



UNIVERSIDAD NACIONAL DEL ALTIPLANO
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA,
ELECTRÓNICA Y SISTEMAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**DISEÑO DE UN SISTEMA DE CONTROL ON/OFF CON
INTERNET DE LAS COSAS PARA LA SUPERVISIÓN DE
TEMPERATURA DEL PROCESO DE COCCIÓN DE LOS
ALIMENTOS AL CARBÓN**

TESIS

PRESENTADA POR:

RICHARD DALY GARNICA CURASI

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PUNO – PERÚ

2024



Reporte de similitud

NOMBRE DEL TRABAJO

DISEÑO DE UN SISTEMA DE CONTROL ON/OFF CON INTERNET DE LAS COSAS PARA LA SUPERVISIÓN DE TEMPERATURA DEL PROCESO DE COCCIÓN DE LOS ALIMENTOS AL CARBÓN

AUTOR

RICHARD DALY GARNICA CURASI

RECuento de palabras

19379 Words

RECuento de caracteres

104621 Characters

RECuento de páginas

104 Pages

Tamaño del archivo

3.3MB

Fecha de entrega

May 9, 2024 3:48 PM GMT-5

Fecha del informe

May 9, 2024 3:50 PM GMT-5

● 14% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 13% Base de datos de Internet
- Base de datos de Crossref
- 8% Base de datos de trabajos entregados
- 1% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● Excluir del Reporte de Similitud

- Material bibliográfico
- Coincidencia baja (menos de 9 palabras)

Christian Augusto Romero Goyzueta
INGENIERO ELECTRÓNICO
CIP. 133009

Carlos Alexander Ccañutu Chirapa
SUB DIRECTOR DE INVESTIGACIÓN
E.P. INGENIERÍA ELECTRÓNICA



DEDICATORIA

Dedico la investigación hacia mis padres, Willy y Marilu, por estar a mi lado en cada paso que doy en mi búsqueda de crecer como persona y profesional. También dedico estas palabras a mi abuela, cuya influencia brilla en el firmamento como una fuente de motivación que me otorga la fuerza para seguir avanzando.

Agradezco sinceramente a mis hermanos por su apoyo constante, con la esperanza de que mi experiencia les sirva como un ejemplo de que cualquier meta es posible. Por último, quiero mencionar a quienes no confiaron en mí, ya que su actitud actuó como un motor que me empujó a seguir adelante con una determinación aún mayor.

Richard Daly



AGRADECIMIENTOS

Expreso mi agradecimiento a la divinidad por haber estado a mi lado y orientarme durante mi trayectoria, por ser mi fuente de fortaleza en los momentos de fragilidad y por otorgarme una vida repleta de enseñanzas, vivencias y, ante todo, alegría.

Richard Daly



ÍNDICE GENERAL

	Pág.
DEDICATORIA	
AGRADECIMIENTOS	
ÍNDICE GENERAL	
ÍNDICE DE FIGURAS	
ÍNDICE DE ANEXOS	
ACRÓNIMOS	
RESUMEN	12
ABSTRACT.....	13
CAPÍTULO I	
INTRODUCCIÓN	
1.1. PLANTEAMIENTO DEL PROBLEMA.....	15
1.2. HIPÓTESIS	16
1.3. OBJETIVO GENERAL	16
1.3.1. Objetivos específicos	16
CAPÍTULO II	
REVISIÓN DE LITERATURA	
2.1. ANTECEDENTES DEL PROYECTO	17
2.2. INTERNET DE LAS COSAS	21
2.3. AUTOMATIZACIÓN	23
2.4. COCIÓN DE ALIMENTOS	24
2.5. CARBÓN	26
2.6. MICROCONTROLADORES.....	28
2.7. ENTORNOS DE PROGRAMACIÓN	30



2.8.	ARDUINO.....	31
2.9.	SHIELD ETHERNET	34
2.10.	SENSORES.....	34
2.11.	ACTUADORES.....	35
2.12.	SENSOR DE TEMPERATURA.....	37
2.13.	SERVIDOR.....	38
2.14.	RED LAN.....	41
	2.14.1. Componentes de una red LAN.....	43
2.15.	RED WAN	48
2.16.	PANTALLA LCD	50
2.17.	PROTOCOLOS DE COMUNICACIÓN EN INTERNET	52
2.18.	LENGUAJE DE PROGRAMACION C	53

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1.	MATERIALES.....	55
	3.1.1. Hardware.....	55
	3.1.2. Software	56
3.2.	DISEÑO Y NIVEL DE LA INVESTIGACIÓN.....	56
	3.2.1. Diseño de investigación	56
	3.2.2. Nivel de la investigación.....	56
3.3.	MUESTRA DE LA INVESTIGACIÓN.....	56
3.4.	UBICACIÓN DE LA INVESTIGACIÓN	57
3.5.	TÉCNICAS E INSTRUMENTOS PARA RECOLECCION DE DATOS ..	57
	3.5.1. Técnicas	57
	3.5.2. Instrumentos.....	58



3.6. PROCEDIMIENTO.....	58
3.6.1. Adquisición de datos de temperatura.....	58
3.6.2. Control on/off del sistema de cocción	62
3.6.3. Exportación de datos de temperatura al servidor.....	66
3.6.4. Desarrollo del código en arduino.....	68
3.6.5. Implementacion fisica del prototipo	78
CAPÍTULO IV	
RESULTADOS Y DISCUSIÓN	
4.1. RESULTADOS DE COCCIÓN DE LOS ALIMENTOS.....	83
4.2. RESULTADOS DEL COMPORTAMIENTO DEL PROTOTIPO.....	85
4.3. RESULTADOS DE ENVÍO DE DATOS DE A LA PÁGINA WEB	90
4.4. COMPARACIÓN DE LA COCCIÓN CON EL PROTOTIPO.....	92
V. CONCLUSIONES.....	94
VI. RECOMENDACIONES	96
VII. REFERENCIAS BIBLIOGRÁFICAS.....	97
ANEXOS.....	101

Área: Telecomunicaciones

Tema: Internet de las Cosas

FECHA DE SUSTENTACIÓN: 21 de mayo del 2024



ÍNDICE DE FIGURAS

	Pág.
Figura 1 Sensor de altas temperaturas	38
Figura 2 Servidor y red de computadoras.....	40
Figura 3 Red LAN	42
Figura 4 Router.....	48
Figura 5 Diagrama de conexión de Arduino con LCD.....	51
Figura 6 Conexión física de Arduino con LCD.....	52
Figura 7 Ciudad de Puno	57
Figura 8 Circuito Interno del módulo MAX 6675.....	61
Figura 9 Conexión de termopares a la placa Arduino	62
Figura 10 Librerías usadas.....	68
Figura 11 Variables.....	70
Figura 12 Objetos	71
Figura 13 Configuración para conexión a internet	72
Figura 14 Setup del código.....	74
Figura 15 Lectura de temperatura de las termocuplas.....	76
Figura 16 Código para conexión a Internet	77
Figura 17 Diagrama del sistema de conexión a Internet.....	79
Figura 18 Conexión de módulos en el sistema de control.....	80
Figura 19 Termocupla ubicada lo más próximo al carbón	80
Figura 20 Soplador acoplado al tubo corrugado.....	81
Figura 21 Dimmer que regula la potencia del soplador.....	82
Figura 22 Tubo corrugado para conducción de aire	82
Figura 23 Resultados de cocción de alimentos.....	85



Figura 24 Carbón	85
Figura 25 Sistema completo	86
Figura 26 Parrilla lista para el funcionamiento.....	87
Figura 27 Soplador próximo al carbón	88
Figura 28 Inicia la cocción de los alimentos	88
Figura 29 Proceso de cocción de alimentos.....	89
Figura 30 Datos en la WEB	89
Figura 31 Verificación del encendido y apagado del soplador.....	90
Figura 32 Pagina web en el servidor.....	92
Figura 33 Cocción estándar	93
Figura 34 Cocción con el prototipo	93



ÍNDICE DE ANEXOS

	Pág.
Anexo 1 Código Fuente de Arduino UNO	101
Anexo 2 Declaración jurada de autenticidad de tesis	103
Anexo 3 Autorización para el depósito de tesis o trabajo de investigación en el repositorio institucional.....	104



ACRÓNIMOS

IoT:	Internet de las cosas
IDE:	Entorno de desarrollo integrado
LAN:	Local Area Network (Red de Área Local)
WAN:	Wide Área Network (Red de Área Amplia)
OSI:	Open Systems Interconnection (Interconexión de Sistemas Abiertos)
MAC:	Media Access Control (Control de Acceso al Medio)
NAT:	Network Address Translation (Traducción de Direcciones de Red)
TCP:	Transmission Control Protocol (Protocolo de Control de Transmisión)
UDP:	User Datagram Protocol (Protocolo de Datagramas de Usuario)
HTTP:	Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)
IP:	Internet Protocol (Protocolo de Internet)
DNS:	Domain Name System (Sistema de Nombres de Dominio)



RESUMEN

El proceso de cocción de alimentos al carbón presenta varios inconvenientes con el control de temperatura, que pueden afectar la calidad de los alimentos. Algunos problemas más comunes durante la cocción de los alimentos al carbón son: la dificultad para mantener una temperatura constante, riesgo de cocción excesiva de los alimentos y falta de precisión en el control de la temperatura. Esta investigación aborda el diseño de un sistema de control ON/OFF utilizando la tecnología de Internet de las Cosas (IoT) para la supervisión de la temperatura durante el proceso de cocción de alimentos al carbón, permitiendo a los usuarios monitorear y controlar la temperatura de manera remota, lo que resulta en una experiencia culinaria más práctica. Este proyecto se centra en la creación de un prototipo de control usando sensores de alta temperatura conocidas como termocuplas, estos sensores son colocados estratégicamente en las parrillas y se usa el microcontrolador ATmega328 para realizar las lecturas periódicamente en el proceso de cocción; para llevar a internet los datos censados se usa un módulo la conexión ethernet y una Raspberry pi, que previamente se configuró como enrutador, finalmente se puede ver los datos censados desde cualquier dispositivo con conexión a internet. Los resultados obtenidos indican que este prototipo es viable y efectivo, esto se verifica usando la técnica de comparación con la cocción al carbón estándar. El sistema permite a los usuarios ahorrar tiempo y esfuerzo al poder monitorear la temperatura desde sus celulares.

