

**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería en Computadores**

(Computer Engineering Academic Faculty)

**Programa de Licenciatura en Ingeniería en Computadores**

(Licentiate Degree Program in Computer Engineering)



**Automatización de horno de biomasa**

(Biomass furnace automation)

**Informe de Trabajo de Graduación para optar por el título de Ingeniero en  
Computadores con grado académico de Licenciatura**

(Report of Graduation Work in fulfillment of the requirements for the degree of Licentiate in Computer  
Engineering)

**Manuel Antonio Bojorge Araya**

**Cartago, Noviembre, 2025**

(Cartago, November, 2025)

# Hoja de aprobación

# Dedicatoria

# Agradecimientos

# Resumen

El presente proyecto consiste en el diseño e implementación de un sistema de control con monitoreo térmico basado en microcontrolador, destinado a optimizar la operación de un horno empleado en procesos térmicos de pequeña escala. El desarrollo se llevó a cabo aplicando los principios de la ingeniería en computadores, integrando componentes de hardware y software para garantizar un funcionamiento confiable, seguro y eficiente.

El sistema incorpora sensores de temperatura, presión y nivel de agua, cuyas lecturas son procesadas por un ESP32 encargado de regular el llenado del tanque y mantener la estabilidad del sistema. La interfaz de usuario permite supervisar el proceso en tiempo real mediante una aplicación web y una pantalla táctil, facilitando la interacción y el control tanto local como remoto.

Durante el desarrollo se consideraron aspectos de protección eléctrica y seguridad operativa, asegurando el cumplimiento de condiciones propias de entornos industriales.

Los resultados obtenidos demostraron que el sistema cumple con las especificaciones planteadas y representa una contribución significativa al campo de la automatización industrial de bajo costo para la empresa donde se desarrolla el proyecto, evidenciando el potencial de las plataformas abiertas para su aplicación en proyectos de ingeniería aplicada.

**Palabras clave:** Automatización, Internet de las cosas (IoT), ESP32, monitoreo remoto, interfaz web.

# Abstract

This project consists of the design and implementation of a microcontroller-based control and thermal monitoring system aimed at optimizing the operation of a furnace used in small-scale thermal processes. The development was carried out by applying the principles of computer engineering, integrating hardware and software components to ensure reliable, safe, and efficient operation.

The system incorporates temperature, pressure, and water level sensors, whose readings are processed by an ESP32, responsible for regulating tank filling and maintaining system stability. The user interface allows real-time monitoring through a web application and a touchscreen, facilitating both local and remote interaction and control.

During development, aspects of electrical protection and operational safety were considered to ensure compliance with conditions typical of industrial environments.

The results demonstrated that the system meets the established specifications and represents a significant contribution to the field of low-cost industrial automation for the company where the project was developed, highlighting the potential of open platforms for their application in applied engineering projects.

**Keywords:** Automation, Internet of Things (IoT), ESP32, remote monitoring, web interface.

# Índice general

	Página
<b>1. Capítulo 1: INTRODUCCIÓN.....</b>	<b>10</b>
1.1. Antecedentes del proyecto.....	10
1.1.1. Descripción de la organización.....	10
1.1.2. Descripción del área de conocimiento del proyecto.....	11
1.1.3. Trabajos similares encontrados.....	11
1.2. Planteamiento del problema.....	12
1.2.1. Contexto del problema.....	12
1.2.2. Justificación del problema.....	13
1.2.3. Enunciado del problema.....	13
1.3. Objetivos del proyecto.....	13
1.3.1. Objetivo general.....	14
1.3.2. Objetivos específicos.....	14
1.4. Alcances, entregables y limitaciones del proyecto.....	14
1.4.1 Alcance.....	14
1.4.2 Entregables.....	14
1.4.3 Limitaciones.....	15
<b>2. Capítulo 2: MARCO DE REFERENCIA TEÓRICO.....</b>	<b>16</b>
2.1 Hardware para Aplicaciones en Condiciones Adversas.....	17
2.1.1. Principios de Medición y Acondicionamiento de Señales.....	17
2.1.2. Aislamiento Galvánico y Protección.....	18
2.2 Comunicaciones para Sistemas IoT.....	19
2.2.1. Protocolos de Comunicación.....	19
2.2.2. Estructura e Intercambio de Datos.....	19
2.2.3. Consideraciones de Tiempo y Desempeño.....	20
2.3 Fundamentos de Control Automático.....	20
2.3.1. Modelado del Comportamiento del Sistema: Máquinas de Estados Finitos.....	20
2.3.2. Teoría de Control para Procesos Térmicos.....	20
2.3.3. Estrategias de Seguridad y Tolerancia a Fallos.....	21
2.4 Arquitectura y herramientas de Software.....	22
2.4.1. Patrones Arquitectónicos.....	22
2.4.2. Lenguajes y Frameworks.....	23
<b>3. Capítulo 3: MARCO METODOLÓGICO.....</b>	<b>24</b>
<b>4. Capítulo 4: DESCRIPCIÓN DEL TRABAJO REALIZADO.....</b>	<b>28</b>
4.1. Descripción del proceso de solución.....	28
4.2. Análisis de los resultados obtenidos.....	34
<b>5. Capítulo 5: CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>37</b>
5.1. Conclusiones.....	37
5.2. Recomendaciones.....	37
<b>6. Apéndices y anexos.....</b>	<b>39</b>
6.1. Apéndice 1: Implicaciones ambientales y de desarrollo sostenible.....	39
<b>7. Referencias bibliográficas.....</b>	<b>42</b>



# Índice de tablas

<b>TABLA No.</b>	<b>Descripción</b>	<b>Página</b>
1	Desglose Metodológico del Proyecto	27
2	Descripción de los estados del sistema	36



# Índice de figuras

<b>FIGURA No.</b>	<b>Descripción</b>	<b>Página</b>
1	Mapa conceptual del marco teórico	16
2	Circuito básico de termocupla	17
3	Optoaislador de fotodiodo/transistor	18
4	Concepto de PID	21
5	Flujo de desarrollo del proyecto	25
6	Boceto del sistema con la ubicación de sensores y actuadores	29
7	Un canal de la interfaz de señales digitales	30
8	Un canal de la interfaz de señales analógicas	30
9	Un canal de salida para controlar los actuadores	31
10	Interfaz gráfica de la aplicación web	32
11	Layout del HMI	33
12	Máquina de estados finitos	33
13	Arquitectura general	34
14	Simulación para la configuración del transistor como seguidor de emisor	35
15	Comparación de la señal de entrada con la señal de salida en la interfaz de señales analógicas	35

# Siglas y acrónimos

- **AC:** Corriente Alterna (*Alternating Current*).
- **ADC:** Convertidor Analógico-Digital (*Analog-to-Digital Converter*).
- **Airbnb:** Plataforma de marketplace de alojamientos turísticos (no requiere definición técnica, pero se incluye por contexto).
- **API:** Interfaz de Programación de Aplicaciones (*Application Programming Interface*).
- **Arduino:** Plataforma de prototipos electrónicos de código abierto (*Open-Source Electronics Prototyping Platform*).
- **CPU:** Unidad Central de Procesamiento (*Central Processing Unit*).
- **DC:** Corriente Directa (*Direct Current*).
- **EMC:** Compatibilidad Electromagnética (*Electromagnetic Compatibility*).
- **EMO:** Parada de Emergencia (*Emergency Machine Off*).
- **ESP32:** Microcontrolador con conectividad Wi-Fi y Bluetooth de *Espressif Systems*.
- **GPIO:** Entrada/Salida de Propósito General (*General Purpose Input/Output*).
- **GUI:** Interfaz Gráfica de Usuario (*Graphical User Interface*).
- **HMI:** Interfaz Hombre-Máquina (*Human-Machine Interface*).
- **HTTPS:** Protocolo Seguro de Transferencia de Hipertexto (*Hypertext Transfer Protocol Secure*).
- **I/O:** Entrada y Salida (*Input/Output*).
- **IDE:** Entorno de Desarrollo Integrado (*Integrated Development Environment*).
- **IoT:** Internet de las Cosas (*Internet of Things*).
- **JSON:** Notación de Objetos de JavaScript (*JavaScript Object Notation*).
- **LED:** Diodo Emisor de Luz (*Light Emitting Diode*).
- **MCU:** Unidad de Microcontrolador (*Microcontroller Unit*).
- **MQTT:** Protocolo de Telemetría de Colas de Mensajes (*Message Queuing Telemetry Transport*).
- **ODS:** Objetivos de Desarrollo Sostenible (*Sustainable Development Goals*).
- **PCB:** Placa de Circuito Impreso (*Printed Circuit Board*).
- **PIC:** Microcontrolador de la serie *Peripheral Interface Controller* (Microchip Technology).
- **PID:** Proporcional, Integral, Derivativo (*Proportional-Integral-Derivative*).
- **PLC:** Controlador Lógico Programable (*Programmable Logic Controller*).
- **PWM:** Modulación por Ancho de Pulso (*Pulse Width Modulation*).
- **REST:** Transferencia de Estado Representacional (*Representational State Transfer*).
- **SaaS:** Software como Servicio (*Software as a Service*).
- **TLS:** Seguridad en la Capa de Transporte (*Transport Layer Security*).
- **UI:** Interfaz de Usuario (*User Interface*).
- **UPS:** Sistema de Alimentación Ininterrumpida (*Uninterruptible Power Supply*).
- **URL:** Localizador Uniforme de Recursos (*Uniform Resource Locator*).
- **USB:** Bus Universal en Serie (*Universal Serial Bus*).
- **UX:** Experiencia de Usuario (*User Experience*).
- **Wi-Fi:** Tecnología de red inalámbrica (*Wireless Fidelity*).

# 1. Capítulo 1: INTRODUCCIÓN

En la era de la transformación digital, incluso los procesos más tradicionales pueden optimizarse para generar mayor eficiencia y valor. Este proyecto aborda el desafío de modernizar el sistema de calentamiento de agua, que actualmente depende de un proceso manual ineficiente en un horno de biomasa de la empresa EcoView, el cual es desarrollado por DigiControl. La automatización de este sistema no solo mejorará significativamente la experiencia de los huéspedes en las propiedades Airbnb, sino que también representa un paso hacia la optimización operativa.

A lo largo de este capítulo se establecen los fundamentos de la solución desarrollada por DigiControl. Se presentan los antecedentes y capacidades de la empresa, el área de conocimiento técnico requerida y trabajos previos que validan el enfoque. Se detalla el problema actual, se justifica la necesidad de la solución y se enuncian formalmente los objetivos y alcances del proyecto de automatización.

## 1.1. Antecedentes del proyecto

El desarrollo del presente proyecto se enmarca dentro de un contexto organizacional y técnico específico que establece los fundamentos para su ejecución. A continuación, se describe el perfil de la organización ejecutora, DigiControl, detallando su trayectoria, capacidades técnicas y experiencia relevante que la habilita para abordar la solución propuesta. También, se define el área de conocimiento multidisciplinario requerido para el desarrollo del sistema y se presentan trabajos previos de naturaleza similar que sirven como referencia y validan el enfoque técnico planteado.

### 1.1.1. Descripción de la organización

La empresa DigiControl cuenta con más de 25 años realizando proyectos de automatización con la última tecnología en el mercado, con proyectos utilizando:

- Programación de PLC [23], Arduino [24], Raspberry Pi [25], PIC [26].
- Diseño y construcción de Paneles eléctricos.
- Automatización de líneas de corte de madera, calderas, marcadores, dosificadoras de combustible.
- Sistemas de captación de datos.
- Realización de capacitación para los operarios.
- Entrega de Planos, diagramas y manuales de operación.

Además se hacen reparaciones a tarjetas electrónicas, realizando un diagnóstico y evaluación de la viabilidad de la reparación. Las reparaciones más frecuentes que se realizan son:

- Variadores de Frecuencia.
- Fuentes de poder.
- Arrancadores suaves.
- Controles en máquinas llenadoras, etiquetadoras, imprentas industriales.
- Pantallas HMI.

- Controladores de Velocidad DC.
- PLC's.

La empresa tiene 2 sedes: en Lourdes de Montes de Oca, 50 metros oeste de la universidad Fidelitas y en peñas blancas de Cachí, 2 kilómetros al sur de la iglesia de Cachí. Para contactos y más información se puede consultar el sitio web [1]

### 1.1.2. Descripción del área de conocimiento del proyecto

La automatización del horno además de electrónica y programación, demanda conocimientos de matemática, física y química. Esto incluye la implementación de un controlador PID (Proportional-Integral-Derivative) para regular la temperatura basado en ecuaciones diferenciales [2], el modelado térmico del serpentín usando la ley de Fourier para la conducción de calor [3] y el análisis sobre la combustión de la biomasa [4] que se quema en el horno para identificar el nivel de temperatura alcanzado y el tiempo que se mantiene.

En este proyecto se aplican principios del Internet de las Cosas (IoT). Para ello, se utilizarán dispositivos comerciales que son útiles por su conectividad integrada de Wi-Fi y bajo consumo. Además permiten controlar dispositivos y adquirir datos por medio de sus salidas y entradas.

Para garantizar su funcionamiento, es necesario un proceso de verificación de hardware y software. Esto conlleva pruebas de estrés del hardware en ambientes extremos de temperatura y humedad, certificación de compatibilidad electromagnética (EMC) [5], así también como la validación del software mediante pruebas unitarias y una simulación completa del sistema antes de su implementación física.

