y Construcción. The main motivation behind this work was to ensure compliance with the parameters established by the ASTM C511 and ISO/IEC 17025 standards, thereby guaranteeing the reliability of the processes carried out in that environment.

Through analysis, the optimal components for the development of this prototype were selected. The system not only monitors but is also integrated with the cloud, which allows for data storage, the generation of historical graphs, and the activation of alerts in the presence of any detected anomalies. In addition, a comparative economic analysis was conducted between the developed prototype and existing commercial solutions.

The project culminated in the successful creation and implementation of a prototype system for monitoring and recording relative humidity and temperature, tailored to the specific needs of the moist room at the Centro de Investigaciones en Vivienda y Construcción. This system provides continuous monitoring reports and real-time alerts, meeting operational demands and monitoring requirements in alignment with the ISO/IEC 17025 standard, which are essential for laboratory accreditation. However, it is important to note that the reliability of the data collected could not be fully confirmed, as the required calibration certificate did not arrive in time to validate the system's measurements.

Keywords: Arduino, CIVCO, cloud integration, economic analysis, ISO/IEC 17025, monitoring, real-time, ASTM C511 standards, wet room.



## ÍNDICE GENERAL

Datos personales .....	ii
Datos de la Empresa .....	ii
Carta de aceptación de la empresa .....	iv
Capítulo 1: Introducción .....	1
1.1 Reseña de la empresa .....	1
1.1.1 Descripción del proceso productivo.....	3
1.2 Antecedentes .....	4
1.2.1 Antecedentes teóricos.....	4
1.4.2 Antecedentes de Campo.....	5
1.3 Descripción del problema.....	6
1.4 Justificación.....	9
1.5 Objetivos del proyecto .....	11
1.5.1 Objetivo General.....	11
1.5.2 Objetivos Específicos.....	11
Capítulo 2: Marco Teórico .....	12
2.1 Cuartos Húmedos.....	12
2.2 Sistemas de monitoreo de humedad y temperatura.....	13
2.3 Microcontrolador Arduino .....	14
2.4 Sensores de humedad y temperatura de precisión.....	15
2.5 Protocolos de comunicación .....	18
2.6 Plataformas de almacenamiento y visualización de datos en la nube.....	18
2.7 Análisis de costos en la implementación de soluciones tecnológicas.....	19
2.8 Conexión Ethernet.....	20
2.9 La nube.....	21
2.10 Llaves API.....	21
2.11 Trabajos Cron.....	22
Capítulo 3: Metodología .....	23
3.1 Metodología para objetivo específico 1 .....	23
3.2 Metodología para objetivo específico 2 .....	24
3.3 Metodología para objetivo específico 3 .....	26

3.4 Metodología para objetivo específico 4 .....	28
Capítulo 4: Desarrollo y Selección de Componentes del Prototipo .....	31
4.1 Variables a monitorear .....	31
4.2 Conceptualización del Prototipo .....	32
4.3 Hardware: Arduino Uno.....	35
4.3.1 Descripción general.....	35
4.3.2 Razones de selección.....	35
4.3.3 Características y especificaciones del Arduino Uno R3 .....	38
4.4 Sensor: DHT 22.....	40
4.4.1 Características del DHT22 .....	40
4.4.2 Justificación de la Elección.....	41
4.4.3 Integración con Arduino .....	44
4.5 Componentes y Materiales.....	44
4.5.1 Pantalla Liquid Crystal 1602.....	44
4.5.2 Módulo I2C FC-113 .....	47
4.5.3 Jumpers Macho-Macho.....	47
4.5.4 Breadboard Mini Modular.....	48
4.5.5 Cable Alarma 4 Hilos.....	49
4.5.6 Adaptador 9VDC .....	49
4.5.7 Caja de Proyecto Personalizada .....	50
4.5.8 Materiales Adicionales.....	51
4.6 Esquema de Conexión.....	51
4.6.1 Descripción del Esquema de Conexión.....	51
4.6.2 Visualización del Esquema .....	52
4.7 Programa para Arduino Uno .....	53
4.7.1 Introducción al Software .....	53
4.7.2 Descripción del Código.....	54
4.8 Prototipo Ensamblado y Operativo .....	58
Capítulo 5: Integración del Sistema de Monitoreo con la Nube .....	61
5.1 Selección de Hardware para Conexión a Internet: Ethernet Shield .....	62
5.1.1 Descripción .....	62
5.1.2 Justificación.....	63
5.1.3 Proceso de Instalación y Configuración.....	64
5.2 Selección de Plataforma de Almacenamiento en la Nube: ThingSpeak .....	66

5.2.1	Descripción de ThingSpeak .....	66
5.2.2	Justificación de selección de la plataforma.....	67
5.2.3	Proceso de Configuración .....	69
5.3	Modificación del Código de Arduino .....	71
5.4	Selección de Plataforma de Notificación: SendGrid.....	75
5.4.1	Descripción de SendGrid .....	75
5.4.1	Justificación de selección.....	75
5.4.2	Proceso de Configuración .....	78
5.5	Desarrollo del Programa de Reportes Históricos .....	79
5.6	Desarrollo del Programa de Alertas .....	82
5.6.1	Sistema de alertas de rangos permitidos en el monitoreo .....	83
5.6.2	Implementación de alertas de desconexión.....	85
5.7	Automatización de Alertas en Google Cloud.....	87
5.7.1	Despliegue del Código en Google Cloud Functions.....	87
5.7.2	Automatización mediante Google Cloud Scheduler.....	89
5.8	Pruebas Iniciales de Funcionamiento.....	91
Capítulo 6: Evaluación del prototipo .....		93
6.1	Implementación y puesta en funcionamiento del prototipo .....	93
6.1.1	Despliegue del prototipo .....	93
6.1.2	Configuración inicial.....	93
6.1.3	Consideraciones eléctricas y diagrama unifilar.....	95
6.1.4	Justificación de la ubicación del sensor .....	98
6.2	Monitoreo y registro de datos .....	99
6.2.1	Proceso de monitoreo.....	99
6.2.2	Herramientas de recuperación de datos.....	100
6.3	Pruebas de sistema de reportes y alertas .....	101
6.4	Análisis estadístico.....	102
6.4.1	Recolección de datos.....	102
6.4.2	Análisis estadístico.....	103
6.4.3	Resultados y documentación.....	103
6.5	Calibración del equipo .....	108
6.5.1	Proceso de Calibración Certificada.....	108
6.5.2	Importancia de la Calibración para CIVCO.....	109
6.5.3	Resultado de la calibración del equipo.....	109

6.6 Observaciones de la evaluación .....	110
Capítulo 7: Análisis comparativo de sistemas de monitoreo .....	111
7.1 Identificación de soluciones comerciales.....	111
7.1.1 Investigación de soluciones.....	112
7.1.2 Selección de soluciones comparables .....	112
7.2 Recopilación de datos de costos.....	118
7.2.1 Costos de soluciones comerciales .....	118
7.2.2 Costo de adquisición del prototipo.....	121
7.3 Análisis comparativo de costos.....	123
7.3.1 Herramientas de visualización .....	124
7.4 Evaluación de mejor opción de sistema .....	125
7.4.1 Costo de adquisición .....	127
7.4.2 Funcionalidad.....	128
7.4.3 Escalabilidad .....	128
7.4.4 Resultado de la Evaluación .....	129
Conclusiones .....	130
Recomendaciones.....	131
Referencias bibliográficas.....	132
Apéndices.....	136
Anexos.....	150

## ÍNDICE DE FIGURAS

<i>Figura 1.1 Descripción del proceso productivo del Centro de Investigaciones en Vivienda y Construcción. Fuente: Elaboración propia.</i> .....	3
<i>Figura 2.1 Vista interna del cuarto húmedo en el Centro de Investigaciones en Vivienda y Construcción (CIVCO), mostrando probetas de concreto y aspersores en funcionamiento.</i> .....	12
<i>Figura 4.1 Diagrama de flujo del proceso de monitoreo en tiempo real. Fuente: Elaboración propia.</i> .....	33
<i>Figura 4.2 Placa de Arduino Uno R3. Arduino (2023).</i> .....	35
<i>Figura 4.3 Sensor de Temperatura y Humedad DHT22. Aosong Electronics Co.,Ltd.</i> .....	40
<i>Figura 4.4 Pantalla Liquid Crystal 1602 para visualización de datos. MicroJPM (2023.)</i> .....	45
<i>Figura 4.5 Módulo I2C FC-113 para Comunicación entre LCD y Arduino. MicroJPM (2023).</i> .....	47
<i>Figura 4.6 Jumpers Macho-Macho de 20 cm para Conexiones Seguras entre Componentes. MicroJPM (2023).</i> .....	48
<i>Figura 4.7 Breadboard mini modular. MicroJPM (2023).</i> .....	48
<i>Figura 4.8 Cable alarma de 4 hilos. MicroJPM (2023).</i> .....	49
<i>Figura 4.9 Adaptador de 9VDC. MicroJPM (2023).</i> .....	50
<i>Figura 4.10 Esquema de conexión para prototipo. Fuente: Elaboración propia.</i> .....	53
<i>Figura 4.11 Bibliotecas usadas para programa en Arduino. Fuente: Elaboración propia.</i> .....	54
<i>Figura 4.12 Definición del sensor en el programa de Arduino. Fuente: Elaboración propia</i> .....	54
<i>Figura 4.13 Asignación a sensor y pantalla. Fuente: Elaboración propia.</i> .....	55
<i>Figura 4.14 Void Setup del programa de Arduino. Elaboración propia</i> .....	55
<i>Figura 4.15 Voidloop del programa de Arduino. Fuente: Elaboración propia</i> .....	56
<i>Figura 4.16 Tiempo de espera del programa. Fuente: Elaboración propia.</i> .....	57
<i>Figura 4.17 Vista interna del prototipo, mostrando el esquema de conexión implementado y los componentes ensamblados. Fuente: Elaboración propia. Fuente: Elaboración propia</i> .....	58
<i>Figura 4.18 Vista frontal del prototipo con la pantalla LCD 1602 en funcionamiento, exhibiendo datos en tiempo real de temperatura y humedad. Fuente: Elaboración propia.</i> .....	59
<i>Figura 5.1 Placa de Ethernet Shield. MicroJPM (2023).</i> .....	63

<i>Figura 5.2 Visualización del montaje integrado de Arduino Uno y Ethernet Shield W5100. Fuente: Elaboración propia.....</i>	<i>65</i>
<i>Figura 5.3 Detalles del canal de ThingSpeak. Fuente: Elaboración propia.....</i>	<i>69</i>
<i>Figura 5.4 Configuración de Parámetros del canal en ThingSpeak. Fuente: Elaboración propia.....</i>	<i>70</i>
<i>Figura 5.5 Código de configuración del ethernet shield. Fuente: Elaboración propia. ....</i>	<i>71</i>
<i>Figura 5.6 Código de principio de configuración del ethernet shield. Fuente: Elaboración propia. ....</i>	<i>72</i>
<i>Figura 5.7 Voidloop del código modificado para conexión con ThingSpeak. Fuente: Elaboración propia.....</i>	<i>73</i>
<i>Figura 5.8 Identidad del remitente de Sendgrip. Fuente: Elaboración propia.....</i>	<i>79</i>
<i>Figura 5.9 Conexión con el canal de ThingSpeak. Fuente: Elaboración propia. ....</i>	<i>80</i>
<i>Figura 5.10 Conversión a zona horaria local. Fuente: Elaboración propia. ....</i>	<i>80</i>
<i>Figura 5.11 Creación de reporte de archivo de texto y dataframe. Fuente: Elaboración propia. ....</i>	<i>81</i>
<i>Figura 5.12 Creación de archivo Excel desde un dataframe. Fuente: Elaboración propia.....</i>	<i>81</i>
<i>Figura 5.13 Código de fragmento del envío de correo electrónico. Fuente: Elaboración propia. ....</i>	<i>82</i>
<i>Figura 5.14 Acceso a datos del canal de ThingSpeak. Fuente: Elaboración propia. ....</i>	<i>83</i>
<i>Figura 5.15 Fragmento de límites permitidos para las alertas. Fuente: Elaboración propia. ....</i>	<i>83</i>
<i>Figura 5.16 Comprobación de rangos permitidos. Fuente: Elaboración propia. ....</i>	<i>84</i>
<i>Figura 5.17 Envío de alertas por correo electrónico mediante SendGrip. Fuente: Elaboración propia. ....</i>	<i>85</i>
<i>Figura 5.18 Fragmento de código de alerta por desconexión en el sistema de monitoreo. Fuente: Elaboración propia.....</i>	<i>86</i>
<i>Figura 5.19 Subida y Configuración del Código en Google Cloud Functions. Fuente: Elaboración propia.....</i>	<i>88</i>
<i>Figura 5.20 Configuración del trabajo cron en Cloud Scheduler. Fuente: Elaboración propia.....</i>	<i>90</i>
<i>Figura 5.21 Sistema en operación, capturando y transmitiendo datos. Fuente: Elaboración propia. ....</i>	<i>91</i>
<i>Figura 5.22 Dashboard de ThingSpeak mostrando datos recibidos del sistema. Fuente: Elaboración propia. ....</i>	<i>92</i>
<i>Figura 6.1 Sistema en funcionamiento dentro de las instalaciones del Centro de Investigaciones en Vivienda y Construcción (CIVCO). Fuente: Elaboración propia .....</i>	<i>94</i>
<i>Figura 6.2 Planta de emergencia garantizando suministro eléctrico al edificio y circuitos específicos. Fuente: Elaboración propia. ....</i>	<i>95</i>

<i>Figura 6.3 Diagrama unifilar de distribución eléctrica del edificio con detalles de alimentación y capacidad. Fuente: Elaboración propia. ....</i>	<i>97</i>
<i>Figura 6.4 Registro de temperatura en el Centro de Investigaciones en Vivienda y Construcción (CIVCO) través de ThingSpeak. Fuente: Elaboración propia. ....</i>	<i>99</i>
<i>Figura 6.5 Registro de humedad relativa en el Centro de Investigaciones en Vivienda y Construcción (CIVCO) a través de ThingSpeak. Fuente: Elaboración propia. ....</i>	<i>100</i>
<i>Figura 6.6 Notificación por correo ante la falta de transmisión de datos del sistema. ....</i>	<i>101</i>
<i>Figura 7.1 Sistema Vaisala modelo HMT120. Vaisala (2023).....</i>	<i>113</i>
<i>Figura 7.2 Sistema Vaisala modelo Indigo202. (Vaisala 2023).....</i>	<i>116</i>
<i>Figura 7.3 Comparativa de inversión inicial en sistemas de monitoreo.....</i>	<i>125</i>

## ÍNDICE DE TABLAS

<i>Tabla 1.3 Resumen de Metodología para el Objetivo 3. Elaboración propia. ....</i>	<i>28</i>
<i>Tabla 2.4 Resumen de Metodología para el Objetivo 4. Elaboración propia. ....</i>	<i>30</i>
<i>Tabla 4.1 Cuadro comparativo de ventajas y desventajas de las opciones de microcontroladores.....</i>	<i>37</i>
<i>Tabla 4.2 Características del sensor DHT-22.....</i>	<i>41</i>
<i>Tabla 4.3 Cuadro comparativo de ventajas y desventajas de las opciones de sensores.....</i>	<i>43</i>
<i>Tabla 4.4 Cuadro comparativo de ventajas y desventajas de las opciones de pantalla. ....</i>	<i>46</i>
<i>Tabla 5.1 Cuadro comparativo de ventajas y desventajas de las opciones de plataforma de almacenamiento.....</i>	<i>68</i>
<i>Tabla 5.2 Cuadro comparativo de ventajas y desventajas de las opciones de plataforma de notificación. ....</i>	<i>77</i>
<i>Tabla 6..1 Resumen estadístico diario de temperatura y humedad con mínimo, máximo y promedio. ....</i>	<i>105</i>
<i>Tabla 7.1 Características del modelo Vaisala HMT120. ....</i>	<i>114</i>
<i>Tabla 7.7 Comparativo de ventajas y desventajas de los requisitos mínimos para las opciones de sistemas de monitoreo.....</i>	<i>127</i>



## **Capítulo 1: Introducción**

Este proyecto abordó la necesidad de monitorear y gestionar datos de humedad relativa y temperatura en la cámara húmeda del Centro de Investigaciones en Vivienda y Construcción (CIVCO) del Instituto Tecnológico de Costa Rica. Dicha cámara es de suma relevancia para realizar investigaciones y pruebas acerca de la resistencia y durabilidad de materiales de construcción (TEC, 2023). El objetivo principal fue mejorar la trazabilidad de los resultados de estas pruebas, asegurando el cumplimiento con la norma ISO/IEC 17025 para la acreditación del laboratorio de ensayos.

Se planteó un sistema de monitoreo en tiempo real para el ambiente de la cámara húmeda que sea en tiempo real y capaz de almacenar datos en la nube. El sistema al estar conectado a la nube mantiene un registro histórico del monitoreo y envía reportes del mismo monitoreo histórico. Proporcionando así trazabilidad de las condiciones ambientales durante el proceso de curado de las probetas de concreto realizado en la sala.

El presente informe escrito detalla y explica todo el procedimiento, análisis y desarrollo elaborado para el cumplimiento de los objetivos del proyecto con el fin de dar una solución al problema actual que tiene el Centro de Investigaciones en Vivienda y Construcción (CIVCO) con sistema de monitoreo para el cuarto húmedo y el deseo del centro de obtener la acreditación respectiva para su laboratorio de ensayos.

### **1.1 Reseña de la empresa**

El Centro de Investigaciones en Vivienda y Construcción (CIVCO) es una entidad adscrita a la Escuela de Ingeniería en Construcción del Instituto Tecnológico de Costa Rica. Situado en Cartago, fue fundado en 1991 con la misión de llevar a cabo investigaciones especializadas y brindar servicios en el campo de la construcción a nivel tanto nacional como internacional.

Desde sus inicios, el Centro de Investigaciones en Vivienda y Construcción (CIVCO) ha destacado por su compromiso con la excelencia y la innovación. Su enfoque multidisciplinario le ha permitido

desarrollar soluciones en áreas tan diversas como materiales, electrónica y forestal. Además de sus labores de investigación, el centro ofrece servicios especializados que abarcan ingeniería, inspección, evaluación de estructuras, estudio de suelos, planificación urbana y administración de la construcción.

El Centro de Investigaciones en Vivienda y Construcción (CIVCO) se organiza en cuatro líneas de investigación principales: abastecimiento de agua, monitoreo de estructuras e infraestructura inteligente, modelación hidrológica e hidráulica, y ambiente construido. A través de estas áreas, el centro busca contribuir al desarrollo sostenible y resiliente de la infraestructura en Costa Rica y la región.

A nivel financiero, el Centro de Investigaciones en Vivienda y Construcción (CIVCO) cuenta con la colaboración y el apoyo de diversos sectores, tanto públicos como privados. Además, ofrece consultorías y asistencia técnica a través del Laboratorio de Construcción (LabCIVCO) y otros programas académicos asociados al Tecnológico de Costa Rica.

Como parte de su compromiso con la educación y la formación continua, el Centro de Investigaciones en Vivienda y Construcción (CIVCO) también ha establecido un programa de extensión en planificación urbana, fortaleciendo su rol como referente en temas relacionados con la construcción y el desarrollo urbano en el país.

### 1.1.1 Descripción del proceso productivo

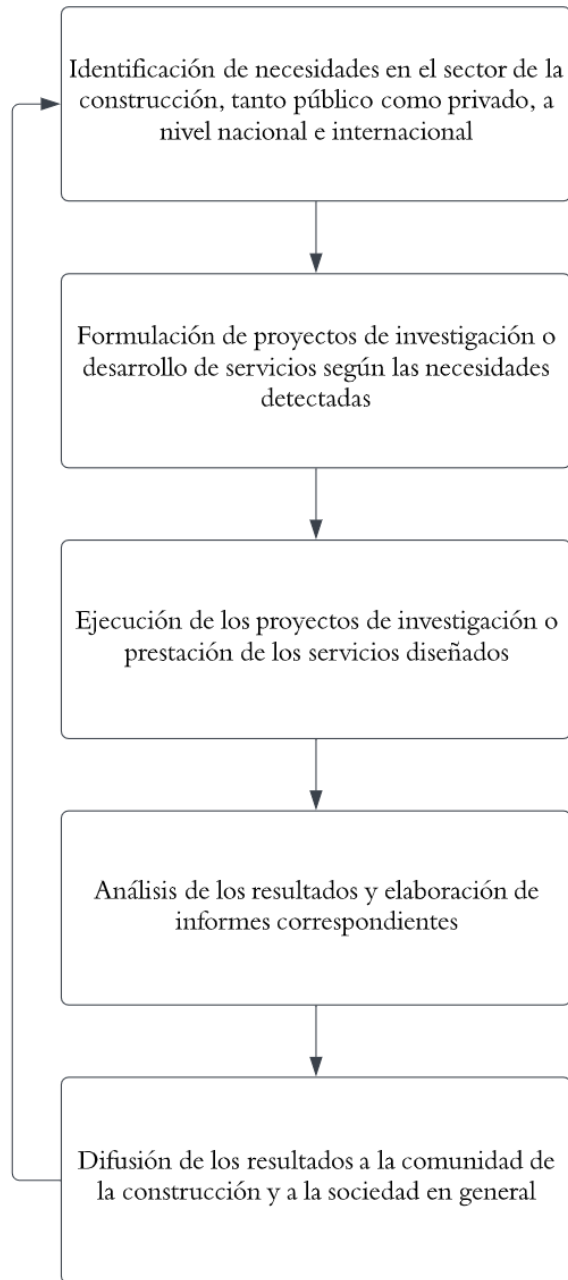


Figura 1.1 Descripción del proceso productivo del Centro de Investigaciones en Vivienda y Construcción. Fuente: Elaboración propia.

## **1.2 Antecedentes**

### **1.2.1 Antecedentes teóricos**

El Centro de Investigaciones en Vivienda y Construcción (CIVCO) del Instituto Tecnológico de Costa Rica dispone de una cámara húmeda utilizada para llevar a cabo investigaciones y pruebas relacionadas con la resistencia y durabilidad de los materiales de construcción bajo condiciones ambientales controladas (TEC, 2023). Esta cámara es un espacio cerrado en el que se monitorean y controlan las condiciones ambientales, manteniendo el ambiente del recinto según las condiciones requeridas (ASTM International, 2013).

El Centro de Investigaciones en Vivienda y Construcción (CIVCO) ofrece una variedad de servicios de investigación y pruebas, incluyendo pruebas de curado de concreto en su laboratorio para evaluar la resistencia y durabilidad de diferentes mezclas de concreto (TEC, 2023). Hay varias normativas aplicables a las pruebas de curado de concreto en cámaras húmedas, incluyendo las normas ASTM C192, ASTM C511 y ASTM C31 de Estados Unidos (ASTM International, 2008).

La ASTM C192 establece prácticas estándar para la preparación y curado de probetas de concreto en laboratorios (ASTM International, 2020), siendo fundamental para garantizar una preparación y curado adecuados del concreto, factores que influyen directamente en la calidad y resistencia del material evaluado.

Por otro lado, la ASTM C31 define prácticas estándar para la preparación y curado de probetas de concreto en campo (ASTM International, 2013). Estos procedimientos son vitales para asegurar la correcta manipulación y tratamiento de las muestras de concreto antes de su almacenamiento y análisis en laboratorio. Aunque la preparación de las muestras se realice en campo según la ASTM C31, los parámetros requeridos en la cámara húmeda para su curado son los mismos que los especificados en la ASTM C192. Ambas normas indican que las probetas deben estar en un ambiente controlado antes de someterse a pruebas de resistencia a la compresión. Además, especifican los requisitos que debe cumplir una cámara húmeda para un proceso

de curado adecuado, incluyendo la necesidad de contar con un instrumento para monitorear la temperatura dentro del recinto (ASTM International, 2013; ASTM International, 2020).

La ASTM C511 especifica las condiciones estándar para los cuartos de mezcla, cámaras húmedas y tanques de almacenamiento de agua utilizados en ensayos de cementos hidráulicos y concretos. Estas especificaciones son las necesarias para garantizar condiciones adecuadas y controladas durante las pruebas y almacenamiento de probetas de cemento o concreto (ASTM International, 2013).

Para obtener acreditación a nivel nacional, el Centro de Investigaciones en Vivienda y Construcción (CIVCO) debe solicitar una evaluación y ser evaluado por el Ente Costarricense de Acreditación (ECA), organismo encargado de acreditar laboratorios de ensayo y calibración en el país (Ente Costarricense de Acreditación, n.d.). Para que la cámara húmeda y el laboratorio de ensayos en general sean fiables y proporcionen resultados precisos, deben cumplir con lo establecido en la norma ISO/IEC 17025:2017 (Rao, 2021). El Ente Costarricense de Acreditación utiliza esta norma para evaluar el laboratorio y determinar su acreditación. Los requerimientos de la ISO/IEC 17025:2017 varían según el proceso.

#### **1.4.2 Antecedentes de Campo**

A nivel nacional, el Ente Costarricense de Acreditación (ECA) acredita los laboratorios de ensayo y calibración. La cámara húmeda en el Centro de Investigaciones en Vivienda y Construcción (CIVCO) tiene como objetivo cumplir con el proceso de curado según lo indicado por la norma ASTM C511. Todos los laboratorios en el país que siguen este proceso de curado de concreto deben adherirse a la norma ISO/IEC 17025 para obtener la acreditación

Estudios internacionales destacan el uso de equipos de bajo costo para monitorear condiciones ambientales. Hardware como Raspberry Pi, Arduino Uno y Arduino Mega pueden monitorear en tiempo real la temperatura y humedad relativa utilizando sensores conectados a los dispositivos (Danladi & Baykara, 2022). La eficacia de estos equipos para monitorear humedad y temperatura se ha demostrado en aplicaciones

como invernaderos (Salleh et al., 2013), cuartos de servidores (Nasution et al., 2019) y centros de datos (Polonelli et al., 2018).

La norma ISO/IEC 17025 y su acreditación correspondiente indican que debe haber una confiabilidad normada en los datos extraídos y en la recolección de datos con un registro histórico (ISO/IEC 17025, 2017). Actualmente, los datos proporcionados por el sistema de monitoreo actual de la cámara húmeda tienen un registro histórico poco fiable; este equipo indica la temperatura y humedad relativa en un momento específico dentro de la cámara. Por lo tanto, el registro histórico se realizaba de manera semi manual, y para obtener un histórico, se realizaban varias mediciones y se creaba un promedio, lo que afecta la confiabilidad directa del registro histórico necesario para la cámara húmeda. Además, existen dudas sobre la precisión de los datos de humedad relativa. Poner en riesgo la acreditación disminuye la confianza del laboratorio de ensayos en el mercado, lo que a su vez desalinea el laboratorio con la misión y visión del centro y de la universidad (TEC, n.d.).

En la actualidad existen soluciones de bajo costo que permiten la captura continua de datos (Olaniyi et al., 2021) y la monitorización constante posibilita tomar acciones inmediatas para corregir desviaciones en el proceso, reduciendo así el riesgo de errores y aumentando la eficiencia en el control de calidad (Rao, 2021).

### **1.3 Descripción del problema**

La situación actual del laboratorio presenta una deficiencia principal identificada en el proyecto es el proceso encontrado de toma de datos de temperatura y humedad relativa en el sistema de monitoreo de la cámara húmeda del Centro de Investigaciones en Vivienda y Construcción (CIVCO), que se realiza de manera semi-manual. El Ente Costarricense de Acreditación evalúa el laboratorio basándose en la norma ISO/IEC 17025, examinando aspectos relacionados con la trazabilidad de resultados y la confiabilidad de las

mediciones (Nurcahyo et al., 2018). Sin embargo, la trazabilidad pasada se encontraba en un estado semi manual y con datos desactualizados, lo que producía resultados con poca fiabilidad.

Debido a este problema surgió la idea de optimizar el sistema de control de humedad relativa y temperatura en la cámara húmeda del Centro de Investigaciones en Vivienda y Construcción (CIVCO). Este esfuerzo buscó asegurar el cumplimiento de las normativas ASTM C511 e ISO/IEC 17025, elementos fundamentales para obtener la acreditación correspondiente.

La optimización se orientó hacia la implementación de un sistema de control ambiental en tiempo real. Este sistema no solo favoreció el registro histórico de datos en la nube, sino que también facilitó la captura de datos con la frecuencia adecuada. De esta forma, se garantizó no solo la precisión en los resultados obtenidos, sino también un seguimiento detallado y continuo del ambiente controlado dentro de la cámara (ASTM International, 2013).

Hay que destacar que, aunque las normas de acreditación no delinear un método específico para el control ambiental durante el curado del concreto, sí enfatizan la necesidad de un registro histórico confiable de datos (ISO/IEC 17025, 2017). Además, la norma ISO/IEC 17025 para laboratorios de ensayo destaca la necesidad de una calibración certificada por un laboratorio certificado con la norma ISO/IEC 17025 para laboratorios de calibración de los instrumentos utilizados.

<b>Debiera</b>	El proceso de curado de probetas de concreto en la habitación húmeda del Centro de Investigaciones en Vivienda y Construcción (CIVCO) del Instituto Tecnológico de Costa Rica debe cumplir con los parámetros establecidos por la norma ASTM C511 y para la acreditación se requiere cumplir con la trazabilidad solicitada por la norma ISO/IEC 17025:2017 evaluada por el Ente Costarricense de Acreditación (ECA) (ASTM International, 2012; ISO/IEC 17025, 2017)	<p>Dato suministrado</p> <p>La norma ISO/ IEC 17025 tiene como requisito clave para laboratorios la trazabilidad de los resultados y confiabilidad de sus mediciones</p>	<p>Referencias bibliográficas</p> <p>ISO/IEC 17025. (2005). General Requirement for the competence of testing and calibration laboratories (2da ed.).</p>
<b>Realidad</b>	El sistema utilizado en la habitación húmeda del Centro de Investigaciones en Vivienda y Construcción para el monitoreo de temperatura y humedad relativa solo suministra la situación actual dentro del recinto en la interfaz. La toma de datos para un registro histórico es tomada de forma semi manual. Además de una falta reportes actualizados del monitoreo dentro de la cámara húmeda.	<p>Dato suministrado</p> <p>El sistema actual de la habitación de humedad del CIVCO cuenta con una trazabilidad insuficiente según el encargado del laboratorio del Centro de Investigaciones en Vivienda y Construcción.</p>	<p>Referencias bibliográficas</p> <p>(R. Cuevas, comunicación personal, marzo de 2023)</p>
<b>Desviación</b>	El proceso actual para la toma de datos de temperatura y humedad relativa con el sistema de monitoreo de la habitación húmeda del Centro de Investigaciones en Vivienda y Construcción es un procedimiento semi manual y no ofrece reportes actualizados a la fecha del monitoreo. El ente costarricense de acreditación encargado de evaluar el laboratorio con la norma ISO/ IEC 17025 evalúa la habitación húmeda aspectos como trazabilidad de resultados en sus mediciones (Nurcahyo et al., 2018)		



## 1.4 Justificación

La justificación de este proyecto radicó en diversas necesidades y objetivos, tanto técnicos como relacionados con los beneficiarios directos e indirectos. La implementación de un sistema de monitoreo de temperatura y humedad en la cámara húmeda del Centro de Investigaciones en Vivienda y Construcción (CIVCO) fue para garantizar el cumplimiento de los estándares internacionales en ensayos de concreto, especialmente en la medición de variables ambientales como la temperatura y la humedad (ASTM International, 2012; ISO/IEC 17025, 2017). Estas condiciones afectan la confiabilidad de los resultados de las pruebas, siendo críticas para el desempeño de los materiales de construcción y su comportamiento en condiciones reales de uso (De et al., 2021).

La acreditación, por lo tanto, se presenta como un elemento de verdadera relevancia. No es indispensable únicamente para el laboratorio de investigación; es importante también para el Instituto Tecnológico de Costa Rica en su conjunto. Este requisito de acreditación se alinea perfectamente con la misión y visión tanto del Instituto Tecnológico de Costa Rica como del Centro de Investigaciones en Vivienda y Construcción (CIVCO). El Instituto Tecnológico de Costa Rica se ha comprometido a formar líderes que contribuyan al desarrollo sostenible del país a través de la creación, transmisión, y aplicación del conocimiento y la innovación en diversos campos, incluyendo ingeniería, tecnología, ciencias y humanidades. Su visión es ser reconocido mundialmente por su excelencia académica, innovación y liderazgo en la formación de profesionales y en la generación de conocimiento (TEC, n.d.).

Por su parte, el Centro de Investigaciones en Vivienda y Construcción (CIVCO) busca generar, transmitir y aplicar conocimientos y tecnologías que mejoren la calidad de vida de la población a través de la investigación y desarrollo en vivienda y construcción, con la visión de ser reconocido por su excelencia, liderazgo y contribución al desarrollo sostenible del país en estos campos (TEC, 2023).

Es por esto por lo que entonces la gran necesidad de que el laboratorio de ensayos esté acreditado por el Ente Costarricense de Acreditación y cumpla con la norma ASTM C511. Al lograrlo, el Centro de

Investigaciones en Vivienda y Construcción (CIVCO) no solo demuestra que cumple con los altos estándares de calidad en sus investigaciones y desarrollos en vivienda y construcción, sino que también respalda y fortalece la misión y visión del Instituto Tecnológico de Costa Rica, promoviendo la formación de profesionales altamente capacitados y fomentando la investigación avanzada en el campo.

Los beneficiarios directos del proyecto incluyen el personal del laboratorio de la institución, estudiantes del Instituto Tecnológico de Costa Rica, profesionales involucrados en proyectos de investigación y desarrollo, y clientes que requieren ensayos de concreto confiables para la ejecución de proyectos y toma de decisiones (TEC, n.d.). El sistema de monitoreo propuesto generó información valiosa, facilitando la identificación de tendencias y patrones que puedan influir en los resultados de las pruebas y en el comportamiento de los materiales almacenados (Olaniyi et al., 2021). Además, proporciona información en tiempo real para detectar y solucionar problemas de manera oportuna, asegurando que las condiciones se mantengan dentro de los rangos establecidos por las normativas aplicables y evitando posibles pérdidas o deterioro de muestras y ensayos (Danladi & Baykara, 2022; Tang et al., 2020).

Investigaciones y desarrollos anteriores han demostrado la factibilidad de estos sistemas en contextos similares, proporcionando un marco conceptual y tecnológico adecuado para la adaptación y aplicación de las soluciones propuestas en el ámbito de las cámaras húmedas de ensayos de concreto (Olaniyi et al., 2021).

## **1.5 Objetivos del proyecto**

### **1.5.1 Objetivo General**

Desarrollar un prototipo de sistema de monitoreo y registro de la humedad relativa y temperatura para el cuarto húmedo del Centro de Investigaciones en Vivienda y Construcción (CIVCO), que cumpla con los requisitos de la norma ISO/ IEC 17025 para la acreditación del laboratorio de ensayos.