Palabras clave: cocción, control on-off, IoT, microcontrolador, temperatura.



ABSTRACT

The charcoal cooking process presents several challenges with temperature control, which can impact food quality. Some common issues during charcoal cooking include difficulty in maintaining a constant temperature, the risk of overcooking, and a lack of precision in temperature control. This research addresses the design of an ON/OFF control system using Internet of Things (IoT) technology to monitor temperature during the charcoal cooking process. This allows users to remotely monitor and control the temperature, resulting in a more practical culinary experience. This project focuses on creating a control prototype using high-temperature sensors known as thermocouples. These sensors are strategically placed on the grills, and the ATmega328 microcontroller is used to periodically read temperatures during the cooking process. To transmit the collected data to the internet, an ethernet connection module and a Raspberry Pi, previously configured as a router, are used. Finally, the collected data can be viewed from any internet-connected device. The obtained results indicate that this prototype is viable and effective, confirmed through a comparison with standard charcoal cooking. The system enables users to save time and effort by monitoring temperature from their smartphones.

Keywords: cooking, IoT, microcontroller, on-off control, temperature.



CAPITULO I

INTRODUCCIÓN

La denominación más familiar de IoT o Internet de las Cosas se refiere a la interconexión de objetos habituales a la infraestructura de Internet. Automóviles, enseres de cocina e inclusive dispositivos de seguimiento cardíaco poseen la aptitud de conectarse mediante esta tecnología. A medida que el ámbito del Internet de las Cosas prosiga su crecimiento en los próximos años, se añadirá una cantidad mayor de dispositivos a esta lista interconectada.

El ámbito del Internet de las Cosas ha progresado gracias a la convergencia de diversas tecnologías, que incluyen sensores básicos, sistemas integrados cada vez más avanzados y aprendizaje automático. Los campos tradicionales como los sistemas integrados, las redes de sensores inalámbricos, los sistemas de control y la automatización, incluida la automatización de hogares y edificios, de manera individual y conjunta, facilitan el desarrollo del Internet de las Cosas.

En este proyecto se explora el control de temperatura con internet de las cosas, con la técnica de control ON/OFF que es un tipo de control automático básico en el que la temperatura se mantiene dentro de un rango específico mediante el uso de un controlador que enciende y apaga un dispositivo de regulación, como un relé o un motor, según sea necesario. Por instancia, en el caso de que la temperatura exceda el valor máximo establecido, el regulador desactiva el mecanismo de control de temperatura hasta que la temperatura caiga por debajo del límite inferior. En ese punto, el regulador vuelve a activarlo.



Lo mencionado previamente tiene pertinencia en el ámbito de la cocina. Se admite la existencia de diversas técnicas para la preparación de alimentos, como el cocimiento o la cocción al vapor, entre otras opciones disponibles. Sin embargo, destaca una de las técnicas más notables, que es la elaboración de alimentos mediante el uso de una parrilla. Esta modalidad trae consigo varios beneficios, entre los cuales se incluye la reducción del contenido graso durante el proceso de cocción. De acuerdo con un reporte presentado por Elsevier, resulta esencial mantener una temperatura moderada en la parrilla, ya que esto evita una cocción excesiva.

Por lo tanto, el propósito de nuestro proyecto radica en desarrollar un sistema de control encendido/apagado utilizando el Internet de las Cosas, con el fin de supervisar la temperatura durante el procedimiento de cocción de alimentos a la parrilla de carbón.

1.1. PLANTEAMIENTO DEL PROBLEMA

La cocción de alimentos al carbón viene a ser un método tradicional para preparar alimentos, especialmente carnes y vegetales, esto gracias al calor que emite la combustión del carbón. Sin embargo, para que los alimentos tengan una buena cocción se requiere que la temperatura sea constante. Es por ello que surge la necesidad de control de temperatura para mejorar el proceso de cocción de alimentos. La finalidad principal de esta investigación es diseñar un sistema de control on/off que nos permita el control de temperatura en la parrilla.

Por ello se plantea la siguiente interrogante:

¿Cómo realizar un sistema de control on/off con internet de las cosas para la supervisión de temperatura del proceso de cocción de los alimentos?



1.2. HIPÓTESIS

El diseño de un sistema de control on/off con internet de las cosas permitirá el control y supervisión de temperatura del proceso de cocción de alimentos al carbón

1.3. OBJETIVO GENERAL

Diseñar un sistema de control on/off con internet de las cosas para la supervisión de temperatura del proceso de cocción de los alimentos al carbón

1.3.1. OBJETIVOS ESPECÍFICOS

- Diseñar el sistema de Internet de las cosas para la supervisión de temperatura.
- Diseñar el sistema de control on/off de temperatura del proceso de cocción de alimentos al carbón.



CAPITULO II

REVISIÓN DE LITERATURA

2.1. ANTECEDENTES DEL PROYECTO

Según la investigación llevada a cabo por Cahuana Aquino, se llega a la conclusión de que, en el año 2020, en el puesto de salud TINGUA, se identifica la carencia de un sistema domótico para la gestión de la temperatura. Las encuestas efectuadas en diversos ámbitos de este establecimiento de salud revelan que el 59% de los empleados señala la imperante necesidad de contar con un sistema de control térmico. Para abordar esta problemática, se concibió un sistema de control de temperatura, cuya interfaz gráfica para dispositivos móviles fue desarrollada a través de la plataforma AppInventor. Los resultados obtenidos fueron sumamente alentadores, lo cual se atribuye a la creatividad e innovación plasmadas en el diseño de la interfaz gráfica. Esta característica ha posibilitado una experiencia de usuario sumamente intuitiva y de fácil manejo, culminando así en una solución exitosa (Cahuana Aquino, 2022).

El modelo que emplea tecnología IoT y servicios en la nube permite la gestión remota del encendido y apagado, así como la supervisión instantánea de la red eléctrica. El diseño del sistema opera eficientemente en el hogar y permite visualizar los resultados o valores de la tensión o voltaje rms, corriente rms y potencia instantánea a través de una página web. La plataforma en línea ofrece una interfaz amigable y de fácil utilización para los usuarios, lo que además posibilita el control remoto de diversos dispositivos como electrodomésticos, bombas de agua y sistemas de calefacción, permitiendo su activación y desactivación de manera remota. Como se puede observar en los ensayos llevados a cabo, el prototipo funciona de manera estable. La transferencia de datos mediante el protocolo MQTT se ejecuta de



manera exitosa, y la visualización en la página web es inmediata gracias a la implementación de websockets. La alternativa basada en la Internet de las Cosas (IoT) y servicios en la nube emerge como una elección sumamente atractiva, y su combinación con diversos sensores ha posibilitado abordar de manera efectiva el objetivo establecido: la comunicación entre el sistema de medición y el servidor alojado en la plataforma AWS en la nube (Jimenes Guaman & Cabrera Mejia, 2020).

Según la investigación realizada por Luis Alonso Sánchez Carlos y Danny Adrián Cruz Irigoin, concluye con el logro exitoso en el desarrollo de un sistema de control de temperatura. Esta indagación condujo a la obtención de un modelo matemático de primer orden, el cual fue posteriormente validado utilizando Matlab. Además, se creó una aplicación de uso sencillo en LabVIEW que permite visualizar las variaciones de temperatura en el módulo diseñado. En última instancia, se destaca la capacidad de observar el comportamiento temporal de las variables y del proceso del sistema, todo ello con el fin de facilitar un análisis más preciso del lazo de control (Sanchez Carlos & Cruz Irigoin, 2018).

Gurum M. Debele y Xiao Qian han concebido y edificado un sistema automatizado destinado a regular la temperatura ambiente, basado en la combinación de Arduino y un sensor DHT11. En este diseño, se ha utilizado un microcontrolador Arduino, un teclado y un sensor DHT11 de manera conjunta para llevar a cabo la gestión y supervisión simultánea tanto del calentador como del ventilador. El ventilador de corriente continua se pone en funcionamiento cuando la temperatura ambiente supera el umbral establecido, y su velocidad se adapta en función de esta variante. En caso de que la temperatura ambiente caiga por debajo del límite mínimo establecido, la lámpara del calentador se enciende y se activa el ventilador de corriente continua. Si la temperatura ambiente se mantiene dentro del rango de



valores establecidos, todas las cargas se desconectan automáticamente. Entre los aspectos más destacados de este sistema se encuentran su accesibilidad económica, facilidad de instalación, simplicidad operativa, consumo energético reducido, dimensiones compactas y su manejo sin complicaciones. En última instancia, este proyecto optimiza de manera efectiva el consumo energético en una habitación, al mismo tiempo que asegura una temperatura agradable en el espacio (Debele & Xiao, 2020).

Erick Escobar Gallardo y Alex Villazón llevaron a cabo la concepción y la realización de lo que se ha denominado chuchusMOTE, un sistema de monitorización energética impulsado por el Internet de las cosas (IoT). Este sistema abarca una red compuesta por sensores y actuadores que interactúan mediante el protocolo MQTT. La estructura desarrollada incluye un servidor alojado en la nube, el cual posibilita también la visualización en tiempo real de los datos recopilados. Los resultados obtenidos evidencian la viabilidad del concepto de IoT, así como la habilidad para programación embebida y la destreza en el desarrollo electrónico empleando componentes sencillos y de fácil disponibilidad (Escobar Gallardo & Villazón, 2018).