Todo lo anterior se integra en el desarrollo de un sistema empujado autónomo. Su arquitectura es modular, dividida en una capa de sensores, una capa de actuadores y una capa de comunicación.

### 1.1.3. Trabajos similares encontrados

En la misma empresa donde se está desarrollando el presente proyecto, Digicontrol, se está desarrollando otro proyecto para la automatización de un horno eléctrico utilizado en el secado de transformadores de alto voltaje que se utilizan en el tendido eléctrico. Esta solución también se basa en una placa ESP32 para el control del sistema e incorpora sensores de temperatura gestionados por un controlador PID, lo que permite estabilizar de manera precisa la temperatura del proceso. Incluye sensores de corriente para detectar interrupciones en la alimentación eléctrica y activar protocolos correctivos. La interfaz de operador o HMI (Human-Machine Interface) se compone de una pantalla táctil que muestra el tiempo del proceso, así como las variables de temperatura y corriente, tanto numéricas como gráficamente. El control de arranque, parada y seguridad depende en gran medida de elementos físicos, como un switch que detecta si la puerta del horno está abierta para impedir el encendido, un botón de parada de emergencia EMO (Emergency Machine Off) para detener el sistema de forma inmediata, y un pulsador de marcha que solo permite el arranque si no existen condiciones de riesgo. Adicionalmente, el sistema cuenta con acceso

remoto a través de los servicios de Arduino Cloud, lo que habilita el monitoreo y la supervisión desde ubicaciones externas.

Otro trabajo similar que sirve como referencia es la automatización de un horno industrial a gas con PLC y su monitoreo a distancia que fue parte de un proyecto de graduación para la carrera de mecatrónica en la universidad politécnica salesiana con sede en Guayaquil, Ecuador [6]. Este sistema se basa en la integración de sensores de procesos, tales como termocuplas y medidores de presión que permiten al PLC realizar algoritmos sofisticados para el control de parámetros como la temperatura y la presión. Para la gestión del sistema se utilizó un método scada para un monitoreo local y para el monitoreo a distancia se lo realizó por medio de Node Red.

Otra referencia significativa es el diseño e implementación de un sistema de control y automatización para un horno eléctrico, realizado como parte de un trabajo de graduación para el título de ingeniero electrónico en la universidad nacional autónoma de México (UNAM) [7] el cual sería utilizado durante el proceso de fabricación de una determinada familia de productos. El sistema diseñado tiene como objetivo principal el control de la temperatura interna del horno, para dicho propósito incorpora sensores de temperatura industriales, y resistencias eléctricas como fuente de calentamiento. Dichos componentes son empleados y monitoreados por un controlador industrial programable, el cual, por medio de un lazo cerrado y utilizando el algoritmo PID, calcula el tiempo de encendido y apagado de las resistencias en un periodo definido para regular la temperatura interna del horno, tomando como referencia las lecturas registradas por los sensores. Se incorporan ventiladores impulsados por motores eléctricos para favorecer la transferencia del calor generado por las resistencias. Además, se incluyeron mecanismos de protección para garantizar la correcta operación de los actuadores, y se instalaron sensores de proximidad inductivos para complementar el monitoreo operativo del horno.

## 1.2. Planteamiento del problema

En el sector de hospedaje turístico, la expectativa de privacidad y comodidad constituye un elemento fundamental para la satisfacción del cliente. Sin embargo, esta expectativa se ve comprometida cuando la prestación de servicios básicos, como el suministro de agua caliente, depende de procesos manuales anticuados que requieren intervención constante. EcoView, empresa dedicada al alojamiento turístico, enfrenta precisamente este desafío en su sistema de calentamiento de agua por biomasa. A continuación, se describe el contexto específico del problema, se presenta una justificación detallada de su relevancia y se formula el enunciado central que este proyecto de automatización busca resolver.

### 1.2.1. Contexto del problema

La empresa DigiControl tiene un cliente llamado EcoView el cual cuenta con casas para alojamiento mediante la plataforma Airbnb. Actualmente, para calentar agua para ducha, jacuzzi y fregadero utiliza un horno de biomasa, principalmente leña que se obtiene de las ramas de los árboles que se cortan debido a que obstruyen el camino y el paso de luz para las plantas del lugar. A través de este horno pasa un serpentín (tubo largo en espiral o en zigzag) usado para facilitar el calentamiento del agua. El agua caliente que sale del serpentín se almacena en un tanque. Este proceso debe hacerse manualmente, es decir,

abrir y cerrar las llaves del paso de agua según el nivel de agua caliente en el tanque (puesto que el nivel varía dependiendo del uso) además de verificar que el horno esté encendido el tiempo necesario a una temperatura adecuada.

### 1.2.2. Justificación del problema

El sistema manual actual genera una serie de consecuencias negativas que justifican plenamente la necesidad de una intervención técnica. La dependencia de al menos un operario para regular el suministro de agua caliente resulta en una prestación de servicio inconsistente que afecta la comodidad y satisfacción de los huéspedes, impactando directamente la experiencia de su estadía.

La ausencia de un sistema de control y retroalimentación automática conlleva a un sobreconsumo de biomasa, ya que el horno podría permanecer encendido por períodos más prolongados de los estrictamente necesarios al no estar su operación ligada a la demanda real ni al estado del tanque de almacenamiento. Esta ineficiencia se traduce directamente en un aumento de los costos operativos y en un uso subóptimo de los recursos energéticos.

La problemática se extiende también al ámbito de la privacidad y autonomía, donde la necesidad de que el personal ingrese constantemente a las áreas de los huéspedes para manipular llaves o verificar el sistema invade la privacidad que se espera en este tipo de alojamientos.

Finalmente, el proceso actual se revela como inherentemente no escalable. Al estar supeditado por completo a la disponibilidad y atención constante de al menos un operario, cualquier aumento en la cantidad de unidades de alojamiento o en la demanda de agua caliente aumentan estas ineficiencias, haciendo inviable el crecimiento futuro del negocio bajo el modelo operativo actual. Por estas razones, la modernización del sistema se presenta no como una mera mejora incremental, sino como una necesidad para garantizar la calidad del servicio, la sostenibilidad operativa y la competitividad a largo plazo del negocio.

### 1.2.3. Enunciado del problema

El proceso manual de regulación de temperatura y flujo de agua en el horno de biomasa en EcoView genera altos costos operativos, ineficiencia energética e inconsistencia en la calidad del servicio, debido a que depende de la presencia de operarios que no deben descuidar el sistema de calefacción de agua.

## 1.3. Objetivos del proyecto

Una vez identificado y justificado el problema central, se establecen los objetivos que guiarán el desarrollo de la solución de automatización. Estos objetivos se han formulado para garantizar que el proyecto aborde de manera integral y metódica cada una de las deficiencias del sistema manual actual. El objetivo general define la meta global del sistema a implementar, mientras que los objetivos específicos detallan las fases críticas de diseño, desarrollo y validación necesarias para asegurar su éxito técnico y operativo. A

continuación, se presentan de manera formal el objetivo general y los objetivos específicos que constituyen el marco de acción de este proyecto.

### 1.3.1. Objetivo general

Implementar un sistema de control automático basado en IoT, utilizando un microcontrolador, sensores y actuadores; que gestionará la temperatura en un horno, el flujo de agua a través de tuberías y el nivel de agua en un tanque; para eliminar la dependencia de operarios y mejorar la experiencia de los usuarios.

### 1.3.2. Objetivos específicos

1. Diseñar la arquitectura hardware del sistema, mediante la selección e integración de los elementos adecuados, para crear una base robusta y fiable que opere bajo las condiciones de funcionamiento.
2. Desarrollar el stack de software que gestionará los datos y hará la comunicación bidireccional entre el usuario y el sistema con la programación del código embebido en el microcontrolador y una aplicación web con interfaz gráfica, con el propósito de habilitar el monitoreo remoto en tiempo real y la notificación de alertas.
3. Validar el funcionamiento del sistema automatizado en condiciones reales de operación, verificando el control de la temperatura, el flujo de agua y el nivel del tanque, así como su eficiencia para compararlo con el proceso manual.

## 1.4. Alcances, entregables y limitaciones del proyecto

### 1.4.1 Alcance

El sistema se centrará en el control de tres variables: la temperatura del horno, el flujo de agua en el serpentín y el nivel de agua en el tanque de almacenamiento.

Dentro de las actividades incluidas se encuentran:

- Diseño e implementación de un circuito con sensores (temperatura, nivel) y actuadores (válvulas) controlados por un dispositivo embebido ESP32.
- La creación de una aplicación web con una interfaz gráfica que permita la visualización de datos en tiempo real.
- La puesta en marcha del sistema y la documentación para su operación.

### 1.4.2 Entregables

Al finalizar el proyecto, se entregarán los siguientes productos:

1. Hardware:.

- a. Diagramas que permitan construir otro sistema con características similares: Esquemáticos del circuito electrónico, planos de conexión y cableado (diagramas de interconexión), y lista de materiales.
  - b. Un sistema de control electrónico funcional: Prototipo físico que incluye el microcontrolador ESP32, sensores de temperatura y nivel, actuadores (válvulas solenoides), fuentes de alimentación y protecciones
2. Software:
- a. Una aplicación web funcional: Interfaz gráfica de usuario accesible desde un navegador web, que muestre las variables del proceso (temperatura y nivel de agua) y el estado de los actuadores.
  - b. Comunicación entre el ESP32 y la aplicación web: Protocolo de comunicación bidireccional que garantice el intercambio de datos entre el dispositivo embebido y la aplicación web
  - c. Código fuente: Repositorio que contenga todo el código desarrollado, debidamente comentado y documentado. Esto incluye el firmware embebido escrito para el ESP32 y el código fuente de la aplicación web.
3. Documentación:
- a. Manual de instalación y configuración del sistema: Documento técnico que guíe el proceso de instalación física de los componentes, el cableado, la configuración inicial del hardware y la instalación del software en el entorno de producción.
  - b. Manual de operación y mantenimiento: Documento dirigido al usuario final y al personal técnico, que describa los procedimientos para la operación diaria del sistema, la interpretación de la interfaz, la resolución de problemas básicos y las actividades de mantenimiento preventivo recomendadas para garantizar la longevidad del sistema.

### 1.4.3 Limitaciones

#### Temporales:

- La fecha de finalización e implementación fijada.

#### Técnicas y Ambientales:

- Todos los componentes electrónicos, en particular los sensores ubicados en la cámara de combustión y el tanque de agua, deben estar especificados para soportar condiciones ambientales extremas, como temperaturas superiores a los 300°C.
- La conectividad a Internet en EcoView podría ser intermitente o de bajo ancho de banda.

#### Recursos:

- El proyecto depende críticamente de que DigiControl provea todos los componentes electrónicos según las especificaciones técnicas acordadas y dentro de los plazos establecidos.
- Se requiere que EcoView facilite el acceso al espacio físico para la instalación y la disponibilidad de su personal para colaborar en el proceso.



## 2. Capítulo 2: MARCO DE REFERENCIA TEÓRICO

El desarrollo de un sistema de automatización con comunicación a Internet para el control del horno de biomasa requiere la integración multidisciplinaria de principios de electrónica, teoría de control, protocolos de comunicación y desarrollo de software. Este marco de referencia teórico tiene como propósito establecer conceptos, definiciones y fundamentos que sustentan técnicamente el diseño, la implementación y la operación del sistema propuesto. El capítulo se estructura a partir de un mapa conceptual que sintetiza las relaciones entre sus componentes principales los cuales son: Hardware, Software, Comunicación y Control.

La Figura 1 presenta el mapa conceptual del sistema, que organiza los conceptos teóricos en categorías y muestra los principales aspectos de cada una de estas.