### **1.5.2 Objetivos Específicos**

1. Elaborar un prototipo de sistema de monitoreo de temperatura y humedad basado en tecnología en tiempo real con Arduino, optimizando la selección de componentes para una relación costo-beneficio eficiente. Indicador de logro: Prototipo armado y en funcionamiento.
2. Integrar el sistema de monitoreo con la nube para almacenamiento de datos, generación de gráficos de monitoreo histórico y alertas en caso de fallas en el sistema para la trazabilidad solicitada por la norma ISO/ IEC 17025. Indicador de logro: Integración del sistema con la nube y reporte de generación de gráficos de monitoreo histórico y alertas.
3. Evaluar el prototipo de monitoreo en la cámara húmeda del Centro de Investigaciones en Vivienda y Construcción (CIVCO) mediante pruebas que validen la capacidad del sistema para la tarea de monitoreos y reportes de manera confiable los parámetros de temperatura y humedad en la nube. Indicador de logro: Reporte de pruebas con datos recopilados en tiempo real y almacenados en la nube que demuestren la exactitud y precisión del prototipo en la medición y reporte de la temperatura y humedad relativa dentro de un recinto.
4. Comparar el coste del prototipo desarrollado con soluciones comerciales existentes, considerando únicamente los costos adquisitivos de inversión. Indicador de logro: Informe comparativo que incluya un análisis de costos iniciales.

## Capítulo 2: Marco Teórico

### 2.1 Cuartos Húmedos

Los cuartos húmedos, también conocidos como cámaras húmedas, son recintos especializados diseñados para mantener condiciones ambientales controladas, con particular énfasis en la temperatura y la humedad. Estas condiciones son vitales para el curado y almacenamiento de muestras de materiales de construcción, tales como concreto (De et al., 2021). Su importancia radica no solo en el desarrollo y evaluación de nuevos materiales de construcción, sino también en el análisis de la resistencia y durabilidad de materiales ya existentes (De et al., 2021).



Figura 2.1 Vista interna del cuarto húmedo en el Centro de Investigaciones en Vivienda y Construcción (CIVCO), mostrando probetas de concreto y aspersores en funcionamiento.

Más allá del concreto, estas cámaras tienen aplicaciones en el estudio y conservación de otros materiales, como morteros, yesos, ladrillos, e incluso materiales biológicos y piezas arqueológicas (Groot et al., 2022). A menudo se emplean en investigaciones para comprender el impacto de distintos aditivos en el

concreto, como fibras y materiales reciclados, y cómo estos influyen en su desempeño general (Neville, 2011).

Para que un cuarto húmedo cumpla con su propósito, debe estar equipado con sistemas adecuados que garanticen la mantención de las condiciones ambientales ideales. La norma ASTM C511 estipula que la temperatura para el curado del concreto en estas cámaras debe oscilar entre 20 y 25 °C, y la humedad relativa debe ser al menos del 95% (ASTM International, 2013).

Sin embargo, simplemente establecer estas condiciones no es suficiente. Es fundamental contar con un sistema de monitoreo y registro que rastree continuamente las condiciones dentro del cuarto. Utilizar dispositivos precisos, como sensores de temperatura y humedad, es acerca de la exactitud y la repetibilidad de los experimentos y pruebas llevadas a cabo en estos espacios (Elsalamawy, Mohamed, & Kamal, 2019).

## **2.2 Sistemas de monitoreo de humedad y temperatura**

El monitoreo preciso de la humedad y la temperatura es necesario en la gestión de cuartos húmedos, especialmente en contextos críticos como los laboratorios de investigación y calibración.

Los sistemas de monitoreo en tiempo real han demostrado ser particularmente eficientes en esta tarea. Mediante el uso de sensores especializados, es posible recopilar datos continuamente y garantizar que las condiciones ambientales se mantengan dentro de los rangos requeridos. Estas soluciones, respaldadas por redes inalámbricas como ZigBee o LoRaWAN, transmiten la información recopilada a servidores centralizados, permitiendo la supervisión remota a través de plataformas web y aplicaciones móviles. Esto no solo posibilita el acceso a datos en tiempo real, sino también la recepción de alertas instantáneas en caso de desviaciones fuera de los parámetros establecidos (Olaniyi et al., 2021; Augustin et al., 2016; Yanshori, Nugraha, & Santi, 2022).

Sin embargo, no todas las soluciones de monitoreo dependen de la conectividad inalámbrica. Los sistemas alámbricos, como los basados en Ethernet Shield, ofrecen una alternativa de menor costo para la

transmisión de datos entre los sensores y las plataformas de monitoreo. Aunque estos sistemas pueden presentar limitaciones en cuanto a la flexibilidad y la distancia de transmisión en comparación con las redes inalámbricas, son especialmente útiles en entornos donde la infraestructura ya está establecida o cuando se busca una solución económica y robusta. La ventaja de soluciones como Arduino Shield reside en su accesibilidad y en la amplia comunidad de desarrolladores, lo que facilita la adaptación y personalización del sistema según las necesidades específicas del cuarto húmedo.

Una innovación reciente en el campo del monitoreo de humedad y temperatura es la integración de inteligencia artificial (IA) y aprendizaje automático (ML). Estas tecnologías ofrecen la posibilidad de analizar grandes conjuntos de datos, identificar patrones y predecir tendencias futuras, potenciando así la eficiencia y precisión de los sistemas de monitoreo (Qavidel Fard, Zomorodian, & Korsavi, 2022).

### **2.3 Microcontrolador Arduino**

La plataforma Arduino emerge como una solución de hardware y software de código abierto, fundamentada en microcontroladores Atmel AVR. Esta ha ganado relevancia en el ámbito de proyectos basados en tecnología en tiempo real (Tang et al., 2020). Dentro de su oferta, Arduino proporciona una variedad de placas de desarrollo, cada una diseñada para satisfacer diferentes necesidades. Algunas de las más destacadas incluyen el Arduino Uno, Arduino Mega y Arduino Nano (Arduino, 2021).

Uno de los atributos más valorados de Arduino es su accesibilidad. Su diseño intuitivo lo convierte en una opción privilegiada tanto para principiantes como para entusiastas en campos como la electrónica y la programación. El entorno de desarrollo integrado (IDE) que ofrece es amigable y compatible con una diversidad de sistemas operativos: desde Windows y macOS hasta Linux (Arduino Team, n.d.). Su estructura programática radica en un lenguaje basado en C/C++, lo cual es ventajoso para aquellos desarrolladores familiarizados con estos lenguajes.

La versatilidad de Arduino lo ha posicionado en la vanguardia de diversos campos de aplicación, tales como el monitoreo ambiental, soluciones domóticas y la agricultura de precisión (Danladi & Baykara, 2022).

A pesar de las múltiples ventajas de Arduino, también presenta ciertas limitaciones. Los microcontroladores Atmel AVR, aunque eficientes, poseen velocidades de reloj y recursos de memoria que, en ocasiones, se quedan cortos, especialmente cuando se enfrentan a aplicaciones más exigentes en términos de rendimiento (Danladi & Baykara, 2022).

## **2.4 Sensores de humedad y temperatura de precisión**

Los sensores de humedad y temperatura son componentes encontrados en aplicaciones de la industria, la agricultura, la meteorología y la domótica, entre otros. Estos sensores permiten medir con precisión las condiciones (Liu, 2020). Este marco teórico aborda los diferentes tipos de sensores de humedad y temperatura de precisión, su funcionamiento y los métodos de calibración utilizados para garantizar la exactitud y confiabilidad de las mediciones.

Existen varios tipos de sensores de humedad y temperatura disponibles en el mercado, cada uno con sus propias características y aplicaciones específicas. A continuación, se describen los tipos de sensores más comunes:

Los sensores capacitivos de humedad son uno de los tipos más utilizados debido a su alta precisión, estabilidad y resistencia a la contaminación (Lorek & Majewski, 2018). Estos sensores funcionan mediante la medición de la capacitancia de un material dieléctrico, cuya constante dieléctrica varía con la humedad relativa del ambiente. Por lo general, el material dieléctrico se encuentra entre dos placas de un capacitor, y la variación de la capacitancia se convierte en una señal eléctrica que se correlaciona con la humedad relativa (Lorek & Majewski, 2018).

Los sensores resistivos de humedad miden la variación de la resistencia eléctrica de un material higroscópico en función de la humedad (Regtien, 2011). Estos sensores suelen ser más económicos que los capacitivos, pero tienen una menor precisión y estabilidad a largo plazo.

Existen varios tipos de sensores de temperatura, como los termistores, los termopares y los sensores de temperatura de resistencia (RTD, por sus siglas en inglés). Los termistores son dispositivos semiconductores cuya resistencia eléctrica varía con la temperatura de manera no lineal (Lorek & Majewski, 2018). Los termopares, por otro lado, están compuestos por dos metales diferentes unidos en un extremo y producen una diferencia de voltaje proporcional a la diferencia de temperatura entre los extremos (Lorek & Majewski, 2018). Los RTD son dispositivos que emplean materiales de alta pureza, como el platino, cuya resistencia varía linealmente con la temperatura.

El funcionamiento de los sensores capacitivos de humedad se basa en la variación de la capacitancia de un material dieléctrico en función de la humedad relativa (Regtien, 2011). Cuando la humedad del ambiente aumenta, las moléculas de agua se adsorben en la superficie del material dieléctrico, lo que provoca un cambio en su constante dieléctrica y, por lo tanto, en la capacitancia del capacitor. Esta variación de capacitancia se mide mediante un circuito oscilador, que genera una señal de frecuencia proporcional a la capacitancia del sensor (Liu, 2020). Luego, la señal de frecuencia se convierte en una señal eléctrica que puede ser procesada y calibrada para obtener la humedad relativa.

El principio de funcionamiento de los sensores resistivos de humedad se basa en la variación de la resistencia eléctrica de un material higroscópico en función de la humedad relativa (Regtien, 2011). Cuando la humedad del ambiente aumenta, las moléculas de agua se adsorben en la superficie del material higroscópico, lo que provoca un cambio en su resistencia eléctrica.

La calibración de sensores de humedad y temperatura es esencial para garantizar la exactitud y confiabilidad de las mediciones (Lorek & Majewski, 2018). Los métodos de calibración varían según el tipo de sensor y las condiciones de aplicación, pero generalmente estos dispositivos ya vienen calibrados.



La calibración de sensores capacitivos y resistivos de humedad se realiza generalmente mediante el uso de cámaras de humedad, son soluciones saturadas de sales que generan ambientes de humedad conocidos (Lorek & Majewski, 2018). Durante la calibración, el sensor se expone a diferentes niveles de humedad y se comparan sus mediciones con las del patrón de referencia o el valor teórico esperado en función de la solución saturada utilizada (Lorek & Majewski, 2018). A partir de esta comparación, se pueden determinar las desviaciones y ajustar los parámetros del sensor para corregir las mediciones.

La calibración de sensores de temperatura también varía según el tipo de sensor, pero generalmente se basa en la comparación de las mediciones del sensor con las de un patrón de referencia o un instrumento de medición de alta precisión a diferentes temperaturas. Por ejemplo, en el caso de los termistores y RTD, se pueden utilizar baños de calibración de temperatura o fuentes de resistencia de precisión para generar condiciones de temperatura conocidas (Liu, 2020). Para los termopares, se pueden utilizar patrones de referencia de termopar o simuladores de termopar para generar voltajes de referencia precisos que correspondan a diferentes temperaturas (Liu, 2020).

En todos estos casos, la calibración implica comparar las mediciones del sensor con las del patrón de referencia o el instrumento de medición de alta precisión y ajustar los parámetros del sensor para corregir las mediciones. Además, la calibración periódica de los sensores de humedad y temperatura es esencial para garantizar la exactitud y confiabilidad de las mediciones a lo largo del tiempo, ya que la deriva y el envejecimiento de los sensores pueden afectar su rendimiento (Lorek & Majewski, 2018; Regtien, 2011).

Si bien es cierto que muchos sensores modernos ya vienen calibrados de fábrica, es importante entender el concepto de calibración para valorar la exactitud y confiabilidad de las mediciones (Lorek & Majewski, 2018). En el pasado, la calibración era una práctica habitual; consistía en exponer el sensor a condiciones controladas y comparar sus lecturas con valores de referencia. Actualmente, gracias a avances tecnológicos, muchos sensores, incluido el DHT-22, eliminan la necesidad de calibraciones adicionales, ofreciendo una precisión garantizada desde su adquisición por el fabricante.

Pero para el ente encargado de la acreditación del laboratorio solicita que los equipos estén calibrados por un laboratorio de calibración que esté acreditado por la norma ISO/IEC 17025 para laboratorios de calibración.

## **2.5 Protocolos de comunicación**

La efectiva comunicación entre dispositivos y sistemas en un contexto de tiempo real es fundamental para garantizar un funcionamiento óptimo y eficiente. Esta comunicación se rige mediante protocolos estandarizados que proporcionan reglas y directrices para la correcta transferencia de datos. Entre estos protocolos, el MQTT ha ganado prominencia por su adaptabilidad a entornos con recursos limitados, especialmente cuando se utiliza en conjunción con dispositivos como Arduino.

El proyecto se benefició del uso del Arduino Ethernet Shield, un dispositivo que permite a las placas Arduino conectarse a internet por cable. Dada la infraestructura del Centro de Investigaciones en Vivienda y Construcción (CIVCO), donde ya existe un cable para internet, optar por esta conexión alámbrica no solo simplifica el proceso, sino que también puede garantizar una comunicación más estable y segura, sin las interrupciones o interferencias típicas de las conexiones inalámbricas.

Al integrar el Arduino, equipado con el Ethernet Shield, con el protocolo MQTT, se facilitó una comunicación efectiva con otros sistemas, como Raspberry Pi, o plataformas en la nube. Esta combinación asegura una transmisión de datos rápida, confiable y segura.

## **2.6 Plataformas de almacenamiento y visualización de datos en la nube**

En la actualidad, la nube se ha convertido en un componente relevante en la infraestructura tecnológica. Las plataformas de almacenamiento y visualización de datos en la nube juegan un papel realmente importante en la gestión de aplicaciones en tiempo real, ofreciendo una estructura robusta que

permite una recopilación, procesamiento y análisis de datos eficiente y escalable, provenientes de una variedad de sensores y dispositivos (MathWorks, 2023; Ubidots, 2023).

ThingSpeak, desarrollada por MathWorks, se destaca por su capacidad de almacenar y procesar datos en tiempo real. La plataforma es compatible con protocolos estándar de comunicación, como HTTP y MQTT. Además, sus herramientas integradas de análisis y visualización facilitan el monitoreo en tiempo real (ThingSpeak, 2023). Su integración con MATLAB potencia aún más su capacidad analítica (Kumar, Kumar, Singh, & Choudhary, 2021).

Además de las soluciones en la nube, los servidores locales gestionados con lenguajes de programación como Python también desempeñan un papel vital en el ecosistema del IoT. Python, conocido por su versatilidad y amplia biblioteca, es una elección común para gestionar servidores locales, especialmente cuando se interconecta con hardware como Arduino.

Las plataformas mencionadas son para proyectos que buscan aprovechar las ventajas de la nube y garantizar una gestión efectiva de datos en tiempo real, especialmente al trabajar con dispositivos como Arduino y Raspberry Pi.

## **2.7 Análisis de costos en la implementación de soluciones tecnológicas**

La implementación de soluciones tecnológicas, especialmente en el ámbito del Internet de las Cosas (IoT) y la computación en la nube, conlleva una serie de gastos que no deben ser pasados por alto. Estos gastos varían en función de múltiples factores.

El hardware representa uno de los principales puntos por considerar. Se hace referencia a dispositivos como Arduino, Raspberry Pi, diversos sensores, actuadores y otros componentes vitales para la recolección y transmisión de datos. Cada decisión tomada en esta área tiene un impacto directo en el presupuesto global.

Aunque existen opciones de libre acceso que pueden parecer económicas inicialmente, hay circunstancias que requieren soluciones de software licenciado o personalizado. Las tarifas de plataformas

en la nube, como AWS IoT, ThingSpeak y Ubidots, pueden fluctuar dependiendo de variables como el volumen de datos procesados o el número de dispositivos interconectados.

El desarrollo y el mantenimiento son aspectos interrelacionados. Desde el diseño y configuración de sistemas y aplicaciones, pasando por el soporte técnico, hasta las inevitables actualizaciones, todos estos elementos tienen un costo asociado. Además, considerar la formación adecuada del equipo para sacar el máximo provecho al sistema.

La seguridad no puede ser dejada de lado. Implementar protocolos y medidas robustas implica un costo, pero las consecuencias de una vulnerabilidad o una brecha pueden resultar devastadoras tanto financieramente como en términos de reputación.

El consumo energético, aunque puede parecer menor, en operaciones prolongadas puede representar un gasto significativo. A medida que el proyecto evoluciona, la escalabilidad se convierte en un factor fundamental, lo que podría implicar gastos adicionales en capacidad, hardware o adaptaciones del software.

## **2.8 Conexión Ethernet**

Ethernet es una familia de tecnologías de redes informáticas, que se utiliza principalmente en redes de área local (LAN), redes de área metropolitana (MAN) y redes de área amplia (WAN). Su nombre proviene del concepto "éter" utilizado en las comunicaciones por radio del siglo XIX y principios del XX, aludiendo a la manera en que las señales se transmiten por el aire o a través de un medio.

La tecnología Ethernet fue desarrollada en los años 70 por Robert Metcalfe, quien estaba trabajando en Xerox PARC (Palo Alto Research Center). Fue introducida al mercado por Xerox, DEC y Intel en 1980. Desde entonces, ha evolucionado en términos de velocidad y métodos de transmisión, pero ha mantenido la misma estructura fundamental.

Aunque inicialmente se usaba una topología de bus, con el tiempo se ha adoptado una topología de estrella debido a su mayor eficiencia y menor propensión a errores.

Ethernet opera en las dos capas más bajas del modelo OSI: la capa física y la capa de enlace de datos. Los estándares Ethernet están definidos por el Institute of Electrical and Electronics Engineers (IEEE) bajo el estándar 802.3.

## **2.9 La nube**

El concepto de "La Nube" o "Cloud Computing" se refiere al suministro de servicios tecnológicos a través de internet, lo que permite a los usuarios acceder a recursos de software y hardware de manera remota, en lugar de tener sistemas físicos o aplicaciones en sus propios dispositivos o infraestructuras locales (Ranger, 2022).

El origen de la nube se remonta a la década de 1960, con la visión de computación en tiempo compartido. Sin embargo, no fue hasta el auge de la internet en los años 90 y 2000 que la computación en nube comenzó a tomar forma.

## **2.10 Llaves API**

Una API, que se traduce del inglés como Interfaz de Programación de Aplicaciones (Application Programming Interface), es un conjunto de definiciones y protocolos que permite a las aplicaciones comunicarse entre sí. Esencialmente, permite que un software pueda utilizar funciones de otro software, sirviendo como un puente entre diferentes sistemas y permitiendo su integración.

Las llaves API, o claves API, son códigos únicos y secretos que se utilizan para identificar la aplicación que está haciendo una solicitud a un servidor o servicio (Rapid, n.d.). Estas llaves actúan como una especie de contraseña, asegurando que solo las aplicaciones autorizadas tengan acceso a ciertos datos o funcionalidades.

## 2.11 Trabajos Cron

El término "Trabajos Cron" se refiere a aquellas tareas programadas que se ejecutan automáticamente en un sistema informático en intervalos regulares de tiempo. El origen del nombre proviene de la palabra "cron", que es una utilidad estándar de los sistemas operativos basados en Unix para programar tareas. En el ámbito del desarrollo y la administración de sistemas, los trabajos Cron son esenciales para automatizar procesos que deben realizarse periódicamente, como copias de seguridad, actualizaciones y análisis de registros, entre otros.

Un trabajo Cron se define mediante una línea en un archivo especial denominado "crontab", que especifica los detalles de cuándo y cómo debe ejecutarse la tarea (Linas L. & Aris B., 2023). Esta línea incluye una combinación de campos que determinan la frecuencia de la tarea (por ejemplo, diaria, semanal o mensual) y el comando que se debe ejecutar.

Es importante destacar que los trabajos Cron deben ser programados con cuidado, ya que una incorrecta configuración puede dar lugar a la ejecución de tareas en momentos no deseados o con una frecuencia mayor a la esperada, lo que podría tener repercusiones negativas en el rendimiento del sistema o en los procesos que dependen de él.

## Capítulo 3: Metodología

### 3.1 Metodología para objetivo específico 1

Objetivo 1: Elaborar un prototipo de sistema de monitoreo de temperatura y humedad basado en tecnología en tiempo real con Arduino, optimizando la selección de componentes para una relación costo-beneficio eficiente.

**Paso 1:** Definición de Parámetros por Medir.

Determinar las variables que se necesitan medir.

**Paso 2:** Selección de Hardware Central.

Elegir el hardware que actuará como el núcleo del sistema.

**Paso 3:** Selección de Sensor.

Elegir un sensor que pueda medir los parámetros deseados dentro de los rangos requeridos.

**Paso 4:** Selección de Pantalla de Visualización

Elegir una pantalla para la visualización en tiempo real de los datos.

**Paso 5:** Diseño del Esquema de Conexión

Establecer un esquema de conexión que integre todos los elementos seleccionados.

**Paso 6:** Desarrollo del Programa

Programar el sistema en Arduino para que realice el monitoreo de los parámetros y los muestre en la pantalla LCD.

**Paso 7:** Ensamblaje y Prueba

Ensamblar los componentes según el diseño del circuito.

Cargar el código en el Arduino.

Verificar que el sistema está en funcionamiento y muestra los datos en tiempo real en la pantalla LCD.

Entregable Final: Prototipo armado y en funcionamiento.

*Tabla 3.1 Resumen de Metodología para el Objetivo 1.*

<b>Objetivo específico planteado</b>	<b>Actividad por realizar</b>	<b>Fuente de información</b>	<b>Análisis de datos con criterios estadísticos</b>	<b>Resultados esperados (Entregables)</b>
Objetivo específico #1: Elaborar un prototipo de sistema de monitoreo de temperatura y humedad basado en tecnología en tiempo real con Arduino, optimizando la selección de componentes para una relación costo-beneficio eficiente.	1. Determinar las variables a medir. 2. Selección de hardware central. 3. Selección de sensor. 4. Selección de pantalla de visualización. 5. Diseño del esquema de conexión. 6. Desarrollo del programa en Arduino. 7. Ensamblaje y prueba del sistema.	1. Documentación técnica de sensores y componentes. 2. Tutoriales y guías de conexión y programación de Arduino.	Análisis descriptivo simple para asegurar la funcionalidad del prototipo	Prototipo armado y en funcionamiento. listo para integración con el sistema de comunicación a la nube.

### 3.2 Metodología para objetivo específico 2

Objetivo específico #2: Integrar el sistema de monitoreo con la nube para almacenamiento de datos, generación de gráficos de monitoreo histórico y alertas en caso de fallas en el sistema para la trazabilidad solicitada por la norma ISO/IEC 17025.

**Paso 1:** Selección de hardware para conexión a internet.

Elegir el hardware que permita la conexión a Internet del sistema de monitoreo.

**Paso 2:** Selección de plataforma de almacenamiento en la nube.

Escoger la plataforma que permita almacenar los datos en la nube y ofrecer una visualización en tiempo real.

**Paso 3:** Modificación del código de Arduino.



Ajustar el código existente para que, además de medir y mostrar los datos, los envíe a la plataforma seleccionada en la nube.

**Paso 4:** Selección de Plataforma de Notificación.

Escoger una plataforma que permita el envío automático de reportes y alertas por correo electrónico.

**Paso 5:** Desarrollo del programa de reportes históricos.

Programar una funcionalidad que permita generar un informe histórico de monitoreo basado en los datos almacenados en la nube y que sea enviado automáticamente por correo electrónico.

**Paso 6:** Desarrollo del programa de alertas.

Programar una funcionalidad que envíe alertas por correo electrónico cuando las mediciones estén fuera del rango permitido según la norma ASTM C511.

**Paso 7:** Ensamblaje final.

Integrar todas las partes y verificar que el sistema funcione según lo esperado, es decir, que almacene datos en la nube, genere informes y envíe alertas.

Entregable Final: Sistema de Monitoreo con Integración a la Nube con funciones de reporte y alertas.

Tabla 3.2 Resumen de Metodología para el Objetivo 2.

Objetivo específico planteado	Actividad por realizar	Fuente de información	Análisis de datos con criterios estadísticos	Resultados esperados (Entregables)
Objetivo específico #2: Integrar el sistema de monitoreo con la nube para almacenamiento de datos, generación de gráficos de monitoreo histórico y alertas en caso de fallas en el sistema para la trazabilidad solicitada por la norma ISO/ IEC 17025.	1. Selección de Hardware para Conexión a Internet. 2. Selección de Plataforma de Almacenamiento en la Nube. 3. Ajustar el código existente para que, además de medir y mostrar los datos, los envíe a la plataforma seleccionada en la nube. 4. Selección de Plataforma de Notificación. 5. Desarrollo del Programa de Reportes Históricos. 6. Desarrollo del Programa de Alertas. 7. Ensamblaje y pruebas.	Especificaciones Técnicas del Hardware. Documentación Oficial de la Plataforma de Almacenamiento en la Nube. Códigos y Tutoriales de Arduino. Reviews y Recomendaciones de Plataformas de Notificación. Documentación de API y SDK para la generación de reportes. Normativa ASTM C511.	No aplica.	Sistema de Monitoreo con Integración a la Nube con funciones de reporte y alertas.

### 3.3 Metodología para objetivo específico 3

Objetivo específico #3: Evaluar el prototipo de monitoreo en la cámara húmeda del Centro de Investigaciones en Vivienda y Construcción (CIVCO) mediante pruebas que validen la capacidad del sistema para la tarea de monitoreo confiable de los parámetros de temperatura y humedad en la nube.

#### **Paso 1:** Implementación y Puesta en Funcionamiento del Prototipo.

Despliegue del prototipo en las instalaciones del Centro de Investigaciones en Vivienda y Construcción (CIVCO).

Reemplazo del sistema previo con el prototipo desarrollado.

Conexión y configuración del hardware necesario para el funcionamiento del prototipo, incluido el sensor dentro del cuarto húmedo.

#### **Paso 2:** Monitoreo y Registro de Datos.

Configuración del sistema para tomar y registrar datos cada cinco minutos.

Verificación del correcto envío, recepción y almacenamiento de los datos en la plataforma en la nube seleccionada.

Desarrollo y utilización de un programa para descargar y recuperar los datos almacenados en la nube.

**Paso 3:** Pruebas de Sistema de Reportes y Alertas.

Activación y monitoreo del sistema de reportes y alertas del prototipo.

Revisión de la consistencia de los reportes generados y las alertas emitidas.

Ajustes necesarios al sistema basados en los resultados de estas pruebas para optimizar la precisión y confiabilidad de los reportes y alertas.

**Paso 4:** Análisis Estadístico y Validación de Confiabilidad.

Recolección de un conjunto de datos emitidos por el prototipo.

Realización de un análisis estadístico de los datos, incluido el cálculo de estadísticas descriptivas

Evaluación de la confiabilidad del sistema a través una prueba de calibración por un laboratorio certificado.

Documentación y presentación de los resultados del análisis, proporcionando evidencia empírica del monitoreo y el resultado de la prueba de calibración.

Indicador de logro: Informe detallado del proceso y resultados de evaluación del prototipo, incluyendo datos descriptivos del monitoreo y el resultado de la prueba de calibración del sistema en laboratorio que respalde la confiabilidad y precisión del sistema en las tareas de monitoreo y reporte de temperatura y humedad.

Tabla 1.3 Resumen de Metodología para el Objetivo 3. Elaboración propia.

Objetivo específico planteado	Actividad por realizar	Fuente de información	Análisis de datos con criterios estadísticos	Resultados esperados (Entregables)
<p>Objetivo específico #3: Evaluar el prototipo de monitoreo en la cámara húmeda del Centro de Investigaciones en Vivienda y Construcción (CIVCO) mediante pruebas que validen la capacidad del sistema para la tarea de monitoreo confiable de los parámetros de temperatura y humedad en la nube.</p>	<p>1. Implementación y puesta de funcionamiento del prototipo. 2. Monitoreo y registro de datos. 3. Prueba de sistemas de reporte y alertas. 4. Análisis estadístico y validación de confiabilidad.</p>	<p>1. Documentación técnica del prototipo y componentes utilizados. 2. Manuales y guías de los servicios de nube y plataformas de reporte y alerta utilizados. 3. Datos recolectados por el prototipo durante el periodo de prueba.</p>	<p>1. Descriptiva: Media, mediana, moda, rango, varianza, desviación estándar de los datos recolectados.</p>	<p>Informe detallado del proceso y resultados de evaluación del prototipo, incluyendo datos descriptivos del monitoreo y el resultado de la prueba de calibración del sistema en laboratorio que respalde la confiabilidad y precisión del sistema en las tareas de monitoreo y reporte de temperatura y humedad.</p>

### 3.4 Metodología para objetivo específico 4

Objetivo específico #4: Comparar el coste del prototipo desarrollado con soluciones comerciales existentes, considerando únicamente los costos adquisitivos de inversión. Indicador de logro: Informe comparativo que incluya un análisis de costos iniciales.

#### **Paso 1:** Identificación de Soluciones Comerciales.

Investigar y listar soluciones comerciales disponibles en el mercado que ofrezcan funcionalidades similares al prototipo desarrollado.

Seleccionar soluciones que sean relevantes y comparables con el prototipo.

**Paso 2:** Recopilación de Datos de Costos.

Recopilar información detallada sobre los costos de adquisición de las soluciones comerciales identificadas.

Calcular el costo total de adquisición del prototipo, sumando el costo de cada componente y herramienta utilizada en el desarrollo.

**Paso 3:** Análisis Comparativo de Costos.

Realizar una comparación directa entre los costos de adquisición del prototipo y cada una de las soluciones comerciales.

Utilizar herramientas gráficas como tablas comparativas o gráficos de barras para visualizar las diferencias de costos.

**Paso 4:** Evaluación Integral de las Opciones.

Analizar la funcionalidad y escalabilidad de cada solución, junto con su coste inicial, para determinar la opción más ventajosa para el Centro de Investigaciones en Vivienda y Construcción.

Incorporar estos criterios en el análisis para brindar una visión más holística que vaya más allá del simple coste adquisitivo.

**Paso 5:** Documentación y Reporte.

Documentar el proceso de comparación y los resultados obtenidos de manera clara y concisa, incluyendo la evaluación de funcionalidad y escalabilidad.

Elaborar un informe comparativo que destaque las diferencias de costos y ofrezca un análisis sobre la viabilidad económica del prototipo frente a las soluciones comerciales.

Entregable Final: Informe comparativo detallado, con análisis y visualizaciones gráficas, que muestre la comparación de costos adquisitivos entre el prototipo desarrollado y las soluciones comerciales existentes seleccionada. Además, un comparativo que incluya un análisis de costos iniciales y una evaluación de funcionalidad y escalabilidad para determinar la mejor opción.

Tabla 2.4 Resumen de Metodología para el Objetivo 4. Elaboración propia.

<b>Objetivo específico planteado</b>	<b>Actividad por realizar</b>	<b>Fuente de información</b>	<b>Análisis de datos con criterios estadísticos</b>	<b>Resultados esperados (Entregables)</b>
Objetivo específico #4: Comparar el coste del prototipo desarrollado con soluciones comerciales existentes, considerando únicamente los costos adquisitivos de inversión.	1. Identificación de Soluciones Comerciales. 2. Recopilación de Datos de Costos. 3. Análisis Comparativo de Costos. 4. Evaluación Integral de las Opciones. 5. Documentación y Reporte.	1. Páginas web oficiales, catálogos y listas de precios de soluciones comerciales. 2. Facturas o recibos de compra de los componentes del prototipo	Descriptivo: Tablas comparativas y/o gráficos de barras que ilustren la comparación de costos entre el prototipo y las soluciones comerciales. Análisis integral que incluya funcionalidad y escalabilidad.	Informe comparativo detallado de los costos adquisitivos con análisis crítico, visualizaciones gráficas y evaluación integral.

## **Capítulo 4: Desarrollo y Selección de Componentes del Prototipo**

El proceso de diseño y elaboración de un prototipo es parte necesaria de la investigación y desarrollo de cualquier solución tecnológica. A través de este, se buscó no solo visualizar una representación física o conceptual de la solución propuesta, sino también verificar su factibilidad, funcionalidad y eficacia en un entorno controlado. En el contexto de este proyecto, el prototipo sirve como una herramienta para monitorear las condiciones ambientales, específicamente la temperatura y humedad, dentro de una cámara húmeda, asegurando el cumplimiento de normas específicas.

La elección de la tecnología y componentes para dicho prototipo requirió un análisis detallado y cuidadoso para garantizar que se seleccionen aquellos componentes que cumplan con los requisitos de precisión, confiabilidad y costo, y que juntos operen de manera óptima. Esta selección se hizo aún más importante cuando se buscó que el prototipo sea una alternativa viable frente a soluciones comerciales ya existentes en el mercado.

Dentro de este capítulo, se describe a fondo el proceso de diseño del prototipo, incluyendo la justificación y selección de cada uno de sus componentes. Adicionalmente, también se aborda la manera en la que cada componente interactúa y contribuye al funcionamiento general del sistema.

### **4.1 Variables a monitorear**

El monitoreo continuo y preciso de las variables ambientales es vital para asegurar el curado adecuado del concreto. Es fundamental definir y comprender las variables a monitorear, junto con sus rangos óptimos de operación, para garantizar registros fiables y consistentes dentro de la cámara húmeda.

La temperatura es un factor crítico en el proceso de curado, influenciando la velocidad de hidratación del cemento, lo cual afecta directamente la resistencia y durabilidad del concreto final. La normativa ASTM C511 exige mantener una temperatura constante de  $23 \pm 0.5$  °C durante el curado.

Mantener una humedad relativa superior al 95% dentro de la cámara es necesario para el proceso de curado. Este nivel de humedad garantiza que haya suficiente agua disponible para la hidratación del cemento durante el curado.

Como se ve, para el proceso llevado a cabo en el cuarto húmedo del curado se requiere de dos variables exclusivamente.

## **4.2 Conceptualización del Prototipo**

La ingeniería, con su naturaleza constructiva, experimental y analítica, ha sido testigo de una constante evolución, sobre todo en áreas donde la precisión es primordial para garantizar la calidad y la conformidad con normativas. En este marco se inscribe nuestro prototipo: una solución tangible y avanzada para una necesidad latente dentro del Centro de Investigaciones en Vivienda y Construcción (CIVCO).

El prototipo está concebido como un sistema de monitoreo en tiempo real de la temperatura y humedad, centrado en la plataforma Arduino. A diferencia de otros sistemas que ofrecen lecturas episódicas y no garantizan un registro continuo de condiciones, este prototipo enfrenta las necesidades actuales del cuarto húmedo proporcionando una monitorización constante y exacta de los parámetros ambientales.

Además de monitorear las condiciones presentes. El sistema facilita el almacenamiento y análisis de datos en tiempo real, proporcionando un historial completo de las condiciones a lo largo del tiempo. Mediante la integración con sistemas de almacenamiento en la nube, se puede revisar tendencias, identificar patrones y responder a desviaciones.

En un entorno donde la norma ISO/IEC 17025 exige registros históricos por medio de equipo calibrado por un laboratorio certificado, el prototipo promete superar las limitaciones del sistema actual de la habitación húmeda.



De acuerdo con las investigaciones en el ámbito del monitoreo en tiempo real, se deduce que la eficacia de tales sistemas descansa en una cadena bien estructurada de operaciones y componentes (Danladi & Baykara, 2022; Tang et al., 2020).