Según Huisacaina Soto, Pesantes Rojas y Herrera Ortiz, concluyeron que, de los resultados gráficos obtenidos, los controles ON/OFF utilizados en este trabajo son satisfactorios utilizando el protocolo de criopreservación de embriones ovinos. La temperatura en el soporte que contiene la muestra biológica se traza en función del tiempo, disminuyendo gradualmente según la curva teórica esperada y permaneciendo casi constante durante el intervalo de tiempo especificado en el protocolo con una variación de aproximadamente ± 1 °C. El DSP utilizado es rápido y tiene una alta potencia de



procesamiento para realizar rápidamente cálculos de cambio de temperatura para el control (Huisacaina Soto et al., 2021).

Las investigadoras Hernández Velásquez Viviana Marcela y Alvarado Bawab María Bernarda desarrollaron un sistema de regulación para el dispositivo que emplea diversos componentes electrónicos en sincronía para controlar en tiempo real el encendido/apagado del magnetrón y la temperatura del material. Las pruebas se realizaron en el montaje físico. Inicialmente, se efectuaron ajustes en los cristales, en la programación del hardware y en otros elementos para resolver el problema de la corriente consumida que obstaculizaba el flujo de voltaje, ocasionando retrasos en el tiempo programado en relación con la unidad de tiempo real. Para medir la temperatura en la superficie del proceso, se usó un sensor infrarrojo MLX90614ESF, que puede medir temperaturas entre -70°C y 350°C y funciona con 5 V. Se colocó a 0,30 m del material en el horno de microondas. Se modificó la tarjeta electrónica y la programación del microprocesador. Para evitar la oscilación excesiva del magnetrón al encenderse/apagarse (afectando su vida útil), se diseñó un sistema de control automático en torno a un rango de temperatura para supervisar el funcionamiento de los dos modos de operación desarrollados en el proyecto. Se llevaron a cabo pruebas técnicas para validar este control y se planea evaluar su rendimiento operativo y energético en futuras investigaciones. Este control se prueba en procesos que van desde el secado de frutas hasta la extracción de aceites de semillas, y se contempla su aplicación en la producción de líquidos combustibles a partir de biomasa, la fundición de vidrio, la aceleración de reacciones químicas, entre otros usos. Hasta la fecha, los resultados indican que las propiedades de cada muestra señalan una temperatura y tasa de radiación óptimas durante el procesamiento, lo que ayuda a minimizar la generación de subproductos y efectos no deseados. Se evaluaron factores como la cantidad



y geometría del material suministrado, así como su disposición espacial en la cavidad. Se logró un rendimiento óptimo en el secado de papaya, con una ratio de 0,014 kg/kJ y una reducción del 86% del peso en una tasa de radiación de 6 x 12 para una rodaja de 1 cm. Estos resultados destacan la importancia del control de encendido/apagado y de temperatura en los procesos asistidos por microondas. También se concluye que la geometría y cantidad de muestra son factores cruciales en los análisis de estos procesos (Hernández Velásquez & Alvarado Bawab, 2017).

2.2. INTERNET DE LAS COSAS

El término Internet de las Cosas (IoT) se emplea para describir la conexión de objetos físicos, conocidos como "cosas", que están equipados con sensores, software y distintas tecnologías, con el fin de establecer comunicaciones y compartir datos con otros dispositivos y sistemas mediante la red mundial de Internet. Estos dispositivos pueden variar desde objetos cotidianos en los hogares hasta herramientas industriales de alta complejidad. En la actualidad, el número de dispositivos IoT conectados supera los 7 mil millones, y se prevé que esta cantidad aumente significativamente a 22 mil millones para el año 2025. En este contexto, Oracle despliega una red de colaboradores en el ámbito de dispositivos conectados.

En la última década, el Internet de las cosas (IoT) ha ganado un lugar de prominencia entre las tecnologías cruciales del siglo XXI. En la actualidad, la capacidad de conectar objetos comunes, desde electrodomésticos de cocina hasta vehículos y monitores para bebés, a la red global mediante dispositivos integrados ha propiciado una comunicación fluida entre personas, procesos y elementos.

A través de la combinación de recursos como la computación a bajo costo, el almacenamiento en la nube, el análisis de grandes volúmenes de datos y las tecnologías



móviles, los objetos físicos pueden compartir y recolectar datos con una interferencia mínima por parte de los seres humanos. En este universo de hiperconexión, los sistemas digitales logran registrar, supervisar y ajustar cada interacción entre los objetos conectados, permitiendo una cooperación efectiva entre el mundo tangible y el mundo digital.

Aunque la noción del IoT ha existido por un tiempo considerable, los recientes avances en varias tecnologías distintas han traducido esta idea en aplicaciones tangibles. Por lo tanto, se han manifestado una serie de ventajas inherentes al concepto del Internet de las cosas (Greengard, 2021).

- Disponibilidad de tecnología de sensores a bajo costo y eficiencia energética: Los sensores económicos y confiables están ampliando las posibilidades del Internet de las cosas (IoT) para un abanico más amplio de fabricantes.
- Interconexión: La diversidad de protocolos de red disponibles en Internet ha simplificado la conexión de sensores tanto con la nube como con otros dispositivos, agilizando la transferencia de datos de manera eficaz.
- Plataformas de cómputo en la nube: La creciente accesibilidad a plataformas basadas en la nube permite a empresas y consumidores acceder a la infraestructura necesaria para expandirse sin la necesidad de gestionar todo por sí mismos.
- Aprendizaje automático y análisis: Los avances en el aprendizaje automático y el análisis, sumados al acceso a grandes volúmenes de datos almacenados en la nube, posibilitan a las empresas recolectar información de manera ágil y sencilla. La convergencia de estas tecnologías sigue ampliando los horizontes del IoT, mientras



que los datos generados por el IoT también enriquecen y robustecen estas capacidades.

- Inteligencia artificial conversacional (IA): Los desarrollos en redes neuronales han incorporado el procesamiento del lenguaje natural (PLN) en los dispositivos del IoT, como los asistentes personales digitales Alexa, Cortana y Siri, convirtiéndolos en opciones atractivas, económicas y viables para ser utilizados en el hogar (King, 2021).

2.3. AUTOMATIZACIÓN

En términos generales, la automatización se puede describir como una tecnología que abarca la ejecución de un proceso mediante instrucciones programadas, junto con un control de retroalimentación automático, con el fin de garantizar la correcta implementación de las pautas establecidas. Como resultado, el sistema resultante tiene la capacidad de funcionar sin necesidad de intervención humana. La progresión de esta tecnología ha estado cada vez más interrelacionada con el uso de computadoras y tecnologías relacionadas con la informática. Por lo tanto, los sistemas automatizados han evolucionado hacia una mayor sofisticación y complejidad. Los sistemas avanzados demuestran un nivel de habilidad y rendimiento que supera en diversos aspectos las habilidades humanas en la ejecución de las mismas tareas.

El propósito fundamental de la automatización no consiste en sustituir a los individuos. Aunque podría haber cierta disminución en la demanda de acción humana al suprimir etapas que involucran interacción, su enfoque y ventajas se centran en elevar la productividad, la uniformidad y la eficacia (Sambamurthy, 2023).

Las ventajas de la automatización.



- Incremento de la eficiencia: Su equipo puede emplear más tiempo para generar un impacto más significativo en su empresa. Delegue las tareas repetitivas al software.
- Reforzamiento de la confiabilidad: Al reducir la intervención humana, se minimizan los errores y problemas. Todos los procedimientos se ejecutan de manera uniforme y constante. Esto permite anticipar cuándo se llevarán a cabo procesos, pruebas, actualizaciones, flujos de trabajo, entre otros.
- Simplificación de la gestión: A medida que aumenta el número de personal, también crece el riesgo de brechas en el conocimiento. Estos vacíos pueden resultar en un desconocimiento de lo que sucede en diferentes áreas de su negocio. La programación completa proporciona un mayor grado de control.

Desafíos inherentes a la automatización.

- Costo: Desarrollar una solución de automatización efectiva requiere tiempo y esfuerzo. Colaborar con un socio de confianza puede ser ventajoso.
- Limitaciones: La automatización no garantiza una inteligencia total. Dependiendo de qué se automatice y cómo se implemente, algunas áreas podrían quedar vulnerables fuera de su alcance. Limitar la automatización en ciertos aspectos o funciones puede atenuar esta preocupación. La inteligencia y seguridad de su automatización dependen de su implementación adecuada (Sands & Ian, 2018).

2.4. COCCIÓN DE ALIMENTOS

La cocción de alimentos es el proceso de calentar los alimentos con el fin de matar los microorganismos nocivos, mejorar el sabor, textura y apariencia de los alimentos. La cocción también ayuda a descomponer las proteínas, carbohidratos y grasas, lo que facilita



la digestión. Existen varios métodos de cocción, como cocinar a la parrilla, al horno, a la plancha, en la olla a presión, en el microondas entre otros. Los alimentos también se pueden cocinar de forma combinada, como cocinar a la parrilla y terminar en el horno. La automatización en la cocción de alimentos se refiere a la utilización de tecnologías para controlar y monitorear el proceso de cocción, como termómetros, sensores y sistemas de control automático para asegurar una cocción consistente y segura.

La cocción a la parrilla consiste en cocinar los alimentos sobre una rejilla caliente, ya sea al aire libre o en una parrilla eléctrica o de gas en el interior de la casa. Esta técnica se utiliza comúnmente para cocinar carnes, pescados, verduras y otros alimentos. El calor se aplica desde arriba y debajo del alimento, lo que le da un sabor ahumado y una textura crujiente en la parte exterior y cocido en el interior. La cocción a la parrilla también es una forma saludable de cocinar, ya que no requiere el uso de grasas adicionales.