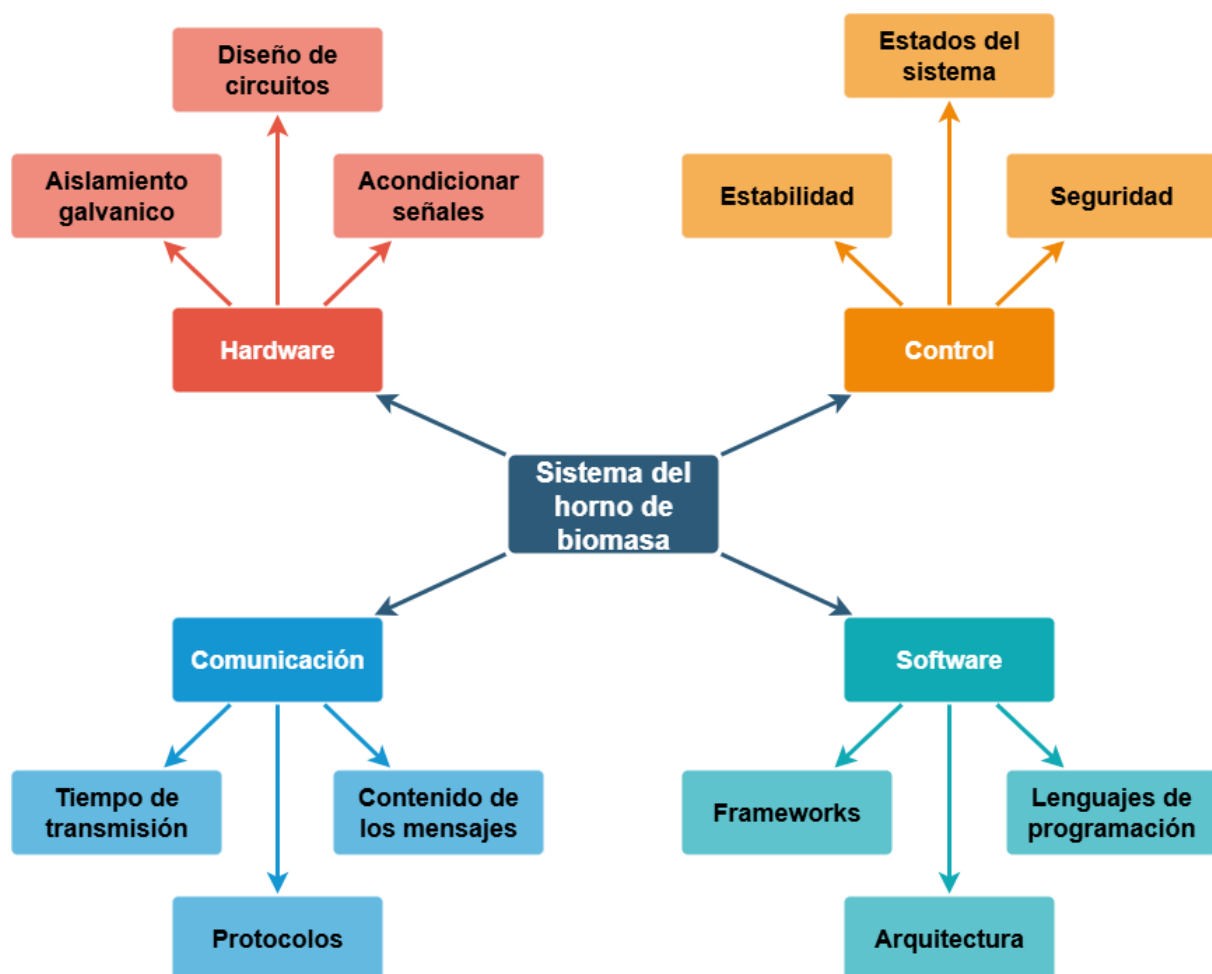


Figura 1: Mapa conceptual (elaboración propia).

A continuación se presenta la fundamentación conceptual que sustenta el diseño del sistema. Esta se estructura en cuatro partes: el hardware, que establece los principios de medición, acondicionamiento de señales y protección para la interacción fiable con el entorno físico; las comunicaciones, que definen los protocolos, formatos de datos y parámetros temporales para el intercambio eficiente y seguro de información en sistemas IoT; el control, que proporciona los modelos formales (como las Máquinas de Estado Finito), algoritmos de regulación (como el control PID) y estrategias de tolerancia a fallos para garantizar la estabilidad y seguridad del proceso; y el software, que establece los patrones arquitectónicos, lenguajes de programación y frameworks para integrar la funcionalidad del sistema y proveer la interfaz de usuario.

## 2.1 Hardware para Aplicaciones en Condiciones Adversas

### 2.1.1. Principios de Medición y Acondicionamiento de Señales

La interacción con el mundo físico en un sistema de automatización requiere de una medición confiable. Esta interacción inicia con la correcta medición de variables analógicas, como la temperatura, y su acondicionamiento para ser interpretadas por los sistemas digitales. Como señalan Figliola y Beasley [8], un sistema de medición toma una cantidad de entrada y la transforma en una señal de salida que puede ser observada o registrada, siendo fundamental comprender la naturaleza de la señal de entrada para seleccionar el sistema de medición adecuado.

Para la medición de temperaturas elevadas, como las presentes en una cámara de combustión, los termopares son sensores ampliamente utilizados. Su principio de operación se fundamenta en el Efecto Seebeck, el cual, como se define como, "la generación de un potencial de voltaje, o fem, en un circuito abierto de termopar debido a una diferencia de temperatura entre las uniones en el circuito" [8] en la Figura 2 se muestra una representación gráfica de este efecto. Una limitación inherente de los termopares es la necesidad de compensar la temperatura de la unión de referencia (unión fría). Para resolver este desafío, existen circuitos integrados especializados que realizan la compensación de unión fría y digitalizan la señal. Uno de estos dispositivos, el MAX31855, es descrito por el fabricante como un conversor que "realiza la compensación de unión fría y digitaliza la señal de un termopar tipo K... [con] una resolución de 14 bits y una precisión de  $\pm 2^{\circ}\text{C}$ " [9]. La integración de este tipo de componentes reduce el tiempo de diseño y mejora la confiabilidad del sistema.

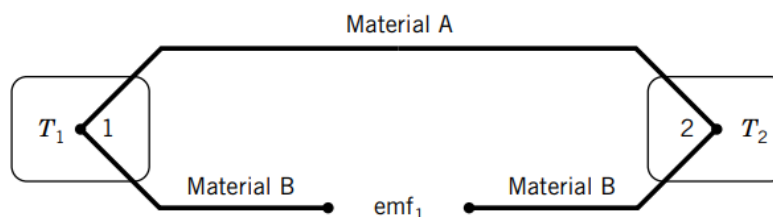


Figura 2: Circuito básico de termocupla [8]. Dos materiales diferentes (A y B) unidos en dos puntos (1 y 2) que se encuentran a temperaturas distintas ( $T_1$  y  $T_2$ ). La diferencia de temperatura entre las uniones genera una fuerza electromotriz ( $emf_1$ ).

Posterior al sensado, el acondicionamiento de señales se deben adaptar las señales del sensor a los rangos de voltaje aceptados por los microcontroladores modernos. Para esta tarea, los amplificadores operacionales (op-amps) son componentes muy útiles. Horowitz y Hill [10] los presentan como "bloques de construcción de ganancia casi perfectos" para el diseño de circuitos analógicos, destacando su papel central en la implementación de sus diferentes configuraciones, como el seguidor de voltaje, este circuito, gracias a su alta impedancia de entrada y baja impedancia de salida, es ideal para evitar la carga del sensor y transferir la señal de manera efectiva a las etapas posteriores de procesamiento [10].

### 2.1.2. Aislamiento Galvánico y Protección

En entornos industriales adversos, la presencia de transitorios de voltaje y diferencias de potencial entre tierras constituye un riesgo significativo para los componentes electrónicos sensibles. La implementación de aislamiento galvánico emerge como una técnica de protección indispensable. Avago Technologies [11] define el propósito principal de un acoplador óptico (optoacoplador) como "proporcionar aislamiento eléctrico y aislamiento de señal".

El aislamiento eléctrico, o capacidad de voltaje de soporte, se determina por la capacidad del dispositivo para "proteger el circuito circundante, así como a sí mismo, contra daños físicos resultantes de diferentes potenciales de voltaje" [11]. Este tipo de aislamiento crea una barrera dieléctrica entre dos partes de un circuito, permitiendo la transferencia de señales o potencia sin una conexión eléctrica directa. Los optoacopladores logran esto utilizando un LED y un fototransistor para transferir la información mediante luz, protegiendo efectivamente componentes de bajo voltaje, como los microcontroladores, de perturbaciones en las etapas de potencia [11]. En la Figura 3 se muestra un circuito simple con un dispositivo de este tipo en cual funciona como un transistor común, pero su corriente de base es controlada por la luz incidente en su región base, esto permite aislar eléctricamente el circuito de entrada del de salida, transmitiendo la señal sin conexión directa.

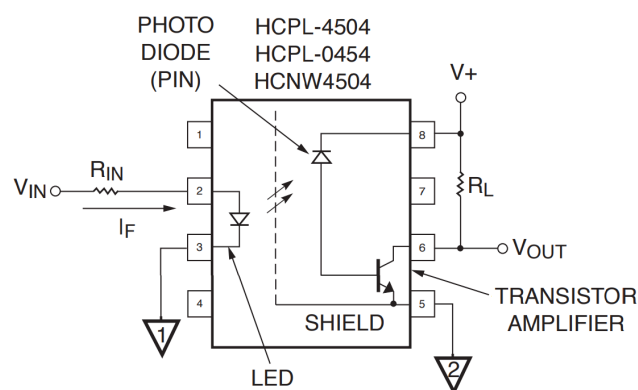


Figura 3: Optoaislador de fotodiodo/transistor [11]. Un LED convierte la señal eléctrica de entrada ( $V_{IN}$ ) en luz, la cual incide sobre un fotodiodo y un transistor que amplifica la señal en el lado de salida ( $V_{OUT}$ ).

## 2.2 Comunicaciones para Sistemas IoT

### 2.2.1. Protocolos de Comunicación

La arquitectura de un sistema IoT depende críticamente de la selección de protocolos de comunicación, los cuales deben satisfacer requisitos de eficiencia, latencia y confiabilidad acordes con el entorno de aplicación. Para el dominio de la telemetría, donde es común el monitoreo de variables desde dispositivos con recursos limitados, el protocolo MQTT (Message Queuing Telemetry Transport) es una opción predominante. Según la especificación de OASIS [12], MQTT es un "protocolo de transporte de mensajes de publicación/suscripción ligero, abierto, simple y diseñado para ser fácil de implementar", lo que lo hace ideal para entornos restringidos como Machine to Machine (M2M) e Internet de las Cosas (IoT). Su arquitectura de publicación/suscripción, gestionada por un *broker* o servidor central, "proporciona una distribución de mensajes uno-a-muchos y desacopla las aplicaciones" [12]. Este desacoplamiento facilita la escalabilidad del sistema. Además, MQTT define tres Calidades de Servicio (QoS) para la entrega de mensajes: "A lo sumo una vez" (pérdida de mensajes aceptable), "Al menos una vez" (pueden ocurrir duplicados) y "Exactamente una vez" [12], ofreciendo flexibilidad para diferentes necesidades de confiabilidad.

Para operaciones que requieren una seguridad robusta, como la autenticación de usuarios o el envío de comandos críticos, HTTPS (Hypertext Transfer Protocol Secure) es el estándar establecido. HTTPS extiende el protocolo HTTP, el cual es fundamental para el desarrollo de aplicaciones web. Como explica Chang y Mac Vittie [13], HTTP es un protocolo de solicitud-respuesta donde un cliente (como un navegador) establece una conexión TCP y envía una solicitud con un método (GET, POST, etc.) para obtener un recurso del servidor. HTTPS incorpora TLS (Transport Layer Security) sobre esta base, proporcionando cifrado, autenticación e integridad de los datos, protegiendo así las transacciones de accesos no autorizados [13].

Otras alternativas relevantes en el dominio IoT incluyen CoAP (Constrained Application Protocol), diseñado específicamente para dispositivos con recursos muy limitados, y WebSockets, que permite una comunicación bidireccional y en tiempo real sobre una única conexión TCP persistente.

### 2.2.2. Estructura e Intercambio de Datos

La eficiencia en la comunicación no solo depende del protocolo, sino también del formato de serialización de datos. Los formatos ligeros y auto-descriptivos son cruciales para minimizar la sobrecarga en la red. JSON (JavaScript Object Notation) es uno de los formatos más populares para la interoperabilidad en sistemas web e IoT. De acuerdo con Beeceptor [14], JSON es un "formato de intercambio de datos ligero, basado en texto e independiente del lenguaje", que se basa en dos estructuras: "una colección de pares nombre/valor (un objeto)

y una lista ordenada de valores (un array)". Esta estructura de pares clave-valor facilita el parsing automático y es "menos verboso que alternativas como XML" [14], lo que lo hace ideal para el intercambio de datos en aplicaciones donde el ancho de banda es un recurso valioso.

### 2.2.3. Consideraciones de Tiempo y Desempeño

Aunque muchos sistemas IoT no operan en tiempo real estricto, es fundamental considerar los parámetros de desempeño que afectan la capacidad de respuesta del sistema. Fielding [15] distingue entre el desempeño de la red (throughput, ancho de banda) y el desempeño percibido por el usuario, cuyas medidas primarias son la latencia y el tiempo de finalización. La latencia se define como "el período de tiempo entre el estímulo inicial y la primera indicación de una respuesta" e involucra retrasos en el procesamiento de la aplicación, la configuración de interacciones y la transmisión en la red [15].

Para garantizar un comportamiento robusto ante fallos en la red, es necesario implementar mecanismos de recuperación. La implementación de *timeouts* configurables y algoritmos de retransmisión adaptativa son esenciales. Estrategias como el backoff exponencial, que aumenta progresivamente el tiempo de espera entre reintentos, optimizan la recuperación de la conectividad sin saturar la red durante intentos repetidos [15]. Esto es particularmente importante para mensajes críticos, como alarmas, donde se deben establecer límites de latencia máxima para asegurar una respuesta oportuna.