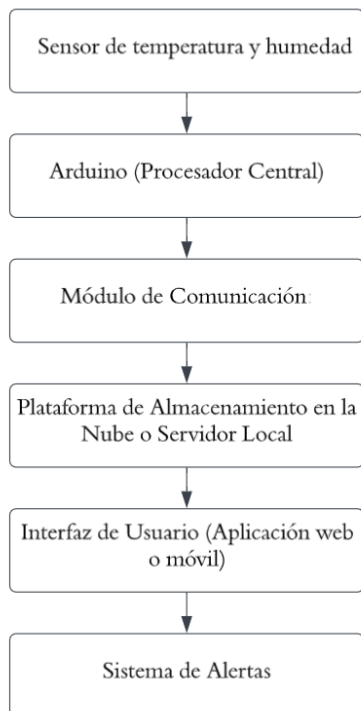


Figura 4.1 Diagrama de flujo del proceso de monitoreo en tiempo real. Fuente: Elaboración propia.

Todo debe comenzar con un dispositivo que sea capaz de medir el parámetro de temperatura y de humedad, o en determinado caso dos dispositivos, uno para cada parámetro. Estratégicamente posicionado en el ambiente de interés, este sensor realiza un seguimiento continuo de las variaciones ambientales. Las mediciones que emite son la base sobre la cual todo el proceso se encuentra.

Luego entra en juego el hardware, que es considerado el corazón procesador del sistema. Al recibir los datos provenientes del sensor, se encarga de almacenar temporalmente las mediciones, procesarlas y adaptarlas a un formato idóneo para su posterior transmisión.

Dicha transmisión es responsabilidad del Módulo de Comunicación. Su función principal consiste en enviar la información procesada hacia el repositorio designado, la nube.

Una vez que los datos son transmitidos, son alojados y gestionados por la Plataforma de Almacenamiento. Este componente no solo almacena la información, sino que también se encarga de realizar reportes. Es fundamental mencionar que, basándose en investigaciones previas, si se detectan valores que se desvíen de los parámetros preestablecidos, la plataforma está preparada para generar alertas.

Finalmente, se encuentra la Interfaz de Usuario. Esta herramienta permite que los interesados puedan interactuar con el sistema, visualizar las mediciones en tiempo real y acceder a registros históricos.

### 4.3 Hardware: Arduino Uno

#### 4.3.1 Descripción general

El Arduino UNO R3, con referencia de producto A000066, es una placa de desarrollo versátil que sirve como una introducción ideal al mundo de la electrónica y la codificación para makers, novatos e industriales. Equipado con los procesadores ATmega328P y ATmega 16U2, este hardware no solo proporciona una experiencia inicial valiosa en el ecosistema de Arduino, sino que también se presta eficazmente para diversas aplicaciones prácticas y experimentales en el ámbito de monitoreo ambiental.

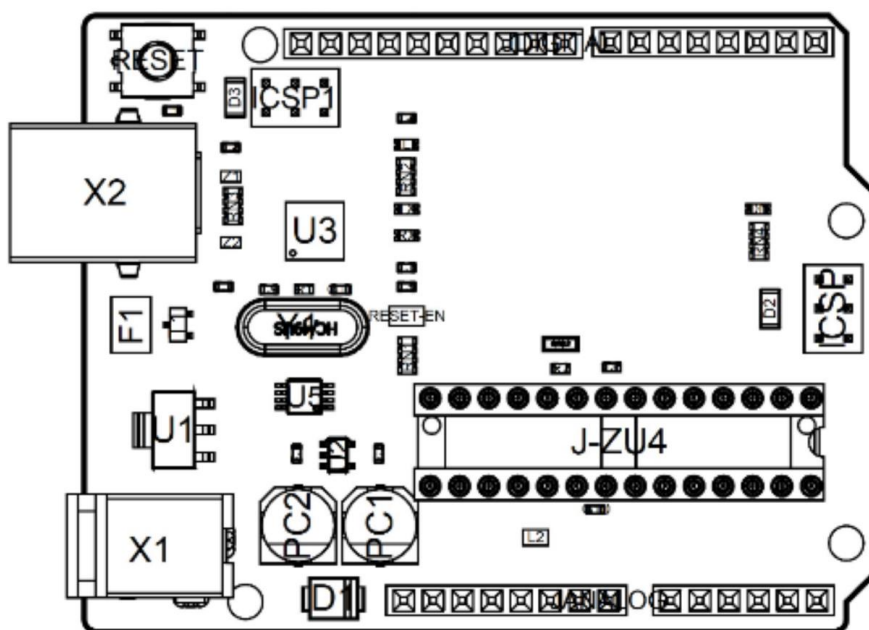


Figura 4.2 Placa de Arduino Uno R3. Arduino (2023)

#### 4.3.2 Razones de selección

El Arduino UNO fue seleccionado como el microcontrolador central por varias razones clave:

Experiencia Previa: La familiaridad previa con el Arduino UNO facilita un desarrollo más fluido y eficiente del prototipo, ya que se comprenden sus capacidades y limitaciones.

**Bajo Costo:** Con un precio accesible, el Arduino UNO ofrece una solución económica sin comprometer significativamente el rendimiento o la funcionalidad.

**Versatilidad:** Capaz de manejar diversas tareas relacionadas con el monitoreo de temperatura y humedad, el Arduino UNO es adecuado para este proyecto específico.

**Fácil Acceso a Recursos:** Hay una abundancia de tutoriales, guías, y foros dedicados a Arduino, facilitando el proceso de desarrollo y solución de problemas (Arduino, 2023).

La siguiente tabla es un cuadro comparativo con otros microcontroladores similares al Arduino Uno, presenta los ventajas y desventajas entre ellos.

Tabla 4.1 Cuadro comparativo de ventajas y desventajas de las opciones de microcontroladores.

Microcontrolador	Ventajas	Desventajas
Arduino Uno	<ol style="list-style-type: none"> <li>1. El más barato de los tres.</li> <li>2. Ideal para principiantes debido a su simplicidad y facilidad de uso.</li> <li>3. Ideal para principiantes debido a su simplicidad y facilidad de uso.</li> </ol>	<ol style="list-style-type: none"> <li>1. Limitada capacidad de memoria y procesamiento.</li> <li>2. Pines de entrada y salidas limitados.</li> <li>3. No es adecuado para proyectos que requieran multitareas o procesamiento avanzado.</li> </ol>
Arduino Mega	<ol style="list-style-type: none"> <li>1. Ofrece una mayor cantidad de pines de E/S en comparación con Arduino Uno.</li> <li>2. Mayor memoria disponible para proyectos más extensos.</li> <li>3. Adecuado para proyectos que requieran una gran cantidad de conexiones, como robots o sistemas de automatización.</li> </ol>	<ol style="list-style-type: none"> <li>1. Más caro en comparación con Arduino Uno.</li> <li>2. Mayor tamaño físico.</li> <li>3. Aunque tiene más capacidades que el Uno, aún está limitado en comparación con soluciones más robustas como Raspberry Pi en términos de multitareas y procesamiento avanzado.</li> </ol>
Raspberry Pi	<ol style="list-style-type: none"> <li>1. Capaz de ejecutar un sistema operativo completo, lo que le permite hacer multitareas.</li> <li>2. Procesador más avanzado que y mayor capacidad de RAM que los otros dos.</li> </ol>	<ol style="list-style-type: none"> <li>1. Requiere una fuente de alimentación más robusta.</li> <li>2. Tiene un costo más elevado en comparación con el Arduino Uno.</li> <li>3. Requiere un nivel de conocimiento más avanzado para su operación y programación.</li> </ol>

Fuente: Elaboración propia.

De la Tabla 4.1 Cuadro comparativo de ventajas y desventajas de las opciones de microcontroladores es importante mencionar que cuando se hace la idea mayor o menor a algo es en referencia los otros dos microcontroladores que se están comparando en la tabla.

Al momento de escoger qué microcontrolador usar se nota que el Raspberry pi es el más completo de los tres, y por otra parte el Arduino Uno es el microcontrolador más básico, para tareas más simples y de menor memoria que los otros dos. Pero no hace falta el uso de un microcontrolador más robusto para la tarea

de monitoreo. Ya queda demostrado en otros trabajos la efectividad que ha tenido el Arduino Uno en tareas de monitoreo en tiempo real. Además, que el programa no llena por completo la memoria del Arduino Uno. No hace falta una mayor cantidad de entradas y salidas para optar en una opción como el Arduino Mega. Además de que el autor del presente trabajo ya tiene experiencia trabajando con Arduino, por lo que usar Raspberry Pi sería una opción que requiere de mayor trabajo al ser un entorno distinto.

### **4.3.3 Características y especificaciones del Arduino Uno R3**

A continuación, se proporciona una tabla con las especificaciones técnicas principales del microcontrolador Arduino Uno R3.

Tabla 4.2 Especificaciones técnicas del microcontrolador Arduino UNO R3.

<b>Especificación</b>	<b>Descripción</b>	<b>Característica</b>
Microcontrolador	ATmega328	Microcontrolador central del prototipo que realiza el procesamiento de datos y ejecuta el código principal.
Voltaje de Operación	5V	Asegura la operación estable y eficiente de los componentes conectados.
Voltaje de Entrada (recomendado)	7-9V	Rango óptimo de alimentación para el funcionamiento seguro y eficiente del Arduino Uno.
Voltaje de Entrada (límites)	6-20V	Límites de voltaje de entrada que el Arduino Uno puede manejar sin dañarse.
Pines de E/S Digitales	14	Permiten la conexión y control de varios componentes y sensores digitales.
Pines de Entrada Analógica	6	Permiten la lectura de señales analógicas de diversos sensores.
Corriente DC por Pin de E/S	40 mA	Límite de corriente que cada pin de E/S puede suministrar, importante para conectar componentes externos.
Corriente DC para Pin 3.3V	50 mA	Máxima corriente suministrable por el pin de 3.3V para alimentar componentes.
Memoria Flash	32 KB	Espacio disponible para cargar y almacenar el código del programa.
SRAM	2 KB	Memoria de acceso aleatorio para el almacenamiento temporal de datos durante la ejecución del programa.
EEPROM	1 KB	Memoria no volátil para almacenamiento persistente de datos entre reinicios.
Velocidad de Reloj	16 MHz	Velocidad a la que opera el microcontrolador, afectando el rendimiento y la velocidad de procesamiento del Arduino Uno.

Nota: Elaboración propia en base a la hoja de datos del Arduino Uno R3.

#### 4.4 Sensor: DHT 22

El DHT22, también conocido como AM2302, ha sido el sensor seleccionado para la implementación del presente prototipo, en virtud de sus características que lo hacen idóneo para la medición confiable de la temperatura y humedad relativa en el ambiente de un cuarto húmedo.

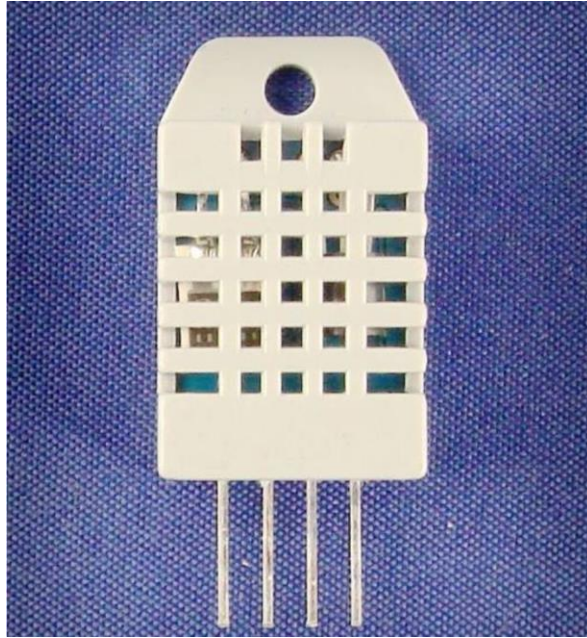


Figura 4.3 Sensor de Temperatura y Humedad DHT22. Aosong Electronics Co.,Ltd.

##### 4.4.1 Características del DHT22

El DHT22 es un sensor digital que ofrece mediciones de temperatura y humedad con un alto nivel de precisión. El rango de medición de temperatura oscila entre -40 y 80 grados Celsius con una exactitud de  $\pm 0.5^{\circ}\text{C}$ , mientras para la humedad relativa, el sensor puede registrar valores desde el 0 hasta el 100%, con una precisión de  $\pm 2\%$ .



Tabla 3.2 Características del sensor DHT-22.

Alimentación	3.3-6V
Corriente en medición	1-1.5mA
Corriente en espera	40-50 uA
Rango de humedad	0-100% RH
Rango de temperatura	-40 - 80°C
Precisión de humedad	±2% RH
Precisión de temperatura	±0.5°C

Nota: Recuperado de manual de usuario de sensor DHT22.

#### 4.4.2 Justificación de la Elección

Para la selección del sensor implicó la revisión y comparación de diversas opciones disponibles en el mercado, incluyendo varios sensores y probes. Las sondas, aunque precisas, fueron descartadas principalmente por su elevado costo, siendo el DHT22 una alternativa coste-eficiente y adecuada para la fase prototipo del proyecto.

El Ente Costarricense de Acreditación exige que los equipos de medición estén debidamente calibrados, es decir que equipo debe presentar un certificado hecho un laboratorio de calibración. Y para ser aceptado por el Ente Costarricense de Acreditación (ECA) la calibración debe ser realizada por un laboratorio que esté acreditado por la norma ISO/IEC 17025 para laboratorios de calibración.

Entre los distintos sensores evaluados, algunos no cumplían con los requisitos necesarios para operar eficientemente dentro del cuarto húmedo, como el alcance a valores de 99% de humedad relativa. El DHT22, en cambio, sí satisface estos requerimientos, lo que justifica su elección para la implementación del prototipo.

Además, el DHT22 se distingue por su facilidad de uso y por la abundante documentación y soporte comunitario disponible, factores que facilitan su integración y programación con la plataforma Arduino.

Tabla 4.3 Cuadro comparativo de ventajas y desventajas de las opciones de sensores.

Dispositivo	Ventajas	Desventajas
Sensor DHT11	<ol style="list-style-type: none"> <li>1. Bajo costo.</li> <li>2. Adecuado para aplicaciones y proyectos caseros.</li> <li>3. Interfaz digital.</li> <li>4. Consumo de energía reducido.</li> </ol>	<ol style="list-style-type: none"> <li>1. Rango de medición más limitado: 0°C a 50°C para temperatura y 20-80% para humedad.</li> <li>2. No tan duradero como el DHT22 o sondas especializadas.</li> <li>3. Susceptible a errores en condiciones de alta humedad.</li> </ol>
Sensor DHT22	<ol style="list-style-type: none"> <li>1. Rango de medición amplio: -40°C a 80°C para temperatura y 0-100% para humedad.</li> <li>2. Precisión relativamente alta: <math>\pm 0.5^{\circ}\text{C}</math> para temperatura y <math>\pm 2\%</math> para humedad.</li> <li>3. Interfaz digital que reduce la susceptibilidad a ruidos.</li> <li>4. Puede funcionar en condiciones de alta humedad.</li> </ol>	<ol style="list-style-type: none"> <li>1. Mayor costo en comparación con el DHT11.</li> <li>2. Menos robusto que algunas sondas profesionales.</li> <li>3. Mayor consumo de energía que el DHT11.</li> </ol>
Sonda de temperatura y humedad.	<ol style="list-style-type: none"> <li>1. A menudo ofrece un rango de medición más amplio, especialmente en modelos profesionales.</li> <li>2. Alta durabilidad, particularmente en sondas diseñadas para aplicaciones industriales.</li> <li>3. Algunos modelos ofrecen salidas tanto analógicas como digitales.</li> <li>4. Aptas para aplicaciones especializadas dependiendo del modelo.</li> </ol>	<ol style="list-style-type: none"> <li>1. Generalmente más costosas que los sensores DHT.</li> <li>2. Las sondas con salida analógica pueden ser más susceptibles a ruidos.</li> <li>3. Pueden requerir circuitos adicionales o componentes para la interfaz con microcontroladores.</li> <li>4. Algunas sondas pueden ser más grandes o difíciles de integrar en ciertos proyectos.</li> </ol>

Nota: Elaboración propia.

De la Tabla 4.3 Cuadro comparativo de ventajas y desventajas de las opciones de sensores. cabe destacar que cuando se hace la idea mayor o menor a algo es en referencia las otras dos opciones de sensores que se están comparando en la tabla.

Ya claro hay destacar que la opción de implementar el sistema desarrollado para el monitoreo es opcional por parte del laboratorio, es por esto que es un prototipo lo que se está desarrollando. Por lo que optar por una opción para el desarrollo de este como una sonda especializa como la que está expuesta en la sección 7.1.2 Selección de soluciones comparables se sale del alcance de costo para la elaboración del prototipo con una sonda especializada. Por esta razón y más ejemplos de trabajos similares del desarrollo de un prototipo de monitoreo se opta por las opciones de los sensores DHT. Pero ocurre que el sensor DHT11 presenta la desventaja que su rango de medición de humedad relativa llega hasta 80%. Es decir, que no llega si quiera a la humedad que especifica la norma ASTM C511 que debe mantenerse dentro del cuarto húmedo, que es de un 95% o mayor.

Por estas razones y que el sensor DHT22 es un sensor que posee la ventaja de trabajar en zonas de alta humedad es por esto que se escoge para el desarrollo del prototipo.

#### **4.4.3 Integración con Arduino**

La integración del DHT22 con el sistema Arduino es sencilla y directa. A través de bibliotecas específicas, es posible comunicar eficientemente ambos dispositivos, permitiendo la lectura y procesamiento adecuada de los datos recabados por el sensor.

### **4.5 Componentes y Materiales**

#### **4.5.1 Pantalla Liquid Crystal 1602**

La Pantalla Liquid Crystal 1602 se seleccionó por ser un componente para visualizar de manera clara y eficiente los datos en tiempo real. Con un fondo azul, tiene la capacidad de mostrar hasta 16 caracteres distribuidos en dos líneas, facilitando una lectura rápida. Impulsado por el chip HD44780, este display asegura una integración sencilla y efectiva con el sistema Arduino.



Figura 4.4 Pantalla Liquid Crystal 1602 para visualización de datos. MicroJPM (2023.)

Tabla 4 Cuadro comparativo de ventajas y desventajas de las opciones de pantalla.

Pantalla	Ventajas	Desventajas
LCD1602	<ol style="list-style-type: none"> <li>1. Sencillez y facilidad de uso en diversos proyectos.</li> <li>2. Bajo consumo de energía.</li> <li>3. Compatible con una amplia variedad de microcontroladores, incluido Arduino.</li> <li>4. Precio asequible.</li> <li>5. Amplia disponibilidad y documentación en la web.</li> </ol>	<ol style="list-style-type: none"> <li>1. Resolución limitada con solo 2 líneas de 16 caracteres.</li> <li>2. Monocromático.</li> <li>3. No tiene capacidad para mostrar gráficos.</li> <li>4. Sin retroiluminación avanzada o personalizable.</li> </ol>
Graphic LCD 128x64 STN LED Backlight [LCD-00710]	<ol style="list-style-type: none"> <li>1. Capacidad para mostrar gráficos,</li> <li>2. Resolución de 128x64, que ofrece más espacio para mostrar información.</li> <li>3. Retroiluminación LED que puede mejorar la visibilidad en condiciones de poca luz.</li> <li>4. Compatible con una variedad de microcontroladores.</li> </ol>	<ol style="list-style-type: none"> <li>1. Puede requerir más pines o una interfaz más compleja para su conexión.</li> <li>2. Potencialmente más costoso que las opciones básicas de LCD.</li> <li>3. Requiere un manejo más complejo para la visualización de gráficos en comparación con los LCD de caracteres simples.</li> </ol>
LCD1604 16x4 Character LCD Display Module Blue Blacklight	<ol style="list-style-type: none"> <li>1. Ofrece 4 líneas de 16 caracteres, lo que permite mostrar más información en comparación con el LCD 1602.</li> <li>2. Retroiluminación azul que mejora la visibilidad.</li> <li>3. Fácil integración en proyectos, similar al LCD 1602.</li> <li>4. Precio menor que el LCD-00710.</li> </ol>	<ol style="list-style-type: none"> <li>1. Aunque tiene más líneas que el LCD 1602, sigue siendo limitado en comparación con las pantallas gráficas.</li> <li>2. Monocromático.</li> <li>3. No tiene capacidad para mostrar gráficos, solo caracteres.</li> <li>4. Puede consumir ligeramente más energía debido a la retroiluminación adicional.</li> </ol>

Nota: Elaboración propia.

Con base en la Tabla 4 Cuadro comparativo de ventajas y desventajas de las opciones de pantalla. es importante aclarar que cuando se hace la idea mayor o menor a algo es en referencia las otras dos opciones de pantalla que se están comparando en la tabla.

De la Tabla 4 Cuadro comparativo de ventajas y desventajas de las opciones de pantalla. se puede deducir que la pantalla más básica y de menor costo es la pantalla lcd1602. Las otras dos opciones ofrecen mayores características como más líneas de caracteres y hasta gráficos. Pero en cuanto a relevancia para el

proyecto y las necesidades del sistema para la pantalla, solo se necesita una que muestre los datos del monitoreo. Por lo que las otras ventajas que presentan las dos pantallas no tienen peso para el prototipo y con la pantalla lcd1602 se cumple la necesidad principal sin incurrir en gastos innecesarios.

#### 4.5.2 Módulo I2C FC-113

El Módulo I2C FC-113 actúa como un puente de comunicación eficaz entre la pantalla LCD y el Arduino. Este componente simplifica la conexión, requiriendo únicamente cuatro cables y utilizando el chip IC PCF8574T para asegurar una transmisión de datos fluida y confiable entre los dispositivos.

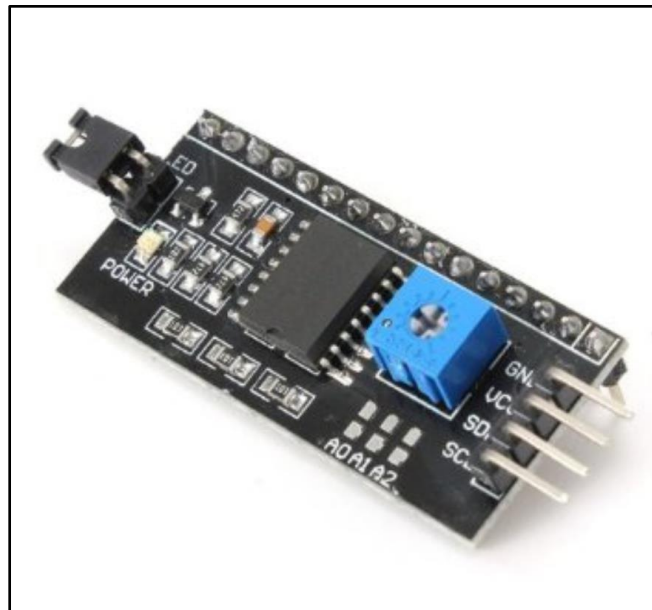


Figura 4.5 Módulo I2C FC-113 para Comunicación entre LCD y Arduino. MicroJPM (2023).

#### 4.5.3 Jumpers Macho-Macho

Los Jumpers Macho-Macho de 20 cm son fundamentales para establecer conexiones seguras y eficientes entre los diferentes componentes del prototipo. Estos cables flexibles y duraderos permiten una conexión rápida y segura, facilitando el proceso de ensamblaje y pruebas del prototipo.

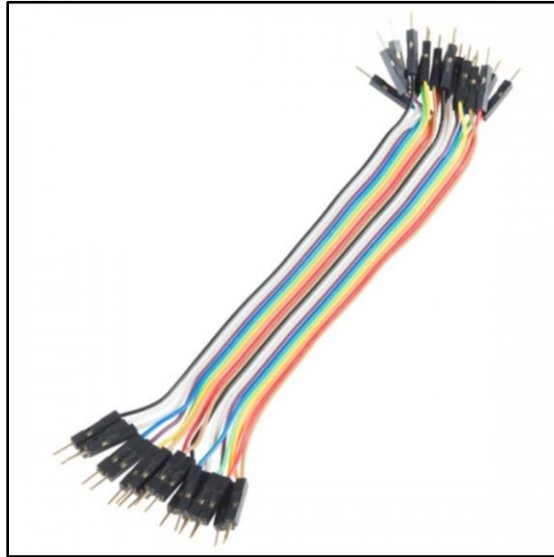


Figura 4.6 Jumpers Macho-Macho de 20 cm para Conexiones Seguras entre Componentes. MicroJPM (2023).

#### 4.5.4 Breadboard Mini Modular

El Breadboard Mini Modular provee un espacio de trabajo adecuado para el prototipado de circuitos. Con 170 puntos de conexión disponibles, este componente facilita la implementación y prueba de circuitos integrados y otros elementos necesarios para el funcionamiento óptimo del prototipo.

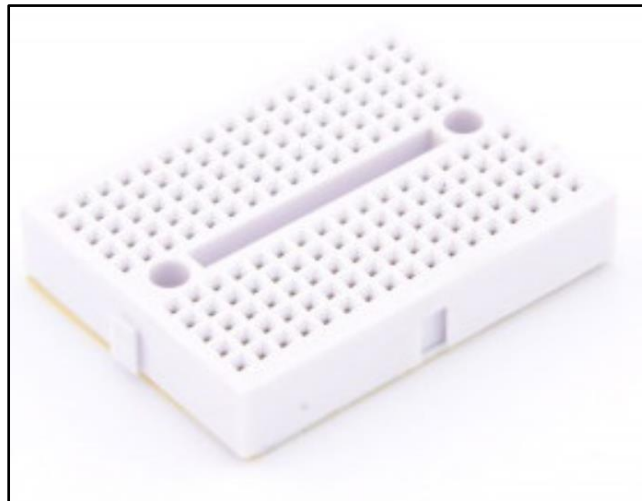


Figura 4.7 Breadboard mini modular. MicroJPM (2023).



#### 4.5.5 Cable Alarma 4 Hilos

Para la conexión segura del sensor DHT22 al sistema principal, se empleó el Cable Alarma 4 Hilos. Este cable posee cuatro conductores de calibre 22AWG, ofreciendo la flexibilidad y resistencia necesarias para una instalación que puede requerir ajustes y movimiento, garantizando una transmisión de datos continua y confiable.



Figura 4.8 Cable alarma de 4 hilos. MicroJPM (2023).

#### 4.5.6 Adaptador 9VDC

El Adaptador 9VDC a 1000mA es la fuente de energía seleccionada para alimentar el prototipo. Este adaptador, certificado por FCC y CE, transforma la corriente alterna de la red eléctrica a una salida continua de 9VDC y 1000mA, proporcionando el flujo de energía necesario para el funcionamiento ininterrumpido y seguro del sistema de monitoreo.



Figura 4.9 Adaptador de 9VDC. MicroJPM (2023).

#### 4.5.7 Caja de Proyecto Personalizada

La caja de proyecto es otro componente utilizado para el prototipo, diseñada para albergar y proteger todos los elementos internos del sistema de monitoreo. Esta caja, fabricada en acrílico resistente, fue elaborada mediante tecnología de corte láser para garantizar precisión en sus dimensiones y acabados.

El diseño de la caja no solo consideró el espacio necesario para alojar los componentes del prototipo de manera segura y ordenada, sino que también tuvo en cuenta los requerimientos específicos de instalación en el Centro de Investigaciones en Vivienda y Construcción (CIVCO). Estos requerimientos son: un gabinete que proteja a los componentes de polvo y que sean de materiales resistentes al contacto con el agua. El resultado es una caja de dimensiones adecuadas que permite una instalación en el espacio previamente destinado al sistema de monitoreo actual.

El color negro de la caja no fue elegido al azar. Este tono no solo proporciona una estética profesional y sobria al prototipo. Para el montaje de la caja personalizada se puede consultar el Apéndice B. Plano de montaje de caja personalizada.

#### **4.5.8 Materiales Adicionales**

Para el ensamblaje se utilizó una soldadura con cautín y cinta aislante para asegurar y proteger las conexiones entre el cable alarma de 4 hilos y los pines del sensor. Estos materiales adicionales son vitales para garantizar un montaje seguro y duradero del prototipo. Su justificación es que en contexto del proyecto el sensor se va a ubicar alejado del sistema. Además de que va a estar en una posición elevada del suelo por lo que es necesario una sujeción confiable para no perder el monitoreo en ningún momento.

### **4.6 Esquema de Conexión**

El esquema de conexión del prototipo diseñado para este proyecto asegura una interacción fluida entre todos los componentes mencionados anteriormente. A continuación, se describe de manera detallada cómo se interconectan estos componentes para formar un sistema de monitoreo de temperatura y humedad eficiente y confiable.

#### **4.6.1 Descripción del Esquema de Conexión**

Fuente de Energía:

Se utiliza un cable jumper para conectar el pin de 5V del Arduino Uno a una fila positiva (VCC) de la mini breadboard.

Se realiza una conexión similar con un cable jumper del pin GND del Arduino a una fila negativa (GND) de la breadboard, estableciendo así una referencia de voltaje común.

Pantalla Liquid Crystal 1602:

La pantalla no está conectada directamente al módulo I2C. Dado que está adherida a una cara de la caja de proyecto, se utilizan cables jumpers para conectar sus pines a la breadboard donde está situado el módulo I2C.

Módulo I2C:

El pin VCC del módulo I2C se conecta a la fila de 5V de la breadboard, mientras que su pin GND se enlaza con la fila de GND. Sus pines SDA y SCL se conectan a los pines correspondientes en el Arduino Uno (A4 y A5 respectivamente), facilitando la comunicación entre la pantalla y el microcontrolador.

Sensor DHT22:

Los pines de VCC y GND del sensor se conectan a las filas de VCC y GND de la breadboard, respectivamente.

El pin de datos del sensor (DATA) se conecta al pin 7 del Arduino Uno, permitiendo la transmisión de datos de temperatura y humedad al microcontrolador para su posterior procesamiento y visualización.

#### **4.6.2 Visualización del Esquema**

Se adjunta a continuación una representación gráfica del esquema de conexión, diseñada en TinkerCAD. Esta ilustración facilita la comprensión visual del enlace entre los diversos componentes y proporciona una guía clara para replicar o modificar el sistema en futuras implementaciones o ajustes.

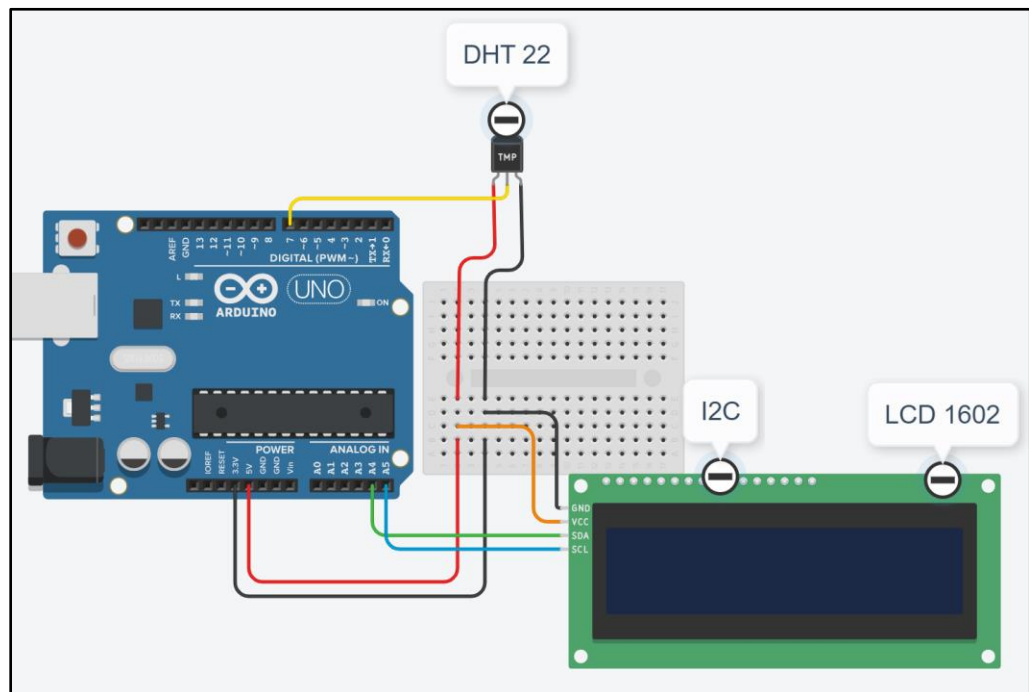


Figura 4.10 Esquema de conexión para prototipo. Fuente: Elaboración propia.

Este esquema de conexión asegura una transmisión de datos eficiente y una alimentación eléctrica adecuada a cada componente, conformando un sistema integrado y coherente para el monitoreo de temperatura y humedad en tiempo real en el Centro de Investigaciones en Vivienda y Construcción (CIVCO).

## 4.7 Programa para Arduino Uno

### 4.7.1 Introducción al Software

El Arduino Uno opera mediante un software específico que se carga en su microcontrolador. Este software, conocido también como programas de sólo lectura, es un conjunto de instrucciones escritas en un lenguaje de programación que el microcontrolador puede interpretar y ejecutar. Para el caso del Arduino Uno, se utiliza un dialecto del lenguaje C++ diseñado especialmente para la plataforma Arduino. El programa de sólo lectura permite al microcontrolador interactuar con los distintos componentes conectados a él, recibir datos de los sensores, procesar esta información, y finalmente mostrar los resultados en una pantalla LCD.

### 4.7.2 Descripción del Código

Para el propósito de este proyecto, se ha desarrollado un código que permite al Arduino Uno leer los datos del sensor DHT22 y mostrar la temperatura y la humedad en la pantalla LCD 1602. A continuación, se describen los segmentos más relevantes del código proporcionado. Hasta el momento el alcance de esta parte del proyecto se concentra en solo esto. Más adelante en el proyecto se comenta la integración de este código a la plataforma que lo conecta a internet y pueda cumplir con los demás objetivos de este trabajo.

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Figura 4.11 Bibliotecas usadas para programa en Arduino. Fuente: Elaboración propia.

Las líneas anteriores incluyen las bibliotecas necesarias para el funcionamiento del prototipo. La biblioteca DHT.h es para el sensor DHT22, Wire.h para la comunicación I2C, y LiquidCrystal\_I2C.h para el control de la pantalla LCD 1602 mediante el protocolo I2C.

```
#define DHT_TYPE DHT22
const int pinDHT = 7;
```

Figura 4.12 Definición del sensor en el programa de Arduino. Fuente: Elaboración propia

Se define el tipo de sensor DHT (en este caso, DHT22) y se asigna el pin 7 del Arduino como el pin de entrada de datos del sensor.

```
DHT dht(pinDHT, DHT_TYPE);  
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

Figura 4.13 Asignación a sensor y pantalla. Fuente: Elaboración propia.

Se inicializan los objetos `dht` y `lcd`, que permitirán interactuar con el sensor y la pantalla LCD, respectivamente.

```
void setup() {  
  Serial.begin(9600);  
  dht.begin();  
  lcd.begin(16, 2);  
  lcd.backlight();  
}
```

Figura 4.14 Void Setup del programa de Arduino. Elaboración propia.

El método `setup()` inicializa la comunicación serie, el sensor DHT22 y la pantalla LCD.