La cocción a la parrilla con Internet de las cosas (IoT, por sus siglas en inglés) combina la técnica tradicional de cocción a la parrilla con la tecnología IoT para proporcionar una experiencia de cocción más precisa y conveniente. Con esta tecnología, se pueden conectar parrillas inteligentes a Internet a través de sensores y dispositivos conectados, como teléfonos inteligentes o tabletas.

Esto permite a los usuarios controlar la parrilla desde cualquier lugar, monitorear el progreso de la cocción, y recibir notificaciones cuando el alimento esté listo para servir. También se pueden programar parrillas inteligentes para seguir recetas específicas y ajustar la temperatura automáticamente. En general, la cocción a la parrilla con IoT permite a los usuarios cocinar con mayor precisión y comodidad, y ofrece una experiencia de cocción más personalizada (Casares, 2022).



2.5. CARBÓN

El carbón constituye un tipo de combustible fósil que se origina a partir de la acumulación de materia vegetal que experimenta altas temperaturas y presiones durante millones de años. Se clasifica en tres variedades: antracita, bituminosa y lignito. Principalmente, se emplea para generar energía eléctrica y calor en centrales térmicas, aunque también tiene aplicaciones en la industria, como en la manufactura de acero y cemento. Sin embargo, su utilización conlleva efectos adversos para el medio ambiente, como la emisión de gases de efecto invernadero y la contaminación del aire y el agua.

La composición química del carbón varía dependiendo del tipo de carbón y de dónde se haya extraído. En general, el carbón está compuesto principalmente de carbono e hidrógeno, pero también contiene pequeñas cantidades de azufre, nitrógeno, oxígeno y trazas de metales como hierro, calcio y fósforo.

- El carbón bituminoso, también conocido como carbón negro, es el tipo más común de carbón y suele tener un contenido de carbono del 60-80%.
- El carbón antracita es un tipo de carbón con un alto contenido de carbono, generalmente del 85-98%.
- El lignito es el tipo menos valorado de carbón, suele tener un contenido de carbono del 60-70%.

Además del carbono e hidrógeno, el carbón también contiene pequeñas cantidades de azufre, nitrógeno, oxígeno y trazas de metales como hierro, calcio y fósforo. El contenido de estos elementos varía dependiendo del tipo de carbón y del lugar de donde se extrajo (Malm, 2020).



El uso del carbón en la cocina tiene algunos beneficios, aunque también tiene algunos riesgos y desventajas asociados. Algunos de los beneficios de cocinar con carbón son los siguientes:

- Sabor: El carbón proporciona un sabor ahumado y auténtico a los alimentos, lo que puede mejorar su sabor y aroma.
- Calor: El carbón es una fuente de calor constante y duradera, lo que permite cocinar los alimentos lentamente y con precisión.
- Tradición: Cocinar con carbón es una tradición en muchas culturas, y puede ser una forma de conectar con la historia y la comunidad.
- Ambiente: Cocinar con carbón es una forma de cocinar en exteriores que puede ser muy agradable en un día cálido y soleado.

Es importante tener en cuenta que el uso del carbón tiene impactos ambientales negativos y también puede generar contaminación del aire y riesgos de incendios. Es recomendable tomar las precauciones necesarias para cocinar de forma segura y cuidar el medio ambiente.

Cocinar en parrilla con carbón es una forma popular de cocinar carne, vegetales y otros alimentos. El proceso implica calentar el carbón en una parrilla hasta que esté listo para cocinar. Es importante tener en cuenta que el carbón debe estar completamente encendido y tener una capa de ceniza antes de comenzar a cocinar. También es recomendable colocar la parrilla a una distancia adecuada del carbón para controlar la temperatura y evitar quemar los alimentos. Para cocinar en parrilla con carbón se necesita tiempo y paciencia, pero puede proporcionar un sabor ahumado y auténtico a los alimentos (Auza y otros, 2020).



2.6. MICROCONTROLADORES

Los microcontroladores se presentan como diminutos aparatos electrónicos programables que incorporan en una sola unidad de chip un microprocesador, memoria y periféricos integrados. Estos componentes encuentran extenso uso en diversos campos, incluyendo la industria automotriz, electrodomésticos, tecnología médica, sistemas de seguridad, control industrial y dispositivos electrónicos de uso cotidiano.

Un microcontrolador incorpora un núcleo de microprocesador que ejecuta las instrucciones de programa almacenadas en su memoria. Además de las instrucciones, esta memoria puede contener datos y configuraciones. Dentro del microcontrolador se encuentran periféricos integrados, como puertos de entrada/salida, convertidores analógicos/digital y capacidades de comunicación inalámbrica, que posibilitan que el microcontrolador interactúe con su entorno externo.

Los microchips son programados utilizando lenguajes de programación específicos, como C o Assembler. Una vez programados, los microcontroladores pueden controlar automáticamente procesos, recolectar y procesar datos, y tomar decisiones en base a esos datos (Ward, 2020).

Existen varios tipos de microcontroladores disponibles en el mercado, cada uno con sus propias peculiaridades y aplicaciones específicas. Algunos de los tipos más comunes de microcontroladores incluyen:

- **Microcontroladores de 8 bits:** Estos microcontroladores tienen un ancho de datos de 8 bits y suelen ser los más económicos y fáciles de usar. Resultan perfectos para usos que requieren poca complejidad y un consumo energético reducido, como en dispositivos de uso cotidiano, juguetes y electrodomésticos.



- **Microcontroladores de 16 bits:** Estos microcontroladores tienen un ancho de datos de 16 bits y suelen ser más potentes que los microcontroladores de 8 bits. Son ideales para aplicaciones de mayor complejidad, como sistemas de control en la industria, dispositivos de uso médico y automoción.
- **Microcontroladores de 32 bits:** Estos microcontroladores tienen un ancho de datos de 32 bits y son los más potentes de todos. Son ideales para aplicaciones de alta complejidad, como sistemas de automatización avanzados, robots y sistemas embebidos.
- **Microcontroladores basados en ARM:** Estos microcontroladores utilizan el conjunto de instrucciones ARM, diseñado por ARM Holdings. Son muy populares en aplicaciones móviles, sistemas embebidos y dispositivos IoT debido al bajo consumo de energía y alta potencia.
- **Microcontroladores basados en AVR:** Estos microcontroladores utilizan el conjunto de instrucciones AVR, desarrollado por Atmel. Son populares en aplicaciones de automatización y control, sistemas embebidos y dispositivos de consumo.

Es importante seleccionar el tipo adecuado de microcontrolador para una aplicación específica teniendo en cuenta el costo, el rendimiento, la memoria y los periféricos necesarios.

En resumen, los microcontroladores representan una tecnología fundamental para la supervisión y la automatización de diversas aplicaciones y dispositivos electrónicos (Purdum & Peter, 2020).



2.7. ENTORNOS DE PROGRAMACIÓN

Un ambiente de programación (IDE, por sus iniciales en inglés) es una utilidad de software que provee un conjunto de atributos y capacidades con el propósito de simplificar la creación de programas. Generalmente, un IDE incorpora un editor de código, un depurador, un compilador y otros recursos para asistir a los programadores en la redacción, evaluación y corrección de sus códigos. Algunas de las funciones comunes que se pueden encontrar en un IDE incluyen:

- **Manipulación del código:** Un editor de código es una utilidad empleada para redactar y modificar el código fuente de un programa. Un IDE incorpora un editor de código que presenta atributos avanzados como el resaltado de la sintaxis, la finalización automática del código y la navegación en el código.
- **Detección de fallos:** Un depurador es una herramienta utilizada para hallar y corregir fallos en el código. Un IDE posee un depurador con atributos avanzados que permiten establecer puntos de parada, realizar un seguimiento de variables y evaluar expresiones.
- **Generación de código ejecutable:** Un compilador es una utilidad utilizada para transformar el código fuente en un código que puede ser ejecutado. Un IDE dispone de un compilador integrado y brinda una interfaz para configurar y llevar a cabo la compilación.
- **Integración con sistemas de control de versiones:** Algunos IDEs se integran con plataformas de control de versiones como git, svn, entre otros, lo que otorga al desarrollador la capacidad de gestionar y supervisar las versiones de su proyecto.



- Interfaz de usuario de fácil comprensión: Una interfaz de usuario de fácil comprensión permite a los programadores navegar sin dificultades a través de los diferentes componentes de su proyecto, como archivos de código, bibliotecas y recursos.

Hay diversos ambientes de programación accesibles para variados lenguajes y plataformas, como Visual Studio, Eclipse, Xcode, PyCharm, entre otros. Los programadores pueden seleccionar el entorno que más concuerde con sus requisitos y gustos (Code Quickly, 2020).

Algunos ejemplos de entornos de programación para microcontroladores son Atmel Studio, Keil uVision, IAR Embedded Workbench, MPLAB X, entre otros. Dependiendo del microcontrolador y plataforma utilizadas hay diferentes opciones de entornos de programación disponibles (Purdum & Peter, 2020).

2.8. ARDUINO

Arduino es una plataforma de desarrollo de código abierto tanto para hardware como para software que capacita la creación de proyectos interactivos haciendo uso de sensores, actuadores y sistemas informáticos. Se apoya en una placa de desarrollo cimentada en un microcontrolador y un entorno de programación unificado (IDE) que permite redactar y cargar el código en dicha placa. Arduino es extensamente empleado en una gama diversa de campos, como la robótica, la automatización, la domótica y el arte interactivo.

Arduino es una plataforma bien conocida para la creación de proyectos de Internet de las Cosas (IoT), gracias a su simplicidad y su asequibilidad. Utilizando Arduino, es factible enlazar sensores y actuadores a una red de dispositivos y gestionarlos mediante conexión a



Internet. Esto habilita la supervisión y el control remoto de los dispositivos, junto con la captura de información y la automatización de operaciones (Stuart, 2020).

Hay muchas placas de desarrollo compatibles con Arduino que vienen con conectividad WiFi y/o Ethernet, permitiendo a los dispositivos conectarse fácilmente a Internet. Además, existen librerías y herramientas de desarrollo específicas para IoT que permiten a los desarrolladores trabajar con protocolos de comunicación como MQTT y HTTP.