## 2.3 Fundamentos de Control Automático

### 2.3.1. Modelado del Comportamiento del Sistema: Máquinas de Estados Finitos

El diseño de la lógica de control para sistemas automatizados complejos requiere de modelos formales que garanticen un comportamiento estructurado y libre de ambigüedades. Las Máquinas de Estados Finitos (FSM) constituyen una herramienta fundamental para este propósito. Como se establece en la literatura sobre automatización de diseño digital [16], "las máquinas de estado finito, como todos las conocemos de los cursos básicos de Diseño Digital, se representan en las entradas a los sistemas CAD mediante grafos de estado o por tablas de estado". Una FSM define un número finito de estados internos en los que el sistema puede existir, las transiciones entre esos estados (disparadas por condiciones de entrada) y las acciones asociadas a cada transición o estado. Este modelo permite especificar el comportamiento del sistema como un conjunto de transiciones de la forma (estado presente, estado de entrada, estado siguiente, estado de salida), proporcionando una base sólida para la implementación del controlador [16].

### 2.3.2. Teoría de Control para Procesos Térmicos

Para la regulación automática de variables de proceso como la temperatura, los controladores PID (Proporcional-Integral-Derivativo) representan el estándar industrial. La potencia de la realimentación (feedback) reside en su capacidad para "reducir los efectos de las perturbaciones, puede hacer que un sistema sea insensible a las variaciones del proceso y puede hacer que un sistema siga los comandos con fidelidad" [2]. El controlador PID es una implementación simple pero extremadamente efectiva de este principio.

El algoritmo PID calcula una acción correctiva basada en tres términos que consideran la evolución temporal del error ( $e = \text{setpoint} - \text{variable\_del\_proceso}$ ). Como lo describe Åström y Hägglund [2]:

- El término Proporcional (P) proporciona una respuesta inmediata al error presente:  
 $u(t) = K e(t)$
- El término Integral (I) elimina el error en estado estacionario mediante la acción acumulativa sobre la historia del error:  $u(t) = K_i \int_0^t e(\tau) d\tau$ . Esta es una propiedad crucial, ya que "si hay un estado estacionario y un controlador tiene acción integral, el error en estado estacionario es siempre cero" [2].
- El término Derivativo (D) proporciona una capacidad anticipatoria al responder a la tasa de cambio del error, actuando como una predicción lineal del error futuro:  
 $u(t) = K T_d \frac{de(t)}{dt}$  [2].

La combinación de estos tres términos ( $u(t) = K[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}]$ ) hace que el controlador PID sea capaz de resolver una amplia gama de problemas de control. La sintonización de las ganancias de cada término ( $K, T_i, T_d$ ) es crítica para lograr un equilibrio entre la velocidad de respuesta y la estabilidad del sistema [2]. La Figura 4 muestra una representación gráfica de este concepto.

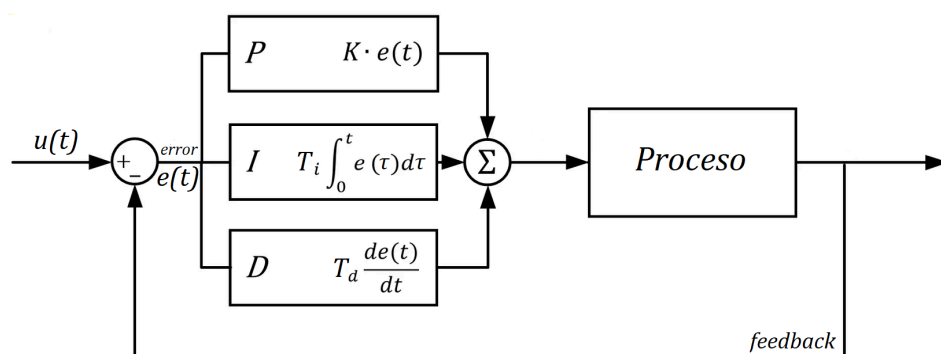


Figura 4: Concepto de PID (elaboración propia).

### 2.3.3. Estrategias de Seguridad y Tolerancia a Fallos

La seguridad operativa en sistemas de automatización industrial se aborda mediante un enfoque de "defensa en profundidad", que implica la implementación de múltiples capas de

protección sucesivas para mitigar riesgos. La ciberseguridad de estos sistemas es particularmente crítica, ya que, como señala el NIST [18], "los sistemas de control por computadora en tiempo real utilizados en aplicaciones de control de procesos tienen muchas características diferentes a los sistemas de procesamiento de información tradicionales", y la seguridad a menudo se omite en favor del rendimiento y la eficiencia.

Este enfoque de defensa en profundidad incluye estrategias para garantizar la estabilidad y el desempeño incluso ante fallos del sistema. La investigación en sistemas de control en tiempo real ha demostrado que, más allá de la estabilidad, es crucial analizar el desempeño y la recuperación del sistema tras eventos adversos, como ráfagas de fallos o pérdida de ciclos de control [17]. Entre las estrategias de tolerancia a fallos se encuentran:

- Mecanismos de Supervisión (Watchdog Timers): Circuitos o temporizadores de software que supervisan la correcta ejecución del programa y reinician el sistema en caso de bloqueo.
- Validación de Rangos y Consistencia: Verificación de que las lecturas de los sensores se encuentran dentro de límites plausibles y son consistentes entre sí, detectando así fallos en sensores o actuadores.
- Protocolos de Parada de Emergencia: Procedimientos predefinidos, a menudo gestionados a través de la Interfaz Humano-Máquina (HMI), para llevar el sistema a un estado seguro ante una condición de fallo crítica [18]. La elección de la estrategia de actuación (por ejemplo, mantener la última señal de control válida o ponerla a cero) durante un fallo afecta significativamente el desempeño del control durante el periodo de recuperación [17].

## 2.4 Arquitectura y herramientas de Software

### 2.4.1. Patrones Arquitectónicos

La arquitectura de software define la estructura fundamental de un sistema, trascendiendo la mera descripción de sus componentes. Como señalan Richards y Ford [19], "la industria en su conjunto ha tenido dificultades para definir con precisión la 'arquitectura de software'", proponiendo que ésta consiste no solo en la estructura, sino también en las características de arquitectura, las decisiones tomadas y los principios de diseño. La estructura, representada por el patrón arquitectónico, es un elemento central.

Los estilos arquitectónicos se clasifican principalmente en dos tipos: monolíticos y distribuidos [19]. Para aplicaciones de escala moderada, una arquitectura monolítica, definida como una "única unidad de despliegue de todo el código", puede ser ventajosa debido a su simplicidad de desarrollo, despliegue y prueba. Este estilo engloba patrones como la arquitectura en capas y la de microkernel. Por otro lado, para sistemas más complejos y con mayores requisitos de escalabilidad, las arquitecturas basadas en microservicios (un estilo distribuido) descomponen la aplicación en múltiples servicios independientes y débilmente acoplados. Si bien los estilos distribuidos ofrecen mayor potencia en términos de rendimiento y escalabilidad, conllevan una complejidad significativamente mayor asociada a las falacias de la computación distribuida [19].

## 2.4.2. Lenguajes y Frameworks

La elección de lenguajes de programación y frameworks está determinada por el dominio de aplicación y por las características intrínsecas de cada tecnología, las cuales contribuyen a su identidad y adecuación para resolver problemas específicos [20]. En el ámbito del desarrollo de software, esta selección influye directamente en aspectos como la eficiencia, la mantenibilidad y la productividad del proceso de ingeniería.

En el caso de los sistemas embebidos, la prioridad recae en la eficiencia y el control directo del hardware, por lo que lenguajes como C y C++ son ampliamente utilizados. Su sintaxis y semántica, definidas por un conjunto de reglas gramaticales formales [20], permiten una gestión precisa de los recursos limitados que caracterizan este tipo de sistemas. Asimismo, los entornos de desarrollo integrados y las librerías específicas de cada plataforma facilitan la creación de aplicaciones reduciendo la complejidad del desarrollo a bajo nivel.

Para el desarrollo de interfaces de usuario y servicios backend, JavaScript se ha establecido como el lenguaje estándar. Una de sus principales ventajas es la capacidad de permitir la unificación del desarrollo tanto en el frontend con frameworks de componentes como React o Vue.js; como en el backend con entornos de ejecución como Node.js. Esta homogeneidad en el stack tecnológico puede mejorar la productividad de los desarrolladores y la mantenibilidad del código [20, 21].

La utilización de frameworks es una práctica clave para lograr software de calidad. Un *framework* se define como "un conjunto integrado de artefactos de software... que colaboran para proporcionar una arquitectura reutilizable para una familia de aplicaciones relacionadas" [21]. Los frameworks proporcionan beneficios como la reutilización del diseño y la implementación, y guían a los desarrolladores a través de los pasos necesarios para crear software complejo de manera exitosa. Características como la inversión de control, donde el framework gestiona el flujo de ejecución y realiza llamadas al código de la aplicación, y su naturaleza de aplicación "semi-completa" los hacen poderosos para estructurar proyectos y amortizar el esfuerzo de desarrollo y validación [21].



### 3. Capítulo 3: MARCO METODOLÓGICO

El presente capítulo detalla la estrategia de ingeniería y el proceso sistemático seguido para materializar la solución de automatización definida en los capítulos anteriores. Aquí se describe la metodología, técnicas, herramientas y secuencia de actividades ejecutadas para cumplir con cada uno de los objetivos específicos del proyecto. El propósito de este marco es proporcionar una guía clara y reproducible que explique cómo se abordó el problema, integrando los fundamentos teóricos del Capítulo 2 para transformar los requisitos del Capítulo 1 en un sistema funcional y validado.

El proceso de desarrollo seguido se basa en los principios de la metodología ágil Scrum, adaptada al contexto del proyecto. Este enfoque se caracteriza por dividir el trabajo en iteraciones cortas o sprints, en las cuales se planifican, diseñan, implementan y prueban incrementos funcionales del sistema; para mantener una comunicación constante entre las etapas del proceso, reduciendo el riesgo de errores acumulativos y garantizando un producto final que responde efectivamente a las necesidades planteadas.

#### **Estrategia de Desarrollo**

La estrategia seleccionada para este proyecto se basa en un desarrollo iterativo e incremental, específicamente mediante el uso de prototipos evolutivos. Esta aproximación fue la más adecuada por las siguientes razones:

- La integración de hardware, software embebido, comunicaciones IoT y una aplicación web presenta una alta complejidad. El desarrollo iterativo permitió verificar el funcionamiento de cada subsistema de manera aislada antes de integrarlos, reduciendo el riesgo de errores en fases avanzadas mediante ciclos cortos de prueba y validación.
- La creación de prototipos evolutivos desde las primeras etapas permitió validar supuestos de diseño y funcionalidades en condiciones simuladas y posteriormente, condiciones reales. Este enfoque facilitó realizar correcciones oportunas y ajustes basados en retroalimentación real antes del despliegue final, asegurando que el sistema cumpliera con los requisitos operativos.
- Esta estrategia se alinea con los entregables definidos en el Capítulo 1, ya que cada iteración genera resultados tangibles como circuitos probados, módulos de software funcionales y versiones de la aplicación web que pueden ser verificados frente a los objetivos específicos. La naturaleza incremental del proceso asegura que cada componente sea validado antes de proceder con la integración del sistema completo.

En la Figura 5 se muestra el proceso seguido para el desarrollo del proyecto, este diagrama de flujo representa el ciclo iterativo, donde el desarrollo de prototipos y la verificación de su funcionalidad constituyen una retroalimentación constante. Si un prototipo no cumple los requisitos establecidos, se retorna a la lista de insumos y luego a la etapa de diseño, promoviendo la mejora continua y la flexibilidad ante cambios en los requerimientos.

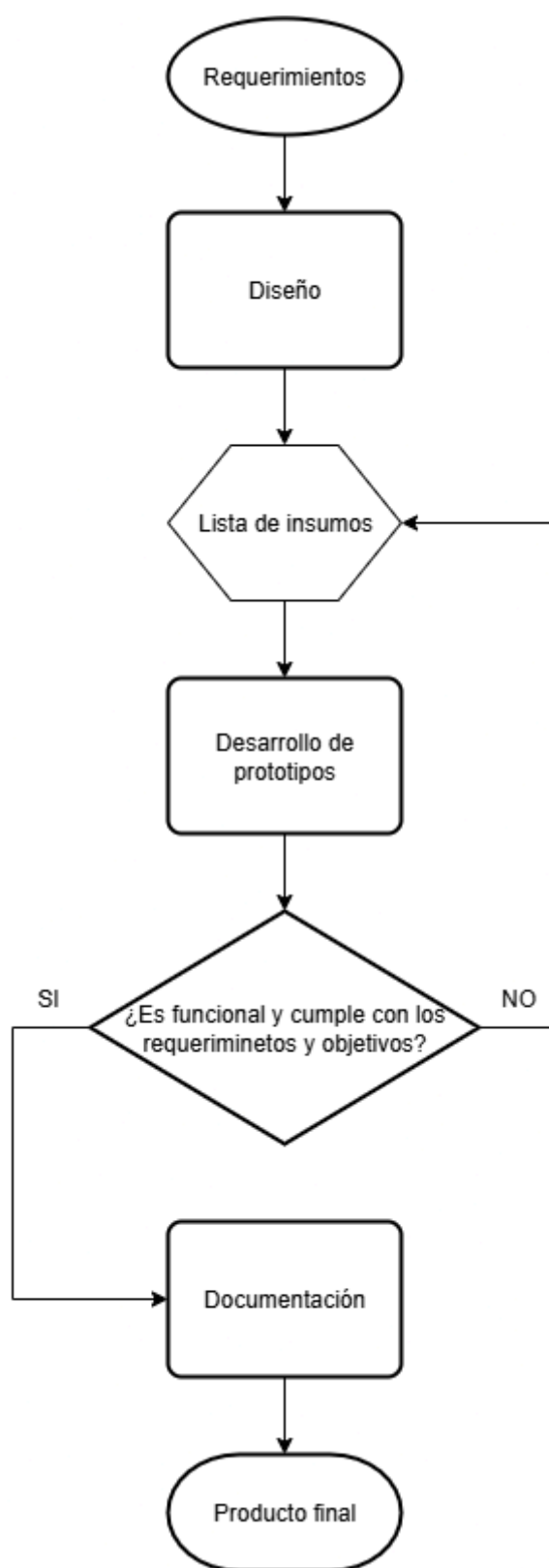


Figura 5: Flujo de desarrollo

## Herramientas elegidas

Las herramientas utilizadas durante el desarrollo del proyecto se seleccionaron considerando los requerimientos técnicos de diseño, simulación, programación, validación y despliegue del sistema.