```

void loop() {
  float temperaturaCelsius = dht.readTemperature();
  float humedad = dht.readHumidity();

  if (isnan(temperaturaCelsius) || isnan(humedad)) {
    Serial.println("Error al leer el sensor DHT");
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Error al leer");
    lcd.setCursor(0, 1);
    lcd.print("el DHT");
  } else {
    Serial.print("Temperatura: ");
    Serial.print(temperaturaCelsius);
    Serial.print(" °C\t");
    Serial.print("Humedad: ");
    Serial.print(humedad);
    Serial.println(" %");

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temp: ");
    lcd.print(temperaturaCelsius);
    lcd.print(" C");
    lcd.setCursor(0, 1);
    lcd.print("Hum: ");
    lcd.print(humedad);
    lcd.print(" %");
  }
}

```

Figura 4.15 Voidloop del programa de Arduino. Fuente: Elaboración propia

El fragmento de código presentado es una sección vital del programa de Arduino, que se ejecuta continuamente dentro de la función `loop()`. Inicialmente, se realizan las lecturas de temperatura y humedad desde el sensor DHT22 mediante las funciones `dht.readTemperature()` y `dht.readHumidity()`, almacenando estos valores en las variables `temperaturaCelsius` y `humedad`, respectivamente.

A continuación, el código verifica si alguno de los valores leídos es "NaN" (Not a Number) a través de la función `isnan()`. Este escenario sucede cuando hay un error en la lectura del sensor. En tal caso, el programa informa sobre el error en la consola serial con `Serial.println("Error al leer el sensor DHT")` y presenta un mensaje de error en la pantalla LCD. Para visualizar el mensaje en la pantalla dicha, primero se



limpia cualquier dato previo con `lcd.clear()`, luego se posiciona el cursor en la esquina superior izquierda con `lcd.setCursor(0, 0)`, y finalmente se imprime el mensaje de error distribuido en las dos líneas de la pantalla.

Si las lecturas del sensor son válidas (es decir, no son NaN), el programa procede a imprimir los valores de temperatura y humedad en la consola serial, ofreciendo una manera de monitorización en tiempo real durante el desarrollo y la depuración. Además, muestra estos valores en la pantalla LCD. Para ello, después de limpiar la pantalla, se posiciona el cursor y se imprimen los valores de temperatura y humedad en formato legible, facilitando así que el usuario pueda leer directamente los datos capturados por el sensor DHT22.

```
delay(300000); // Espera 5 minutos
```

Figura 4.16 Tiempo de espera del programa. Fuente: Elaboración propia.

La instrucción `delay(300000)`; incorporada en el código implica una demora deliberada de 300,000 milisegundos, equivalentes a 5 minutos. Este intervalo temporal ha sido establecido cuidadosamente para moderar la frecuencia de las operaciones de lectura del sensor DHT22 y de actualización en la pantalla LCD. Este margen de cinco minutos entre lecturas consecutivas no es arbitrario ya que evita la generación de lecturas ininterrumpidas que podrían saturar tanto la consola serial como la pantalla LCD con un flujo excesivo de datos. Es relevante señalar que el Arduino Uno, con su microcontrolador ATmega328P, tiene un flujo de datos máximo, para comunicación serial, de hasta 115200 bps, aunque comúnmente se utiliza una tasa de 9600 bps por su estabilidad (Arduino, nd). En términos de tiempo de lectura, el tiempo mínimo entre lecturas analógicas es de aproximadamente 100 microsegundos, lo que teóricamente permitiría hasta 10,000 lecturas por segundo. Sin embargo, considerando el procesamiento, comunicación y otras tareas, es prudente mantener un intervalo adecuado, como los cinco minutos propuestos, para garantizar la estabilidad y precisión del sistema

#### 4.8 Prototipo Ensamblado y Operativo

El proceso de ensamblaje del prototipo inicia con la carga del programa previamente desarrollado y explicado en el apartado 4.7 en la placa de microcontrolador Arduino Uno. Este paso es necesario, ya que es el software el que orquesta la sinfonía de componentes y circuitos, haciendo posible la recolección, procesamiento y exhibición de los datos de temperatura y humedad en tiempo real.

Una vez cargado el programa, se procedió al ensamblaje físico del prototipo siguiendo fielmente el esquema de conexión delineado en la sección 4.6. Este esquema de conexión sirvió como una guía invaluable, asegurando que cada componente se interconectara en la posición correcta, garantizando así la funcionalidad y seguridad del sistema.

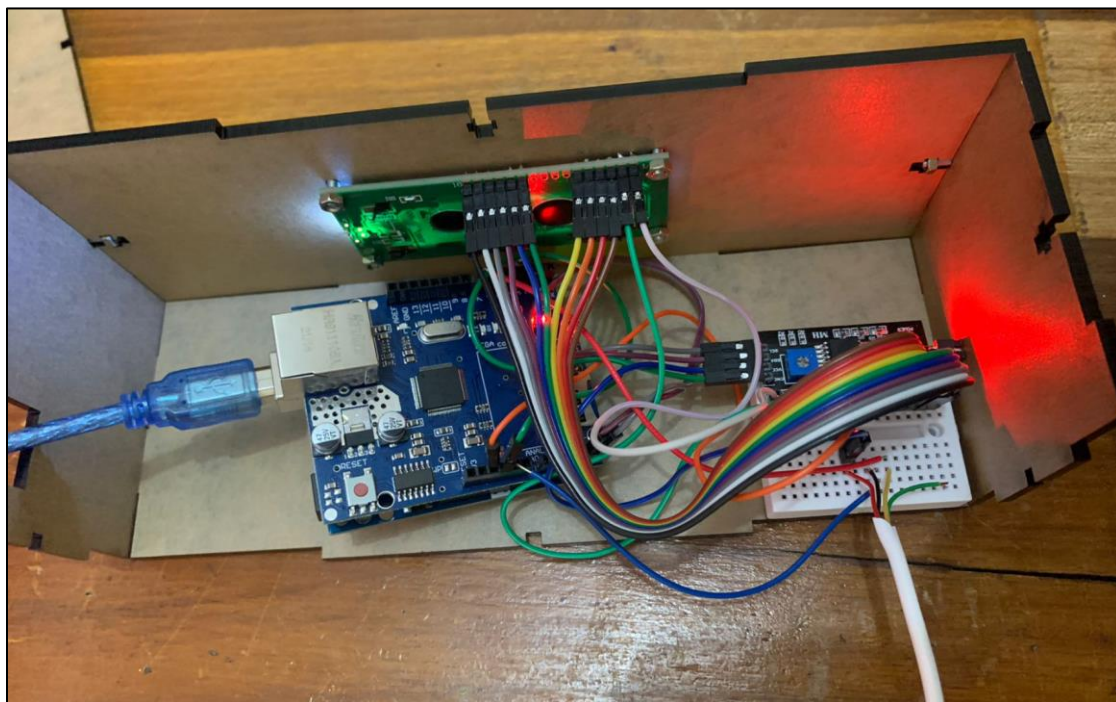


Figura 4.17 Vista interna del prototipo, mostrando el esquema de conexión implementado y los componentes ensamblados. Fuente: Elaboración propia. Fuente: Elaboración propia.

La Figura 4.17 Vista interna del prototipo, mostrando el esquema de conexión implementado y los componentes ensamblados. Fuente: Elaboración propia. que se presenta en este apartado ilustra claramente la conexión efectiva de todos los componentes del sistema. Esta imagen no solo sirve como un registro visual del trabajo completado, sino que también funciona como una herramienta educativa para aquellos interesados en replicar o entender el funcionamiento intrínseco del prototipo.

Posteriormente, se procedió a albergar cuidadosamente los componentes ensamblados dentro de la caja de proyecto diseñada a medida, como se explicó en el apartado 4.5. La caja no solo ofrece protección a los delicados circuitos, sino que también proporciona un medio para visualizar la información relevante a través de la pantalla LCD 1602.



Figura 4.18 Vista frontal del prototipo con la pantalla LCD 1602 en funcionamiento, exhibiendo datos en tiempo real de temperatura y humedad. Fuente: Elaboración propia.

La segunda imagen muestra el prototipo ya completamente ensamblado y operativo, con la pantalla LCD iluminando los resultados en tiempo real del monitoreo de temperatura y humedad. En esta imagen, el prototipo no es solo un conjunto de componentes electrónicos y cables, sino un sistema cohesivo y funcional.

Cabe resaltar que el diseño presentado del prototipo constituye una fase inicial y no terminal del proyecto integral. En esta versión preliminar, aunque operativa, aún no se ha incorporado un elemento necesario para la funcionalidad completa del sistema como es el shield o dispositivo que posibilita la conexión a Internet. Dicha conexión es fundamental para lograr un monitoreo remoto y en tiempo real, permitiendo el acceso y análisis de los datos desde cualquier lugar y momento. La integración de este componente demandará, inevitablemente, ajustes y modificaciones adicionales en el código cargado en el Arduino. Este proceso de mejora y expansión de capacidades del prototipo será abordado en el siguiente capítulo del proyecto, asegurando un sistema de monitoreo más robusto, conectado y adaptado a las necesidades de monitoreo y registro continuo de datos ambientales.

## Capítulo 5: Integración del Sistema de Monitoreo con la Nube

En este capítulo, se aborda el proceso de integración del sistema de monitoreo desarrollado con servicios de almacenamiento en la nube. Este paso logra facilitar el acceso remoto a los datos generados por el sistema, posibilitando la visualización en tiempo real, el almacenamiento de datos históricos y la recepción de alertas pertinentes en caso de condiciones anómalas o fallas detectadas por el sistema. Este nivel de conectividad y acceso remoto no solo amplía la utilidad y la accesibilidad de los datos recolectados, sino que también proporciona herramientas vitales para el monitoreo continuo y la respuesta rápida a posibles problemas identificados. Además de que ya mencionado en reiteradas ocasiones que un sistema con fácil acceso al registro histórico facilita la tarea de trazar los datos de monitoreo.

Se detalla y justifica la selección de hardware específico para la conexión a Internet, en este caso, el módulo Ethernet Shield. Este componente permite una comunicación fluida y confiable entre el Sistema de Monitoreo y la plataforma de almacenamiento en la nube seleccionada, ThingSpeak. Se explicará el proceso de configuración y las modificaciones realizadas al código original de Arduino para asegurar una transmisión de datos exitosa.

Posteriormente, se introduce la plataforma de notificación SendGrid, seleccionada para el envío de alertas y reportes históricos vía correo electrónico. Se describirán las razones detrás de esta elección y los pasos seguidos para la configuración y puesta en marcha de esta herramienta de notificación.

Finalmente, este capítulo aborda el desarrollo y validación de los programas de reportes históricos y alertas. Estos programas son para el monitoreo efectivo y la trazabilidad de las condiciones dentro del laboratorio, en concordancia con los requisitos de la norma ISO/IEC 17025.

A través de la integración cuidadosa de estas tecnologías y herramientas, el Capítulo 5 delinea cómo se ha mejorado el sistema de monitoreo para ofrecer un servicio más robusto, confiable y accesible, brindando así un soporte invaluable para las operaciones diarias y la garantía de calidad en el laboratorio.

## **5.1 Selección de Hardware para Conexión a Internet: Ethernet Shield**

### **5.1.1 Descripción**

El Ethernet Shield W5100 para Arduino es un módulo que facilita la conexión de la placa Arduino a la Internet. Este hardware específicamente es la versión genérica del Arduino Ethernet Shield V1. Al emplear este módulo, la placa Arduino puede vincularse a la red mediante un cable RJ45 (no provisto con el módulo). El Ethernet Shield opera con un voltaje de 5V, suministrado directamente por la placa Arduino.

El componente clave de este Ethernet Shield es el controlador Ethernet W5100. Este controlador cuenta con un buffer interno de 16K y tiene una velocidad de conexión que puede variar entre 10 y 100Mb. El W5100 es un chip robusto que proporciona una pila de red (IP) con capacidad para manejar protocolos TCP y UDP, y permite hasta cuatro conexiones de socket simultáneas.

El módulo se conecta al puerto SPI del Arduino, y este diseño permite que se puedan apilar otros shields encima del Ethernet Shield. Para facilitar el almacenamiento y la transferencia de archivos a través de la red, el shield incorpora una ranura para tarjetas micro-SD.

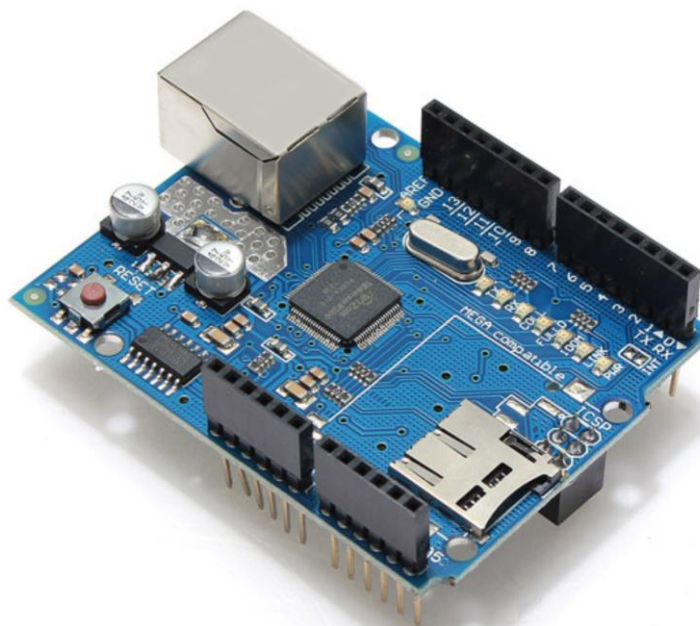


Figura 5.1 Placa de Ethernet Shield. MicroJPM (2023).

### 5.1.2 Justificación

Inicialmente, la concepción del proyecto gravitaba en torno a la tecnología LoRaWAN, un protocolo de comunicación diseñado para sistemas de largo alcance y bajo consumo de energía. No obstante, en el transcurso de la fase de investigación y planificación, se identificaron ciertos elementos tomados en cuenta que motivaron una reconsideración de esta elección inicial.

El proyecto se desarrolla en el contexto específico del Centro de Investigación de Vivienda y Construcción (CIVCO). En este entorno, se observó que ya existía una infraestructura de red Ethernet estable y accesible, con un punto de conexión situado convenientemente cerca del sistema de monitoreo propuesto, permitiendo una conexión directa mediante un breve cable Ethernet de 0,5 metros.

Esta circunstancia favoreció la consideración de alternativas tecnológicas que pudieran aprovechar la infraestructura de red existente. En este proceso exploratorio, se descubrió el Ethernet Shield W5100, un

dispositivo no solo compatible con la placa Arduino Uno —utilizada en este proyecto— sino que además presentaba varias ventajas estratégicas:

- **Economía:** Comparado con soluciones basadas en LoRaWAN, el Ethernet Shield W5100 se presentó como una opción más económica y coste-eficiente.
- **Facilidad de Integración:** El Shield puede integrarse sin dificultades con Arduino Uno.
- **Adaptabilidad al Entorno:** Dadas las condiciones específicas del laboratorio, el Ethernet Shield W5100 no solo respondía a las necesidades del proyecto, sino que se alineaba de manera óptima con las características de la infraestructura de red ya disponible en el lugar.
- **Estabilidad de Conexión:** Aprovechando la red Ethernet existente, se garantiza una conexión estable y confiable, fundamental para la transmisión en tiempo real y el almacenamiento de datos del sistema de monitoreo.

La idea de LoRaWAN se descarta completamente a pesar de una opción altamente usada en proyectos similares y la posibilidad de conexión a internet en lugares remotos ya que es una opción más costosa, no solo por el costo inicial de adquisición, sino que también requiere pagos periódicos para el internet. Y el proyecto se enfoca para las necesidades del Centro de Investigación en Vivienda y Construcción (CIVCO) que ya tiene una estructura de ethernet en sus instalaciones del laboratorio.

### **5.1.3 Proceso de Instalación y Configuración**

El término “shield” refiere a un dispositivo concebido para ensamblarse de forma directa y complementaria sobre la placa Arduino Uno. Tal diseño permite no sólo una integración física sino también la unión funcional entre ambos aparatos. Una vez incorporado el Ethernet Shield, las conexiones que



inicialmente convergían en el Arduino son redirigidas hacia los terminales correspondientes del mencionado escudo, proporcionando al sistema integral capacidades de conectividad a redes Ethernet.

Esta mecánica de instalación no solo optimiza la disposición física y operativa del hardware, sino que, además, favorece un esquema de trabajo armonizado y cohesivo entre los distintos componentes del sistema.

Tras culminar el proceso de instalación se procedió con las labores de configuración y adecuación del entorno de programación. El software residente en el Arduino debe ser modificado para incorporar y capitalizar las funcionalidades introducidas por el Ethernet Shield.

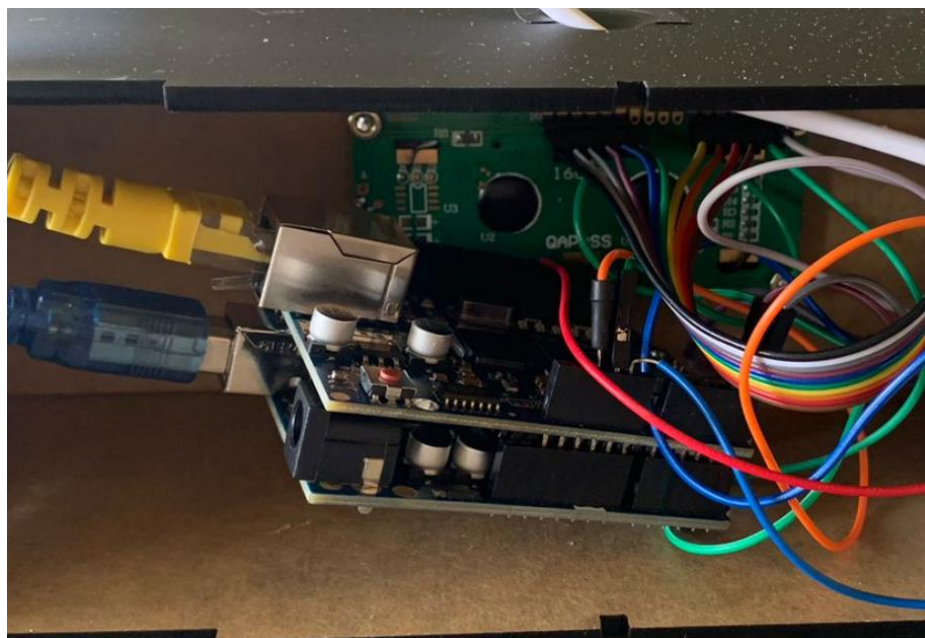


Figura 5.2 Visualización del montaje integrado de Arduino Uno y Ethernet Shield W5100. Fuente: Elaboración propia.

## 5.2 Selección de Plataforma de Almacenamiento en la Nube: ThingSpeak

### 5.2.1 Descripción de ThingSpeak

ThingSpeak es una plataforma de aplicación de Internet de las Cosas (IoT) diseñada para permitir la recopilación, análisis y visualización de datos en tiempo real. Este recurso es útil para cualquier proyecto que implique la recopilación y el análisis de datos desde múltiples fuentes, siendo de especial relevancia en contextos de monitoreo ambiental, control domótico y proyectos educativos o de investigación.

#### Características Principales:

- **Recolección de Datos:** ThingSpeak proporciona la capacidad de recoger datos de numerosos dispositivos, incluyendo dispositivos Arduino, y consolidar estos datos en la nube. Estos datos pueden ser variables de sensores, tiempo de ejecución y cualquier otro tipo de datos numéricos o de texto.
- **Análisis en Tiempo Real:** Una vez los datos son recopilados, ThingSpeak ofrece herramientas para analizarlos en tiempo real. Esto es necesario para identificar patrones, analizar tendencias y tomar decisiones basadas en los datos recogidos.
- **Visualización:** La plataforma permite la creación de visualizaciones de los datos recolectados. Los usuarios pueden crear gráficos, tablas y otras visualizaciones que faciliten la interpretación de los datos.
- **API Abierta:** La plataforma proporciona una API abierta que facilita la integración con otros sistemas y dispositivos. Esta característica permite a los usuarios expandir y personalizar sus proyectos de manera efectiva.
- **Comunidad de Desarrolladores:** ThingSpeak cuenta con una comunidad que contribuyen con nuevo código, tutoriales y soporte usuarios.

## 5.2.2 Justificación de selección de la plataforma

La decisión de seleccionar ThingSpeak como plataforma de almacenamiento en la nube para el presente proyecto de investigación se fundamentó en una serie de factores relevantes que atienden tanto a las necesidades específicas del proyecto como a consideraciones de eficiencia y costo.

En primer lugar, el proyecto requería de una solución confiable y eficaz que permitiera no solo enviar datos de monitoreo sino también almacenarlos de manera segura y accesible. ThingSpeak responde a estas exigencias ofreciendo un servicio sólido y continuo de recolección y almacenamiento de datos en la nube. Esto asegura que la información generada por el sistema de monitoreo sea preservada y esté disponible para su análisis y revisión en cualquier momento, aspecto fundamental para la trazabilidad y el control de la información en el marco del presente estudio.

Una consideración fundamental en la fase de prototipo del proyecto es el manejo eficiente de los recursos y el control de costos. En este sentido, ThingSpeak emerge como una opción sumamente ventajosa. El servicio proporciona un plan gratuito que permite el almacenamiento de hasta un millón de datos durante cada año de uso, lo que se traduce en un apoyo significativo para proyectos en fase inicial o de prototipo, permitiendo explorar las funcionalidades del sistema sin incurrir en costos elevados.

Adicionalmente, es importante destacar que ThingSpeak no solo se limita a ser un servicio de almacenamiento de datos. La plataforma ofrece un conjunto de herramientas para el análisis en tiempo real y la visualización de los datos almacenados, lo cual facilita enormemente el proceso de interpretación y presentación de la información recolectada. Esto, unido a su integración nativa con MATLAB, brinda un entorno potente para el manejo, procesamiento y visualización de datos, ofreciendo un marco de trabajo integral y robusto para el proyecto.

Por último, la decisión se vio respaldada por la existencia de una comunidad activa y colaborativa en torno a ThingSpeak. Esta comunidad, compuesta por desarrolladores y usuarios de la plataforma, representa

un recurso valioso de soporte y asistencia en el uso y manejo del servicio, proporcionando una red de conocimiento y experiencia a la que recurrir en caso de necesidad.

Tabla 5 Cuadro comparativo de ventajas y desventajas de las opciones de plataforma de almacenamiento.

Plataforma	Ventajas	Desventajas
ThingSpeak	<ol style="list-style-type: none"> <li>1. Integración sencilla con una amplia variedad de dispositivos, incluidos los de Arduino y Raspberry Pi.</li> <li>2. Proporciona visualizaciones de datos en tiempo real.</li> <li>3. Funciones de análisis y procesamiento de datos incorporadas.</li> <li>4. Permite la integración con MATLAB.</li> <li>5. Es gratuito para un uso básico.</li> </ol>	<ol style="list-style-type: none"> <li>1. Las características avanzadas requieren una suscripción paga.</li> <li>2. La frecuencia de actualización gratuita es relativamente baja.</li> <li>3. Menos opciones de personalización en comparación con algunas otras plataformas.</li> </ol>
Blynk	<ol style="list-style-type: none"> <li>1. Interfaz de usuario altamente personalizable.</li> <li>2. Proporciona una aplicación móvil que facilita la monitorización y control remoto.</li> <li>3. Amplia biblioteca de widgets para diferentes funcionalidades.</li> </ol>	<ol style="list-style-type: none"> <li>1. La interfaz del usuario puede ser menos intuitiva en comparación con algunos competidores.</li> <li>2. Algunas características avanzadas requieren un plan premium.</li> <li>3. Limitaciones en la cantidad de correos electrónicos en la versión gratuita.</li> </ol>
Ubitots	<ol style="list-style-type: none"> <li>1. Interfaz fácil de usar.</li> <li>2. Ofrece herramientas para visualización de datos.</li> <li>3. Alta frecuencia de actualización y capacidad de almacenamiento.</li> <li>4. Posee características avanzadas como eventos y alertas.</li> <li>5. Proporciona opciones de integración con otros servicios y herramientas.</li> </ol>	<ol style="list-style-type: none"> <li>1. La versión gratuita tiene limitaciones en cuanto a dispositivos y actualizaciones.</li> <li>2. La personalización completa y las integraciones empresariales pueden tener un costo adicional.</li> </ol>

Nota: Elaboración propia.

Sobre Tabla 5 Cuadro comparativo de ventajas y desventajas de las opciones de plataforma de almacenamiento. Se aclara que cuando se menciona mayor o menor a algo es en referencia las otras dos opciones de plataformas.

Sobre la elección de qué plataforma utilizar para el proyecto. Las tres opciones propuestas ofrecen similitudes en sus características. Las tres ofrecen planes gratuitos que se pueden adaptar a las necesidades como plataforma de almacenamiento de datos. Por lo que la necesidad del proyecto en cuanto a plataforma de almacenamiento en la nube recae en la cantidad de datos que puede almacenar. Y en este punto la plataforma de ThingSpeak destaca por la cantidad de datos que ofrece almacenar en la nube con su plan gratuito. Este plan ofrece almacenar un millón de datos por cada año. Ajustándose a la necesidad de la configuración del programa de Arduino detallada en la sección 4.7 Programa para Arduino Uno que menciona que tiene un periodo de 5 minutos entre cada dato que envía a la nube.

### 5.2.3 Proceso de Configuración

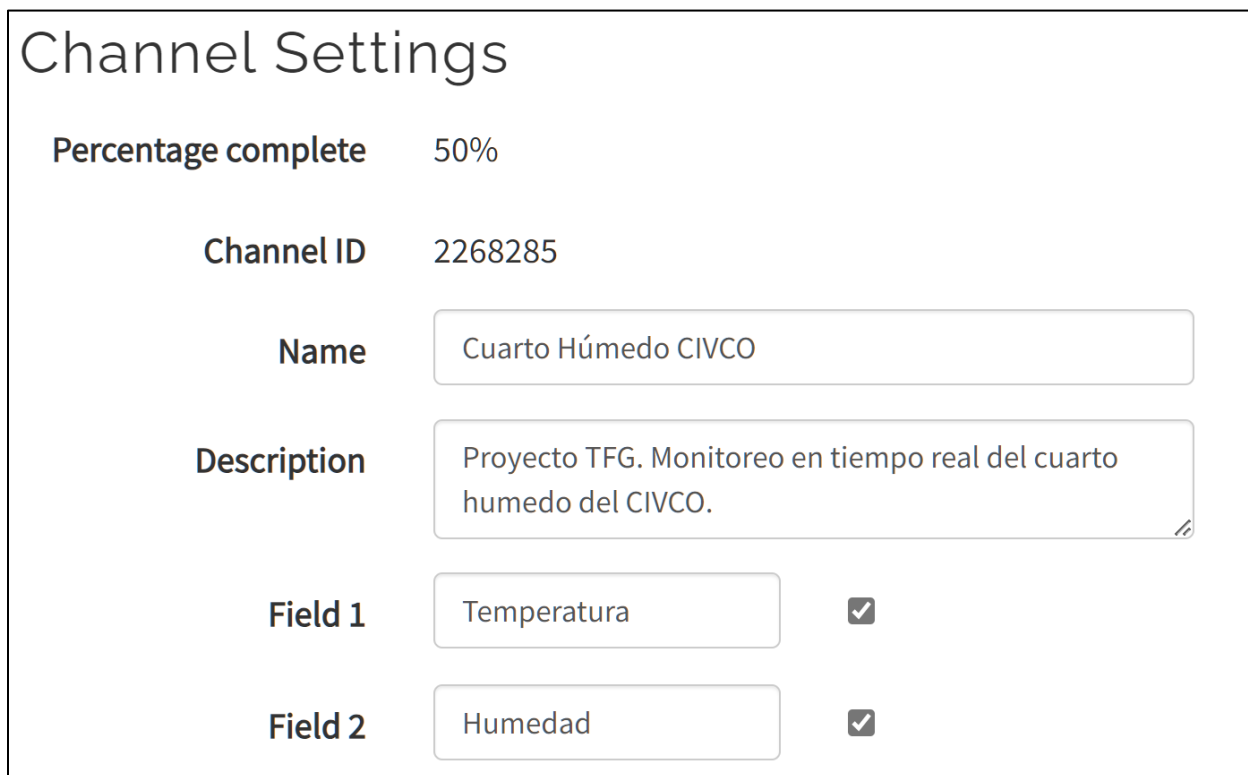
Para configurar ThingSpeak con el sistema de monitoreo, primero se creó una cuenta en MathWorks, ya que ThingSpeak es un producto de esta empresa.

Se estableció un canal en ThingSpeak con el nombre “Cuarto Húmedo CIVCO”, que se describe como “Proyecto TFG. Monitoreo en tiempo real del cuarto húmedo del CIVCO”. Este canal fue configurado para recibir y visualizar datos de temperatura y humedad del cuarto húmedo del CIVCO en tiempo real.



Figura 5.3 Detalles del canal de ThingSpeak. Fuente: Elaboración propia.

Se configuraron dos campos en el canal. “Field 1” se destinó a las mediciones de temperatura y “Field 2” a los datos de humedad. Esta configuración facilita la interpretación de los datos y asegura un registro ordenado.



<h2>Channel Settings</h2>	
Percentage complete	50%
Channel ID	2268285
Name	<input type="text" value="Cuarto Húmedo CIVCO"/>
Description	<input type="text" value="Proyecto TFG. Monitoreo en tiempo real del cuarto humedo del CIVCO."/>
Field 1	<input type="text" value="Temperatura"/> <input checked="" type="checkbox"/>
Field 2	<input type="text" value="Humedad"/> <input checked="" type="checkbox"/>

Figura 5.4 Configuración de Parámetros del canal en ThingSpeak. Fuente: Elaboración propia.

Se generó una clave API de escritura (Write API Key) que ThingSpeak proporciona para el canal creado. Esta clave, que se introdujo en el código del Arduino, permite la transmisión de datos desde el Arduino a ThingSpeak, asegurando que los datos se registren en el canal correspondiente.

### 5.3 Modificación del Código de Arduino

La modificación del código de Arduino es necesaria para integrar el sistema de monitoreo con ThingSpeak mediante el uso del Ethernet Shield. Este segmento describe las modificaciones significativas realizadas al código previamente mencionado, permitiendo el envío de datos al canal de ThingSpeak.

```
// Configuración del Ethernet Shield
#include <Ethernet.h>

byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
EthernetClient client;

char server[] = "api.thingspeak.com";
String apiKey = "API KEY";
```

Figura 5.5 Código de configuración del ethernet shield. Fuente: Elaboración propia.

En esta sección se introduce la biblioteca Ethernet.h para el manejo del Ethernet Shield. Se define una dirección MAC para el Shield, se instancia un objeto EthernetClient, se establece el servidor de ThingSpeak y se introduce la clave API (Write API Key) de ThingSpeak asignada al canal previamente configurado.

```

void setup() {

  Serial.println("Iniciando configuracion de Ethernet con DHCP...");
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Fallo al configurar Ethernet usando DHCP");
    while (true);
  }

  Serial.print("Direccion IP asignada por DHCP: ");
  Serial.println(Ethernet.localIP());
  delay(1000);
}

```

Figura 5.6 Código de principio de configuración del ethernet shield. Fuente: Elaboración propia.

En la función `setup()`, el sistema inicia el proceso de conexión a la red mediante DHCP, como se indica con el mensaje serial "Iniciando configuración de Ethernet con DHCP...". El protocolo DHCP es utilizado para asignar automáticamente una dirección IP al Ethernet Shield, facilitando su incorporación a la red.

El segmento `if (Ethernet.begin(mac) == 0)` intenta inicializar la conexión de Ethernet con la dirección MAC proporcionada. Si este intento no es exitoso, lo que se denota por un retorno de 0 por parte de `Ethernet.begin(mac)`, el sistema imprime un mensaje de error: "Fallo al configurar Ethernet usando DHCP". Este escenario de fallo detiene la ejecución adicional del código, ingresando en un bucle infinito `while (true);`, dado que sin una dirección IP válida, el dispositivo no puede continuar con las operaciones de red planificadas.

Si se obtiene una IP válida, la dirección asignada se imprime en la consola serial para referencia y diagnóstico, utilizando `Serial.println(Ethernet.localIP())`. Este paso es para verificar que el dispositivo ha sido correctamente asignado y reconocido dentro de la red, permitiendo así el flujo subsiguiente de datos hacia la plataforma ThingSpeak.