Arduino se ha convertido en una plataforma ampliamente utilizada para la creación de proyectos de IoT, gracias a su sencillez, coste asequible y extensa comunidad de programadores. Utilizando Arduino, es factible establecer conexiones de sensores y actuadores con la red mundial, lo que posibilita la supervisión, control a distancia y automatización de diversas tareas.

Para programar en Arduino, se emplea el lenguaje de programación C/C++ en conjunto con el entorno de desarrollo integrado (IDE) proporcionado por Arduino. Este IDE es una aplicación de código abierto disponible para su descarga gratuita desde el sitio web oficial de Arduino. Una vez instalado el IDE, se puede escribir, cargar y depurar código en la placa de desarrollo utilizando el IDE. El código se escribe en el editor de texto del IDE y se carga en la placa de desarrollo mediante un cable USB.

El lenguaje de programación implementado en Arduino se asemeja al lenguaje C/C++, con algunas pequeñas variaciones y una biblioteca de funciones específicamente diseñada para interactuar con los pines de entrada/salida y otras capacidades de la placa de desarrollo. El esquema fundamental de un "sketch" (programa) en Arduino está compuesto



por las funciones `setup()` y `loop()`. La función `setup()` se ejecuta solamente una vez al inicio del "sketch", en contraste, la función `loop()` se ejecuta continuamente después de la finalización de la función `setup()`.

En resumen, la programación en Arduino se lleva a cabo mediante el uso del lenguaje de programación C/C++ y el entorno de desarrollo IDE de Arduino. La estructura elemental de un "sketch" radica en las funciones `setup()` y `loop()`. Arduino utiliza principalmente dos tipos de señales: señales digitales y señales analógicas.

- Señales digitales: son señales binarias que solo tienen dos estados posibles, alto (HIGH) o bajo (LOW). Estas señales se utilizan para controlar dispositivos que solo requieren una señal binaria, como LED, relés, buzzers, entre otros.
- Señales analógicas: son señales de voltaje variables que pueden tener un rango de valores entre 0 y 5 voltios. Estas señales se utilizan para controlar dispositivos que requieren una señal de voltaje variable, como motores, servomotores, pantallas de visualización, entre otros.

Además de estas señales, Arduino también puede manejar señales de comunicación como I2C, SPI, entre otras.

De manera general, las interfaces digitales de entrada y salida en Arduino se emplean para señales digitales, en contraste, las interfaces de entrada y salida analógicas se destinan a señales de índole analógica. Aunque es esencial señalar que algunas placas Arduino cuentan con pines que pueden ser adaptados para funcionar tanto como entradas como salidas, ya sean digitales o analógicas (Blum, 2019).

2.9. SHIELD ETHERNET

Una shield Ethernet de Arduino es un módulo de expansión que se conecta con Arduino y permite comunicarse con redes Ethernet. Esta placa adicional brinda a la placa Arduino una interfaz Ethernet, posibilitando la conexión de la placa a una red y la transferencia bidireccional de datos por medio de dicha conexión.

La placa Ethernet se enlaza con la placa Arduino mediante los pines de extensión y establece comunicación a través del protocolo SPI (Interfaz Periférica en Serie). Una vez integrada, la placa provee una dirección IP fija a la placa Arduino, posibilitando la conexión a una red local o a Internet. Este componente también incluye un chip de memoria flash (EEPROM) para retener la configuración de red, así como un conector RJ-45 para conectar un cable Ethernet.

Con la shield Ethernet, los proyectos Arduino pueden acceder a servidores web, enviar y recibir correos electrónicos, acceder a bases de datos en línea, entre otras posibilidades. Existen diferentes modelos de shield Ethernet de Arduino en el mercado, algunas de las más populares son la shield Ethernet W5100 y la shield Ethernet ENC28J60 (Ganazhapa, 2021).

2.10. SENSORES

Los sensores constituyen dispositivos que evalúan o perciben variaciones en una variable física específica (como temperatura, luminosidad, movimiento, entre otros), transformándola en una señal eléctrica aprovechable por sistemas electrónicos para el monitoreo o ajuste de dicha variable. Estos componentes son empleados en diversas aplicaciones, que abarcan desde la industria y el sector automotriz hasta campos como la medicina y la robótica.



La historia de los sensores se remonta a la antigüedad, cuando los humanos utilizaron herramientas básicas como termómetros de mercurio y termómetros de alcohol para medir la temperatura. Sin embargo, el desarrollo de los sensores modernos comenzó en el siglo XIX con el invento del primer sensor de temperatura eléctrico, el termopar, por el físico Thomas Seebeck en 1821.

A medida que la tecnología avanzaba, surgieron nuevos tipos de sensores, como los sensores de presión, de humedad, de luz y de movimiento. En el siglo XX, la introducción de la electrónica y la informática permitió el desarrollo de sensores más precisos y sofisticados, así como la capacidad de procesar y transmitir la información recolectada por los sensores.

Durante la década de 1960, los sensores encontraron aplicaciones diversas en el ámbito industrial, tales como la automatización de plantas de producción y el monitoreo de procedimientos en la industria química. Hacia las décadas de 1980 y 1990, los sensores comenzaron a integrarse en contextos móviles, como teléfonos celulares y vehículos. Finalmente, en los años 2000, los sensores se volvieron un componente fundamental en los campos de la robótica y el Internet de las Cosas (IoT).

En el presente, los sensores desempeñan un papel significativo en una diversidad de contextos, abarcando desde vehículos autónomos hasta dispositivos de salud portátiles, y se prevé que su utilización siga expandiéndose en los próximos años (Tsiatsis et al., 2018).

2.11. ACTUADORES

Los actuadores son dispositivos mecánicos o electrónicos encargados de convertir una señal de control en una acción física concreta. Estos elementos tienen aplicaciones en



una amplia variedad de campos, como la automatización industrial, la robótica, el monitoreo y control de procesos, la aviónica y la industria automotriz.

Hay varios tipos de actuadores, cada uno con peculiaridades específicas y ventajas.

Algunos ejemplos de actuadores son:

- Actuadores de tipo eléctrico: Emplean electricidad como fuente de energía para generar movimiento. Ejemplos de estos son los motores eléctricos y los solenoides.
- Actuadores neumáticos: Utilizan aire comprimido para producir movimiento. Ejemplos de estos son los cilindros neumáticos y las válvulas neumáticas.
- Actuadores hidráulicos: Usan un fluido, típicamente aceite, para inducir movimiento. Ejemplos de estos son los cilindros hidráulicos y las bombas hidráulicas.
- Actuadores mecánicos: Utilizan mecanismos de transmisión para generar movimiento. Los engranajes, poleas y manivelas son ejemplos de actuadores mecánicos.

Los actuadores son una parte esencial de los sistemas automatizados, ya que permiten que los dispositivos controlados realicen una tarea específica. Su elección depende de las características de las tareas que se espera realice, como la velocidad, la fuerza, la precisión y el rango de movimiento.

Los actuadores comunes incluyen motores, servos, solenoides, y relés. Arduino es una plataforma de desarrollo de hardware y software libre que permite controlar estos dispositivos mediante el uso de su placa y su lenguaje de programación. Los actuadores



pueden ser controlados con Arduino utilizando los pines de salida y los comandos de control de movimiento específicos (Tsiatsis et al., 2018).

2.12. SENSOR DE TEMPERATURA

Existen varios tipos de sensores de temperatura que se pueden utilizar con una placa de Arduino, algunos de ellos son:

- Sensor termistor: Este género de sensor de temperatura emplea un material semiconductor que modifica su resistencia eléctrica al experimentar variaciones térmicas.
- Sensor termopar: Este tipo de sensor de temperatura se basa en la unión de dos materiales conductores distintos para medir la diferencia de potencial termoeléctrico y evaluar la temperatura.
- Sensor de RTD (Resistencia térmica en detenimiento): Este tipo de sensor de temperatura utiliza una resistencia metálica que cambia su resistencia eléctrica al variar la temperatura.

Sensor LM35: Este dispositivo de detección de temperatura es un chip integrado que emplea una mínima cantidad de energía para generar una salida de voltaje en relación a la temperatura.

Estos sensores se conectan a los pines analógicos o digitales de Arduino y se utiliza la librería de Arduino para leer y procesar la señal de salida del sensor (Khan & Yuce, 2019).

Figura 1

Sensor de altas temperaturas



Nota: (Khan & Yuce, 2019)

2.13. SERVIDOR

Un servidor es un equipo o un programa que suministra servicios a otros dispositivos o programas en una red. Estos servidores se emplean para alojar, procesar y transferir información, aplicaciones y prestaciones a través de una red. Pueden ser físicos o virtuales, y su uso se extiende a diversas aplicaciones, como almacenamiento de archivos, hospedaje de sitios web, correo electrónico, bases de datos, ejecución de transacciones y administración de redes.

Algunos ejemplos de servidores son:

- Servidores web: engloban aquellos que suministran servicios en línea, incluyendo el hospedaje de páginas web y el respaldo a aplicaciones de la web.
- Servidores de correo: son aquellos que ofrecen prestaciones relacionadas con el correo electrónico, como el almacenamiento de mensajes y la transferencia de correos.



- Servidores de bases de datos: engloban aquellos que brindan servicios de almacenamiento y recuperación de datos, gestionando el almacenamiento de información en una base de datos y la obtención de dicha información a través de consultas.
- Servidores de aplicaciones: son aquellos que ofrecen servicios de aplicaciones, abarcando el hospedaje y el soporte para aplicaciones específicas.

En palabras simples, un servidor es un dispositivo o software que se encarga de proveer servicios a través de una red (LaCroix, 2022).