Para el desarrollo de la arquitectura de hardware, en primer lugar se elige KiCad para la elaboración de los esquemáticos eléctricos y el diseño de las tarjetas de circuito impreso (PCB). Esta herramienta permite definir las interconexiones entre los distintos componentes, además de generar la lista de materiales y los planos listos para manufactura. Su naturaleza de código abierto, junto con su compatibilidad con una amplia variedad de librerías y componentes, la convierte en una opción idónea para este proyecto.

Complementariamente, Draw.io se usa para crear diagramas de bloques y bocetos para tener clara la idea del diseño del sistema, además Multisim se utiliza como entorno de simulación de circuitos electrónicos antes de su implementación física. Esta herramienta facilita la validación del comportamiento tanto de componentes analógicos como digitales, especialmente en lo relacionado con el acondicionamiento de señales provenientes de los sensores.

Asimismo, se utilizaron diversas herramientas de laboratorio, tales como el osciloscopio, multímetro, generador de funciones, fuentes de poder, protoboard, cables, cortadora de cable, cautín y estaño. Estos instrumentos son esenciales para la implementación y validación práctica del sistema.

GitHub se utiliza como sistema de control de versiones y repositorio principal del proyecto. Su uso permite mantener un registro de los cambios en el código fuente y coordinar las diferentes etapas del desarrollo de forma organizada.

El Arduino IDE es la plataforma utilizada para el desarrollo del firmware del microcontrolador ESP32. Este entorno permite programar el dispositivo en lenguaje C++, gestionar librerías y realizar la carga directa del código al hardware.

Por otra parte, el entorno Visual Studio Code se utiliza para el desarrollo tanto del frontend como del backend de la aplicación web, empleando tecnologías como HTML, CSS y JavaScript. Esta herramienta destaca por su flexibilidad, sus amplias extensiones para control de versiones y su integración con GitHub.

Finalmente, para el despliegue de la aplicación web se utiliza el servicio Render (render.com), el cual permite alojar la interfaz y establecer la comunicación remota con el dispositivo IoT. Esta plataforma de hosting en la nube es una solución práctica y estable para la etapa de desarrollo, pues facilita la accesibilidad del sistema desde cualquier navegador web.

En la Tabla 1 se resume para cada objetivo específico las actividades a realizar, los entregables asociados, las técnicas y herramientas utilizadas, y las estrategias para verificar y validar los resultados.

Tabla 1: Desglose Metodológico del Proyecto

Objetivo Específico	Entregables Asociados	Actividades	Técnicas y Herramientas	Estrategias de Verificación y Validación
1. Diseñar la arquitectura hardware del sistema.	1.a Diagramas (esquemáticos, planos de conexión, lista de materiales)	1.1. Especificar requisitos técnicos de sensores (temperatura, nivel) y actuadores (válvulas). 1.2. Seleccionar componentes electrónicos tolerantes a altas temperaturas (>300°C). 1.3. Diseñar esquemáticos electrónicos y planos de interconexión.	<b>Técnicas:</b> Diseño de circuitos y simulación.  <b>Herramientas:</b> Draw.io, KiCad, Multisim, herramientas de laboratorio.	<b>Verificación:</b> Revisión de planos y pruebas de circuitos según especificaciones.  <b>Validación:</b> Prueba del prototipo integrado bajo condiciones simuladas.
2. Desarrollar el stack de software.	2.a Aplicación web funcional 2.b Protocolo de comunicación 2.c Código fuente	2.1. Programar el firmware del ESP32 2.3. Implementar el protocolo para la comunicación. 2.4. Desarrollar la interfaz gráfica. 2.5. Integrar y probar la comunicación.	<b>Técnicas:</b> Programación embebida y desarrollo web full-stack  <b>Herramientas:</b> Arduino IDE, Visual Studio Code, render, GitHub.	<b>Verificación:</b> Pruebas unitarias y de integración del firmware y la aplicación web.  <b>Validación:</b> Demostración de comunicación estable y visualización correcta de datos.
3. Validar el funcionamiento del sistema.	1.b Sistema de control electrónico funcional 3.a Manual de instalación y configuración 3.b Manual de operación y mantenimiento	3.1. Integrar hardware y software en un prototipo. 3.2. Realizar pruebas 3.3. Validar el control de las variables en el entorno real. 3.5. Redactar documentación técnica y manuales.	<b>Técnicas:</b> Pruebas de integración y pruebas de estrés  <b>Herramientas:</b> banco de pruebas simulado y luego en el entorno real en EcoView	<b>Verificación:</b> Control de variables dentro de rangos y revisión de documentación.  <b>Validación Final:</b> Operación autónoma y confiable en el entorno real de EcoView.

## 4. Capítulo 4: DESCRIPCIÓN DEL TRABAJO REALIZADO

En esta sección se detalla el proceso de desarrollo e implementación del sistema propuesto, desde el diseño de la arquitectura hardware y software hasta la validación del funcionamiento. El contenido se estructura en dos partes: la primera describe el proceso seguido para alcanzar la solución final, la segunda presenta el análisis de los resultados obtenidos, los cuales comprueban el cumplimiento de los objetivos establecidos y la eficacia del sistema desarrollado.

### 4.1. Descripción del proceso de solución

Primero se hace la toma de requerimientos del sistema en el cual se especifica que el proyecto se debe realizar con ESP32, debe poder consultarse la información de manera remota y su principal función es regular la circulación de agua para que el usuario siempre disponga de agua caliente, tomando en cuenta las medidas de seguridad ante dos eventos críticos los cuales son que el horno esté muy caliente o que no haya agua disponible para llenar el tanque. Ya se contaba con algunos recursos disponibles como el horno, el serpentín, el tanque de agua y bombas de presión (para impulsar el agua es decir, promover la circulación de agua).

Luego se realiza un diseño general, el cual considera sensar la entrada de agua fría al tanque, también se considera medir al menos 3 niveles de agua dentro del tanque, en cuanto a los sensores de temperatura debido a que el principal producto del sistema es entregar agua caliente el primer sensor mide la temperatura dentro del tanque, por cuestiones de seguridad el segundo sensor mide la temperatura del horno el cual es la fuente de calor del sistema, debido a que el serpentín (por el cual pasará el agua para calentarse) estará dentro de la cámara de humos se decide colocar otro sensor ahí y para verificar la temperatura de agua que tendrá el usuario se coloca otro sensor de temperatura a la salida. Por razones de seguridad es importante que se mantenga la circulación de agua mientras el horno está caliente es por eso que se decide tener disponibles al menos 2 bombas que alternan su funcionamiento cada cierto tiempo.

La Figura 6 presenta un boceto del sistema que ilustra la integración de los diferentes componentes con base en los requerimientos. El diseño incorpora los cuatro sensores de temperatura explicados anteriormente. Asimismo, se incluye un sensor de presión en la tubería de entrada para detectar la disponibilidad de agua, función crítica para prevenir daños por falta de agua. Además, el sistema cuenta con dos bombas para la circulación de agua y sensores de nivel en el tanque que indican si se debe hacer la apertura o cierre de la válvula de llenado con agua fría.



física, en la Figura 7 se muestra uno de los canales (la interfaz cuenta con 8 canales iguales) el cual a su entrada tiene un pulsador que representa la señal en alto o en bajo, una resistencia para limitar la corriente y un LED para tener una indicación visual de cuál canal está en alto o activo; luego está el opto-transistor que acopla la señal con la siguiente etapa, en la salida hay una resistencia de pull-down en el emisor que es de donde se toma la señal que entrará al ESP32. Se consideró hacer un circuito más simple con una sola fuente y sin opto-acoplador para contrastar seguridad contra costo; se concluyó que estos elementos extra hacen un circuito más robusto y en caso de que el transistor quedara en cortocircuito las pérdidas económicas podrían ser mayores.

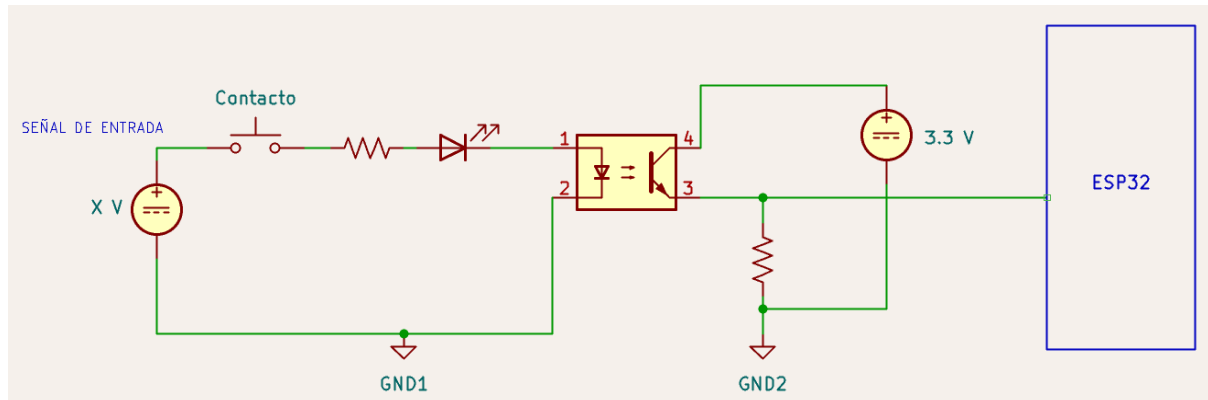


Figura 7: Un canal de la interfaz de señales digitales

Las señales analógicas provienen del sensor de presión el cual da una salida de 0V a 10V, esta señal necesita ser acondicionada, para esto se utiliza un circuito con un amplificador operacional en su configuración de seguidor de voltaje para aprovechar sus características de alta impedancia de entrada con el fin de evitar la carga del sensor y baja impedancia de salida para transferir la señal de manera efectiva; en la entrada positiva del operacional se coloca un capacitor de baja capacitancia para filtrar ruido que se pueda genera. Posterior a la salida del operacional se utiliza un transistor BJT que hará el escalamiento de la señal a un rango de 0V a 3.3V que es el voltaje que puede manejar el microcontrolador, en la Figura 8 se muestra uno de los canales de esta interfaz.

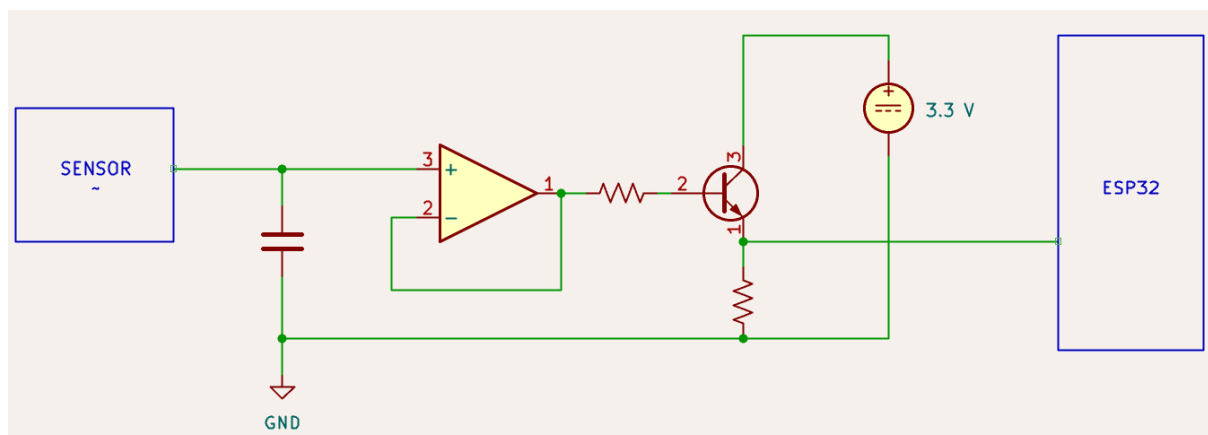


Figura 8: Un canal de la interfaz de señales analógicas

**Salidas:** Se programa y se arman/ensamblan los circuitos necesarios para encender y apagar las electroválvulas y las bombas. Estos elementos requieren de mayor potencia por lo tanto se necesita de elementos que puedan manejar estas características, entonces se decide utilizar relés los cuales serán activados con las señales que salen del ESP32 y nuevamente se hace uso de opto-acopladores y también un diodo de rodamiento en paralelo a la bobina del relé por cuestiones de seguridad, en la Figura 9 se ilustra uno de los canales para las salidas que activan o desactivan los actuadores.