```

void loop() {

  Serial.println("Intentando conectar a ThingSpeak...");

  int intentos = 0;
  while (!client.connect(server, 80) && intentos < maxReintentos) {
    Serial.println("Error al intentar conectar a ThingSpeak. Reintentando...");
    intentos++;
    intentosTotales++;
    delay(1000); // Espera un segundo antes de reintentar
  }

  if (intentos < maxReintentos) {
    Serial.println("Conectado exitosamente a ThingSpeak.");
    intentosTotales = 0; // Resetear intentos totales fallidos ya que la conexión fue exitosa

    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(temperaturaCelsius);
    postStr += "&field2=";
    postStr += String(humedad);
    postStr += "\r\n\r\n";

    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);

    delay(2000); // Espera un poco para que el servidor responda
    while (client.available()) {
      char c = client.read();
      Serial.print(c);
    }

    client.stop();
  } else {
    Serial.println("Fallo después de varios intentos. Pasando al siguiente ciclo.");
    if (intentosTotales >= maxReintentosTotales) {
      Serial.println("Maximos intentos fallidos alcanzados. Reiniciando conexion Ethernet...");
      client.stop();
      Ethernet.begin(mac);
      intentosTotales = 0; // Resetear contador de intentos totales
      delay(5000); // Esperar un poco antes de continuar
    }
  }
}
delay(300000); // Espera 5 minutos
}

```

Figura 5.7 Voidloop del código modificado para conexión con ThingSpeak. Fuente: Elaboración propia.

Dentro de la función loop(), el código maneja la conexión continua con ThingSpeak, la plataforma en la nube seleccionada para el almacenamiento y análisis de datos. Este segmento de código es vital para

garantizar que los datos recopilados por el Arduino sean transmitidos y almacenados de manera efectiva en la plataforma online.

Primero, el sistema intenta establecer una conexión con ThingSpeak. Para asegurar la conexión del sistema ante posibles fallos de conexión, se implementa un ciclo que realiza tres intentos de conexión en un lapso de un segundo por cada intento en caso de fracaso inicial. Este proceso de reintentos proporciona una robustez adicional al sistema, permitiendo manejar situaciones temporales de pérdida de conexión sin perder datos importantes ni requerir intervención manual.

Cada intento de conexión es seguido por un mensaje de diagnóstico impreso en la consola serial, proporcionando feedback sobre el estado de la conexión en tiempo real. En caso de fallar en establecer conexión después de tres intentos, el sistema pausa brevemente por un segundo antes de intentar de nuevo, minimizando el riesgo de sobrecargar la red o los servidores de ThingSpeak con solicitudes repetidas.

Una vez que se establece la conexión, el código procede a preparar los datos recopilados para su transmisión. Los valores de temperatura y humedad se adjuntan a una cadena de texto, que se estructura de acuerdo con los requisitos de la API de ThingSpeak para la transmisión de datos. Esta cadena de texto se envía luego al servidor de ThingSpeak utilizando un protocolo de transmisión de datos (HTTP POST), asegurando que los datos sean recibidos y almacenados correctamente en la plataforma.

Luego de enviar los datos, el sistema espera una respuesta del servidor de ThingSpeak. Los datos recibidos en respuesta son leídos y mostrados en la consola serial para fines de diagnóstico y monitoreo. Finalmente, la conexión con ThingSpeak se cierra hasta el próximo ciclo de 5 minutos de transmisión.

En situaciones donde la conexión falla repetidamente, el código está diseñado para reiniciar la conexión Ethernet automáticamente después de tres determinado de intentos fallidos, esto garantiza que cualquier problema persistente de conectividad pueda ser resuelto reiniciando la interfaz de red del sistema, facilitando la recuperación de la conexión y la reanudación de la transmisión de datos.

## **5.4 Selección de Plataforma de Notificación: SendGrid**

### **5.4.1 Descripción de SendGrid**

SendGrid es una plataforma de servicio en la nube que proporciona servicios de entrega de correo electrónico transaccional. En términos simples, permite a los desarrolladores enviar correos electrónicos de manera programática desde sus aplicaciones, ya sea para notificar a los usuarios, confirmar registros, enviar alertas, y más. SendGrid es particularmente útil para aplicaciones que requieren la entrega de grandes volúmenes de correos electrónicos, proporcionando herramientas robustas y un servicio confiable para garantizar que los correos lleguen a sus destinatarios de manera oportuna y sin problemas relacionados con la filtración de spam.

La plataforma de SendGrid ofrece un conjunto de APIs que facilitan la integración con aplicaciones y sistemas diversos. En el contexto de este proyecto, SendGrid será empleado para enviar notificaciones y alertas generadas por el sistema de monitoreo, permitiendo así la comunicación efectiva y automática de condiciones críticas o datos relevantes detectados por el dispositivo Arduino en tiempo real.

La utilidad de SendGrid en el proyecto radica en su capacidad para automatizar el proceso de notificación, permitiendo que los usuarios reciban información puntual y relevante directamente en sus buzones de correo electrónico sin intervención manual continua. Este servicio es para mantener informados a las personas relacionadas al proyecto acerca del estado y funcionamiento del sistema de monitoreo, así como para alertar sobre situaciones que requieran atención o intervención inmediata.

### **5.4.1 Justificación de selección**

La selección de SendGrid como plataforma de notificación para este proyecto se fundamenta en una serie de consideraciones pragmáticas y estratégicas. En primera instancia, el proyecto exige una solución que facilite tanto la entrega de alertas instantáneas como la distribución de reportes periódicos de monitoreo, los cuales deben ser accesibles y fácilmente interpretables para los usuarios finales.

Uno de los requerimientos del sistema es la capacidad de enviar reportes que puedan ser descargados en formatos como Excel o CSV. Estos formatos son indispensables ya que permiten una manipulación de datos flexible y detallada, posibilitando la creación de gráficos y análisis estadísticos que son para el proceso de trazabilidad, un elemento que se encuentra en los protocolos de acreditación requeridos en el ámbito de investigación en el que se sitúa el proyecto.

Además, es fundamental considerar factores económicos. El proyecto, en su fase actual, tiene un carácter de prototipo, por lo que optar por soluciones de bajo coste que permitan validar la viabilidad y eficacia del sistema antes de contemplar inversiones más significativas. SendGrid emerge como una opción atractiva en este sentido, ya que proporciona un plan gratuito que se alinea perfectamente con las necesidades de esta fase del proyecto, ofreciendo servicios sin incurrir en costos adicionales.

Otros métodos de notificación evaluados, como los mensajes SMS, involucran gastos que se prefieren evitar en esta etapa. Además, las alternativas como aplicaciones dedicadas, aunque viables, no ofrecen la misma conveniencia y accesibilidad que el correo electrónico para la recepción y gestión de reportes de datos.

Tabla 6 Cuadro comparativo de ventajas y desventajas de las opciones de plataforma de notificación.

Plataforma	Ventajas	Desventajas
Sendgrip	<ol style="list-style-type: none"> <li>1. Capacidades sólidas de entrega de correos electrónicos con tasas de entrega elevadas.</li> <li>2. Proporciona análisis detallados y en tiempo real del rendimiento del correo electrónico.</li> <li>3. Capacidad para escalarse y manejar grandes volúmenes de correos electrónicos.</li> <li>4. Plan gratuito limitado a 6000 correos por cada mes.</li> </ol>	<ol style="list-style-type: none"> <li>1. Algunas características avanzadas requieren un plan de pago.</li> <li>2. Limitaciones en la cantidad de correos electrónicos en la versión gratuita.</li> </ol>
Mailgun	<ol style="list-style-type: none"> <li>1. Escalable para manejar grandes volúmenes de correos electrónicos.</li> <li>2. Soporte para pruebas de validación de correo electrónico.</li> <li>3. Ofrece un plan "Pay-as-you-go" para controlar los costos según el uso.</li> </ol>	<ol style="list-style-type: none"> <li>1. Puede ser menos amigable para usuarios sin experiencia técnica.</li> <li>2. No ofrece un plan gratuito.</li> <li>3. Los costos pueden aumentar rápidamente con el volumen de correos electrónicos.</li> </ol>
Mailchimp	<ol style="list-style-type: none"> <li>1. Interfaz de usuario intuitiva y amigable.</li> <li>2. Amplia gama de plantillas y herramientas de diseño.</li> <li>3. Capacidades sólidas de segmentación y automatización.</li> <li>4. Proporciona análisis detallados y herramientas de informes.</li> </ol>	<ol style="list-style-type: none"> <li>1. Algunas características avanzadas están reservadas para planes más caros.</li> <li>2. Plan gratuito limitado a 1000 correos por mes.</li> <li>3. Las políticas estrictas pueden llevar al cierre de cuentas si no se siguen las mejores prácticas de envío.</li> </ol>

Nota: Elaboración propia.

La

Tabla 6 Cuadro comparativo de ventajas y desventajas de las opciones de plataforma de notificación. sirve como referencia tomada en cuenta para la elección de Sendgrip como plataforma de envío de notificaciones (alertas y reportes). Lo más importante para este apartado y la característica de mayor peso en la escogencia de plataforma es la cantidad de envíos de correos electrónicos en sus planes. Mailgun no ofrece un gratuito, y por otra parte Mailchimp tiene un límite de 1000 correos electrónicos por cada mes en su plan gratuito. A diferencia de Sendgrip que su límite es 6000 correos electrónicos, ajustándose de sobra para los requerimientos de envíos de correos electrónicos, que van a ser utilizados solamente para un correo por mes de reporte y los necesarios para los correos de alertas.

#### **5.4.2 Proceso de Configuración**

El proceso de configuración para SendGrid comienza con la creación de una cuenta y un emisor, el cual se mostrará como remitente en los correos electrónicos enviados. Para este fin, se selecciona un correo electrónico dedicado exclusivamente a las notificaciones, facilitando así su identificación por parte de los destinatarios.

Tras la creación de la cuenta, se procede a establecer el emisor. SendGrid requiere comprobar que el usuario tiene permiso para enviar mensajes desde la dirección de correo electrónico seleccionada. Este paso es necesario para garantizar la autenticidad de los mensajes y reducir las posibilidades de que sean marcados como spam. La validación se realiza mediante un correo electrónico de confirmación enviado a la dirección seleccionada. Este emisor es muestra en Figura 5.8 Identidad del remitente de Sendgrip., que es la identidad que usa sendgrip para enviar los correos de forma automática.

SENDERS	ADDRESS	NICKNAME	VERIFIED
<b>Monitoreo Cuarto Húmedo CIVCO</b> <b>FROM</b> mchcivco@gmail.com <b>REPLY</b> mchcivco@gmail.com	CIVCO, Tecnológico de Costa Rica	MonitoreoCIVCO	✓

Figura 5.8 Identidad del remitente de Sendgrip. Fuente: Elaboración propia.

Una vez confirmado el remitente, es necesario generar una API Key. Esta clave será utilizada como credencial en las solicitudes enviadas desde los scripts del sistema a SendGrid. La API Key debe integrarse en los códigos responsables de los reportes y alertas, permitiendo así el envío de correos electrónicos con la información generada por el sistema.

Es aconsejable realizar pruebas de envío para verificar que la configuración es correcta y que los mensajes llegan adecuadamente a los destinatarios. En los scripts, además de integrar la API Key, se deben definir el destinatario, el asunto y el cuerpo del mensaje, asegurando que la información enviada sea comprensible para los receptores.

## 5.5 Desarrollo del Programa de Reportes Históricos

En esta sección se presenta el desarrollo y validación del programa de reportes históricos, un componente para monitorear y documentar las variables ambientales de temperatura y humedad en el tiempo.

El código proporcionado se desarrolla en el lenguaje de programación Python y tiene la función de obtener datos de ThingSpeak, procesarlos y enviar reportes por correo electrónico mediante la plataforma SendGrid. Se proporciona un desglose paso a paso del código para explicar su funcionamiento:

**Obtención de datos de ThingSpeak:** Se define el identificador del canal y la clave API de lectura para acceder a los datos. Luego, se realiza una solicitud a la URL de ThingSpeak para obtener los espacios del canal, seleccionando datos de los últimos 30 días.

```

# Detalles de ThingSpeak
CHANNEL_ID = "CHANNEL_ID"
READ_API_KEY = "API_KEY"

# URL para obtener datos del canal
url = f"https://api.thingspeak.com/channels/{CHANNEL_ID}/feeds.json?api_key={READ_API_KEY}&days=30"

response = requests.get(url)
data = response.json()
feeds = data['feeds']

```

Figura 5.9 Conexión con el canal de ThingSpeak. Fuente: Elaboración propia.

Procesamiento de datos: Los datos recibidos se procesan para extraer información relevante como temperatura, humedad, fecha y hora. Se realiza una conversión de la zona horaria a la local (Costa Rica en este caso).

```

for feed in feeds:
    # Convertir la fecha/hora a la zona horaria de Costa Rica
    date_time = datetime.strptime(feed['created_at'], '%Y-%m-%dT%H:%M:%SZ')
    date_time = utc_zone.localize(date_time).astimezone(costa_rica_zone)

    # Convertir a representación naive (sin info de zona horaria)
    date_time = date_time.replace(tzinfo=None)

    # Separar fecha y hora
    dates.append(date_time.date())
    times.append(date_time.time())

    temp = float(feed['field1'])
    humidity = float(feed['field2'])

    entries.append((date_time, temp, humidity))

```

Figura 5.10 Conversión a zona horaria local. Fuente: Elaboración propia.

Sobre la creación del reporte de texto. Se genera un informe de texto que registra la temperatura y la humedad junto con la fecha y hora correspondientes. El informe se guarda como un archivo de texto (.txt) con el nombre que incluye la fecha actual.



```

# Crear el texto del "diario"
diary_text = f"Reporte de Monitoreo de Temperatura y Humedad\nIntervalo: {start_date} - {end_date}\n\n"
for entry in entries:
    date_time, temp, humidity = entry
    diary_text += f"Fecha y hora: {date_time}\n"
    diary_text += f"Temperatura: {temp}°C\n"
    diary_text += f"Humedad: {humidity}%\n\n"

# Nombre de archivos con fecha
current_date = datetime.now().strftime("%Y-%m-%d")
txt_filename = f"reporte_{current_date}.txt"
xlsx_filename = f"reporte_{current_date}.xlsx"

# Guarda el texto en un archivo
with open(txt_filename, "w") as report_file:
    report_file.write(diary_text)

# Crear un DataFrame con los datos
df = pd.DataFrame({
    'Fecha': dates,
    'Hora': times,
    'Temperatura (°C)': temperaturas,
    'Humedad (%)': humedades
})

```

Figura 5.11 Creación de reporte de archivo de texto y dataframe. Fuente: Elaboración propia.

Creación de un marco de datos (DataFrame): Se utiliza la biblioteca pandas para crear un marco de datos (Dataframe) que contiene las fechas, horas, temperaturas y humedades extraídas. Este marco de datos se guarda como un archivo Excel (.xlsx) con un nombre similar al archivo de texto.

```

# Guardar el DataFrame en un archivo Excel
df.to_excel(xlsx_filename, index=False)

```

Figura 5.12 Creación de archivo Excel desde un dataframe. Fuente: Elaboración propia.

Preparación y envío de correos electrónicos: Se utiliza la llave API de SendGrid para enviar el correo electrónico. Se definen los detalles del correo, como el remitente, el destinatario, el asunto y el contenido. Se adjuntan los archivos de texto y Excel al correo electrónico, y finalmente, se envía el correo.

```

# Enviar el correo usando SendGrid
SENDGRID_API_KEY = "
EMAIL_FROM = "mchcivco@gmail.com"
EMAIL_TO = ""

message = Mail(
    from_email=EMAIL_FROM,
    to_emails=EMAIL_TO,
    subject=f"Reporte de Monitoreo de Temperatura y Humedad - {current_date}",
    plain_text_content="Por favor, consulte los archivos adjuntos para el reporte de monitoreo."
)

# Función para adjuntar archivos
def attach_file_to_message(message, filepath, mime_type):
    with open(filepath, "rb") as f:
        data = f.read()
    encoded = base64.b64encode(data).decode()
    attachment = Attachment()
    attachment.file_content = FileContent(encoded)
    attachment.file_type = FileType(mime_type)
    attachment.file_name = FileName(filepath.split("/")[-1]) # Obtiene el nombre del archivo
    attachment.disposition = Disposition('attachment')
    message.attachment = attachment

# Adjuntar archivos
attach_file_to_message(message, txt_filename, 'text/plain')
attach_file_to_message(message, xls_filename, 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet')

# Enviar el correo
sg = SendGridAPIClient(SENDGRID_API_KEY)
response = sg.send(message)

print(response.status_code)

```

Figura 5.13 Código de fragmento del envío de correo electrónico. Fuente: Elaboración propia.

Este programa automatiza el proceso de generación y envío de reportes, facilitando el monitoreo continuo y la documentación de las condiciones ambientales en el área de estudio. La generación automática de reportes es fundamental para tener registros históricos precisos y actualizados, los cuales son necesarios para la trazabilidad que se le solicita al laboratorio para los procesos normados. Estos reportes pueden ser fácilmente descargados y utilizados para generar gráficos y estadísticas necesarias para el proceso de acreditación mencionado anteriormente. En Apéndice D Ilustración de parte recortada del reporte de monitoreo en archivo de Excel. muestra una imagen de como es este reporte.

## 5.6 Desarrollo del Programa de Alertas

Se presenta el código escrito en Python cuyo propósito principal es monitorear en tiempo real los datos ambientales, generando alertas cuando los valores medidos de temperatura y humedad se desvían de un rango predefinido.

Acceso a Datos de ThingSpeak: Se establecen identificadores de canal y claves API, facilitando el acceso a los datos más recientes almacenados en ThingSpeak. Mediante una solicitud específica a la API de ThingSpeak, el programa extrae la última entrada de datos del canal correspondiente.

```

# Detalles de ThingSpeak
CHANNEL_ID = ""
READ_API_KEY = ""

# URL para obtener datos del canal
url = f"https://api.thingspeak.com/channels/{CHANNEL_ID}/feeds.json?api_key={READ_API_KEY}&results=1"

response = requests.get(url)
data = response.json()
feed = data['feeds'][0]

# Convertir datos de temperatura y humedad a float
temp = float(feed['field1'])
humidity = float(feed['field2'])

```

Figura 5.14 Acceso a datos del canal de ThingSpeak. Fuente: Elaboración propia.

### 5.6.1 Sistema de alertas de rangos permitidos en el monitoreo

Definición de Rangos Permitidos: Se establecen los límites de temperatura y humedad. Para temperatura se determina un máximo de 25° C y un mínimo de 20° C. Para la humedad relativa se puso un límite inferior de 93% hasta comprobar la calibración del sensor. Cualquier medición que se desvíe de estos rangos predeterminados activará una alerta, lo cual se diseñó para el monitoreo en tiempo real y la respuesta rápida a condiciones ambientales no óptimas.

```

# Rangos permitidos
TEMP_MIN = 20
TEMP_MAX = 25
HUM_MIN = 93

```

Figura 5.15 Fragmento de límites permitidos para las alertas. Fuente: Elaboración propia.

Generación de Alertas: El código evalúa las condiciones ambientales en tiempo real. Si las lecturas de temperatura y/o humedad están fuera de los rangos permitidos, se genera un mensaje de alerta. Este mensaje incluye información detallada sobre las condiciones actuales y se estructura para facilitar su

comprensión y respuesta rápida. En el Apéndice E Ilustración de correo de alerta sobre desconexión del sistema, se muestra como se ve una alerta de desconexión.

```
# Inicializar variables
alert_text = "Alerta de Monitoreo en Cuarto Húmedo\n\n"
subject = "Alerta de Monitoreo: "

# Verificar si los valores están fuera del rango
temp_alert = temp < TEMP_MIN or temp > TEMP_MAX
hum_alert = humidity < HUM_MIN

if temp_alert or hum_alert:
    if temp_alert:
        subject += "Temperatura "
        if temp < TEMP_MIN:
            alert_text += f"La temperatura está baja: {temp}°C\n"
        elif temp > TEMP_MAX:
            alert_text += f"La temperatura está alta: {temp}°C\n"

    if hum_alert:
        if temp_alert: # Si ya hay una alerta de temperatura, añadir 'y' al asunto
            subject += "y "
        subject += "Humedad "
        if humidity < HUM_MIN:
            alert_text += f"La humedad está baja: {humidity}%\n"

subject += "en Cuarto Húmedo"
```

Figura 5.16 Comprobación de rangos permitidos. Fuente: Elaboración propia.

El envío de alertas por correo electrónico utilizando la llave API de SendGrid, esta sección del código automatiza el proceso de envío de alertas por correo electrónico a un destinatario predefinido. Se estructura el mensaje de correo electrónico, se añade el mensaje de alerta como contenido y se envía el correo a la dirección especificada.

```

# Enviar el correo usando SendGrid
SENDGRID_API_KEY = ""
EMAIL_FROM = "mchcivco@gmail.com"
EMAIL_TO = ""

message = Mail(
    from_email=EMAIL_FROM,
    to_emails=EMAIL_TO,
    subject=subject,
    plain_text_content=alert_text
)

sg = SendGridAPIClient(SENDGRID_API_KEY)
response = sg.send(message)

print(response.status_code)

```

Figura 5.17 Envío de alertas por correo electrónico mediante SendGrip. Fuente: Elaboración propia.

Este código es fundamental para el sistema de monitoreo propuesto en este trabajo ya que proporciona una respuesta inmediata a condiciones que podrían ser perjudiciales para el ambiente monitoreado, facilitando intervenciones rápidas y medidas correctivas. Su implementación garantiza que se mantenga un registro constante y se generen alertas en tiempo real, para una gestión eficiente del entorno en estudio.

### 5.6.2 Implementación de alertas de desconexión

La importancia de asegurar una monitorización constante en tiempo real de las condiciones ambientales es crítica para prevenir posibles desviaciones que puedan impactar negativamente en el ambiente controlado. La integridad y continuidad de los datos es un aspecto necesario. Por lo tanto, se debe implementar un sistema de alertas que notifique automáticamente al usuario en caso de una desconexión del sistema de sensores.

La desconexión de los sensores o interrupciones en la transmisión de datos son identificadas comparando la hora actual con la marca de tiempo de la última entrada de datos enviada por los sensores al canal de ThingSpeak. Este proceso se realiza utilizando la librería `datetime` de Python y convirtiendo las

cadena de tiempo de las entradas de datos a objetos de tiempo, permitiendo calcular la diferencia entre el último registro y el momento actual.

Una vez detectada una desconexión, definida por un período de inactividad superior a un límite preestablecido de 15 minutos el sistema genera una alerta automática. El motivo de elección de 15 minutos es para no saturar la computadora de virtual de Google Cloud y no haga solicitudes de comprobación de conexión tan seguido y que el tiempo de 15 minutos es un tiempo razonable para alertar al usuario y que no haya gran pérdida del monitoreo. El mensaje de alerta incluye detalles pertinentes acerca de la interrupción en la transmisión de datos, instando a una revisión inmediata del sistema y hardware correspondiente.

El envío de alertas se realiza mediante el uso de la API de SendGrid, la cual permite automatizar el proceso de envío de emails de manera eficiente y segura. Mediante el envío de una solicitud HTTP POST a la API de SendGrid con los detalles del email, se garantiza que las alertas son entregadas inmediatamente al buzón del destinatario predefinido.

A continuación, se presenta un fragmento del código Python responsable de detectar desconexiones y enviar alertas en consecuencia.

```
else:
    now = datetime.now(timezone.utc)
    last_feed_time = datetime.strptime(feed['created_at'], '%Y-%m-%dT%H:%M:%S').replace(tzinfo=timezone.utc)
    time_difference = now - last_feed_time
    print(f"Time difference: {time_difference}") # Imprime o registra la diferencia de tiempo para depuración
    if time_difference > timedelta(minutes=15):
        should_send_alert = True
        subject = "Alerta de Monitoreo: Desconexión del Sistema"
        alert_text = "El sistema de monitoreo no ha enviado datos en los últimos 15 minutos. Por favor verifica la conexión y el funcionamiento del hardware."
```

Figura 5.18 Fragmento de código de alerta por desconexión en el sistema de monitoreo. Fuente: Elaboración propia.

## **5.7 Automatización de Alertas en Google Cloud**

La infraestructura de Google Cloud Platform (GCP) fue utilizada para implementar y automatizar el sistema de alertas del proyecto, asegurando un monitoreo continuo y confiable de las condiciones ambientales sin depender de un sistema local. En especial para el caso de las alertas ya que el sistema por sí mismo no tiene la capacidad de conectarse y enviar las alertas. Además de que en el caso de que pierda alguna conexión no sería capaz de comunicar este estado. A continuación, se describen los pasos y estrategias empleadas para la automatización del sistema de alertas en GCP.

### **5.7.1 Despliegue del Código en Google Cloud Functions**

Lo primero para automatizar el programa de alertas fue el uso de Google Cloud Functions que es un entorno de ejecución sin servidor que permite crear y ejecutar aplicaciones. La utilización de esta plataforma permite enfocarse en la lógica del código, mientras que Google Cloud se encarga de la gestión de los recursos computacionales.

La primera etapa del despliegue en Google Cloud Functions (GCF) implicó la configuración de un nuevo proyecto en la plataforma de Google Cloud, asegurando que todas las API necesarias, como las de GCF y SendGrid, estuviesen habilitadas. Posteriormente, el código del proyecto, desarrollado en Python y responsable de monitorear las variables ambientales y emitir alertas, fue subido al entorno de GCF.

El fragmento de código previamente desarrollado (ver sección 5.6.2) fue ligeramente modificado para adaptarse al entorno sin servidor de Google Cloud Functions. Dicha adaptación involucra ajustes en la lógica de ejecución y manejo de peticiones HTTP que son propias de la estructura de las funciones en la nube.

Cloud Functions Detalles de la función EDITAR BORRAR COPIAR

check\_sensor\_data 2ª gen. (Se implementó el 11 oct 2023, 18:36:58) URL: [https://us-central1-steady-cat-400305.cloudfunctions.net/check\\_sensor\\_data](https://us-central1-steady-cat-400305.cloudfunctions.net/check_sensor_data)

MÉTRICAS DETALLES FUENTE VARIABLES ACTIVADOR PERMISOS REGISTROS PRUEBA

Entorno de ejecución: Python 3.11 Punto de entrada: check\_sensor\_data EDITAR