Figura 2

Servidor y red de computadoras



Nota: (LaCroix, 2022)

La placa adicional Shield Ethernet de Arduino le agrega la capacidad de conectarse a una red Ethernet. Una vez conectado a una red Ethernet, Arduino puede actuar como un servidor y proporcionar servicios a través de la red.

Un ejemplo de cómo se puede utilizar la Shield Ethernet con Arduino como servidor es con el uso de un protocolo de red como TCP/IP para escuchar las solicitudes entrantes y proporcionar respuestas a través de una conexión de red. Por ejemplo, se puede crear un servidor web en Arduino que responda a solicitudes HTTP y proporcione páginas web almacenadas en la memoria de Arduino.

Otro ejemplo es el uso de protocolos de comunicación como MQTT, donde el servidor de Arduino puede publicar y suscribirse a mensajes mediante una conexión Ethernet. Esto permite conectar y controlar dispositivos IoT.



En resumen, la Shield Ethernet permite conectar Arduino a una red Ethernet y utilizarlo como servidor para proporcionar servicios a través de la red mediante el uso de protocolos de red específicos (Ganazhapa, 2021).

2.14. RED LAN

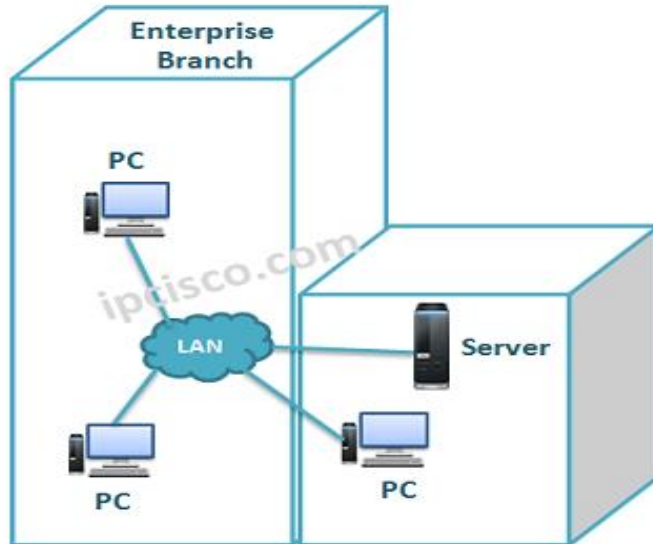
Una red se define por su estructura característica, la cual simplifica la interconexión de sus diversos componentes. A partir de esta definición, se pueden distinguir varios tipos de redes.

En el ámbito de la informática, una red informática implica la agrupación de computadoras (u ordenadores), periféricos y otros dispositivos interconectados con el propósito de permitir el intercambio de sus recursos. La clasificación de las redes informáticas puede variar en función de la forma de interconexión, la relación entre sus componentes y otras variables. Entre estas categorizaciones se encuentra la ampliamente reconocida red LAN, que es un acrónimo de "Local Area Network" (Red de Área Local). Estas redes conectan computadoras situadas en un espacio físico limitado, como una oficina o un edificio. La interconexión se establece a través de cables u ondas.

Los dispositivos informáticos que se enlazan a una red de área local (LAN) son conocidos como nodos, y cada uno de ellos representa una computadora individual. Mediante esta red, los usuarios de estas computadoras pueden intercambiar documentos y también acceder a periféricos compartidos, como una impresora. (Raghunandan, 2022).

Figura 3

Red LAN



Nota: (Raghunandan, 2022)

La instauración de una red LAN en una compañía o incluso en una residencia conlleva una multitud de beneficios. Por ejemplo, al compartir una impresora, se obvia la necesidad de que cada usuario cuente con su propia unidad, dando pie a un marcado ahorro de recursos. Además, la facilidad de entrada a documentos guardados en cualquier punto de la red LAN resulta especialmente provechosa para la colaboración en tareas.

A menos que ciertos contextos específicos lo requieran, la velocidad de traslado de datos en una red LAN tiende a sobrepasar notablemente la obtenida vía Internet. Los dispositivos que vinculan computadoras u otros dispositivos en un edificio suelen superar significativamente la velocidad de la red global. A manera de ejemplo, es común implementar un router capaz de alcanzar velocidades de descarga de hasta 300 Mb/s (Megabits por segundo), incluso para servicios que no exceden los 12 Mb/s.



En una situación de este tipo, si bien cada nodo podría descargar datos a un máximo de 12 Mb/s desde Internet, si un usuario necesitara acceder a archivos en la red local, teóricamente podría lograrlo a una velocidad tope de 300 Mb/s, equivalente a una velocidad 25 veces superior. Además, durante el uso de la conexión local, no se agota el tráfico del servicio de Internet, lo cual es de gran valor en instancias de planes de datos limitados o cuando se realiza la transferencia de volúmenes considerables de información (Dodd, 2019).

2.14.1. Componentes de una red LAN

Una red de información representa un método para enlazar diversos dispositivos entre sí mediante medios de transmisión y dispositivos de coordinación, siguiendo el protocolo establecido para el conjunto de la configuración.

- **Switch:** Un switch, también conocido como conmutador, es un dispositivo digital esencial en la interconexión de redes informáticas. Opera en la capa 2 del modelo OSI, que es el nivel de enlace de datos. Su función principal radica en la creación de vínculos entre múltiples segmentos de red, de manera semejante a los puentes (bridges), posibilitando la transferencia de datos de un segmento a otro basándose en las direcciones MAC de destino presentes en la red. Los switches cuentan con la habilidad de aprender y retener las direcciones MAC de nivel 2 de los dispositivos conectados a cada uno de sus puertos. Por ejemplo, cuando un dispositivo se conecta a un puerto del switch, este almacena su dirección MAC. Esto tiene el efecto de permitir que los datos dirigidos a un dispositivo sean transmitidos directamente desde el puerto de origen al puerto de destino, una mejora en comparación con los concentradores o hubs. En el caso de conectar dos switches o un switch y un concentrador, cada switch adquiere conocimiento



de las direcciones MAC de los dispositivos que son accesibles mediante sus puertos. Como resultado, las direcciones MAC de los dispositivos del otro switch quedan registradas en el puerto de interconexión (Carthern et al., 2021).

- **Router:** El enrutador juega un papel esencial en nuestra red al ser el dispositivo responsable de proveernos acceso a Internet y llevar a cabo una variedad de funciones críticas de seguridad en el entorno en línea.

Un enrutador, también conocido como router, es un componente de hardware diseñado para facilitar la interconexión de redes de computadoras, operando en el tercer nivel del modelo OSI. Su tarea primordial es asegurar el direccionamiento adecuado de paquetes entre distintas redes y determinar la ruta que un paquete de datos debe seguir. En situaciones en las que se establece una conexión a Internet, generalmente se necesita un módem para conectar un solo dispositivo, o un enrutador en el caso de conectar múltiples dispositivos dentro de la red interna. El simple hecho de emplear un enrutador, debido a las funciones inherentes que desempeña, ya implica un nivel de seguridad más elevado en comparación con las capacidades de un módem.

El enrutador comúnmente es el dispositivo encargado de establecer la conexión a Internet y nos brinda la posibilidad de ajustar los parámetros necesarios, así como supervisar la calidad de la conexión.

Existen otros tipos de conexiones que requieren un módem de manera obligatoria. El módem lleva a cabo la transformación del medio, convirtiendo la tecnología entrante (cable, fibra, WiFi, WiMAX, etc.) en una forma adecuada



para la red informática. En estos casos, se utilizan enrutadores neutrales, los cuales realizan la adaptación del medio, ya que tanto la entrada como la salida se efectúan mediante el mismo medio.

La Traducción de Dirección de Red (NAT) se refiere a una función que implica la conversión de direcciones IP y es en gran medida una tarea del enrutador. Esta función permite que, teniendo una única conexión a Internet con una dirección IP pública única, podamos conectar múltiples computadoras en la red interna, asignando a cada una dirección IP interna distinta, lo que optimiza el uso de la conexión.

En términos metafóricos, podríamos decir que en Internet se utiliza un lenguaje y en nuestra red interna se emplea otro diferente, y el enrutador actúa como un traductor simultáneo que facilita la comunicación entre las computadoras de ambos lados, las cuales utilizan lenguajes diversos.

Más adelante, nos encontramos con el PAT (Traducción de Dirección y Puerto), una característica derivada del NAT que lleva a cabo la traducción de conexiones TCP y UDP realizadas por un host y un puerto desde una red externa a otra dirección y puerto en la red interna. Esto permite que una única dirección IP externa pueda responder a hasta 64000 direcciones internas.

Como hemos observado, el uso de un enrutador genera la necesidad de traducir direcciones, y la combinación de las funciones NAT y PAT proporciona automáticamente un cierto nivel de seguridad, protegiéndonos de ciertos tipos de ataques provenientes de Internet. En la actualidad, la mayoría de los enrutadores



incorporan funciones básicas de cortafuegos para protegerse de ataques típicos desde Internet, además de ofrecer algunas medidas de control interno de seguridad (Carthern et al., 2021).

- **Punto de acceso:** En el contexto de las redes informáticas, un punto de acceso inalámbrico (WAP) o, en términos más amplios, un punto de acceso (AP), se refiere a un dispositivo de hardware de red que habilita la conexión de otros dispositivos Wi-Fi a una red que posee cableado. Aunque un AP puede funcionar de manera independiente al enlazarse mediante un cable a un enrutador, en el contexto de un enrutador inalámbrico, también puede formar parte integral del propio enrutador. La diferencia entre un AP y un punto de acceso radica en que este último corresponde a un sitio físico en el cual se brinda acceso a la red WiFi.