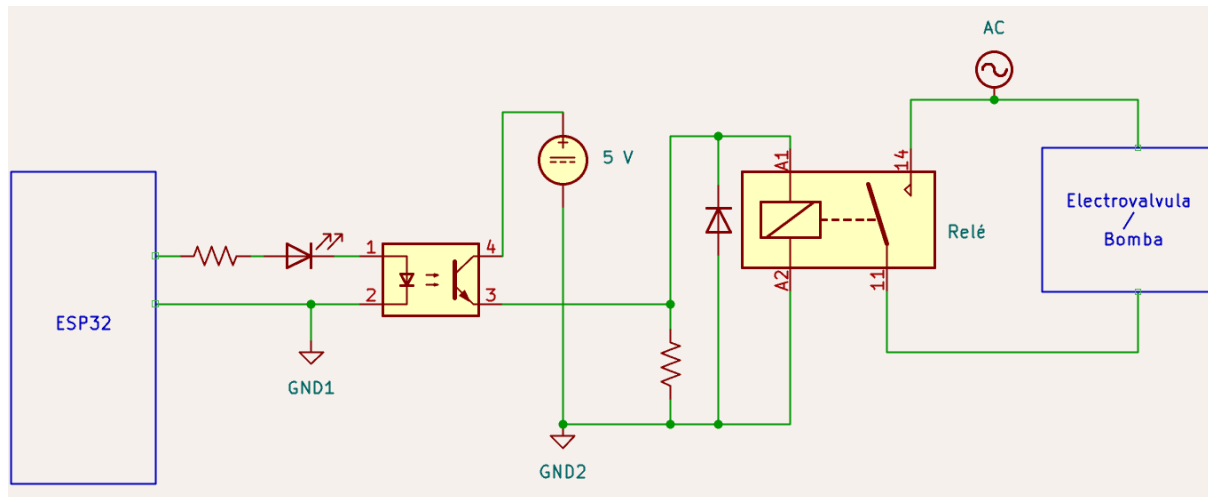


Figura 9: Un canal de salida para controlar los actuadores

**Interfaz web:** Se crea la aplicación para desplegar la información que proviene del microcontrolador. Se implementa una arquitectura monolítica por simplicidad debido a que es una aplicación pequeña además de reducir costos en el alojamiento en la nube en comparación a una arquitectura orientada a servicios. Se desarrolla utilizando el framework de Javascript Nodejs en conjunto con HTML5 y CSS3. En la Figura 10 se presenta la apariencia de la aplicación web, donde se visualizan los parámetros principales del sistema, tales como temperaturas, niveles de agua, presión y estado de los actuadores.



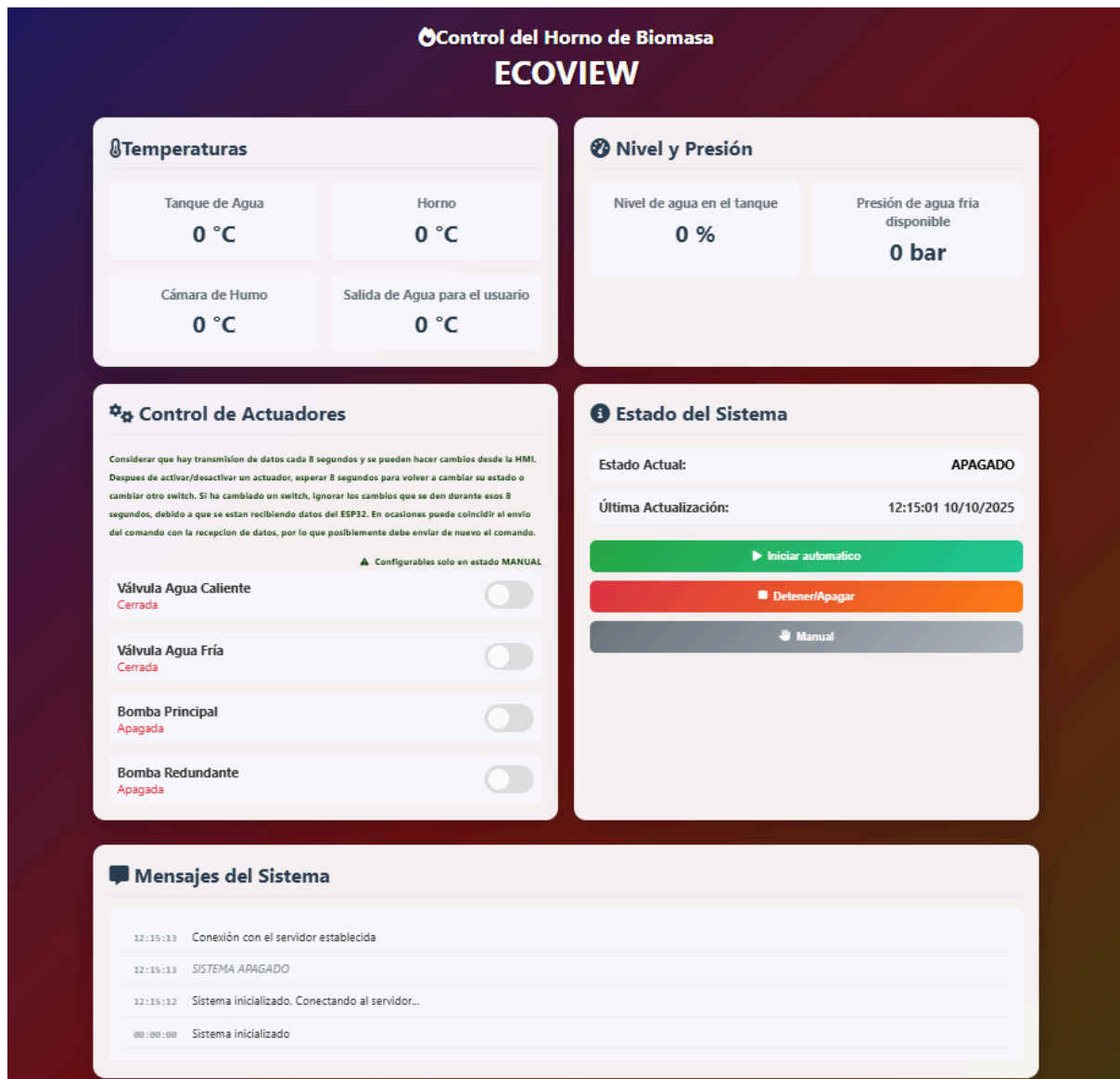


Figura 10: Interfaz gráfica de la aplicación web

**Comunicación:** Se hace la programación en el microcontrolador y en la aplicación web necesaria para crear la comunicación bidireccional. En este caso se tiene la posibilidad de utilizar ambos protocolos MQTT y HTTPS, sin embargo debido a las limitaciones del hosting el cual se ejecuta en la web bajo HTTPS, la seguridad de este protocolo bloquea los mensajes MQTT además de que el tiempo en el que se puede transmitir la información no es crítico es que se opta por utilizar HTTPS para la comunicación remota, pero con la posibilidad de usar MQTT (para una mayor eficiencia en la transmisión de información para otro proyecto que lo requiera) en una red local o en un servidor web realizando algunos ajustes.

**HMI:** La interfaz gráfica se diseña en Nextion Editor, herramienta propia para las pantallas de esta marca. La comunicación entre la pantalla HMI y el ESP32 se realiza mediante el protocolo UART (Universal Asynchronous Receiver-Transmitter), que permite el intercambio continuo de datos. Dado que la información mostrada requiere actualización constante, el canal UART permanece ocupado de forma permanente, lo cual dificulta la recepción de

datos provenientes de los botones de la pantalla. Para resolverlo, se implementa una calendarización en el envío de datos, permitiendo intervalos suficientes para que el microcontrolador reciba las señales generadas por la interacción del usuario. En la figura 11 se muestra la interfaz gráfica del HMI.



Figura 11: Layout del HMI

**Control:** Se gestiona el comportamiento del sistema con la implementación de una máquina de estados finitos, este comportamiento se ilustra en la Figura 12.

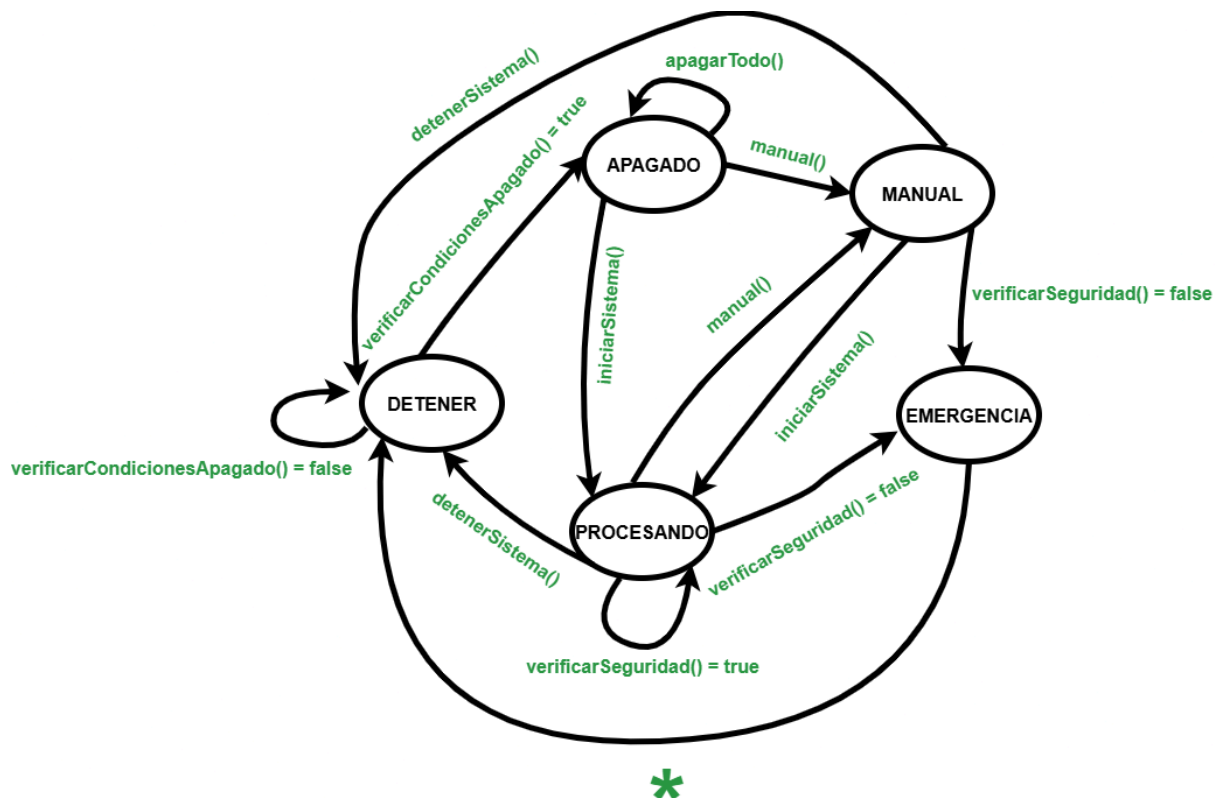


Figura 12: Máquina de estados finitos.

Con cada módulo se realiza un proceso iterativo de validación y verificación respecto a los requerimientos y objetivos establecidos. En caso de no cumplir con ellos, se evalúa la necesidad de materiales adicionales y se procede a ajustar o reconstruir el prototipo del módulo antes de repetir las etapas de validación y verificación.

Una vez validados y verificados los diferentes módulos, se procede a integrar el sistema completo. Con el sistema unificado, se realizan nuevas pruebas utilizando archivos que simulan distintos valores de los sensores, con el fin de inducir los diferentes estados de operación y efectuar los ajustes necesarios. Finalmente, una vez que el sistema funciona correctamente, se lleva a cabo la documentación de los manuales técnico y de usuario para el sistema.

## 4.2. Análisis de los resultados obtenidos

En la Figura 13 se presenta la arquitectura general del sistema desarrollado, donde la dirección de las flechas indica el sentido de la comunicación. Esta es bidireccional tanto para el HMI como para la aplicación web. La lectura de los sensores se realiza cada 2 segundos, mientras que las señales hacia los actuadores se envían inmediatamente después de que la instrucción llega al microcontrolador, dependiendo del origen del comando (HMI o aplicación web). Los datos enviados al HMI se transmiten con un intervalo de 100 milisegundos entre cada uno, mientras que en la aplicación web el envío y la consulta de información se alternan cada 4 segundos, repitiéndose el ciclo completo cada 8 segundos. A partir de las pruebas realizadas, se observó que ocasionalmente era necesario volver a presionar el botón de envío de comando, tanto en el HMI como en la aplicación web, debido a coincidencias temporales entre los procesos de envío y recepción de datos que podían impedir la correcta transmisión del comando.