```

1 import requests
2 from datetime import datetime, timezone, timedelta
3 from sendgrid import SendGridAPIClient
4 from sendgrid.helpers.mail import Mail
5
6 def check_sensor_data(request):
7     # Detalles de ThingSpeak
8     CHANNEL_ID = "2268285"
9     READ_API_KEY = "PERM6705ZSMWUAPG"
10
11     # Rangos permitidos
12     TEMP_MIN = 20
13     TEMP_MAX = 25
14     HUM_MIN = 90
15
16     # URL para obtener datos del canal
17     url = f"https://api.thingspeak.com/channels/{CHANNEL_ID}/feeds.json?api_key={READ_API_KEY}&results=1"
18     response = requests.get(url)
19     data = response.json()
20     feed = data['feeds'][0]
21
22     # Convertir datos de temperatura y humedad a float
23     temp = float(feed['field1'])
24     humidity = float(feed['field2'])
25

```

Figura 5.19 Subida y Configuración del Código en Google Cloud Functions. Fuente: Elaboración propia.

Para activar la función de la nube en respuesta a eventos específicos, se crearon disparadores de HTTP. Esto significa que la función se ejecutará en respuesta a solicitudes HTTP GET o POST. Estos triggers se configuraron proporcionando una URL única generada por Google Cloud Functions, la cual se utiliza en el Google Cloud Scheduler para invocar la función en intervalos regulares.

Una vez desplegada la función, se procedió a quitar la conexión de corriente del prototipo para que la función de Google Cloud mandase una alerta de desconexión a como se esperaba que respondiera adecuadamente a las peticiones. La plataforma de Google Cloud proporciona opciones de visualización de registros y monitoreo de la actividad de la función, herramientas que se emplearon para validar y depurar las operaciones de esta.

Se prestó especial atención a la seguridad y gestión de acceso a la función desplegada. Para asegurar que solo las entidades autorizadas puedan activar la función, se aplicaron políticas de IAM (Identity and



Access Management) y se aseguró que las claves API y credenciales utilizadas estuviesen seguras y gestionadas adecuadamente para prevenir accesos no autorizados.

### **5.7.2 Automatización mediante Google Cloud Scheduler**

Con el código ya integrado en Google Cloud Functions ahora el siguiente paso es que se ejecute el programa cada cierto tiempo para que en el caso de se active una situación alerta lo detecte el programa y envíe la alerta al usuario. Esto es posible mediante Google Cloud Scheduler que permite la configuración y automatización de tareas que deben ejecutarse en intervalos de tiempo predefinidos, actuando como un trabajo cron totalmente gestionado y alojado en la nube. Este servicio es especialmente útil para situaciones donde las funciones o procesos deben ser ejecutados regularmente sin intervención humana, asegurando la precisión y reduciendo la posibilidad de errores operativos.

En el contexto de este proyecto, Cloud Scheduler es utilizado para invocar periódicamente la función alojada en Google Cloud Functions, que se encarga de verificar los datos del sensor, analizar si hay condiciones que requieran el envío de alertas, y si es necesario, enviar esas alertas por correo electrónico. La siguiente secuencia detalla cómo se integró Google Cloud Scheduler para automatizar el sistema de alertas:

Lo primero es crear un trabajo cron en Cloud Scheduler, definiendo una expresión cron que especifica la frecuencia con la que el trabajo debe ejecutarse. Se definió un tiempo de 15 minutos por lo que la expresión cron quedo así: `*/15 * * * *`.

- **Define el programa**

Región  
us-central1

Descripción  
Alertas del civco que se debe correr el programa cada cierto tiempo. Y si se disj

Frecuencia \*  
15 \* \* \* \* ?

Las programaciones se especifican con el formato cron de Unix. P. ej., cada minuto: "\* \* \* \* \*", cada 3 horas: "0 \*/3 \* \* \*", todos los lunes a las 9:00 a.m.: "0 9 \* \* 1". [Más información](#)

• Minuto:  
◦ Igual a 15

Zona horaria \*  
Casey Time (AWST)

Figura 5.20 Configuración del trabajo cron en Cloud Scheduler. Fuente: Elaboración propia.

El trabajo cron se configura para desencadenar la función en Google Cloud Functions, utilizando una URL HTTP que se genera automáticamente cuando se despliega la función. Este enlace asegura que, cada vez que el trabajo cron se ejecuta, la función en Cloud Functions se invoca y realiza su lógica de comprobación y alerta.

Y también Cloud Scheduler proporciona registros detallados de las ejecuciones de los trabajos cron a través de Stackdriver Logging, permitiendo la visualización y monitorización de cada invocación, y facilitando la identificación y corrección de posibles problemas.

Con este enfoque se consigue que las tareas de alertas y envío de correos electrónicos automáticos no los tenga que hacer una computadora local que esté encendida todo el tiempo para ejecutar el monitoreo de posibles alertas. Con esto se consigue que esta tarea sea ejecutada siempre a través de una máquina virtual.

## 5.8 Pruebas Iniciales de Funcionamiento

Las pruebas iniciales de funcionamiento forman parte del desarrollo del proyecto, permitiendo validar el correcto funcionamiento del sistema en su conjunto, asegurando que el hardware configurado es capaz de capturar y enviar datos a ThingSpeak de manera esperada.

En esta etapa, se procedió a encender el sistema y permitir su operación por un período extendido de dos días. Las Figura 5.21 Sistema en operación, capturando y transmitiendo datos. Fuente: Elaboración propia. y la Figura 5.22 Dashboard de ThingSpeak mostrando datos recibidos del sistema. ilustran el sistema en operación y el display mostrando datos de temperatura y humedad que están siendo enviados a ThingSpeak en tiempo real.



Figura 5.21 Sistema en operación, capturando y transmitiendo datos. Fuente: Elaboración propia.

La verificación en línea en la plataforma ThingSpeak evidenció la recepción exitosa de los datos enviados por el sistema.

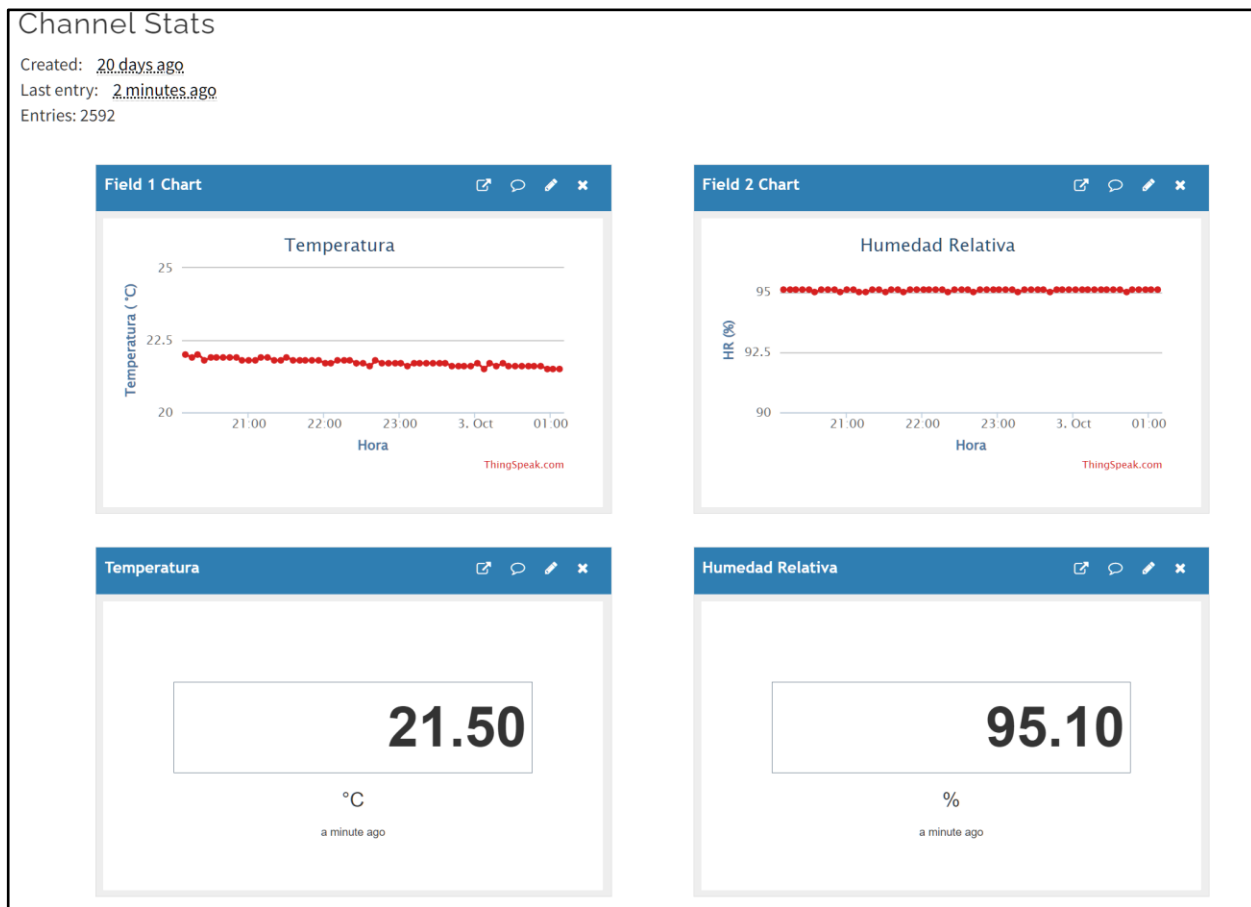


Figura 5.22 Dashboard de ThingSpeak mostrando datos recibidos del sistema. Fuente: Elaboración propia.

Adicionalmente, se confirmó que el sistema maneja adecuadamente las conexiones a Internet, reconectándose automáticamente en casos de interrupción del servicio de red, asegurando una transmisión de datos continua y confiable a la plataforma en la nube.

Esta fase de pruebas iniciales de funcionamiento permitió corroborar la robustez y eficacia del sistema desarrollado en el proyecto, sentando las bases para las evaluaciones y análisis subsecuentes detallados en las secciones siguientes.

## **Capítulo 6: Evaluación del prototipo**

### **6.1 Implementación y puesta en funcionamiento del prototipo**

#### **6.1.1 Despliegue del prototipo**

El despliegue del prototipo se ejecutó en el Centro de Investigaciones en Vivienda y Construcción (CIVCO), marcando una transición desde el sistema de monitoreo preexistente. Un aspecto primordial de este despliegue implicó el desmontaje del sistema previo sin desconexión y la instalación del nuevo prototipo.

**Desmontaje del Sistema Anterior:** La extracción del hardware preexistente se realizó con precaución para evitar alteraciones no deseadas en el entorno.

**Instalación del Nuevo Prototipo:** El nuevo sistema fue colocado, asegurando que todos los componentes, especialmente el sensor, fueran instalados de manera segura y en ubicaciones estratégicas.

**Verificación Inicial:** Se efectuó una revisión preliminar para confirmar la operatividad del prototipo. Mediante la conexión del sistema a la corriente e internet y la recepción de datos en la nube enviados por el propio sistema.

#### **6.1.2 Configuración inicial**

La configuración inicial del prototipo se realizó garantizando una base sólida para la obtención de datos precisos sobre la humedad relativa del cuarto húmedo en el Centro de Investigaciones en Vivienda y Construcción (CIVCO).

Al comenzar, se debió garantizar una correcta conexión de todo el hardware del prototipo. Con especial énfasis en el sensor de humedad relativa, se revisaron de manera visual las conexiones, garantizando que los dispositivos estuvieran operativos y listos para transmitir datos.

La ubicación del sensor en el cuarto húmedo fue un aspecto crítico. Dado que los aspersores del cuarto están constantemente funcionando, se buscó un sitio que, si bien estuviera influenciado por esta fuente de humedad, no quedara tan cerca como para saturar el sensor. Se eligió una ubicación que reflejara

adecuadamente la humedad del ambiente general, considerando también las áreas con buena circulación de aire y evitando zonas con posibles interferencias. Para conocer más acerca de la ubicación del sensor se puede consultar la sección 6.1.4 Justificación de la ubicación del sensor.

En el ámbito del software, se programó el intervalo de mediciones para que se realicen lecturas cada cinco minutos. Los cinco minutos es para que en la sección 6.4 Análisis estadístico se pueda tener la mayor cantidad de datos sin saturar el sistema de almacenamiento de la nube. Además, se estableció la comunicación con la plataforma ThingSpeak, asegurando que los datos se enviaran, recibieran y almacenaran correctamente en la plataforma en línea.

Con estos ajustes iniciales, el prototipo quedó adecuadamente configurado, listo para monitorear las condiciones del cuarto húmedo y contribuir con datos valiosos para la investigación.



Figura 6.1 Sistema en funcionamiento dentro de las instalaciones del Centro de Investigaciones en Vivienda y Construcción (CIVCO). Fuente: Elaboración propia

### 6.1.3 Consideraciones eléctricas y diagrama unifilar

Dentro del contexto de un sistema eléctrico, es importante tener en cuenta las contingencias que pueden surgir debido a interrupciones inesperadas en el suministro eléctrico. La continuidad de la energía es fundamental para garantizar el funcionamiento ininterrumpido de sistemas críticos, y, por ende, es una de las preocupaciones mayores para el caso de este trabajo.

El edificio donde está instalado el sistema está preparado para enfrentar tales contingencias, gracias a la presencia de una planta de emergencia. Esta planta no sólo garantiza el suministro eléctrico general del edificio, sino que también tiene en cuenta circuitos específicos, como el circuito TP/5. Es en este circuito donde se encuentra el tomacorriente al que está conectado el sistema. Que es un tomacorriente perteneciente al circuito mencionado que lo conforman también más tomacorrientes de habitaciones y laboratorios del edificio.



Figura 6.2 Planta de emergencia garantizando suministro eléctrico al edificio y circuitos específicos.  
Fuente: Elaboración propia.

La relevancia de este diseño radica en la importancia de mantener el sistema operativo incluso en el caso de fallos eléctricos. La planta de emergencia entra en acción de forma automática ante un corte de luz, en un tiempo máximo de 1 minuto. Asegurando que el circuito TP/5, y por ende el sistema, continúe recibiendo la energía necesaria para su funcionamiento.

El diagrama unifilar ilustra la configuración eléctrica del edificio. La fuente principal proviene de la red primaria de JASEC con una capacidad de 3.375 KVA y una tensión de 416 kV. Esta alimentación se divide en dos tensiones: 208 VAC y 120 VAC.

A partir del interruptor principal de 600 A con fusibles de 400 A, la distribución se realiza hacia tres tableros:

TABLERO C: Alimentación de iluminación con conductores de 4 #6 TW.

TABLERO P: Alimentación de tomacorrientes y equipos con conductores de 4 #2 TW.

TABLERO M: Alimentación específica para equipos con conductores de 3 #3/0 THW y 1 #2 TW.

Dentro del TABLERO P se encuentra el circuito TP/5. Este circuito tiene una capacidad de Polos/Amp de 1/30 y una carga de 2400 Watts. Está alimentado por conductores de 3 #10.

En caso de interrupción en la red primaria, el edificio cuenta con una planta de emergencia que respalda al circuito TP/5, entre otros. Esto asegura la alimentación al tomacorriente donde está conectado el sistema en cuestión.



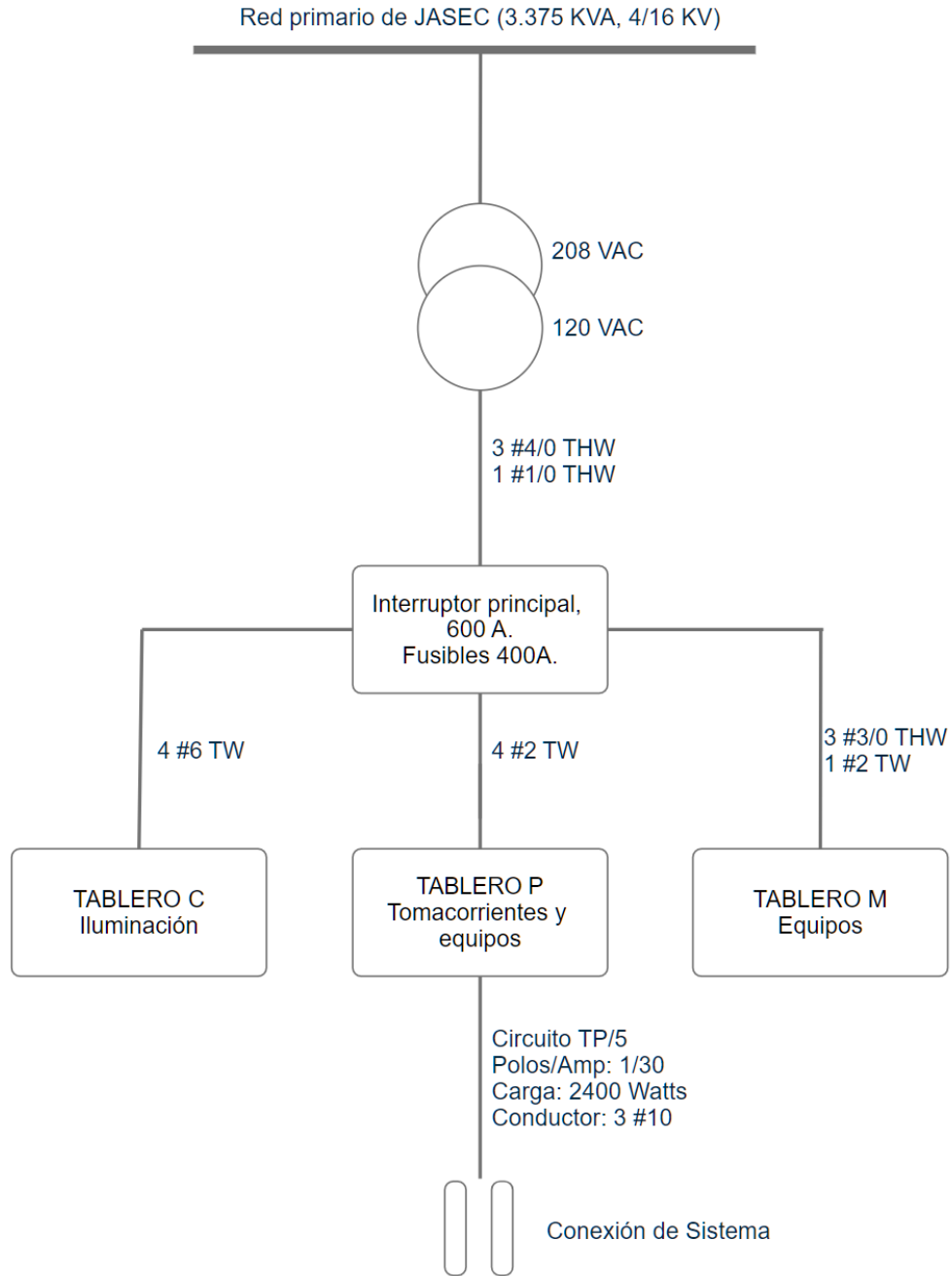


Figura 6.3 Diagrama unifilar de distribución eléctrica del edificio con detalles de alimentación y capacidad. Fuente: Elaboración propia.

#### 6.1.4 Justificación de la ubicación del sensor

La elección estratégica del emplazamiento del sensor en el cuarto húmedo es un aspecto determinante para garantizar la precisión y fiabilidad de las mediciones de humedad relativa. La ubicación del sensor fue determinada considerando múltiples factores técnicos y ambientales, detallados a continuación:

1. Proximidad a los aspersores: Se debe mantener una distancia prudencial entre el sensor y los aspersores, la utilizada fue una distancia de 0,8 m. Una proximidad excesiva puede causar una saturación del sensor debido a la acumulación directa de agua, lo que podría arrojar lecturas imprecisas o incluso dañar el equipo. Por lo tanto, la ubicación del sensor se determinó para garantizar una medición correcta de la humedad relativa sin la interferencia directa de las gotas de agua de los aspersores.

2. Altura: La ubicación del sensor a una altura determinada del suelo busca evitar interferencias que puedan surgir del contacto directo con posibles fuentes de calor o frío en el nivel del suelo. Además, a medida que el aire caliente tiende a ascender y el aire frío a descender, una ubicación elevada permite obtener una lectura más representativa del ambiente general del cuarto (Isobox Systems, 2023). Por lo que se escogió una altura de 2,7 m tomando en cuenta que la altura del cuarto húmedo es de 2,8 m. Esta altura escogida está por encima de los aspersores por lo que se protege de una lectura errónea en el caso de que una gota se introduzca de manera directa por los aspersores.

3. Espacio cerrado: El cuarto húmedo, al ser un espacio cerrado, puede presentar variaciones en la distribución de la humedad y temperatura. Pero de temperatura no existe una diferencia considerable al tratarse de un espacio cerrado. (Elsalamawy, Mohamed, & Kamal, 2019).

4. Comparación con el sistema preexistente: El sensor fue emplazado en una posición similar al sistema de monitoreo anterior. Esto permite una comparación más directa entre los datos recopilados por ambos sistemas.

Se buscó obtener mediciones de humedad relativa confiables. Es importante recordar que cualquier variación en la posición del sensor puede influir en las lecturas, por lo que la colocación óptima está para con mediciones certeras. .

## 6.2 Monitoreo y registro de datos

### 6.2.1 Proceso de monitoreo

El sistema de monitoreo ambiental diseñado para el Centro de Investigaciones en Vivienda y Construcción (CIVCO) estuvo operando de manera continua, registrando y enviando datos a la plataforma ThingSpeak durante el periodo de 20 días. Las gráficas presentadas reflejan el registro histórico de dos parámetros: temperatura y humedad relativa.

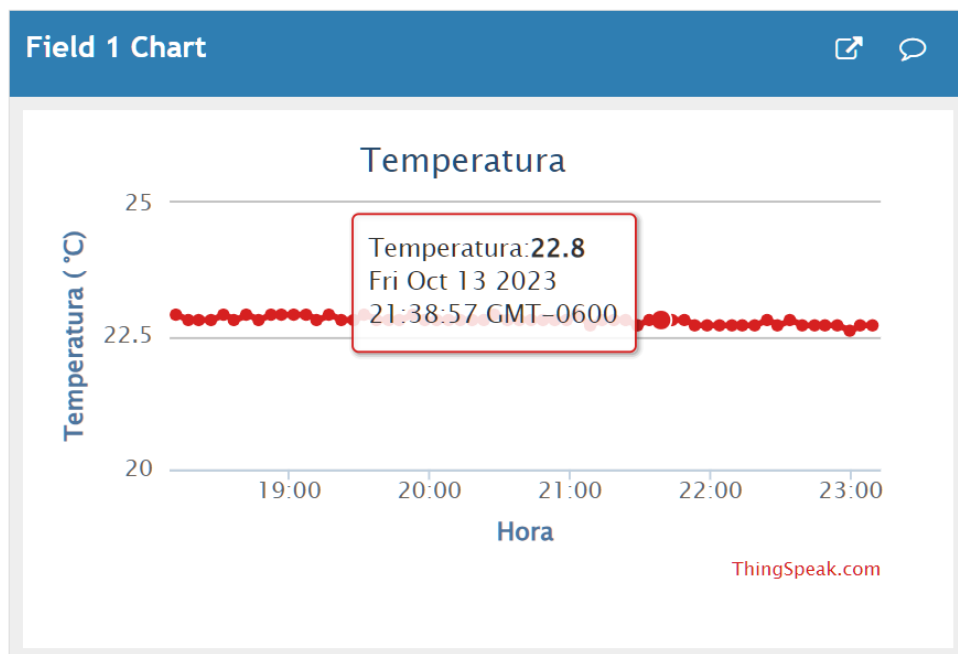


Figura 6.4 Registro de temperatura en el Centro de Investigaciones en Vivienda y Construcción (CIVCO) través de ThingSpeak. Fuente: Elaboración propia.

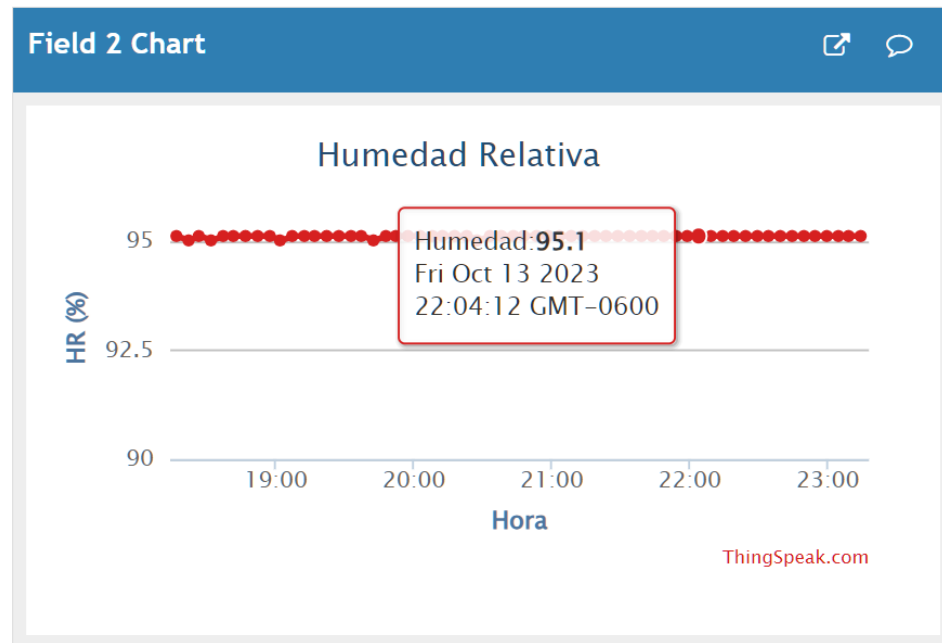


Figura 6.5 Registro de humedad relativa en el Centro de Investigaciones en Vivienda y Construcción (CIVCO) a través de ThingSpeak. Fuente: Elaboración propia.

Las imágenes proceden directamente de ThingSpeak y muestran los datos registrados durante periodos específicos. A partir de estas gráficas, se puede observar la capacidad del sistema para transmitir de manera constante la información recopilada.

Es relevante mencionar que la confiabilidad y precisión de estos datos fueron evaluados posteriormente mediante ensayos de calibración realizados por un laboratorio dedicado a esto (Lorek & Majewski, 2018; Regtien, 2011). Por el momento, estas gráficas sirvieron como evidencia de la operación continua del sistema y su capacidad para registrar y transmitir datos.

## 6.2.2 Herramientas de recuperación de datos

Dentro del monitoreo ambiental que se realiza en el Centro de Investigaciones en Vivienda y Construcción (CIVCO), es fundamental poder acceder y recuperar los datos recolectados en tiempo real y de

manera histórica. En esta sección, se presenta la eficacia de un programa desarrollado para tal fin, evidenciando su funcionamiento en situaciones reales.

### 6.3 Pruebas de sistema de reportes y alertas

Una vez que los datos son recolectados y almacenados en la nube, el programa desarrollado se encarga de recuperarlos de forma programada. Esta función garantiza que la información sea accesible para análisis y reportes en cualquier momento.

Registro histórico: Esta función permite enviar correos periódicos con un resumen de los datos recopilados durante un período determinado.

Para el caso de envío de alertas se muestra la siguiente imagen. Esta imagen muestra una alerta activada cuando el sistema no transmite datos en un lapso de 15 minutos. Y fue enviada en una situación real donde el sistema estuvo por un tiempo desconectado sin enviar datos.

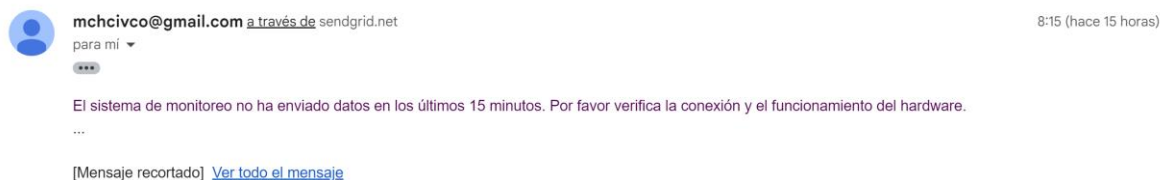


Figura 6.6 Notificación por correo ante la falta de transmisión de datos del sistema.

Con esta notificación sugiere al usuario verificar la conexión y el estado del hardware, ya que la falta de transmisión podría señalar problemas con la conexión, fallos del sistema o interrupciones en el servicio de Thingspeak

El programa ha demostrado ser una herramienta eficiente en la gestión y recuperación de datos. Los correos de alerta y registro histórico son testamentos de su efectividad y capacidad para operar en condiciones reales. Parte del registro histórico puede ser revisado en Apéndice D Ilustración de parte recortada del reporte

de monitoreo en archivo de Excel. donde se muestra una parte del registro de monitoreo que realizó el sistema durante el tiempo de los 20 días completos que estuvo realizando su función en el laboratorio de ensayos donde está el cuarto húmedo del CIVCO.

## **6.4 Análisis estadístico**

### **6.4.1 Recolección de datos**

La recolección de datos está diseñada para capturar información relevante emitida por el prototipo. Inicialmente, se establece una conexión con el canal de ThingSpeak para extraer datos relacionados con temperatura y humedad. A través de esta plataforma, los datos se recopilan y almacenan durante un período de 30 días.

Una vez extraídos, los datos se procesan y se adaptan a la zona horaria de Costa Rica. Estos registros se estructuran en entradas que contienen fecha, hora, temperatura y humedad. Esta información se consolida y organiza en un formato textual, que ofrece una descripción de las mediciones a lo largo del intervalo de fechas de recopilación, que es un rango todos los días del mes.

Para facilitar su consulta y análisis, la información se almacena en dos formatos diferentes: un archivo de texto (.txt) y una hoja de cálculo (.xlsx). El nombre de estos archivos se genera automáticamente con la fecha actual para una fácil identificación.

Finalmente, una vez generados estos archivos, se envían por correo electrónico al usuario a través del servicio SendGrid, este correo electrónico puede ser consultado en Apéndice C Correo electrónico recibido por el programa de reportes de monitoreo. El correo incluye los archivos adjuntos con los registros detallados y se notifica al usuario para su revisión. La automatización de este proceso asegura una recolección y entrega eficiente de los datos, permitiendo una monitorización constante del prototipo.

## 6.4.2 Análisis estadístico

Se llevó a cabo un análisis estadístico de los datos recolectados durante el período de estudio, que fue un plazo de 20 días que estuvo el sistema trabajando en la situación real de trabajo. Se calculan diversas estadísticas descriptivas, como medias, medianas, desviaciones estándar y otros valores relevantes que ofrecen una visión clara y objetiva de la distribución y comportamiento de los datos. De forma de esclarecer mejor los resultados se muestra cada estadística descriptiva por cada día de los datos que fue recolectando cada 5 minutos que el sistema censaba y enviaba el dato al almacenamiento de la plataforma de Thingspeak.

## 6.4.3 Resultados y documentación

Tabla 6.1 Resumen estadístico diario de temperatura y humedad con mediana, moda y desviación estándar.

Día	Fecha	mediana temp. (°C)	moda temp (°C)	moda humedad (%)	Desviación estandar temp (°C)	Desviación estandar hum. (%)
1	2023-09-26	21,7	21,6	94,5	20,61158879	626,2562617
2	2023-09-27	21,5	21,8	94,4	31,29793103	263,0297318
3	2023-09-28	21,3	20,8	94,3	32,756	193,8286316
4	2023-09-29	21,5	21,8	94,5	33,04196491	11,02196491
5	2023-09-30	21,5	22	94,8	36,65458042	2,717762238
6	2023-10-01	21,7	21,4	94,9	23,53312281	0,573122807
7	2023-10-02	21,3	20,8	94,3	32,70996491	194,0724211
8	2023-10-03	21,6	21,4	95,1	26,27312281	1,780491228
9	2023-10-04	21,5	21,4	95,2	6,871438596	3,792280702
10	2023-10-05	21,4	22	95,5	46,85312281	61,08442105
11	2023-10-06	21,8	22,3	94,2	38,2794386	17,89368421
12	2023-10-07	22	22,4	93,7	28,34785965	14,1682807
13	2023-10-08	22,1	22,5	94,2	38,61263158	20,69873684
14	2023-10-09	22,2	21,9	93,8	23,7282807	16,51207018
15	2023-10-10	22,4	22,5	94,3	8,777754386	9,189122807
16	2023-10-11	21,9	21,9	94,1	15,82727915	22,72826855
17	2023-10-12	21,7	21,4	94,2	8,293333333	32,63674797
18	2023-10-13	22,7	22,8	95,1	44,14994565	4,668695652
19	2023-10-14	22,4	22,6	95,2	14,94732394	1,636161972
20	2023-10-15	22,4	22,6	95,2	21,08732394	2,229119718
21	2023-10-16	21,8	21,7	95,3	7,808432836	0,517089552

Nota: Elaboración propia.

La tabla anterior es una representación de las mediciones de temperatura y humedad tomadas durante el periodo ya explicado. Cada fila se dedica a un día específico, y las columnas detallan diferentes aspectos de estas mediciones. La columna "Fecha" indica el día al que corresponden las mediciones. La "Mediana de

temperatura" muestra el punto medio de todas las temperaturas registradas en ese día, dividiendo los datos en dos mitades iguales. A su lado, las columnas de "Moda" indica los valores que se presentaron con mayor frecuencia, tanto para la temperatura como para la humedad. Estas modalidades nos dan una idea de los valores más típicos o recurrentes.

Por otro lado, las columnas de "Desviación estándar" proporciona una imagen clara de la variabilidad de los datos. En el caso de la temperatura y la humedad, estas desviaciones explican cómo se dispersaron las mediciones alrededor de su promedio. Una desviación estándar que se aleje del promedio indica fluctuaciones notables, las cuales podrían ser el resultado de factores externos, variabilidades naturales o incluso inconsistencias con el equipo de medición. Por otro lado, una desviación estándar cercana al promedio sugiere estabilidad en las mediciones de ese día.

La utilidad de esta tabla radica en su capacidad para ofrecer, un vistazo, tanto una visión general de las condiciones diarias como una comprensión profunda de la confiabilidad y variabilidad de los datos. Estos detalles son para la interpretación correcta de los resultados y para fundamentar decisiones informadas basadas en ellos.



Tabla 6.2 Resumen estadístico diario de temperatura y humedad con mínimo, máximo y promedio.

Día	Fecha	media temp. (°C)	media humedad (%)	minimo temp (°C)	minimo humedad (%)	maximo temp. (%)	maximo humedad (%)
1	2023-09-26	21,82336449	93,88224299	21,3	76,3	24,2	94,6
2	2023-09-27	21,4908046	93,79118774	20,7	85,9	22,7	94,5
3	2023-09-28	21,22	94,04631579	20,7	84,3	22,3	94,4
4	2023-09-29	21,47298246	94,37298246	20,8	93,4	22	94,7
5	2023-09-30	21,56188811	94,73356643	21	94,5	22,1	94,9
6	2023-10-01	21,75824561	94,89508772	21,3	94,8	22,3	95
7	2023-10-02	21,21964912	94,05052632	20,7	84,3	22,3	94,4
8	2023-10-03	21,66175439	95,11017544	21,1	94,9	22,2	95,3
9	2023-10-04	21,51859649	95,25087719	21,2	95,1	21,8	95,5
10	2023-10-05	21,49508772	94,99052632	20,7	92,5	22,1	95,6
11	2023-10-06	21,90140351	93,97894737	21,4	93,5	22,4	94,3
12	2023-10-07	22,05403509	93,93087719	21,5	93,5	22,6	94,3
13	2023-10-08	22,12631579	93,88210526	21,5	93,3	22,7	94,3
14	2023-10-09	22,19754386	93,7845614	21,7	93,4	22,8	94,3
15	2023-10-10	22,40280702	94,16491228	21,9	93,9	22,8	94,5
16	2023-10-11	21,85088339	94,2975265	21,4	93,9	22,3	94,9
17	2023-10-12	21,66666667	94,54065041	21,2	94	22,1	95,5
18	2023-10-13	22,40054348	95,16304348	21,2	95	22,9	95,8
19	2023-10-14	22,36901408	95,13908451	21,9	95	22,7	95,3
20	2023-10-15	22,33098592	95,14823944	21,8	95	22,8	95,3
21	2023-10-16	21,91268657	95,28432836	21,5	95,1	22,4	95,4

Nota: Elaboración propia.

La Tabla 6.2 Resumen estadístico diario de temperatura y humedad con mínimo, máximo y promedio. es un análisis de las mediciones diarias de temperatura y humedad. Las dos columnas, "Media de temperatura" y "Media de humedad", representan los valores promedio de las mediciones tomadas durante ese día, ofreciendo un panorama general de las condiciones ambientales.

Pasando a un enfoque más específico, la Tabla 6.2 Resumen estadístico diario de temperatura y humedad con mínimo, máximo y promedio. también proporciona los valores extremos registrados. Las columnas "Mínimo de temperatura" y "Mínimo de humedad" indican las condiciones más bajas alcanzadas, lo que podría ser particularmente útil para identificar si hubo momentos de frío extremo o de baja humedad relativa que podrían afectar a estar fuera de lo indicado por la norma para el proceso de curado de concreto. Por otra parte, las columnas "Máximo de temperatura" y "Máximo de humedad" muestran los picos más altos registrados, momentos en los que las condiciones podrían haber sido demasiado cálidas o húmedas.

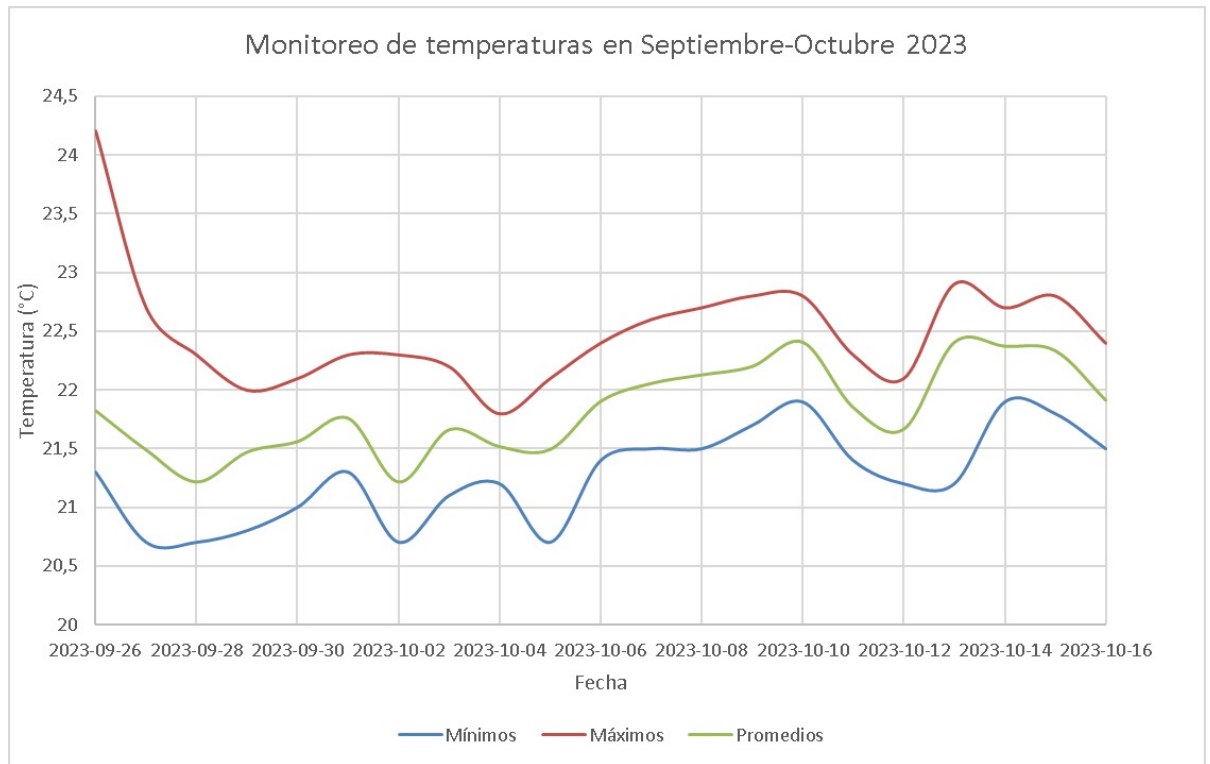


Figura 6.8 Variaciones de temperatura diarias con mínimos, máximos y promedios en septiembre-octubre del 2023. Fuente: Elaboración propia.

El gráfico presentado revela el seguimiento de las temperaturas registradas en el cuarto húmedo durante un período de tres semanas, abarcando desde fines de septiembre hasta mediados de octubre de 2023.

Para cumplir con las normativas establecidas, la temperatura del cuarto húmedo debe fluctuar entre los 20°C y 25°C. Al observar las líneas que representan los valores mínimos y máximos registrados se aprecia que el mínimo estuvo en 20,7 °C de temperatura y 76,3 % de humedad relativa. Por otra parte, el máximo de temperatura estuvo en 24,4 y de humedad relativa estuvo en 95,6 %. Es evidente que el sistema de monitoreo ha logrado mantenerse en gran medida dentro del rango óptimo. Las veces que estuvo fuera del rango permitido fue debido a la instalación que estuvieron apagados los aspersores y la puerta abierta, o es ocasiones donde estuvo la puerta abierta.

Este monitoreo de las temperaturas no solo cumple con los estándares requeridos, sino que también garantiza la integridad y calidad de los procesos y materiales alojados en el cuarto húmedo.

Es importante recalcar que este proyecto se enfocó solo en el monitoreo de la sala, no en control. Es decir que con la implementación del sistema ocurre que si llegan días muy fríos o por el otro lado muy calientes que lleven las temperaturas dentro del cuarto fuera del rango admitido por la norma. Este solo podrá alertar sobre la situación que sucede dentro del cuarto con respecto a la temperatura.

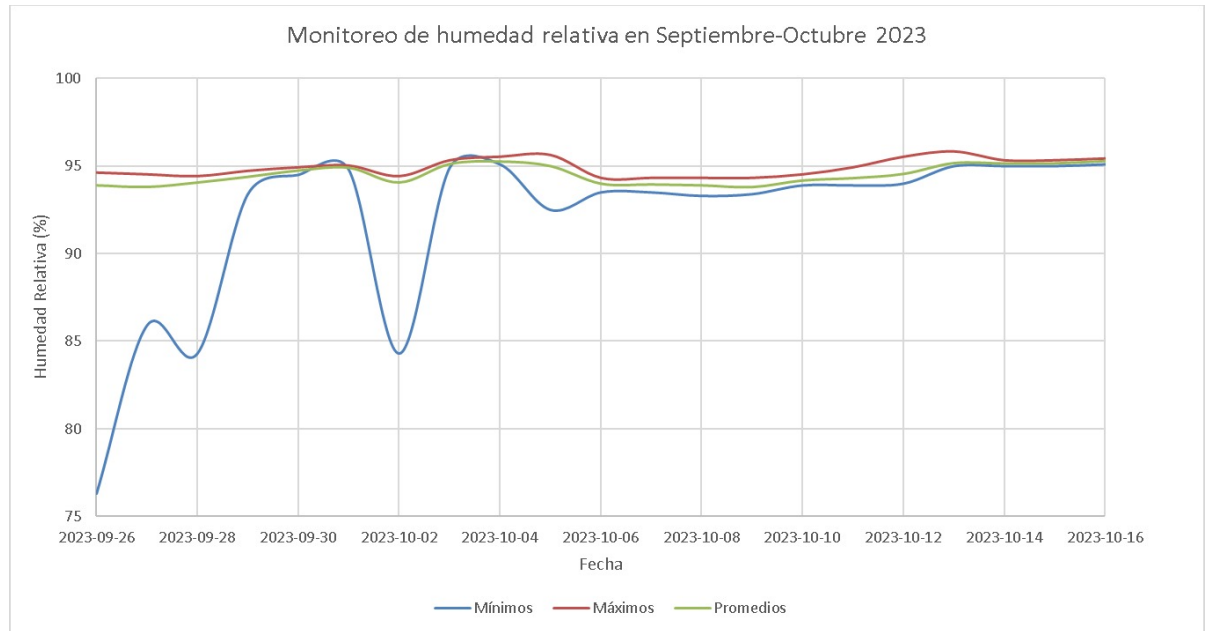


Figura 6.9 Variaciones de humedad relativa diarias con mínimos, máximos y promedios en septiembre-octubre del 2023. Fuente: Elaboración propia.

La Figura 6.9 Variaciones de humedad relativa diarias con mínimos, máximos y promedios en septiembre-octubre del 2023 tiene un gráfico que muestra el monitoreo de la humedad relativa en el cuarto de curado de probetas de concreto. Según las normas establecidas, es esencial que la humedad relativa se mantenga entre el 95% y 100% para garantizar condiciones óptimas de curado (ASTM International, 2013).

En una inspección inicial, se destaca que los valores promedio, denotados por la línea verde, mantienen una consistencia razonable dentro del rango ideal, gravitando en su mayoría alrededor del 97%. Sin embargo, se observan dos detalles que merecen especial atención:

Máximos Aproximándose al Límite Inferior: En varios días, la línea roja, que representa los valores máximos, se acerca peligrosamente al límite inferior del 95%. Este comportamiento plantea dos posibles escenarios:

El sensor puede estar presentando inexactitudes al medir la humedad relativa, proporcionando lecturas ligeramente inferiores a la realidad.

Existe la posibilidad de que el cuarto no esté proporcionando el ambiente óptimo, posiblemente debido a fallos en el sistema de humidificación o a factores externos que afecten la humedad relativa dentro del cuarto.

Descensos Pronunciados en los Valores Mínimos: Se observan notables caídas en la humedad relativa en ciertos días, lo que podría ser motivo de preocupación. Sin embargo, gracias a la información adicional proporcionada, entendemos que estos descensos se deben a la pausa temporal de los aspersores y a la apertura del cuarto por cuestiones propias de laboratorio. Esta práctica, aunque necesaria, puede comprometer temporalmente las condiciones ideales de curado.

## **6.5 Calibración del equipo**

Para garantizar que las mediciones realizadas por el sistema de monitoreo desarrollado sean precisas y confiables, es esencial someter el equipo a un proceso de calibración certificada. Dado que el objetivo principal del proyecto es desarrollar un sistema que cumpla con los requisitos de la norma ISO/IEC 17025 para la acreditación del laboratorio de ensayos, la calibración no es solo una recomendación, sino un requisito esencial.

### **6.5.1 Proceso de Calibración Certificada**

La norma ISO 17025 estipula que las mediciones deben estar respaldadas por un laboratorio certificado de calibración. Este proceso implica:

**Selección del Laboratorio:** Es fundamental elegir un laboratorio que esté certificado bajo la norma ISO 17025. Esta certificación asegura que el laboratorio cumple con los estándares internacionales en términos de competencia, operaciones y calidad del servicio.

**Envío del Equipo:** El sistema de monitoreo se enviará al laboratorio seleccionado, donde será sometido a una serie de pruebas y procedimientos de calibración.

**Documentación y Certificación:** Al finalizar el proceso de calibración, el laboratorio proporcionará la documentación y certificación que demuestra que el equipo ha sido calibrado de acuerdo con los estándares ISO 17025. Esta certificación es necesaria para la acreditación del laboratorio de ensayos del CIVCO.

### **6.5.2 Importancia de la Calibración para CIVCO**

Para que el Centro de Investigaciones en Vivienda y Construcción (CIVCO) obtenga la acreditación de su laboratorio de ensayos, es imperativo que todos sus equipos de monitoreo estén debidamente calibrados. Esto garantiza la precisión y confiabilidad de las mediciones, asegurando que los resultados obtenidos durante los ensayos sean válidos y reconocidos internacionalmente.

### **6.5.3 Resultado de la calibración del equipo**

A pesar de que el sistema de monitoreo fue enviado a METCAL, un laboratorio certificado conforme a la norma ISO 17025, la calibración del equipo no se completó dentro del plazo del proyecto. Esto ha dado lugar a una situación en la que, aunque el prototipo ha funcionado adecuadamente dentro de las instalaciones del CIVCO, la falta de la documentación certificada impide verificar plenamente la fiabilidad de los datos recogidos por el sistema durante su operación.

Por consiguiente, el objetivo específico número 3 no se ha logrado en su totalidad. Aunque el prototipo fue evaluado dentro de la cámara húmeda del CIVCO, y se realizaron pruebas que muestran su capacidad para el monitoreo y la generación de informes, la ausencia del resultado de calibración certificada

impide afirmar con certeza la exactitud y precisión de las mediciones de temperatura y humedad. En consecuencia, el indicador de logro, que requiere un reporte de pruebas con datos recopilados y almacenados en tiempo real que confirmen la precisión del prototipo, no se ha podido cumplir en su integridad.

## **6.6 Observaciones de la evaluación**

A partir de la implementación y evaluación del prototipo en el Centro de Investigaciones en Vivienda y Construcción (CIVCO), se han derivado las siguientes observaciones:

La configuración inicial del prototipo permitió establecer una base para el monitoreo de la humedad relativa en el cuarto húmedo. Las consideraciones eléctricas garantizan un funcionamiento continuo del sistema, minimizando las interrupciones.

La ubicación estratégica del sensor, lejos de los aspersores y a una altura adecuada, ha sido un aspecto de alta relevancia para obtener mediciones correctas y así evitar interferencias, tanto por gotas directas de agua, como presentaba el sistema de monitoreo actual que era inundado por los aspersores y arrojaba lecturas erróneas.

A través de herramientas estadísticas, se ha proporcionado una visión detallada de las condiciones diarias, destacando la variabilidad de los datos.

A pesar de que los valores promedio de humedad relativa se mantienen consistentemente cerca del rango ideal, se han observado máximos que rozan el límite inferior y descensos pronunciados en los valores mínimos. Estas observaciones pueden ser indicativas de posibles inexactitudes del sensor o fallos en el sistema de humidificación. Además, las interrupciones en la humidificación debido a la pausa de los aspersores y la apertura del cuarto pueden comprometer temporalmente las condiciones de curado.

## **Capítulo 7: Análisis comparativo de sistemas de monitoreo**

La elección adecuada de herramientas y plataformas puede marcar una diferencia sustancial tanto en el rendimiento del sistema como en el presupuesto destinado a su implementación. Específicamente, se evaluarán los servicios de ThingSpeak, SendGrid y Google Cloud, pilares fundamentales que, en conjunto, permiten la operatividad de este sistema.

La esencia de este análisis radica en comparar las ventajas, limitaciones y costos adquisitivos de la propuesta con alternativas presentes en el mercado. Se considerarán aspectos como el almacenamiento de datos en la nube, la capacidad de monitoreo en tiempo real desde dispositivos móviles, la generación de reportes de monitoreo y la implementación de sistemas de alertas.

Dado que el mercado ofrece una amplia gama de soluciones, para propósitos de este estudio se han seleccionado dos alternativas reconocidas por su confiabilidad y por ofrecer funcionalidades similares a las requeridas por el Centro de Investigaciones en Vivienda y Construcción (CIVCO) en un sistema de monitoreo. A través de este análisis comparativo, se busca esclarecer cuál es la mejor opción en términos de coste, funcionalidad y escalabilidad, brindando así una base sólida para tomar una decisión acertada sobre qué equipo conviene adquirir. Y si el sistema desarrollado puede competir con el mercado o se ve limitado.

### **7.1 Identificación de soluciones comerciales**

La implementación de sistemas de monitoreo en cuartos húmedos no puede ser realizada por cualquier dispositivo de medición, demanda una solución integral que combine precisión, funcionalidad y adaptabilidad a las condiciones ambientales. A estas necesidades, se realizó una búsqueda de soluciones comerciales disponibles que se asemejen o superen las funcionalidades propuestas por el prototipo desarrollado.

### **7.1.1 Investigación de soluciones**

Ante la demanda de herramientas tecnológicas en los laboratorios de ensayos, múltiples empresas han desarrollado sistemas de monitoreo específicos para cuartos húmedos. Las características deseables en estas soluciones son presentadas a continuación.

**Display Integrado:** Facilita la lectura instantánea de las condiciones ambientales, permitiendo intervenciones rápidas si es necesario.

**Monitoreo de Humedad y Temperatura:** Fundamental para cuartos húmedos porque la norma ASTM 511 así lo solicita para el proceso de curado.

**Sensor de Larga Distancia:** A diferencia de sistemas compactos, un sensor de larga distancia garantiza que el dispositivo principal esté protegido de las condiciones de humedad, al tiempo que monitorea adecuadamente el espacio.

**Conexión a Internet:** La capacidad de enviar datos en tiempo real a una plataforma en línea para su análisis, visualización y generación de reportes.

**Alertas Instantáneas:** Fundamental para prevenir problemas a tiempo, especialmente en ambientes donde las variaciones pueden causar daños.

**Calibración Certificada:** Cumplir con la norma 17025 asegura que las mediciones estén respaldadas por un laboratorio certificado de calibración.

### **7.1.2 Selección de soluciones comparables**

Entre las opciones identificadas, el primer producto para comparar es el HMT120 de Vaisala, junto con su software viewLinc. Este sistema presenta una pantalla integrada para ofrecer una visualización clara e inmediata de las condiciones monitorizadas. La herramienta cumple con las demandas de monitoreo de humedad y temperatura, en línea con lo estipulado por la norma ASTM 511. El diseño del equipo incorpora un sensor remoto, protegiendo así el dispositivo principal y garantizando un monitoreo efectivo del ambiente



de interés. La conectividad y capacidad de transmitir datos en tiempo real por medio de viewLinc se presenta como una ventaja comparada con otros sistemas del mercado. Este permite un seguimiento y análisis detallado en cualquier momento. Asimismo, las alertas inmediatas y la calibración certificada van acorde a lo que la norma ISO/IEC 17025 solicita.



Figura 7.1 Sistema Vaisala modelo HMT120. Vaisala (2023)

Entre las características que posee este sistema de monitoreo se presentan en la siguiente tabla.

Tabla 7.1 Características del modelo Vaisala HMT120.

Atributo	Descripción
Nombre del modelo	HMT120
Tipo de instrumento	Transmisor de humedad y temperatura
Rango de medición de humedad	0% a 100% Humedad relativa (dependiendo de la sonda)
Rango de medición de temperatura	-40°C a +180°C (dependiendo de la sonda)
Precisión de medición de humedad	±1.5% HR (en el rango de 0% a 90% HR)
Precisión de medición de temperatura	±0.2°C (a 20°C)
Salida	4 a 20 mA o 0 a 20 mA
Alimentación	10 a 35 VDC
Material de la carcasa	Acero inoxidable
Aplicaciones Típicas	HVAC, instalaciones industriales, museos, etc

Nota: Elaboración propia en base a la hoja de datos del modelo Vaisala HMT120.

Se observa que los rangos de medición están dentro de lo requerido para el monitoreo que se requiere. Además de las funciones ya presentadas queda este sistema con alcance muy parecido al sistema desarrollado.

El siguiente sistema presentado para la comparación de este capítulo es de la misma marca Vaisala. Es el modelo Indigo202 transmisor digital. La elección de otra vez escoger un producto del mismo fabricante es debido a la popularidad de este en el mercado nacional. Y más específico en las preferencias actuales del propio Centro de Investigaciones en Vivienda y Construcción (CIVCO) para sus sistemas de monitoreo. En

las instalaciones del laboratorio se encuentran varios sistemas ya del mismo fabricante. Por lo que según intereses de este se tienen tendencia a trabajar con este tipo de productos.

Además de que este modelo trabaja de igual forma con Viewlinc como software del fabricante para aplicaciones de monitoreo remoto, gráficos de tendencias, reportes históricos y alertas. A diferencia de otros fabricantes, Vaisala ofrece una gama más amplia de funcionalidades de monitoreo que se alinean con las necesidades y objetivos del Centro en relación con su sistema de monitoreo en la cámara húmeda.

El Indigo202 se distingue por ser más avanzado el otro modelo. Funciona como un dispositivo anfitrión con comunicación Modbus y se adapta a una variedad de sondas Indigo. Entre estas sondas, se incluyen las de la serie GMP para medición de CO<sub>2</sub>, las de la serie HPP para vapor de peróxido de hidrógeno, las de la serie HMP para humedad y temperatura y, por último, las de la serie TMP específicas para temperatura. Estas sondas pueden conectarse directamente al transmisor o mediante un cable especial si se prefiere una conexión remota.



Figura 7.2 Sistema Vaisala modelo Indigo202. (Vaisala 2023)

Además, el Indigo202 presenta una interfaz de usuario basada en un navegador inalámbrico, lo que facilita la configuración y la visualización de datos. En la siguiente se muestran las características principales de este modelo.

Tabla 7.2 Características del modelo Vaisala Indigo202.

Atributo	Descripción
Nombre del modelo	Indigo202
Tipo de instrumento	Transmisor con sondas intercambiables
Rango de medición de humedad	Depende de sonda seleccionada
Rango de medición de temperatura	Depende de sonda seleccionada

Precisión de medición de humedad	No especificado
Precisión de medición de temperatura	No especificado
Salida	RS-485 Modbus RTU
Alimentación	15 a 30 VDC
Material de la carcasa	PC/ABS plástico
Aplicaciones Típicas	Uso de interiores

Nota: Elaboración propia en base a la hoja de datos del modelo Vaisala Indigo202.

Para el sistema Indigo202 de Vaisala, se consideró su capacidad de adaptabilidad y precisión. Cabe mencionar que los dispositivos Indigo no incluyen, por defecto, una sonda, por lo que es necesario adquirir una por separado. De todas las opciones disponibles en el catálogo de Vaisala, se optó por la sonda HMP3, capaz de censar la temperatura y humedad relativa a la vez.

La sonda HMP3 de Vaisala es versátil y está diseñada específicamente para diversos procesos industriales. Una de sus características más destacadas es la estructura de la sonda que permite reemplazar el sensor sin necesidad de herramientas.

En cuanto a las especificaciones, la cabeza de la sonda tiene una temperatura operativa que va entre  $-40\text{ }^{\circ}\text{C}$  y  $+120\text{ }^{\circ}\text{C}$ , mientras que el cuerpo de la sonda opera en un rango de  $-40\text{ }^{\circ}\text{C}$  a  $+80\text{ }^{\circ}\text{C}$ .

No obstante, un factor decisivo en la elección de esta sonda es que cada unidad viene con un certificado de calibración estándar ISO 9001 directamente de fábrica. Para laboratorios en busca de la certificación ISO/IEC 17025, este aspecto es de suma relevancia.

Ahora, ya se mencionó que ambos sistemas pueden trabajar con el mismo software de monitoreo Viewlinc. Este software ofrece una amplia cantidad de funciones, es compatible con diversos registradores de datos Vaisala y dispositivos Modbus, ya sean cableados o inalámbricos. Su capacidad para conectar y gestionar desde instalaciones pequeñas con unos pocos puntos de medición hasta sistemas amplios con miles

de ubicaciones. Una de sus características destacadas es la garantía de integridad de los datos, proporcionando un rastro de auditoría seguro, controles de acceso y cifrado avanzado.

El software también se distingue por su versatilidad de idiomas, con versiones disponibles en varios idiomas, como inglés, español y francés, entre otros, permitiendo su implementación en empresas con alcance global. Además, ViewLinc cuenta con funcionalidades de alarmas programables que se pueden configurar según horarios específicos, como turnos de trabajo o periodos de mantenimiento, y se pueden enviar a través de diversos medios, como email o mensaje de texto (SMS).

En cuanto a su operatividad, ViewLinc se integra como un servicio en sistemas Windows, con la capacidad de reiniciarse automáticamente si es necesario.

## 7.2 Recopilación de datos de costos

### 7.2.1 Costos de soluciones comerciales

Para el sistema HMT120 de Vaisala se configuraron según las opciones que se consideraron adecuadas según las necesidades mencionadas y se obtuvo el siguiente estimado como precio de adquisición del producto.

Tabla 7.3 Características del sistema Vaisala modelo HMT120. Vaisala (2023)

Ítem	Detalles
Sistema	HMT120 de Vaisala
Descripción	Transmisor de humedad y temperatura
Sonda	Acero Inoxidable HPM110 (Humedad relativa y temperatura)
Rejilla Plastica	Con filtro PTFE para sonda de acero
Cable	3 m para sonda remota

Rango de humedad relativa	0% a 100%
Rango de temperatura	-40 °C a 60 °C
Prensaestopas	Para cable de 2 a 6mm de diámetro
Unidades	Métricas
Pantalla	Con pantalla
Salida	4mA a 20mA
<b>Coste del producto (USD)</b>	\$845,00
<b>Coste del producto (CRC)</b>	₡450 570,90

Nota: Elaboración propia en base a la hoja de datos del modelo Vaisala HMT120 y página de compra de Vaisala.

Envío e impuestos: A determinar. Tras completar la orden, Vaisala envía un correo electrónico de confirmación de la orden. En un plazo de dos días hábiles, se recibirá un reconocimiento formal de la orden con la fecha de entrega estimada. Una vez que los productos hayan sido enviados, se recibirá un tercer correo electrónico con un código de seguimiento para monitorear el envío. Por lo tanto, los costos de envío e impuestos serán definidos en una proforma formal proporcionada por Vaisala posterior a la orden.

Nota: Es importante considerar que los costos de envío e impuestos no están incluidos en el precio mostrado y serán calculados de manera adicional por Vaisala en un documento formal posterior.

En el análisis de la solución Indigo202, sobre la adquisición del equipo no se limita únicamente al transmisor en sí. Es necesario añadirle una sonda adecuada. En este contexto, se ha optado por el modelo HMP3, que se adecua a las condiciones de trabajo las cuales va a estar expuesto. A continuación, se presenta un desglose de los costos asociados con la adquisición de estos componentes en una tabla.

Tabla 7.4 Características del sistema Vaisala modelo Indigo202.

Ítem	Detalles
Sistema	Indigo202 de Vaisala
Descripción	Transmisor digital de humedad y temperatura
Sonda	HMP3 (Humedad relativa y temperatura)
Protocolo	Modbus RTU over RS-485 Serial
Cable	3 m para sonda remota
Rango de humedad relativa	0% a 100%
Rango de temperatura	-40 °C a 80 °C
Unidades	Métricas
<b>Coste de transmisor</b>	\$692,00
<b>Coste de sonda</b>	\$1 387,00
<b>Coste del producto (USD)</b>	\$2 079,00
<b>Coste del producto (CRC)</b>	₡1 180 564,38

Nota: Elaboración propia en base a la hoja de datos del modelo Vaisala HMT120 y página de compra de Vaisala..

Nota: Al igual que con el otro sistema de Vaisala, es importante considerar que los costos de envío e impuestos no están incluidos en el precio mostrado. Estos serán calculados de manera adicional por Vaisala y se proporcionarán en un documento formal posterior.

Es necesario mencionar que, para que ambos sistemas (tanto el HMT120 como el Indigo202) cuenten con funciones completas de monitoreo continuo, monitoreo remoto, reportes en línea y alertas, es necesario añadir el costo de la licencia del software ViewLinc. Este software también ofrece la ventaja de permitir la configuración remota del sistema.



Es de alta relevancia mencionar que a pesar de contactar y conversar con personas de servicio al cliente del fabricante que ofrece las licencias del software no se consiguió un estimado del precio por este servicio. Se solicitó con código de compra un estimado por el equipo presentado junto a las licencias del software, pero no se consiguió una respuesta por el precio de estas licencias, solo por el equipo. Por lo que el precio final por el equipo solo contempla la adquisición del mismo, mas no la licencia que permite el uso de un monitoreo continuo remoto.

### 7.2.2 Costo de adquisición del prototipo

Se ha realizado un desglose detallado del costo total para la adquisición de los componentes del prototipo. Esta tabla contempla cada uno de los elementos y herramientas que fueron utilizados en el desarrollo y ensamble final del prototipo.

Dentro de este desglose, se encuentran componentes como el Arduino Uno, el sensor DHT22, la pantalla Liquid Crystal 1602, entre otros. Cada ítem ha sido detallado en términos de cantidad, precio unitario en dólares (USD) y colones costarricenses (CRC), así como el precio total en CRC.

El costo total estimado para la adquisición de todos los componentes del prototipo fue de ₡51 736,73. Este monto brinda una visión de la inversión inicial necesaria para llevar a cabo la creación del prototipo presentado en este proyecto.

N°	Componente	Cantidad	Precio Unitario (USD)	Precio Unitario (CRC)	Precio Total (CRC)
1	Arduino Uno	1	\$18,00	₡9 597,96	₡9 597,96
2	Sensor DHT22	1	\$10,95	₡5 838,76	₡5 838,76
3	Pantalla Liquid Crystal 1602	1	\$6,95	₡3 705,88	₡3 705,88

4	Ethernet Shield W5100	1	\$32,95	₡17 569,60	₡17 569,60
5	Modulo I2C FC-113	1	\$3,50	₡1 866,27	₡1 866,27
6	Jumpers	1	\$2,00	₡1 066,44	₡1 066,44
7	Cable alarma	1	\$1,10	₡586,54	₡586,54
8	Adaptador 9VDC	1	\$10,25	₡5 465,51	₡5 465,51
9	Caja de proyecto personalizada	1	\$9,38	₡5 000,00	₡5 000,00
10	Breadboard Mini Modular	1	\$1,95	₡1 039,78	₡1 039,78
				<b>Total</b>	<b>₡51 736,73</b>

Nota: Elaboración propia.

Al evaluar las herramientas y plataformas de software utilizadas para el funcionamiento integral del sistema, se identificó que la mayoría operan bajo planes gratuitos, aunque con ciertas restricciones.

SendGrid: Esta plataforma se empleó bajo un plan gratuito, el cual, aunque tiene un límite de 6000 correos electrónicos que pueden enviarse al mes, es adecuado para el volumen actual del sistema.

ThingSpeak: Se optó por un plan gratuito que permite recibir hasta un millón de datos por cada año. Esta cantidad es suficiente para cubrir las necesidades del sistema durante cada año.

Google Cloud: Esta plataforma ofrece un plan gratuito que proporciona \$300 en créditos para utilizar en sus servicios. A pesar de contar con este límite monetario, el sistema, con su plan de alertas actual, no ha consumido un solo crédito.

Para los costos de mano de obra, se realizó una evaluación de las horas invertidas en el desarrollo del prototipo. Se contabilizaron un total de 68 horas de trabajo directo, las cuales no incluyen el tiempo adicional empleado en pruebas de laboratorio o ajustes indirectos. El Centro de Investigaciones en Vivienda y Construcción (CIVCO) proporcionó una tarifa estándar de 1,500 colones por hora como valor de referencia,

equivalente a la tarifa de asistente estudiantil. Por consiguiente, la estimación de los costos por mano de obra asciende a ¢102,000.

Teniendo en cuenta todos estos factores, el costo integral de software, adquisición de equipo necesario y estimación de mano de obra da como resultado un total de ¢153 736,73.

### **7.3 Análisis comparativo de costos**

El propósito de esta sección es realizar una evaluación objetiva y clara de los costos asociados al prototipo desarrollado en comparación con otras soluciones similares disponibles en el mercado. Dicha evaluación permitió determinar qué sistema conviene adquirir. Para el objetivo de conseguir la acreditación para el laboratorio de ensayo.

### 7.3.1 Herramientas de visualización

Tabla 7.6 Comparativa de Características de Sistemas de Monitoreo.

Característica	Sistema Propio	Vaisala HMT120	Vaisala Indigo202
Rango de medición de humedad	0% a 100%	0% a 100%	0% a 100%
Rango de medición de temperatura	-40 °C a 80 °C	-40 °C a 60 °C	-40 °C a 80 °C
Alimentación	9 VDC	10 a 35 VDC	15 A 30 VDC
Material de carcasa	Acrílico	Acero inoxidable	Plástico PC/ABS
Certificado de calibración	Requiere de prueba en laboratorio de calibración	Requiere de prueba en laboratorio de calibración	Incluido con la adquisición
Pantalla	Muestra solo monitoreo actual	Muestra solo monitoreo actual	Muestra monitoreo actual y configuraciones como gráficos de registro historico

Nota: Elaboración propia.

La siguiente gráfica ilustra una comparativa de los costos de adquisición de tres diferentes sistemas de monitoreo; el sistema propio, Vaisala HMT120 y Vaisala Indigo202. Estos costos representan una inversión inicial en cada tecnología, siendo un factor determinante en la decisión de implementación.

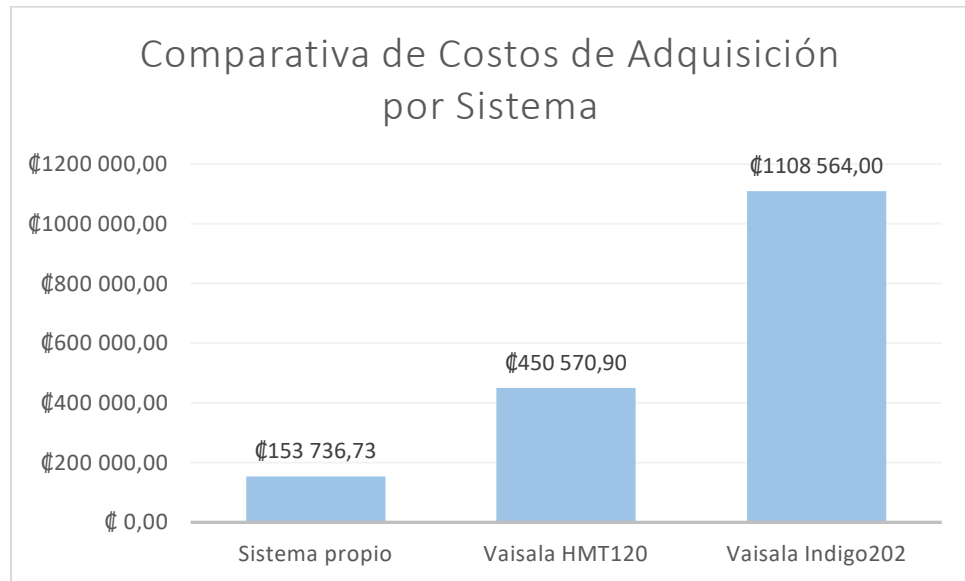


Figura 7.3 Comparativa de inversión inicial en sistemas de monitoreo. Fuente: Elaboración propia.

#### 7.4 Evaluación de mejor opción de sistema

La elección adecuada de una solución de monitoreo para el Centro de Investigaciones en Vivienda y Construcción (CIVCO) se basa tanto en términos de coste como de funcionalidad y escalabilidad. Para tomar una decisión informada y sólida, se ha evaluado estas tres dimensiones y otorgado puntuaciones a cada opción: el sistema Vaisala HMT120, el sistema Vaisala Indigo202 y el desarrollo propio del sistema.

Además, con base en los datos recopilados, se ha preparado un reporte que destaca notoriamente las diferencias de costos entre el prototipo propio y las soluciones comerciales Vaisala HMT120 y Vaisala Indigo202. Al analizar estos costos, se evidencia que el prototipo, con un costo de adquisición de €153 73,73 es un 65.86% más barato que el Vaisala HMT120 y un 86.11% en comparación con el Vaisala Indigo202.

Los tres sistemas que se compararon son útiles para el Centro de Investigaciones en Vivienda y Construcción (CIVCO) para cumplir con el objetivo de un monitoreo apto en el cuarto húmedo que garantice la posibilidad de cumplir con lo evaluado por el Ente Costarricense de Acreditación (ECA). El precio aumenta según las posibilidades de cada sistema, mientras el Vaisala HMT120 tiene un precio de adquisición

de ¢450 570.90 y el Vaisala Indigo202 se eleva hasta ¢1 108 564,00, la diferencia de precios es para tomarlo en cuenta.

Las soluciones comerciales, a pesar de ser más costosas, gozan del respaldo y reconocimiento de fabricantes consolidados, así como de garantías y asistencia técnica, aspectos que pueden ser determinantes para ciertas organizaciones. Pero, si el principal objetivo es obtener mediciones aceptada por el organismo acreditador sin grandes desembolsos, el prototipo emerge como una alternativa factible. Este último responde al desafío de ofrecer una herramienta apta para monitoreos y posibles calibraciones en el laboratorio. Según la norma ISO/IEC, los equipos de medición requieren calibración en un laboratorio certificado por la norma ISO 17025 y una trazabilidad de los datos, condiciones que nuestro sistema cumple.

A continuación presentan los tres apartados dando una puntuación según el orden de mejor opción (3 puntos) a peor opción (1 punto). Como respaldo de la puntuación otorgada por cada punto se presenta la siguiente tabla para comparar en sus requisitos mínimos las ventajas y desventajas de cada sistema.

Tabla 7.7 Comparativo de ventajas y desventajas de los requisitos mínimos para las opciones de sistemas de monitoreo.

Requisito mínimo	Sistema Propio		Vaisala HMT120		Vaisala Indigo202	
	Ventajas	Desventajas	Ventajas	Desventajas	Ventajas	Desventajas
Sensor capaz de ser calibrado por un laboratorio de calibración certificado.	Sí ofrece.		Sí ofrece.		Sí ofrece. Con la compra ya incluye una certificación de calibración del sensor.	
Pantalla integrada	Sí incluye.	Limitado a 16x2 caracteres	Sí incluye		Sí incluye.	Ofrece más funciones que no son necesarias para el laboratorio.
Monitoreo remoto	Sí incluye por página web. Gratuito.		Incluido con software.	El software es por licencia de paga.	Incluido por software.	El software es por licencia de paga.
Configuración del sistema	Sí ofrece.	Requiere conexión directa a una computadora y conocimiento del código de programación del sistema.	Sí ofrece. Posibilidad de ser remoto.	Sí es remoto es por medio de un software con licencia de paga.	Sí ofrece. Posibilidad de ser remoto.	Sí es remoto es por medio de un software con licencia de paga.
Reporte histórico de monitoreo	Sí ofrece.		Sí ofrece.		Sí ofrece.	

Nota: Elaboración propia.

#### 7.4.1 Costo de adquisición

Desarrollo Propio: 3 puntos (El más económico).

Vaisala HMT120: 2 puntos (Precio intermedio).

Vaisala Indigo202: 1 punto (El más caro de los tres).

La puntuación refleja que el Desarrollo Propio es la opción más económica de adquirir, lo que lo convierte en una alternativa atractiva desde una perspectiva financiera.

#### **7.4.2 Funcionalidad**

Desarrollo Propio: 3 puntos (Altamente personalizado a las necesidades del CIVCO). Su desarrollo cumple con lo requerido por el laboratorio y no incurre en funciones que no son de interés para el laboratorio.

Vaisala HMT120: 2 puntos (Ofrece funcionalidad adecuada, pero requiere software de paga externo).

Vaisala Indigo202: 1 punto (Ofrece funcionalidades de más que son innecesarias para el CIVCO). Este sistema sí el más robusto de los tres, por lo que puede parecer de mayor funcionabilidad que los otros sistemas. Pero las funciones de más que ofrece este dispositivo no son necesarias para el laboratorio.

La puntuación indica que el sistema de Desarrollo Propio ofrece la funcionalidad más alineada con las necesidades específicas del CIVCO, sin depender de software externo y permitiendo una alta personalización.

#### **7.4.3 Escalabilidad**

Vaisala Indigo202: 3 puntos (Mayor escalabilidad). Al ser el más robusto de los tres sistemas ofrece mayores opciones de escalabilidad.

Vaisala HMT120: 2 puntos (Buena escalabilidad). Ya que ofrece menos trabajos para un cambio de componente o si se piensa adquirir una nueva sonda. Es de solo quitar y poner.

Desarrollo Propio: 1 punto (Requiere trabajo adicional para cambios significativos). Es decir, para que el sistema se adecue a necesidades en un futuro, puede incurrir en mayores tareas como cambios de componentes y requiere de mayor esfuerzo la programación del sistema. Las otras dos opciones ofrecen un software que permite la posibilidad de ajustar parámetros como tiempo entre mediciones, ajuste de rangos de alerta y demás.



Según nuestra evaluación, el Vaisala Indigo202 destaca por su escalabilidad y certificaciones, lo que lo convierte en una opción robusta a medida que las necesidades del CIVCO evolucionen.

#### **7.4.4 Resultado de la Evaluación**

Tras evaluar y puntuar cada opción en términos de coste, funcionalidad y escalabilidad, se llegó al siguiente resultado:

Desarrollo Propio: 7 puntos

Vaisala HMT120: 6 puntos

Vaisala Indigo202: 5 puntos

El Desarrollo Propio se presenta como la elección más adecuada para el Centro de Investigaciones en Vivienda y Construcción CIVCO en términos de coste, funcionalidad y escalabilidad.

Este proceso de evaluación y selección proporciona una base sólida para tomar una decisión informada sobre la adquisición de un sistema de monitoreo que cumpla con los requisitos requeridos y, al mismo tiempo, sea coste-efectivo y adaptable a las necesidades actuales y las que pueden aparecer en el futuro.

## Conclusiones

- Se ha desarrollado un prototipo de sistema de monitoreo de humedad relativa y temperatura con la intención de cumplir con los requisitos de la norma ISO/IEC 17025 para el cuarto húmedo del Centro de Investigaciones en Vivienda y Construcción. Sin embargo, la ausencia de los resultados de la calibración certificada dentro del plazo del proyecto impide confirmar su conformidad con dicha normativa. Este hecho limita la posibilidad de que el prototipo pueda contribuir en el proceso de acreditación del laboratorio de ensayos hasta que se complete la calibración y se corrobore la fiabilidad de sus mediciones.
- Se ha logrado elaborar un prototipo basado en Arduino que monitoriza en tiempo real la temperatura y humedad. La optimización en la selección de componentes garantizó una relación costo-beneficio eficiente. Los pasos aplicados en el ensamblaje y prueba del prototipo culminaron con un sistema en funcionamiento que muestra datos en tiempo real.
- El sistema de monitoreo ha sido integrado con éxito en la nube, lo que permite un almacenamiento de datos, generación de gráficos de monitoreo histórico y alertas en caso de fallas. A pesar de una selección adecuada de hardware y software, así como modificaciones efectuadas en el código Arduino para garantizar la trazabilidad, la confiabilidad de los datos recogidos no ha podido ser confirmada debido a la ausencia del certificado de calibración del sistema.
- Al comparar el coste inicial del prototipo con soluciones comerciales, el sistema de desarrollo propio es la de menor coste. Además, tras una evaluación que tomó en cuenta costo inicial, funcionalidad y escalabilidad, la alternativa del sistema de desarrollo propio se establece como la más ventajosa para el Centro de Investigaciones en Vivienda y Construcción (CIVCO).

## Recomendaciones

- Se sugiere revisar y recalibrar periódicamente los sensores del prototipo para garantizar que las mediciones sean confiables. Además de que debe estar certificado el sensor por un laboratorio de calibración para que el resultado de monitoreo sea válido por el Ente Costarricense de Acreditación (ECA).
- Se sugiere la implementación de gabinete para intemperie IP65.
- Dada la observación de que el sistema solo puede monitorear, pero no corregir activamente las condiciones, se sugiere explorar opciones para integrar capacidades de intervención. Esto podría implicar el control automático de aspersores o sistemas de humidificación.
- Establecer protocolos de respaldo para asegurar la integridad y disponibilidad de los datos en caso de fallas o interrupciones en la conexión a internet.
- Realizar un estudio más profundo sobre los costos operativos a largo plazo del prototipo para compararlo de manera más holística con soluciones comerciales. Esto incluiría mantenimiento, calibración, actualizaciones y posibles fallas.
- Asegurarse de mantener una documentación actualizada del sistema, sus componentes y su operación. Esto facilitará cualquier tarea de mantenimiento o reparación futura.
- Brindar capacitaciones regulares al personal involucrado para garantizar el correcto uso y comprensión del sistema.
- Se sugiere la utilización del Apéndice A. Manual del Prototipo para el correcto uso del sistema de monitoreo.

## Referencias bibliográficas

- Amazon Web Services. (2023). AWS IoT. [En línea]. Disponible en: <https://aws.amazon.com/iot/>
- Arduino Team. (n.d.). Getting started with Arduino. Recuperado el 1 de mayo de 2023, de <https://docs.arduino.cc/learn/starting-guide/getting-started-arduino/>
- Arduino. (2021). Arduino Boards. Recuperado el 29 de abril de 2023, de <https://www.arduino.cc/en/Main/Products>
- ASTM International. (2008). Annual Book of ASTM Standards. Section 4: Construction.
- ASTM International. (2012). ASTM C31, Standard Practice for Making and Curing Concrete Test Specimens in the Field. In Book of Standards (Vol. 04.02).
- ASTM International. (2013). ASTM C511, Standard Specification for Mixing Rooms, Moist Cabinets, Moist Rooms, and Water Storage Tanks Used in the Testing of Hydraulic Cements and Concretes. In Book of Standards (Vol. 04.01).
- ASTM International. (2020). ASTM C 192: Standard Practice for Making and Curing Test Specimens in the Laboratory.
- Augustin, A. et al. (2016). A study of LoRa: Long-range & low-power networks for the Internet of Things. *Sensors*, 16(9), 1466.
- Danladi, M. S., & Baykara, M. (2022). Design and implementation of temperature and humidity monitoring system using LPWAN technology. *Ingénierie des systèmes d'information*, 27(4), 521–529.
- De, J., Iglesia, L., Murillo, M., & Borrero Restrepo, D. (2021). Study of the importance of core cylinders in the monitoring of the compressive strength of the concrete. *IOP Conference Series: Materials Science and Engineering*, 1126(1), 012050.
- Dwiyaniti, M., Nitisasmita, K. M., & Tohazen. (2019). Energy efficiency on smart street lighting using Raspberry Pi based on SCADA and internet of things (IOT). *Journal of Physics: Conference Series*, 1364(1), 012034.

- Elsalamawy, M., Mohamed, A. R., & Kamal, E. M. (2019). The role of relative humidity and cement type on carbonation resistance of concrete. *Alexandria Engineering Journal*, 58(4), 1257–1264.
- Ente Costarricense de Acreditación. (n.d.). Generando confianza. Recuperado el 11 de abril de 2023, de <https://eca.or.cr/>
- Franksson, R., & Liljegren, A. (2018). *Measuring a LoRa Network Performance, Possibilities and Limitations*. Karlskrona, Sweden: Blekinge Institute of Technology.
- Gay, W. (2018). *Advanced raspberry pi: Raspbian linux and Gpio Integration*. Berkeley, CA: Apress.
- Groot, C., Veiga, R., Papayianni, I., Van Hees, R., Secco, M., Alvarez, J. I., Faria, P., & Stefanidou, M. (2022). Rilem TC 277-LHS report: Lime-based mortars for restoration—a review on long-term durability aspects and experience from practice. *Materials and Structures*, 55(10).
- Isobox Systems. (2023). *Requisitos de los sensores de salas blancas de centros de datos*.
- ISO/IEC 17025. (2017). *General Requirement for the competence of testing and calibration laboratories* (2da ed.).
- Kumar, A., Kumar, A., Singh, A. K., & Choudhary, A. K. (2021). IOT based energy efficient agriculture field monitoring and smart irrigation system using nodemcu. *Journal of Mobile Multimedia*.
- Linus L., & Aris B. (2023). *What Is a Cron Job: Understanding Cron Syntax and How to Configure Cron Jobs*. VPS.
- Liu, H.-Y. (2020). Design of a wireless temperature and humidity monitoring system based on Zigbee Technology. *Journal of Physics: Conference Series*, 1601(4), 042044. <https://doi.org/10.1088/1742-6596/1601/4/042044>
- LoRaAlliance. (2015). *A technical overview of lora® and lorawan™*. Technical Marketing Workgroup 1.0.
- Lorek, A., & Majewski, J. (2018). Humidity measurement in carbon dioxide with capacitive humidity sensors at low temperature and pressure. *Sensors*, 18(8), 2615.

- MathWorks. (2023). ThingSpeak: The IoT Platform. [En línea]. Disponible en: <https://www.mathworks.com/products/thingspeak.html>
- Nasution, T.H., Muchtar, M.A., Seniman, S., & Siregar, I. (2019). Monitoring temperature and humidity of server room using LattePanda and ThingSpeak. *Journal of Physics: Conference Series*.
- Neville, A. (2011). *Concrete Technology: Properties and Testing of Aggregates*. Springer.
- Nurcahyo, R., Gabriel, D. S., Ivan, E., Habiburrahman, M., & Sari, I. P. (2018). ISO/IEC 17025 implementation at Testing Laboratory in Indonesia. En 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS).
- Olaniyi, A. O., Watanabe, T., Umeda, K., & Hashimoto, C. (2021). IOT-web-based integrated wireless sensory framework for non-destructive monitoring and evaluation of on-site concrete conditions. En *Lecture Notes in Mechanical Engineering* (pp. 527-536).
- Pereira, E., Pinto, R., Reis, J., & Gonçalves, G. (2019). MQTT-RD: A mqtt based resource discovery for machine to machine communication. *Proceedings of the 4th International Conference on Internet of Things, Big Data and Security*.
- Polonelli, T., Brunelli, D., Bartolini, A., & Benini, L. (2018). A lorawan wireless sensor network for data center temperature monitoring. In *International Conference on Applications in Electronics Pervading Industry, Environment and Society* (pp. 169-177).
- Qavidel Fard, Z., Zomorodian, Z. S., & Korsavi, S. S. (2022). Application of machine learning in Thermal comfort studies: A review of methods, performance and challenges. *Energy and Buildings*, 256, 111771.
- Ranger, S. (2022). What is cloud computing? Everything you need to know about the cloud explained.
- Rao, D. D. (2021). ISO/IEC 17025: Accreditation standard for testing and Calibration Laboratories. *Radiation Protection and Environment*, 44(3), 121.
- Rapid. (n.d.). API Key – What is an API Key?

- Raspberry Pi Foundation. (2021). Raspberry Pi Products. Recuperado el 29 de abril de 2023, de <https://www.raspberrypi.org/products/>
- Regtien, P. P. (2011). Humidity sensors. *Measurement Science and Technology*, 23(1), 010103.
- Salleh, A., Ismail, M.K., Mohamad, N.R., Abd Aziz, M.A., Othman, M.A., & Misran, M.H. (2013). Development of greenhouse monitoring using wireless sensor network through ZigBee technology. *International Journal of Engineering Science Invention*, 2(7), 6-12.
- Tang, X., Tan, C., Chen, A., Li, Z., & Shuai, R. (2020). Design and implementation of temperature and humidity monitoring system for small cold storage of fruit and vegetable based on Arduino. *Journal of Physics: Conference Series*, 1601(6), 062010.
- Tang, X., Tan, C., Chen, A., Li, Z., & Shuai, R. (2020). Design and implementation of temperature and humidity monitoring system for small cold storage of fruit and vegetable based on Arduino. *Journal of Physics: Conference Series*, 1601(6), 062010.
- TEC. (2023, febrero 2). Centro de Investigaciones en vivienda y construcción. <https://www.tec.ac.cr/civco>.
- TEC. (n.d.). Misión y Visión. Recuperado el 11 de abril de 2023, de <https://tec.ac.cr/mision-vision>
- ThingSpeak. (2023). Communication Protocols. [En línea]. Disponible en: <https://www.mathworks.com/help/thingspeak/communication-protocols.html>
- Ubidots. (2023). Ubidots IoT Application Development Platform. [En línea]. Disponible en: <https://ubidots.com/platform/>
- Ubidots. (2023). Ubidots IoT Platform. [En línea]. Disponible en: <https://ubidots.com/>
- Vignesh, R., & Abdul Rahim, A. (2022). Mechanical and microstructural properties of Quaternary Binder system containing OPC-GGBS-Metakaolin-Lime. *Materials Today: Proceedings*, 64, 970–975.
- Yanshori, D. W. Nugraha, & Santi, D. (2022). Lot-based temperature and humidity monitoring system for Smart Garden. *IOP Conference Series: Materials Science and Engineering*, 1212(1), 012047.

## Apéndices

Apéndice A. Manual del Prototipo

### *Manual del Prototipo: Monitoreo y registro en tiempo real de humedad relativa y temperatura*



Desarrollado por: Luis Hernesto Pérez Buendía.



## **1. Introducción**

Este es el manual de usuario del sistema de monitoreo en tiempo real basado en tecnología Arduino. Este sistema ha sido diseñado con el objetivo de ofrecer una herramienta eficiente y fácil de usar para la monitorización en tiempo real de diferentes variables en el área de monitoreo de condiciones ambientales de lugares cerrados.

La principal característica de este sistema es su capacidad de conectarse a la red a través de la tecnología Ethernet, gracias al uso del Ethernet Shield, permitiendo así una transmisión de datos constante y en tiempo real.

Este manual está dirigido, en especial, a los operarios del Centro de Investigación en Vivienda y Construcción, quienes se beneficiarán directamente de las capacidades y funcionalidades de este prototipo. No obstante, cualquier persona o entidad interesada en sistemas de monitoreo basados en Arduino y su interconexión a través de Ethernet encontrará en este manual una guía para el uso de este sistema.

Mi deseo es que, a través de este manual, los usuarios puedan comprender, configurar y utilizar este sistema de manera óptima y sacar el máximo provecho a todas sus características.

## 2. Especificaciones Técnicas

Especificación	Descripción	Característica
Dimensiones	0,22x0,08x0,08	Metros
VARIABLES POR MONITOREAR		
Temperatura	23± °C	Normativa ASTM C511: mantener a 23± °C durante curado
Humedad relativa	+95%	Mantener superior al 95% dentro de la cámara.
Arduino Uno R3		
Chip	ATmega328	Microcontrolador central del prototipo.
Voltaje de Operación	5V	Asegura operación estable.
Voltaje de Entrada (recomendada)	7-9V	Rango óptimo de alimentación.
Pines de E/S Digitales	14	Conexión y control de componentes
Pines de Entrada Analógica	6	Lectura de señales analógicas.
Memoria Flash	32 KB	Espacio para cargar y almacenar el código.
Velocidad de Reloj	16 MHz	Velocidad de operación del microcontrolador.
Sensor: DHT 22		
Alimentación	3.3-6V	Rango de alimentación.
Rango de humedad	0-100% RH	Intervalo de detección de humedad.
Rango de temperatura	-40°C a 80°C	Intervalo de detección de temperatura.
Precisión de humedad	±2% RH	Exactitud en las mediciones de humedad.
Precisión de temperatura	±0.5°C	Exactitud en las mediciones de temperatura.
Ethernet Shield		
Controlador Ethernet	W5100	Microcontrolador central del shield.
Voltaje de Operación	5V	Compatible con Arduino Uno y Mega.
Velocidad de Red	10/100Mb	Velocidad de transferencia de datos
Conexión	RJ-45	Estándar para conexiones de red Ethernet.

### **3. Instalación y Configuración**

#### **3.1 Instalación del Prototipo**

##### **a) Instalación del Prototipo:**

Se debe ubicar el prototipo en el lugar designado, garantizando que todos sus componentes estén fijados de manera segura. Es esencial prestar especial atención a cualquier sensor o componente crítico, para asegurar su correcta instalación en una ubicación óptima.

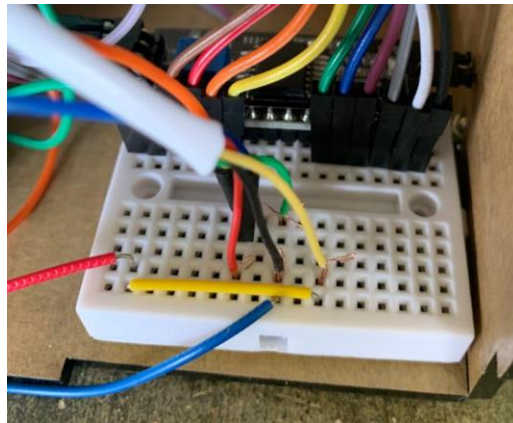
##### **b) Verificación Inicial:**

Antes de proceder con la configuración, se realiza una comprobación general para confirmar que el prototipo funciona adecuadamente y que todas las conexiones están firmes.

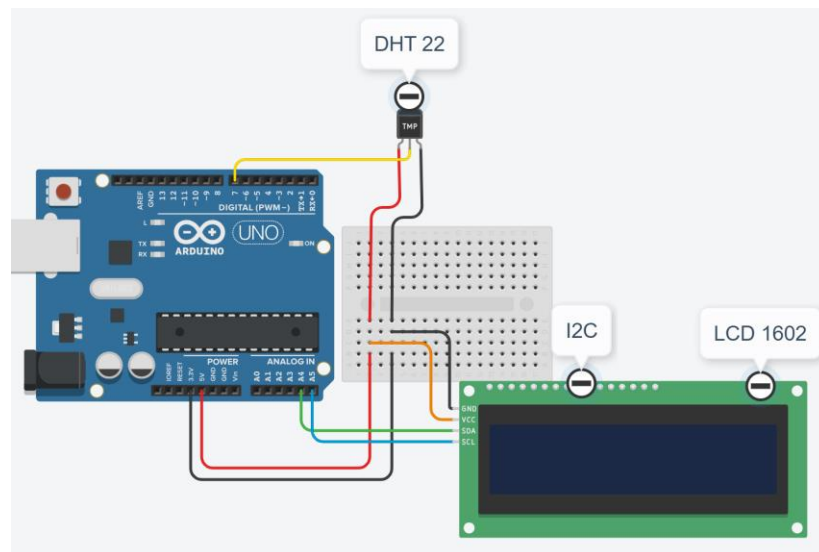
### 3.2 Configuración Inicial del Prototipo.

#### a) Conexión del Hardware:

Se deben revisar todas las conexiones físicas del prototipo, incluyendo cualquier cableado o conexión entre componentes. Se asegura que todo esté conectado de manera segura y en los puertos adecuados. La siguiente imagen muestra las conexiones existentes de la breadboard.



Se debe seguir el siguiente esquema de conexión para garantizar el correcto funcionamiento.



### **b) Posicionamiento del Sensor:**

En caso de que el prototipo cuente con sensores, estos se ubican en lugares donde puedan realizar mediciones precisas sin interferencias, evitando zonas con fluctuaciones extremas o posibles obstrucciones.

### **c) Configuración del Software:**

Tiempo de envío de datos a la nube:

Dentro del código correspondiente de arduino, hay una línea específica que determina el intervalo de tiempo con el cual se envían datos a la plataforma ThingSpeak. Esta línea está configurada en términos de milisegundos.