Usualmente, los AP desempeñan una función principal al simplificar la conectividad con la red, dejando que labores como el enrutamiento y la asignación de direcciones sean manejadas por servidores, enrutadores y conmutadores. En su mayoría, los AP siguen las pautas de comunicación 802.11 establecidas por IEEE, lo que asegura su compatibilidad con una amplia variedad de dispositivos inalámbricos. Algunos equipos llevan a cabo tareas adicionales, como la configuración de enrutamiento, la asignación de puertos, la seguridad y la gestión de usuarios, todas ellas basadas en una configuración preestablecida. Reforzando la interoperabilidad entre servidores y AP, se pueden lograr mejoras en los servicios proporcionados, como la adaptación dinámica a cambios en la red y el ajuste de la configuración de los dispositivos.



Los AP actúan como un enlace entre las redes con cableado y las inalámbricas. La implementación de varios AP posibilita la funcionalidad de itinerancia. Estos dispositivos han contribuido a reducir la necesidad de realizar nuevas instalaciones de cableado. Un AP con el estándar IEEE 802.11b tiene un alcance aproximado de 100 metros. Son los responsables de establecer la red y siempre están disponibles para atender a nuevos clientes. El punto de acceso recibe, almacena y transmite información entre la red de área local inalámbrica (WLAN) y la red de área local (LAN) con cableado.

Un AP individual puede brindar soporte a un pequeño grupo de usuarios y operar en un rango que va desde unos treinta metros hasta varios cientos. Por lo general, el AP o su antena se colocan en lugares elevados, aunque también pueden ubicarse en cualquier posición que asegure la cobertura de radio deseada.

Los usuarios finales acceden a la red WLAN a través de adaptadores en sus dispositivos (como computadoras, tabletas, teléfonos inteligentes, televisores inteligentes, radios por internet, etc.). Estos adaptadores establecen una interfaz entre el sistema operativo de red del cliente (NOS: network operating system) y las señales de radio, mediante el uso de una antena inalámbrica (Raghuhandan, 2022).

Figura 4

Router



Nota: (Carthern et al., 2021)

2.15. RED WAN

La tecnología WAN (Wide Área Network) es la tecnología que conecta las áreas geográficas lejanas. Se puede utilizar para que un cliente se conecte a la red corporativa, conexiones entre las sucursales de una empresa y conexiones de área amplia similares, etc. Existen otros tipos de redes más pequeñas como MAN, LAN y PAN.

Los protocolos relacionados con la tecnología de redes WAN operan en las capas inferiores del modelo OSI, es decir, en la capa física y en la capa de enlace de datos. La capa física se encarga de establecer las conexiones, mientras que la capa de enlace de datos facilita la transmisión de información de forma encapsulada. En el ámbito de las redes WAN, se utilizan diversos protocolos en la capa de enlace de datos, como HDLC, PPP, Frame-Relay, ATM, y otros similares (Odom, 2019).

Hay varios protocolos WAN que se utilizan entre diferentes ubicaciones de diferentes redes. Estos protocolos son:



- HDLC (Control de Enlace de Datos de Alto Nivel) constituye un protocolo de nivel 2 para la encapsulación en redes WAN, específicamente en enlaces de datos sincrónicos. Es la solución de conectividad WAN más sencilla, capaz de enlazar oficinas remotas mediante líneas alquiladas. Presenta tanto un estándar de la industria como una versión propietaria desarrollada por Cisco.
- El protocolo PPP (Protocolo Punto a Punto) también desempeña un rol como método de encapsulación WAN, derivado de HDLC, aunque podría considerarse como una mejora evolutiva de este último. PPP agrega varias funciones extra, como autenticación, soporte para múltiples enlaces, detección de errores y control de calidad.
- Por su parte, Frame Relay se configura como otro protocolo de nivel 2. Se fundamenta en X.25 y ofrece conexiones basadas en circuitos virtuales. Aunque tuvo popularidad en el pasado, su utilización en la actualidad es escasa.
- ATM (Modo de Transferencia Asíncrono) constituye un protocolo de nivel 2 en redes WAN basado en celdas. Se emplea en entornos de alta velocidad de transmisión, como T3, E3 y SONET.

La instalación de una red WAN proporciona a las compañías una serie de ventajas esenciales para aumentar la productividad y la eficiencia en el marco de su infraestructura de red empresarial. Las ventajas derivadas de la expansión de la red, la sincronización y el intercambio de información, así como la posibilidad de compartir software y dispositivos, contribuyen a establecer la red WAN como una herramienta fundamental para aquellas empresas que tienen varios usuarios en su red (Dodd, 2019).

2.16. PANTALLA LCD

Un panel de cristal líquido (LCD) es un dispositivo plano u otro componente óptico que funciona mediante la modulación electrónica de las características de los cristales líquidos en combinación con polarizadores. En lugar de emitir luz directamente, los cristales líquidos aprovechan una fuente de luz de fondo o un reflector para crear imágenes en colores o en blanco y negro. Estos paneles LCD pueden ser utilizados para mostrar diversas imágenes, como en monitores de uso general en computadoras, o imágenes estáticas con contenido reducido que pueden ser mostradas u ocultadas. Ejemplos de dispositivos con esta clase de pantallas incluyen relojes digitales que muestran palabras predefinidas, dígitos y pantallas de siete segmentos. Aunque comparten la misma base tecnológica, las imágenes arbitrarias son generadas a partir de una matriz de píxeles pequeños, mientras que en otros tipos de pantallas los elementos son más grandes. La propiedad de estar encendido (positivo) o apagado (negativo) en los paneles LCD depende de la disposición del polarizador (Hallberg, 2020).

Arduino es una plataforma de código abierto tanto en hardware como en software que faculta a los usuarios para diseñar proyectos interactivos que involucran sensores, actuadores y otros elementos electrónicos. Se puede conectar un LCD a un Arduino para mostrar información en la pantalla. Para hacer esto, se necesitan algunos componentes adicionales, como una placa de expansión para el LCD y algunas resistencias, así como código de programación en el Arduino para controlar la pantalla.

La librería LiquidCrystal es una biblioteca de software para el Arduino que permite controlar un LCD (Liquid Crystal Display) de una manera sencilla y conveniente. Esta

librería proporciona una serie de funciones que permiten enviar instrucciones al LCD y mostrar información en él.

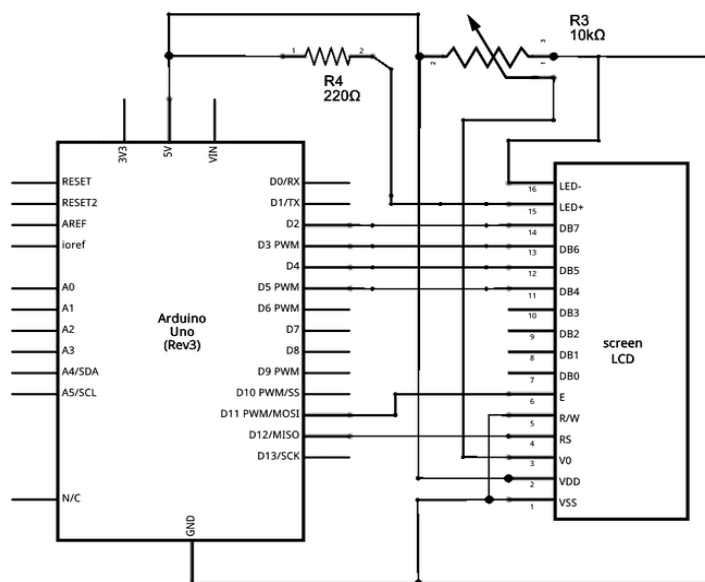
Con la librería LiquidCrystal, se puede hacer cosas como:

- Configurar el tamaño del LCD (número de columnas y filas)
- Mostrar texto en el LCD
- Establecer la posición del cursor en el LCD
- Controlar el brillo y el contraste del LCD

La librería LiquidCrystal es muy fácil de usar y permite agregar una pantalla de visualización a tus proyectos de Arduino en muy poco tiempo (Natheem, 2021).

Figura 5

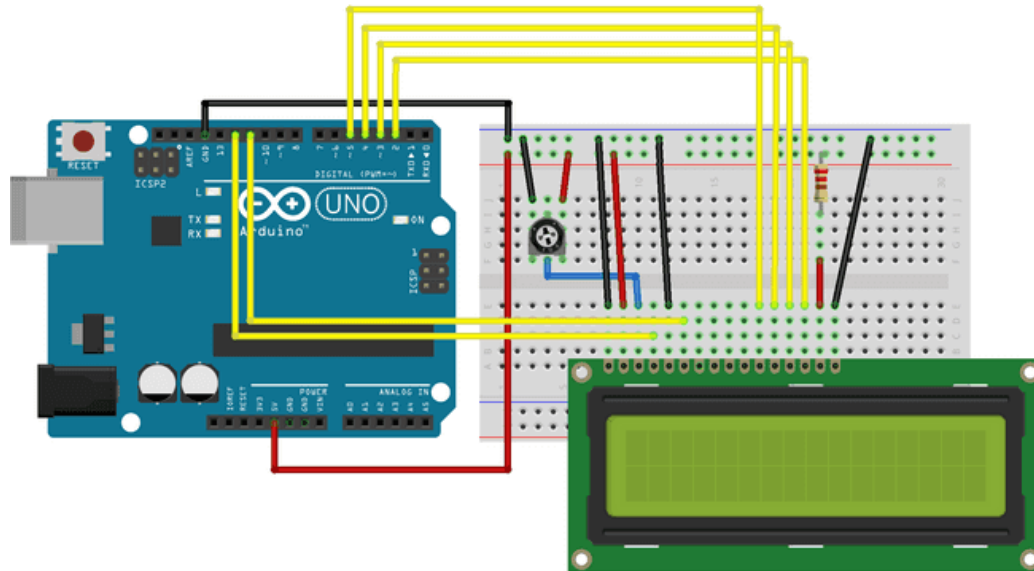
Diagrama de conexión de Arduino con LCD



Nota: (Natheem, 2021)

Figura 6

Conexión física de Arduino con LCD



Nota: (Natheem, 2021)

2.17. PROTOCOLOS DE COMUNICACIÓN EN INTERNET

Los protocolos de comunicación en Internet consisten en normativas y acuerdos que detallan el proceso de envío y recepción de datos en una red de computadoras. Estas normas resultan vitales para asegurar una comunicación eficiente y fiable entre dispositivos y redes. Aquí hay algunos de los protocolos de comunicación más comunes en Internet:

- TCP (Protocolo de Control de Transmisión): Este protocolo se encarga de asegurar la transmisión confiable de datos en Internet. TCP segmenta los datos en paquetes y los transfiere de un equipo a otro. Si ocurre la pérdida o corrupción de un paquete durante la transmisión, TCP lo identifica y lo reenvía hasta su recepción correcta.
- UDP (Protocolo de Datagramas de Usuario): Este protocolo es más simple y rápido que TCP, aunque no garantiza la entrega de los datos. UDP divide los datos en paquetes y los envía sin verificar si llegaron correctamente. Se emplea en aplicaciones



que necesitan transmisiones rápidas y eficientes, como juegos en línea o transmisiones de video en tiempo real.