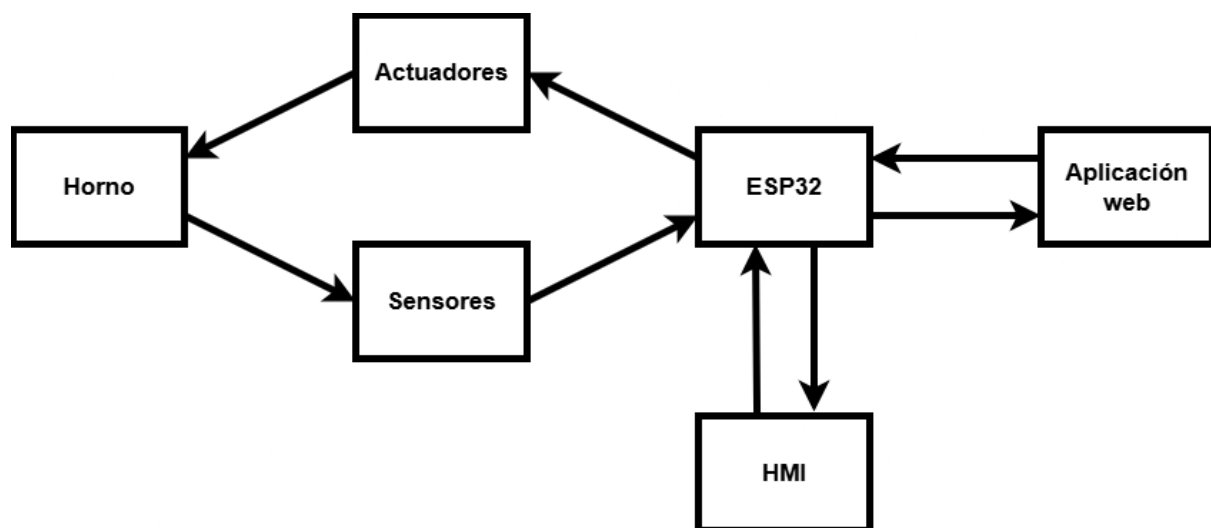


Figura 13: Arquitectura general

En la Figura 14 se muestra la simulación para el acondicionamiento y escalamiento de la señal analógica, en donde se aplica una señal que varía de 0 V a 10 V (5 voltios pico) a la base de un transistor BJT en configuración de emisor seguidor, el resultado obtenido en la salida del circuito es una señal que oscila entre 0 V y 3.1 V.

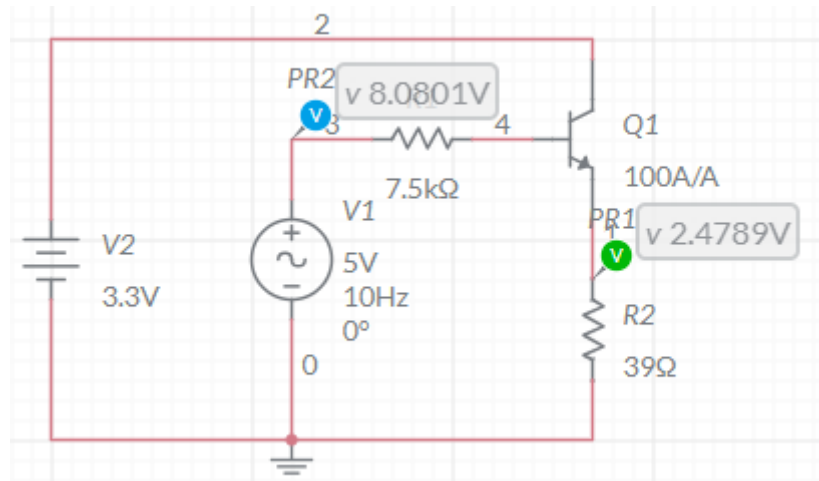


Figura 14: Simulación para la configuración del transistor como seguidor de emisor.

En la Figura 15 se muestra la comparación entre las señales de entrada (en color azul) y de salida (en color verde) en una gráfica de voltaje frente al tiempo. Se aprecia que la señal de salida conserva la misma forma que la de entrada, aunque con una amplitud menor. El circuito diseñado podría, de manera similar, recibir una señal con una amplitud distinta, por ejemplo, de 1 V, y mantener una salida dentro del rango de 0 V a 3.1 V. No obstante, para este proyecto únicamente se realizaron simulaciones y mediciones con osciloscopio utilizando las señales específicas empleadas en el sistema.

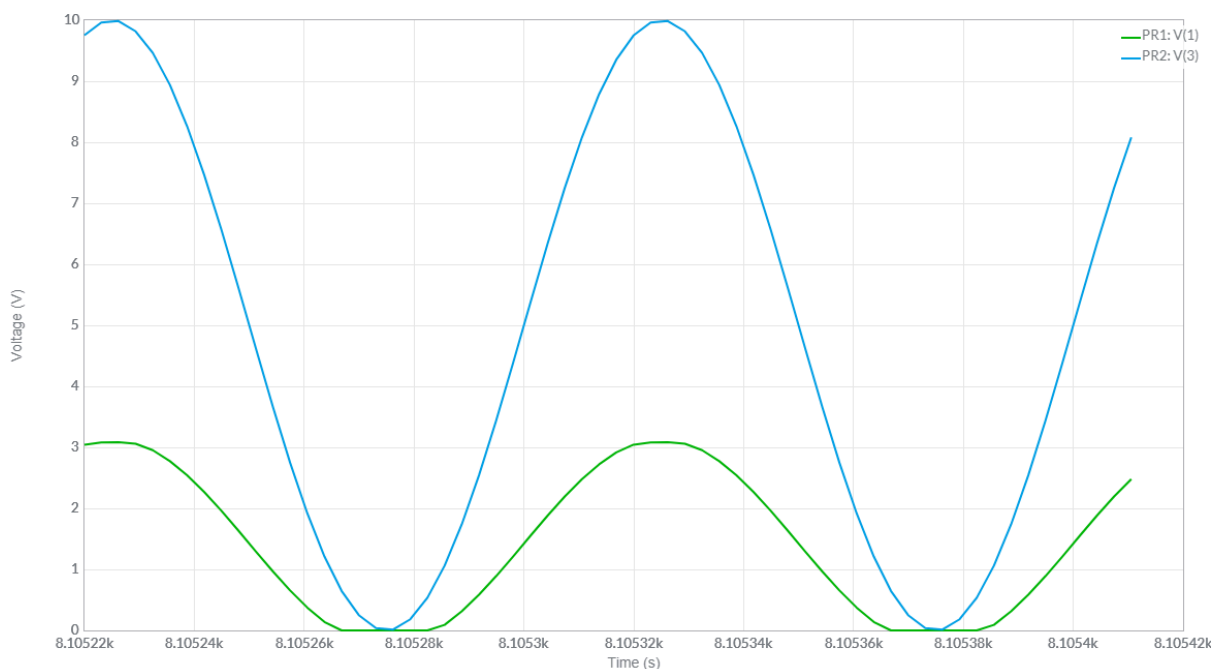


Figura 15: Comparación de la señal de entrada con la señal de salida de la interfaz de señales analógicas

En cuanto al funcionamiento del sistema como máquina de estados finitos, el comportamiento obtenido se resume en la Tabla 2, donde se hace una descripción general, las condiciones asociadas a cada estado y sus respectivas transiciones, confirmando que el algoritmo implementado opera conforme a lo esperado.

Tabla 2. Descripción de los estados del sistema

Estado	Descripción general	Condiciones de activación	Comportamiento de actuadores	Transición a otros estados
<b>APAGADO</b>	Estado de reposo total del sistema. Todos los actuadores permanecen desenergizados.	Se activa cuando el horno está frío o el sistema se apaga manualmente (después de que se enfríe el horno).	Bombas y válvulas apagadas.	Si las condiciones dejan de ser seguras (aumento de temperatura en el horno), pasa a <b>DETENER</b> .
<b>DETENER</b>	Estado intermedio de apagado controlado. Mantiene la circulación de agua para enfriar el sistema de forma segura.	Se activa cuando el horno aún conserva temperatura o en condiciones anormales.	Bombas encendidas alternadamente para circulación; válvulas según nivel del tanque.	Cuando el horno alcanza temperatura segura, pasa a <b>APAGADO</b> .
<b>PROCESANDO</b>	Estado normal de operación automática. Controla llenado del tanque y circulación del agua según temperatura y nivel.	Se activa al cumplir condiciones mínimas de inicio: temperatura adecuada, presión suficiente y nivel de agua aceptable.	<ul style="list-style-type: none"> <li>- Si el tanque está vacío: abre válvula de entrada.</li> <li>- Si el nivel es intermedio o alto, alterna las bombas para circulación.</li> <li>- Si el horno está frío: mantiene todo apagado hasta alcanzar la temperatura mínima.</li> </ul>	Si las condiciones dejan de ser seguras (baja presión, nivel vacío, sobrecalentamiento), pasa a <b>EMERGENCIA</b> .
<b>EMERGENCIA</b>	Estado de protección ante fallas críticas.	Se activa por baja presión, sobrecalentamiento del horno o tanque vacío con horno caliente.	Se apagan todas las bombas y válvulas para evitar daños.	Pasa automáticamente a <b>DETENER</b> .
<b>MANUAL</b>	Estado de control manual del sistema. Permite operar bombas y válvulas desde el HMI o la aplicación web.	Se activa por orden del usuario desde la interfaz únicamente si hay condiciones seguras.	Los actuadores responden directamente a los comandos recibidos.	Retorna a <b>APAGADO</b> o <b>PROCESANDO</b> según las condiciones y comandos del usuario.

## 5. Capítulo 5: CONCLUSIONES Y RECOMENDACIONES

Al finalizar el desarrollo del proyecto se cumplió con los objetivos planteados en el anteproyecto, tanto a nivel técnico como formativo.

A lo largo de su ejecución, se integraron los conocimientos adquiridos en las áreas de electrónica, control, programación y comunicaciones, demostrando la aplicabilidad del perfil del ingeniero en computadores en proyectos de automatización industrial con enfoque sostenible.

### 5.1. Conclusiones

En primer lugar, el proyecto consiguió integrar sensores de temperatura, presión y nivel, junto con actuadores (válvulas y bombas), de forma confiable y estable, validando la funcionalidad de los circuitos de acondicionamiento de señales diseñados. Las pruebas experimentales confirmaron que las señales eléctricas presentan una respuesta lineal, estable y dentro de los márgenes operativos definidos.

Asimismo, se logró automatizar el proceso de calentamiento de agua, eliminando la necesidad de supervisión manual con excepción de la carga de leña dentro del horno y encender fuego para quemar la biomasa para calentar el horno. El sistema desarrollado monitorea la temperatura del horno, controla el flujo de agua y gestiona la seguridad mediante una máquina de estados programada en el microcontrolador ESP32, lo que garantiza una operación segura y eficiente.

Desde el punto de vista de la interfaz, la implementación de una pantalla HMI y una aplicación web permite ofrecer un sistema amigable, intuitivo y de fácil monitoreo. La comunicación bidireccional se estableció correctamente.

En síntesis, el sistema desarrollado cumple con todos los objetivos específicos y entregables establecidos, demostrando la viabilidad técnica y económica de implementar soluciones de automatización inteligente basadas en microcontroladores.

El trabajo representa un aporte tangible para la empresa DigiControl y su cliente EcoView, consolidando una base tecnológica replicable y adaptable a nuevos escenarios de aplicación.

### 5.2. Recomendaciones

#### **Optimización del monitoreo:**

Se recomienda ampliar la funcionalidad de la aplicación web para incluir almacenamiento histórico de datos, gráficos de consumo y alertas automáticas mediante notificaciones o correo electrónico, de modo que se facilite la supervisión continua del sistema.

#### **Optimización del firmware:**

Implementar mejoras en la máquina de estados para permitir la reconfiguración de

parámetros como las temperaturas umbral, tiempos de bombeo y otros ajustes directamente desde la interfaz web o el HMI, sin necesidad de reprogramar el microcontrolador.

**Estandarización de protocolos IoT:**

Para futuras versiones, se sugiere migrar la comunicación a MQTT o WebSocket seguro, lo cual reduciría el uso de ancho de banda y mejoraría la escalabilidad del sistema para manejar múltiples instalaciones simultáneamente.

**Fuentes de energía renovables:**

Evaluar la posibilidad de incorporar fuentes de alimentación renovables como paneles solares para alimentar el sistema de control.

**Gestión de ciberseguridad:**

Dada la naturaleza conectada del sistema, se recomienda integrar autenticación por rol para prevenir accesos no autorizados.

**Expansión funcional:**

Se podría extender la automatización a otros subsistemas de las propiedades, como iluminación, control de acceso o monitoreo ambiental, aprovechando la infraestructura IoT existente para ofrecer soluciones integrales de domótica sostenible.

**Validación prolongada:**

Ejecutar pruebas de funcionamiento continuo durante varios meses permitirá recopilar datos reales de consumo, confiabilidad y mantenimiento, para fortalecer la validación del sistema.

## 6. Apéndices y anexos

### 6.1. Apéndice 1: Implicaciones ambientales y de desarrollo sostenible

Esta sección analiza los impactos y proyecciones del proyecto de automatización del horno de biomasa en los ámbitos social, económico, ambiental, legal, de seguridad y de salud. Se examina su contribución a los Objetivos de Desarrollo Sostenible (ODS) definidos por la Organización de las Naciones Unidas (ONU), así como las implicaciones que tiene para la sostenibilidad y el ejercicio profesional de la Ingeniería en Computadores.