```
// Enviar datos a ThingSpeak cada 5 minutos
if (millis() - lastUpdate >= 300000) {
    Serial.println("Intentando conectar a ThingSpeak...");
    .
    .
    .
}
```

Para ajustar la frecuencia con la que se transmiten los datos, simplemente se debe localizar y modificar esta línea en el código. El valor que se especifique en esta línea determinará el tiempo entre envíos sucesivos de datos. Por ejemplo, si se desea enviar datos cada 10 segundos, el valor que se debe ingresar es 10000 (ya que 10 segundos equivalen a 10,000 milisegundos).

Es esencial tener en cuenta que al modificar este valor, se está ajustando la periodicidad con la que el sistema se comunica con ThingSpeak. Por lo tanto, es importante encontrar un equilibrio entre la frecuencia de envío y las necesidades, para no sobrecargar la plataforma

### **d) Conexiones Finales:**

Por último, se conecta el prototipo a las fuentes de alimentación y comunicación requeridas, como el cable negro de corriente del convertidor de 9VDC y el cable ethernet al puerto correspondiente.

Una vez completados estos pasos, el prototipo comenzará a desplegar datos en la plataforma ThingSpeak correspondiente. Si solo muestra datos en el display y no envía información a ThingSpeak, es

indicativo de problemas con la conexión a internet. Por otro lado, si el display muestra un error relacionado con la detección del DHT, indica que no está recibiendo datos del sensor.

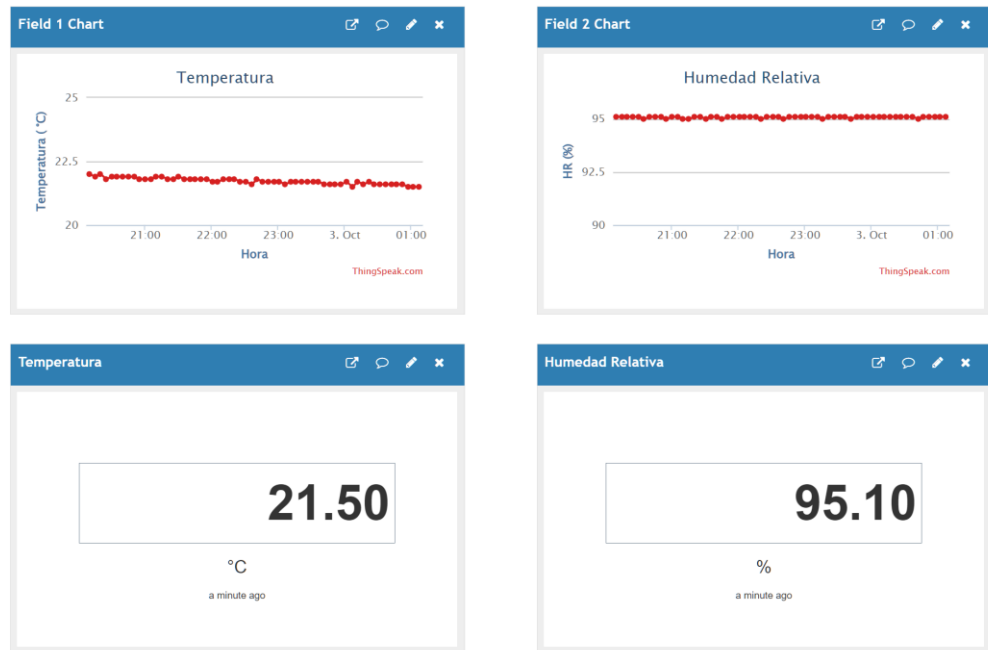
#### 4. Funcionamiento

##### Descripción de la interfaz de usuario:

Al acceder al canal ThingSpeak, el usuario se encuentra con un "Dashboard" o panel de control. Aquí, se pueden ver gráficos y tablas que representan los datos enviados por los sensores en tiempo real. Estos gráficos se pueden personalizar según las preferencias del usuario, permitiendo elegir entre diferentes tipos de visualizaciones, como gráficos de líneas, barras o puntos.

##### Channel Stats

Created: [20 days ago](#)  
 Last entry: [2 minutes ago](#)  
 Entries: 2592



En la parte superior de la interfaz, hay una barra de herramientas que ofrece acceso rápido a diversas funciones, como agregar un nuevo canal, configurar alertas o acceder a análisis avanzados. Cada canal en ThingSpeak corresponde a un conjunto específico de datos de sensores. Al seleccionar un canal, el usuario puede ver un resumen detallado de esos datos, así como estadísticas relevantes y gráficos históricos.

**Ingreso de correo electrónico destinatario de las alertas:**

```
# Enviar el correo usando SendGrid  
EMAIL_FROM = "mhcivco@gmail.com"  
EMAIL_TO = ""
```

En la línea EMAIL\_TO = "", se debe ingresar la dirección de correo electrónico donde se desea recibir las alertas. Es importante que se proporcione una dirección válida para garantizar la recepción adecuada de las notificaciones.

Este sistema ha sido integrado con Google Cloud. Al recibir el prototipo, se le proporcionarán cuentas y códigos específicos para que pueda configurarlo conforme a sus preferencias.

**Ingreso de correo electrónico destinatario de reporte de monitoreo:**

Dentro del programa desarrollado, al ejecutarlo, se genera automáticamente un informe histórico basado en el monitoreo de los últimos 30 días. Este informe es posteriormente enviado por correo electrónico para su revisión.

Para asegurar la correcta recepción de este reporte, necesario que el usuario proporcione una dirección de correo electrónico válida. Para hacerlo, debe dirigirse a la línea EMAIL\_TO = "" y, en el espacio provisto entre las comillas, ingresar la dirección donde desea recibir el informe.

```
# Enviar el correo usando SendGrid  
EMAIL_FROM = "mhcivco@gmail.com"  
EMAIL_TO = ""
```

## 5. Mantenimiento y Cuidados

A continuación, se presentan una serie de recomendaciones para garantizar el correcto funcionamiento de su prototipo a lo largo del tiempo:

### a. Limpieza:

**Superficie:** Utilice un paño suave y ligeramente húmedo para limpiar la superficie del Arduino, shield y otros componentes. Asegúrese de que no entre líquido en ninguna de las partes electrónicas.

**Conexiones y Puertos:** De vez en cuando, limpie puertos y conexiones, eliminando el polvo y otros residuos que puedan acumularse.

### b. Almacenamiento:

Guarde el prototipo en un lugar seco y libre de polvo. La humedad y el polvo pueden causar cortocircuitos o interferir con la funcionalidad del sistema.

### c. Uso de Cables:

No doble ni retuerza los cables en exceso. Esto puede dañar las conexiones internas y reducir la eficiencia de la transmisión de datos o energía.

Regularmente, revise los cables en busca de desgaste o daño y reemplace los que presenten signos evidentes de deterioro.

### d. Manejo de Sensores:

Trate el sensor con cuidado para evitar daños físicos.

### e. Inspección Regular:

Realice inspecciones visuales regularmente para detectar signos de daño o desgaste en las conexiones, soldaduras y otros componentes. Las reparaciones tempranas pueden prevenir fallos más grandes en el futuro.

### f. Evite Sobrecargas:

No exceda la capacidad recomendada del Arduino y otros componentes, ya que esto puede resultar en sobrecalentamiento o daño.



## **7. Contacto y Soporte**

Si tiene alguna duda, comentario o necesita soporte relacionado con el uso del prototipo, se puede contactar de las siguientes maneras:

Luis Hernesto Pérez

Correo Electrónico: Para consultas generales o asuntos técnicos, [luishpb@gmail.com](mailto:luishpb@gmail.com).

Teléfono: +506 8588 1632

Apéndice B. Plano de montaje de caja personalizada.

The diagram shows an exploded view of a custom box assembly. The components are numbered 1 through 8. The assembly is shown in a perspective view, with grid lines A-F and 1-4 indicating dimensions and positions. Callout 1 points to the front panel, 2 to the top, 3 to the left side, 4 to the right side, 5 to the back, 6 to the bottom, 7 to the screws, and 8 to the nuts.

N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Parte frontal pantalla	1
2	Parte superior	1
3	Parte lateral izquierda	1
4	Parte lateral derecha	1
5	Parte trasera	1
6	Parte inferior	1
7	DIN EN 24017 - M2 x 10-N	11
8	Hexagon Nut ISO 4036 - M2 - N	11

SINO SE INDICA LO CONTRARIO: ACABADO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:				REBARBAR Y ROMPER ARDITAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
NOMBRE	FIRMA	FECHA			TÍTULO: Plano de montaje caja personalizada	
DIBUJ.	L. Pérez				Nº DE DIBUJO: Ensamblaje	
VERIF.					A4	
APROB.					ESCALA: 1:2	
FABR.					HOJA 1 DE 1	
CALID.				MATERIAL: Acrílico		
				PESO:		

Apéndice C Correo electrónico recibido por el programa de reportes de monitoreo.

# Reporte de Monitoreo de Temperatura y Humedad - 2023-10-17

Recibidos x



mhcivco@gmail.com a través de sendgrid.net

para mí ▾

Por favor, consulte los archivos adjuntos para el reporte de monitoreo.

2 archivos adjuntos • Analizado por Gmail ⓘ

	A	R	O	D	E	F
1	Fecha	Hora	Temperatura	Humedad (%)		
2	2023-10-17 08:00		26.2	76.2		
3	2023-10-17 08:07		26.2	76.3		
4	2023-10-17 08:14		26.2	76.2		
5	2023-10-17 08:21		26.3	76.2		
6	2023-10-17 08:28		26.3	76.3		
7	2023-10-17 08:35		26.2	76.3		
8	2023-10-17 08:42		26.2	76.2		
9	2023-10-17 08:49		26.2	76.3		
10	2023-10-17 08:56		26.2	76.3		
11	2023-10-17 09:03		26.2	76.2		
12	2023-10-17 09:10		26.3	76.3		
13	2023-10-17 09:17		26.2	76.2		
14	2023-10-17 09:24		26.2	76.3		
15	2023-10-17 09:31		26.2	76.3		
16	2023-10-17 09:38		26.2	76.5		
17	2023-10-17 09:45		26.2	76.2		
18	2023-10-17 09:52		26.1	76.4		

X reporte\_2023-10-...




Apéndice D Ilustración de parte recortada del reporte de monitoreo en archivo de Excel.

	A	B	C	D	E
4965	2023-10-12	14:24:08		21,9	95,2
4966	2023-10-12	14:29:11		22,1	95,1
4967	2023-10-12	14:34:14		22	95,2
4968	2023-10-12	14:39:17		22	95,2
4969	2023-10-12	14:44:20		22,1	95,3
4970	2023-10-12	14:49:23		22,1	95,3
4971	2023-10-12	14:54:26		22,1	95,3
4972	2023-10-12	14:59:29		22	95,4
4973	2023-10-12	15:04:32		22	95,3
4974	2023-10-12	15:09:34		22	95,4
4975	2023-10-12	15:14:37		22	95,4
4976	2023-10-12	15:19:40		22,1	95,4
4977	2023-10-12	15:24:43		22	95,4
4978	2023-10-12	15:29:46		22	95,4
4979	2023-10-12	15:34:49		22	95,4
4980	2023-10-12	15:39:52		22	95,4
4981	2023-10-12	15:44:54		22	95,4
4982	2023-10-12	15:49:57		22,1	95,4
4983	2023-10-12	15:55:00		22	95,4
4984	2023-10-12	16:00:03		22	95,4
4985	2023-10-12	16:05:06		22	95,4
4986	2023-10-12	16:10:09		22	95,4
4987	2023-10-12	16:15:12		22,1	95,4
4988	2023-10-12	16:20:15		22,1	95,4
4989	2023-10-12	16:25:17		22	95,4
4990	2023-10-12	16:30:20		22	95,4
4991	2023-10-12	16:35:23		22	95,4
4992	2023-10-12	16:40:26		22	95,4
4993	2023-10-12	16:45:29		22	95,5
4994	2023-10-13	08:31:29		21,3	95,8
4995	2023-10-13	08:36:32		21,2	95,7
4996	2023-10-13	08:41:35		21,3	95,6
4997	2023-10-13	08:46:38		21,4	95,6
4998	2023-10-13	08:51:41		21,4	95,6
4999	2023-10-13	08:56:44		21,3	95,6
5000	2023-10-13	09:01:47		21,5	95,6

## Apéndice E Ilustración de correo de alerta sobre desconexión del sistema.

Alerta de Monitoreo: Desconexión del Sistema Σ Recibidos x

 **mchcivco@gmail.com** a través de sendgrid.net mar, 24 oct, 20:15 (hace 5 días)  
para mi ▼

El sistema de monitoreo no ha enviado datos en los últimos 15 minutos. Por favor verifica la conexión y el funcionamiento del hardware.  
...

[Mensaje recortado] [Ver todo el mensaje](#)

## Anexos

Anexo 1 Cambio de divisas de dólar estadounidense a colón costarricense de la fecha de 10 de octubre de 2023.

1 Dólar estadounidense =

**533,22 Colón costarricense**

Última actualización · 10 de octubre, 4:07 UTC

USD \$ - Dólar estadounidense	⇌	CRC ₡ - Colón costarricense
1,00		533,22

Anexo 2 Datos técnicos del sensor DHT22.

### Aosong Electronics Co.,Ltd

Your specialist in innovating humidity & temperature sensors

1. Feature & Application:

- \* Full range temperature compensated
- \* Relative humidity and temperature measurement
- \* Calibrated digital signal
- \* Outstanding long-term stability
- \* Extra components not needed
- \* Long transmission distance
- \* Low power consumption
- \* 4 pins packaged and fully interchangeable

#### 2. Description:

DHT22 output calibrated digital signal. It utilizes exclusive digital-signal-collecting-technique and humidity sensing technology, assuring its reliability and stability. Its sensing elements is connected with 8-bit single-chip computer.

Every sensor of this model is temperature compensated and calibrated in accurate calibration chamber and the calibration-coefficient is saved in type of programme in OTP memory, when the sensor is detecting, it will cite coefficient from memory.

Small size & low consumption & long transmission distance(20m) enable DHT22 to be suited in all kinds of harsh application occasions.

Single-row packaged with four pins, making the connection very convenient.

#### 3. Technical Specification:

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +-2%RH(Max +-5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+0.3%RH
Long-term Stability	+0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

Fuente: Aosong Electronics Co.,Ltd.

## Anexo 3 Datos generales del microcontrolador Arduino Uno R3.



Arduino® UNO R3

---

### Features

- **ATMega328P Processor**
  - **Memory**
    - AVR CPU at up to 16 MHz
    - 32KB Flash
    - 2KB SRAM
    - 1KB EEPROM
  - **Security**
    - Power On Reset (POR)
    - Brown Out Detection (BOD)
  - **Peripherals**
    - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
    - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
    - 1x USART with fractional baud rate generator and start-of-frame detection
    - 1x controller/peripheral Serial Peripheral Interface (SPI)
    - 1x Dual mode controller/peripheral I2C
    - 1x Analog Comparator (AC) with a scalable reference input
    - Watchdog Timer with separate on-chip oscillator
    - Six PWM channels
    - Interrupt and wake-up on pin change
- **ATMega16U2 Processor**
  - 8-bit AVR® RISC-based microcontroller
- **Memory**
  - 16 KB ISP Flash
  - 512B EEPROM
  - 512B SRAM
  - debugWIRE interface for on-chip debugging and programming
- **Power**
  - 2.7-5.5 volts

Fuente: Arduino (2023)

## Anexo 4 Descripción de pines del microcontrolador Arduino Uno R3.

### 5.1 JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

### 5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

Fuente: Arduino (2023)



## Anexo 5 Costo de microcontrolador Arduino Uno R3.

**Cesta de compra**

Producto	Cantidad	Precio por unid.	Total
<a href="#">Arduino UNO R3 [VERSION GENERICA]</a>	<input type="text" value="1"/>	US\$18,00	US\$18,00 ✖
<b>Total</b>			<b>US\$18,00</b>

[Sigue comprando](#)
[Actualizar cesta](#)
[Pasar por caja](#)

Fuente: MicroJPM (2023)

## Anexo 6 Costo de sensor DHT22.

**Cesta de compra**

Producto	Cantidad	Precio por unid.	Total
<a href="#">DHT22 Humidity Temperature Sensor</a>	<input type="text" value="1"/>	US\$10,95	US\$10,95 ✖
<b>Total</b>			<b>US\$10,95</b>

[Sigue comprando](#)
[Actualizar cesta](#)
[Pasar por caja](#)

Fuente: MicroJPM (2023)

## Anexo 7 Costo de display liquid crystal 1602.

**Cesta de compra**

Producto	Cantidad	Precio por unidad.	Total
<a href="#">Basic 16x2 Character LCD (Color Azul)</a>	<input type="text" value="1"/>	US\$6,95	US\$6,95 ✖
<b>Total</b>			<b>US\$6,95</b>

[Sigue comprando](#)
[Actualizar cesta](#)
[Pasar por caja](#)

Fuente: MicroJPM (2023)

## Anexo 8 Costo de módulo de interfaz I2C para display 1602.

**Cesta de compra**

Producto	Cantidad	Precio por unidad.	Total
<a href="#">I2C Serial Interface Module For 1602</a>	<input type="text" value="1"/>	US\$3,50	US\$3,50 ✖
<b>Total</b>			<b>US\$3,50</b>

[Sigue comprando](#)
[Actualizar cesta](#)
[Pasar por caja](#)

Fuente: MicroJPM (2023)

## Anexo 9 Costo de jumpers macho-macho.

Cesta de compra

Producto	Cantidad	Precio por unid.	Total
<a href="#">Wire Jumper Macho-Macho (20 Unidades)</a>	<input type="text" value="1"/>	US\$2,00	US\$2,00 ✖
<b>Total</b>			<b>US\$2,00</b>

[Sigue comprando](#)
[Actualizar cesta](#)
[Pasar por caja](#)

Fuente: MicroJPM (2023)

## Anexo 10 Costo de breadboard mini modular.

Cesta de compra

Producto	Cantidad	Precio por unid.	Total
<a href="#">Breadboard - Mini Modular (White) - 170 Points</a>	<input type="text" value="1"/>	US\$2,95	US\$2,95 ✖
<b>Total</b>			<b>US\$2,95</b>

[Sigue comprando](#)
[Actualizar cesta](#)
[Pasar por caja](#)

Fuente: MicroJPM (2023)

## Anexo 11 Costo de cable alarma 4 hilos.

**Cesta de compra**

Producto	Cantidad	Precio por unidad.	Total
<a href="#">D006 Miyako Cable Alarma 4 Hilos [METRO] (METRO)</a>	<input type="text" value="1"/>	US\$1,10	US\$1,10 ✖
<b>Total</b>			<b>US\$1,10</b>

[Sigue comprando](#)
[Actualizar cesta](#)
[Pasar por caja](#)

Fuente: MicroJPM (2023)

## Anexo 12 Costo de adaptador 9VDC.

**Cesta de compra**

Producto	Cantidad	Precio por unidad.	Total
<a href="#">250-SA9V1A JCM Adaptador 9VDC.@1000mA</a>	<input type="text" value="1"/>	US\$10,25	US\$10,25 ✖
<b>Total</b>			<b>US\$10,25</b>

[Sigue comprando](#)
[Actualizar cesta](#)
[Pasar por caja](#)

Fuente: MicroJPM (2023)

## Anexo 13 Costo de modulo ethernet shield 5100.

Cesta de compra			
Producto	Cantidad	Precio por unid.	Total
<a href="#">Ethernet Shield Module W5100 Board For Arduino [Version Generica]</a>	<input type="text" value="1"/>	US\$32,95	US\$32,95 <span>✖</span>
Total			US\$32,95
Sigue comprando		Actualizar cesta	Pasar por caja

Fuente: MicroJPM (2023)

## Anexo 14 Datos técnicos del producto Vaisala HMT120.

## Technical data

### Models

Model	Measured parameters	Output
HMT120	RH + T	2 analog outputs, 4 ... 20 mA (loop-powered)
HMT130	RH + T	2 analog outputs, 0 ... 1 V, 0 ... 5 V, 0 ... 10 V, or user-defined between 0 ... 10 V
TMT120	T	1 analog output, 4 ... 20 mA (loop-powered)
TMT130	T	1 analog output, 0 ... 1 V, 0 ... 5 V, 0 ... 10 V, or user-defined between 0 ... 10 V

### Measurement performance

#### Relative humidity <sup>1)</sup>


Measurement range	0 ... 100 %RH
Accuracy <sup>2) 3)</sup>	
At 0 ... +40 °C (+32 ... +104 °F)	±1.5 %RH (0 ... 90 %RH) ±2.5 %RH (90 ... 100 %RH)
At -40 ... 0 °C and +40 ... +80 °C (-40 ... +32 °F and +104 ... +176 °F)	±3.0 %RH (0 ... 90 %RH) ±4.0 %RH (90 ... 100 %RH)
Factory calibration uncertainty at +20 °C (+68 °F)	±1.1 %RH (0 ... 90 %RH) ±1.8 %RH (90 ... 100 %RH)
Humidity sensor types	HUMICAP® 180R HUMICAP® 180V
Stability	±2 %RH over 2 years
Stability in typical HVAC applications	±0.5 %RH per year

#### Temperature

Measurement range	-40 ... +80 °C (-40 ... +176 °F)
Accuracy over temperature range:	
At +15 ... +25 °C (+59 ... +77 °F)	±0.1 °C (±0.18 °F)
At 0 ... +15 °C and +25 ... +40 °C (+32 ... +59 °F and +77 ... +104 °F)	±0.15 °C (±0.27 °F)
At -40 ... +0 °C and +40 ... +80 °C (-40 ... +32 °F and +104 ... +176 °F)	±0.4 °C (±0.72 °F)
Temperature sensor	Pt1000 RTD Class F0.1 IEC 60751


Fuente: Vaisala (2023)

### Anexo 15 Coste de producto Vaisala HMT120.



Humidity and Temperature Transmitter,  $\pm 1.5$  %RH,  
4...20 mA  
Product Code: HMT120  
Configuration Code: HMT120KB1A1B12A1A0A

Quantity: 1 ▼ USD\$845.00



---

Humidity and Temperature, HMP110 Stainless Steel Probe  
Plastic Grid with PTFE filter for Steel Probe  
3 m Cable for Remote Probe  
Channel 1 RH 0...100%RH  
Temperature Range T -40...+60 °C (-40...+140°F)  
Cable Gland for 2...6 mm Diameter Cable  
Metric Units  
With Display  
Output 4...20mA, 2-wire loop-powered

---

🕒 Delivery 7 days 📄

Subtotal	USD\$845.00
Taxes <span>📄</span>	USD\$0.00
Shipping	USD\$0.00
<b>Order Total</b>	<b>USD\$845.00</b>

Fuente: Vaisala (2023)

## Anexo 16 Datos técnicos del producto Vaisala Indigo202.

# Technical data

## General

- Color display (Indigo201: optional non-display version)
- USB connection to Vaisala Insight PC software for easy access to configuration and monitoring options.

## Indigo-compatible probes

Measurement type	Probe models
Humidity and temperature	HMP1, HMP3, HMP4, HMP5, HMP7, HMP8, HMP9
Temperature	TMP1
Dew point	DMP5, DMP6, DMP7, DMP8
Carbon dioxide	GMP251, GMP252
Vaporized hydrogen peroxide	HPP271, HPP272
Moisture in oil	MMP8



## Operating environment

Operating temperature	With display -20 ... +60 °C (-4 ... +140 °F) Without display -40 ... +60 °C (-40 ... +140 °F)
Storage temperature	-40 ... +70 °C (-40 ... +158 °F)
Chemical tolerance	Temporary exposure during cleaning: <ul style="list-style-type: none"> <li>• H<sub>2</sub>O<sub>2</sub> (6000 ppm, non-condensing)</li> <li>• Alcohol-based cleaning agents such as ethanol and IPA (max. 70 % concentrate)</li> </ul>
IP rating	IP65
Indoor/outdoor use	Indoor use

Fuente: Vaisala (2023)



Anexo 17 Coste del producto Vaisala Indigo202 junto a la sonda Vaisala HMP.

	Relative Humidity and Temperature Probe for General Purpose Product Code: HMP3	Quantity: 1	USD\$1,387.00
	INDIGO202 Digital Transmitter Product Code: INDIGO202 Configuration Code: INDIGO2021A2A0	Quantity: 1	USD\$692.00
• •			
			🕒 Delivery 7 days ⓘ
Subtotal			USD\$2,079.00
Taxes ⓘ			USD\$0.00
Shipping			USD\$0.00
<b>Order Total</b>			<b>USD\$2,079.00</b>

Fuente: Vaisala (2023)