#### Aspecto Social

- **Identificación y Análisis:**

Al garantizar un suministro constante de agua caliente sin intervención manual, se eleva significativamente el nivel de confort y se preserva la privacidad de los huéspedes, factores determinantes en la satisfacción del cliente dentro del sector de hospedaje turístico. Esta mejora operativa motiva la implementación de otros sistemas automatizados, lo cual fortalece la competitividad y reputación de la empresa EcoView, impulsando indirectamente el desarrollo económico regional a través de una oferta turística más atractiva y sostenible.

- **Relación con los ODS:**

La iniciativa se alinea con el ODS 9: Industria, Innovación e Infraestructura, al integrar tecnologías de vanguardia como IoT y automatización en un proceso tradicional, sentando las bases para infraestructuras turísticas más resilientes, inteligentes y adaptadas a las demandas actuales.

- **Proyecciones Futuras:**

Este sistema sienta un precedente para el desarrollo de infraestructuras turísticas de bajo impacto, como eco-lodges en zonas rurales, donde la tecnología potencia la experiencia sin comprometer el entorno. Para el ingeniero en computadores, esto es un campo de acción creciente en el diseño de soluciones tecnológicas, conectadas y centradas en el usuario final, que armonicen la innovación con la sostenibilidad social.

#### Aspecto Económico

- **Identificación y Análisis:**

La automatización introduce ganancias de eficiencia que se traducen en beneficios económicos directos. Al optimizar el consumo de biomasa y eliminar la necesidad de supervisión humana permanente, se reducen sustancialmente los costos operativos de EcoView, mejorando su rentabilidad. Paralelamente, para la empresa desarrolladora DigiControl, el proyecto consolida su portafolio en el mercado de IoT, posicionándose como un referente en la automatización de infraestructuras para el sector de hospedajes.

- **Relación con los ODS:**

Este impacto se enmarca en el ODS 8: Trabajo decente y crecimiento económico, al



demostrar cómo la innovación tecnológica es un motor para la optimización de recursos, el aumento de la productividad y la generación de nuevos modelos de negocio en economías locales.

- **Proyecciones Futuras:**

La evolución del proyecto podría apuntar hacia un modelo de Software como Servicio (SaaS), donde el valor se traslada del producto físico al monitoreo continuo y al mantenimiento predictivo remoto. Este escenario demandará del ingeniero en computadores competencias híbridas, integrando el conocimiento técnico con la analítica de datos, la gestión de servicios digitales y una comprensión de los modelos de negocio sostenibles.

## Aspecto Ambiental

- **Identificación y Análisis:**

El proyecto presenta un balance ambiental positivo, aunque con matices. Su principal contribución radica en la optimización del uso de la biomasa, el cual es un recurso renovable, minimizando el desperdicio y reduciendo las emisiones contaminantes asociadas a un uso excesivo e ineficiente de este recurso. No obstante, es imperativo reconocer y gestionar el impacto del ciclo de vida de los componentes electrónicos, mediante reciclaje y una cuidadosa selección de materiales.

- **Relación con los ODS:**

La iniciativa contribuye al ODS 7: Energía asequible y no contaminante, al promover el uso eficiente de fuentes renovables, y al ODS 12: Producción y consumo responsables, al establecer un modelo de gestión energética que minimiza el desperdicio y fomenta la economía circular al buscar reducir los residuos y la contaminación desde la etapa de diseño.

- **Proyecciones Futuras:**

El futuro de la ingeniería en computadores en este ámbito estará dominado por el diseño sostenible de hardware y los principios del Green IoT. Esto implica priorizar la eficiencia energética extrema, la modularidad para facilitar la reparación, el uso de materiales reciclados y el desarrollo de algoritmos de control que optimicen el proceso.

## Aspecto Legal

- **Identificación y Análisis:**

El despliegue del sistema abarca la seguridad eléctrica al aislar eléctricamente las diferentes etapas del circuito y rigiéndose bajo las regulaciones del uso de recursos hídricos por parte del proveedor de este servicio. En el ámbito digital, la aplicación web no gestiona datos personales personales, pero los datos de los sensores y comando se transmiten con seguridad mediante el protocolo cifrado como HTTPS.

- **Relación con los ODS:**

Este cumplimiento se relaciona con el ODS 9, al asegurar que la infraestructura tecnológica sea confiable y de calidad, y el ODS 16: Paz, justicia e instituciones sólidas, al fomentar la transparencia, la ética profesional y el respeto por los marcos legales en el desarrollo de tecnologías emergentes.

- **Proyecciones Futuras:**

La creciente interconexión de sistemas críticos exigirá que el ingeniero en computadores domine normativas internacionales especializadas, como la serie IEC 62443 sobre ciberseguridad industrial. El profesional deberá integrar la ciberseguridad y el cumplimiento legal como requisitos no negociables desde la fase inicial de diseño, asegurando la integridad y resiliencia de los sistemas automatizados.

## Aspecto de Seguridad

- **Identificación y Análisis:**

El sistema prioriza la seguridad operacional mediante una máquina de estados finitos que previene de forma autónoma condiciones de riesgo como sobrecalentamientos o funcionamiento en seco. Sin embargo, la conectividad a Internet introduce un vector de riesgo cibernético: un acceso no autorizado podría manipular parámetros críticos, pero esto, nuevamente, se mitiga con la seguridad operacional, pues aunque el sistema esté en el estado MANUAL, si se llegan a dar condiciones de riesgo, el sistema va a gestionar el apagado automático.

- **Relación con los ODS:**

Este enfoque en la seguridad integral sustenta el ODS 9, cuyo fin es construir infraestructuras resilientes, confiables y seguras, capaces de soportar y mitigar fallos operativos y cibernéticos.

- **Proyecciones Futuras:**

La seguridad se consolidará como un pilar transversal en el desarrollo de IoT. El ingeniero en computadores deberá adoptar de manera sistemática el paradigma de "Security by Design", que incorpora protocolos de comunicación robustos como el TLS, autenticación multifactorial y pruebas continuas de penetración desde la concepción misma del sistema, fomentando una cultura de confianza en la automatización.

## Aspecto de Salud

- **Identificación y Análisis:**

El impacto en la salud ocupacional es positivo. La automatización elimina la exposición directa de los operarios a fuentes de calor intenso reduciendo el riesgo de accidentes y enfermedades. Para el usuario final, el sistema garantiza un suministro de agua a temperaturas controladas y seguras.

- **Relación con los ODS:**

Esta contribución se vincula directamente con el ODS 3: Salud y bienestar, al crear entornos laborales más seguros y proteger el bienestar físico tanto de los trabajadores como de los huéspedes.

- **Proyecciones Futuras:**

Los futuros sistemas IoT evolucionarán de protectores pasivos a guardianes activos de la salud. El rol del ingeniero en computadores se expandirá hacia el diseño de sistemas que salvaguarden de manera integral la salud humana y ambiental.

## 7. Referencias bibliográficas

- [1] Digicontrol CR. Accedido el 1 de julio de 2025. [En línea]. Disponible: <https://digicontrolcr.com/>
- [2] "Advanced PID Control", p. 1, 2006. Accedido el 10 de agosto de 2025. [En línea]. Disponible: [https://www.isa.org/getmedia/fb0e41bc-e4f3-422a-9f67-b9bd31340e16/Advanced-PID-Contr ol\\_AstromHagglund\\_Chapter1-Introduction.pdf](https://www.isa.org/getmedia/fb0e41bc-e4f3-422a-9f67-b9bd31340e16/Advanced-PID-Contr ol_AstromHagglund_Chapter1-Introduction.pdf)
- [3] J. R. Welty, G. L. Rorrer y D. G. Foster, *Fundamentals of Momentum, Heat, and Mass Transfer*, 7ª ed. s. f. Accedido el 10 de agosto de 2025. [En línea]. Disponible: [https://dl.konkur.in/2025/04/Momentum-Heat2020-\[www.konkur.in\].pdf](https://dl.konkur.in/2025/04/Momentum-Heat2020-[www.konkur.in].pdf)
- [4] W Emmons, H., & Atreya, A. (s.f.). *The science of wood combustion*. <https://www.ias.ac.in/public/Volumes/sadh/005/04/0259-0268.pdf>
- [5] H. W. Ott, *Electromagnetic compatibility engineering*. 2009. Accedido el 11 de agosto de 2025. [En línea]. Disponible: <https://daskalakispiros.com/files/Ebooks/Electromagnetic%20Compatibility%20Engineering.p df>
- [6] "Automatización de horno industrial a gas con PLC y su monitoreo remoto", Universidad Politécnica Salesiana, 2024. Accedido el 12 de agosto de 2025. [En línea]. Disponible: <https://dspace.ups.edu.ec/bitstream/123456789/27761/1/UPS-GT005249.pdf>
- [7] "Diseño de sistema de control y automatización para horno eléctrico industrial", Universidad Nacional Autónoma de México, 2023. Accedido el 14 de agosto de 2025. [En línea]. Disponible: <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/RepoFi/18916/Tesis.pdf?sequen ce=8&isAllowed=y>
- [8] S. Figliola, R., & E. Beasley, D. (s.f.). *Theory and Design for Mechanical Measurements* (5ª ed.). [https://www.awa2el.net/sites/default/files/mech.measurm.6th.eng\\_.abdullahalkilani@gmail.com\\_.pdf](https://www.awa2el.net/sites/default/files/mech.measurm.6th.eng_.abdullahalkilani@gmail.com_.pdf)
- [9] Maxim Integrated. (s.f.). *Cold-Junction Compensated Thermocouple-to-Digital Converter. Mixed-signal and digital signal processing ICs | Analog Devices*. <https://www.analog.com/media/en/technical-documentation/data-sheets/max31855.pdf>
- [10] Horowitz, P., & Hill, W. (s.f.). *The Art of Electronics* (3ª ed.). [https://kolegite.com/EE\\_library/books\\_and\\_lectures/Електроника/\\_The%20Art%20of%20Electronics%203rd%20ed%20\[2015\].pdf](https://kolegite.com/EE_library/books_and_lectures/Електроника/_The%20Art%20of%20Electronics%203rd%20ed%20[2015].pdf)
- [11] Avago technologies. (s.f.). *Optocouplers Designer's Guide. Support Documents and Downloads*. <https://docs.broadcom.com/doc/AV02-4387EN>

- [12] OASIS OPEN. (s.f.). MQTT Version 3.1.1. <https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf>
- [13] Chang, P., & Mac Vittie, L. (s.f.). The Fundamentals of HTTP. <https://www.f5.com/pdf/white-papers/http-fundamentals-wp.pdf>
- [14] Web Data Serialization: JSON, XML, YAML & More Explained. (s.f.). Beeceptor - Rest & SOAP API Mock Server. <https://beeceptor.com/docs/concepts/data-exchange-formats/>
- [15] Fielding, R. (s.f.). Architectural Styles and the Design of Network-based Software Architectures [UNIVERSITY OF CALIFORNIA, IRVINE]. [https://ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation.pdf](https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf)
- [16] Portland State University. (s.f.). DIGITAL DESIGN AUTOMATION FINITE STATE MACHINE DESIGN. <https://web.cecs.pdx.edu/~mperkows/PerkowskiGoogle/finite-sm.pdf>
- [17] Vreman, N., Cervin, A., & Maggio, M. (s.f.). Stability and Performance Analysis of Control Systems Subject to Bursts of Deadline Misses. <https://drops.dagstuhl.de/storage/00lipics/lipics-vol196-ecrts2021/LIPIcs.ECRTS.2021.15/LIPIcs.ECRTS.2021.15.pdf>
- [18] National Institute of Standards and Technology (NIST). (s.f.). IT Security for Industrial Control Systems. [https://tsapps.nist.gov/publication/get\\_pdf.cfm?pub\\_id=821684](https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=821684)
- [19] Richards, M., & Ford, N. (2020). Software Architecture as a Set of Architectural Design Decisions. <https://mrce.in/ebooks/Software-Fundamentals%20of%20Software%20Architecture.pdf>
- [20] Alves, J., Costa Neto, A., Varanda Pereira, M. J., & Henriques, P. R. (s.f.). Characterization and Identification of Programming Languages. <https://drops.dagstuhl.de/storage/01oasics/oasics-vol113-slate2023/OASlcs.SLATE.2023.13/OASlcs.SLATE.2023.13.pdf>
- [21] Vanderbilt University. (s.f.). Frameworks: Why They Are Important and How to Apply Them Effectively. <https://www.cs.wm.edu/~dcschmidt/PDF/Queue-04.pdf>
- [22] 330ohms. (2024, 30 de julio). ¿Qué es un control PID? - 330ohms. <https://www.330ohms.com/blogs/blog/que-es-un-control-pid>
- [23] What is a PLC? How does it Work? (s.f.). <https://www.onlogic.com/blog/what-is-a-plc-how-does-it-work/>
- [24] What is Arduino? (s.f.). Arduino. <https://www.arduino.cc/en/Guide/Introduction/>
- [25] What is a Raspberry Pi? (s.f.). Opensource.com. <https://opensource.com/resources/raspberry-pi>

[26] *What is a PIC Microcontroller: The Harvard Architecture.* (s.f.). PCB Design and Analysis Center.

<https://resources.pcb.cadence.com/blog/2020-what-is-a-pic-microcontroller-the-harvard-architecture>