- IP (Protocolo de Internet): Este protocolo tiene la responsabilidad de encaminar los datos a través de Internet. IP segmenta los datos en paquetes y los dirige a través de la red hasta su destino final. Cada paquete lleva la dirección IP de origen y destino para facilitar el enrutamiento.
- HTTP (Protocolo de Transferencia de Hipertexto): Este protocolo es fundamental en la World Wide Web. Se usa para transmitir los datos que componen las páginas web desde los servidores web a los navegadores web.
- FTP (Protocolo de Transferencia de Archivos): Este protocolo se emplea para transferir archivos de un servidor a otro o de una computadora a otra. FTP proporciona una interfaz eficiente y sencilla para el intercambio de archivos.

Estos son solo algunos de los protocolos de comunicación en Internet más comunes. Hay muchos otros protocolos que son imprescindibles para el buen funcionar de Internet, como DNS, SMTP, POP3 y otros. Todos estos protocolos trabajan juntos para lograr la comunicación efectiva y segura de datos a través de Internet.

2.18. LENGUAJE DE PROGRAMACION C

C es un lenguaje de programación de alto nivel con un propósito general, reconocido por su eficiencia y versatilidad. Su creación se atribuye a Dennis Ritchie en la década de 1970, mientras trabajaba en Bell Labs. Distinguido por su uso extendido en la programación de sistemas y aplicaciones, C también ha tenido una influencia significativa en el desarrollo de otros lenguajes notables, como C++ y Java. C es conocido por ser un lenguaje de bajo



nivel, lo que le permite tener un control detallado sobre el hardware de un sistema, pero también puede ser utilizado para desarrollar aplicaciones de alto nivel con una sintaxis clara y concisa (Szuhay, 2020).

- Directivas de preprocesador: son instrucciones que se procesan antes de que el compilador vea el código fuente. Estas instrucciones incluyen la inclusión de archivos de cabecera y la definición de constantes.
- Declaraciones de variables: son las líneas en las que se declaran las variables que se van a utilizar en el programa.
- Funciones: son bloques de código que realizan tareas específicas. La función principal es la función donde comienza la ejecución del programa.
- Sentencias de control de flujo: son las instrucciones que permiten controlar el flujo de ejecución de un programa, como if, while, y for.
- Comentarios: son líneas de texto que se ignoran por el compilador y que sirven para describir el código y facilitar su comprensión.
- Espacios en blanco y saltos de línea: son caracteres utilizados para separar elementos en el código y mejorar su legibilidad.

Estos elementos son combinados para crear programas complejos que realizan tareas específicas. Es importante seguir buenas prácticas de programación y utilizar una sintaxis clara y concisa para facilitar la mantenibilidad y la comprensión del código (Team Codex, 2022).



CAPITULO III

MATERIALES Y MÉTODOS

3.1. MATERIALES

3.1.1. Hardware

Ordenador:

- Lenovo Slim Intel i7 de 12^a generación (Windows 11 Home)
- Pantalla de 16 pulgadas
- Memoria de 16 GB
- Almacenamiento de 1 TB
- Gráficos Intel Arc A370M

Arduino:

- ATmega328
- 14 pines digitales de entrada/salida
- Oscilador de vidrio de 16 MHz
- Conexión USB
- EEPROM 1 KB
- SRAM 2 KB
- Flash Memory 32 KB



3.1.2. Software

- Arduino IDE
- Sistema Operativo de 64 bits con Windows 11 Home
- Microsoft Word 2016 profesional

3.2. DISEÑO Y NIVEL DE LA INVESTIGACIÓN

3.2.1. Diseño de investigación

El tipo de investigación es experimental, ya que busca monitorear en tiempo real la temperatura de la parrilla, para lo cual se implementó un prototipo electrónico con internet de las cosas y se desarrolló un sistema web para la visualización de valores en una red de internet. Los experimentos pueden realizarse tanto en entornos de laboratorio como en situaciones de la vida real. Estas pruebas son particularmente efectivas para investigaciones de naturaleza aplicada y explicativa, las cuales a menudo se restringen a temas en los que es posible manipular las condiciones en las que se encuentran las personas. (Hernández-Sampieri & Mendoza, 2020).

3.2.2. Nivel de la investigación

El nivel de investigación adoptado es exploratorio, ya que este enfoque se emplea para estudiar fenómenos que no han sido investigados anteriormente y existe un interés en analizar sus atributos. (Ramos Galarza, 2020).

3.3. MUESTRA DE LA INVESTIGACIÓN

El muestreo por conveniencia, el cual es tipo de muestra no probabilístico indica que la muestra se elige de acuerdo con la conveniencia de investigador, le permite elegir de manera arbitraria el número de participantes que puede haber en el estudio. En esta investigación se considera como muestra al prototipo realizado.

3.4. UBICACIÓN DE LA INVESTIGACIÓN

El trabajo de investigación se realizará en el distrito Puno en la provincia de Puno en la región de Puno.

Figura 7

Ciudad de Puno



Elaboración propia

3.5. TÉCNICAS E INSTRUMENTOS PARA LA RECOLECCION DE DATOS

3.5.1. Técnicas

En este estudio se emplea el método de observación, que representa la manera más estructurada y lógica de registrar de forma visual y verificable aquello que se busca comprender. En términos sencillos, esta técnica facilita la captura imparcial de los sucesos en el mundo real, ya sea con el propósito de describirlos, analizarlos o explicarlos desde una perspectiva científica. Este enfoque difiere de cómo las personas en su vida cotidiana utilizan información y datos observados de manera práctica en el



mundo concreto, con el fin de resolver problemas o satisfacer sus necesidades. Los datos tienen que ser verificados en el proceso de la investigación, en base a ello se hará el sistema de control de temperatura.

3.5.2. Instrumentos

Un investigador puede utilizar varios instrumentos para recolectar datos mediante la técnica de observación, por ejemplo:

- Registros de observación: son notas escritas tomadas por un observador sobre un fenómeno o comportamiento específico.
- Cámaras de video: permiten grabar la observación para su revisión posterior.
- Herramientas de medición: incluyen medidores de tiempo, medidores de distancia y otros instrumentos que permiten medir variables específicas durante la observación.

3.6. PROCEDIMIENTO

3.6.1. Adquisición de datos de temperatura

La temperatura se representa mediante un valor numérico que indica el nivel de calor o frío presente en un objeto, sustancia o entorno específico. Este valor numérico está relacionado con la energía cinética y el movimiento de las partículas que componen dicha sustancia. La medición de la temperatura se puede realizar utilizando varias escalas, siendo las más comunes la escala Celsius ($^{\circ}\text{C}$) y la escala Fahrenheit ($^{\circ}\text{F}$). En la escala Celsius, el punto de congelación del agua se establece en 0°C , mientras que su punto de ebullición a nivel del mar es 100°C . En la escala Fahrenheit, el punto de congelación del agua es 32°F y el punto de ebullición a nivel del mar es 212°F .



Además, la temperatura puede expresarse en la escala Kelvin (K), que es una escala absoluta donde el cero absoluto, que representa la falta total de calor, se encuentra en 0 Kelvin (-273.15°C). La temperatura ejerce una influencia considerable en distintas propiedades físicas y químicas de las sustancias, como la densidad, la viscosidad, la presión de vapor y la velocidad de las reacciones químicas. También desempeña un papel crucial en los seres vivos, ya que muchas especies son sensibles a los cambios térmicos y tienen rangos óptimos de temperatura para sobrevivir y funcionar de manera adecuada.

En esta investigación se medirá la temperatura en grados Celsius ($^{\circ}\text{C}$), para medir este parámetro en nuestra parrilla se usará una termocupla, el cual es un sensor de temperatura utilizado para medir y monitorear cambios en la temperatura de forma precisa. Consiste en dos cables metálicos diferentes que se unen en un extremo para formar una unión de medición. Los metales utilizados en las termocuplas pueden variar, siendo los más comunes el hierro, el cobre, el níquel y el cromo. Al aplicar una diferencia de temperatura a lo largo de la termocupla, se genera una pequeña corriente eléctrica en el circuito gracias al efecto Seebeck. Esta corriente es proporcional a la diferencia de temperatura entre la unión de medición y el extremo de referencia de la termocupla. Al medir esta corriente, se puede determinar con precisión la temperatura en la unión de medición. Las termocuplas son ampliamente utilizadas en diversas industrias debido a su amplio rango de temperatura de funcionamiento, su resistencia y su capacidad para soportar condiciones adversas, como ambientes corrosivos o alta presión. Además, pueden utilizarse en aplicaciones de alta temperatura, incluso superiores a los 1000°C .



En esta investigación, se utiliza un termopar de tipo K en conjunto con un dispositivo llamado MAX6675. Este dispositivo, que funciona como un convertidor analógico a digital diseñado específicamente para termopares de tipo K, permite la conexión de un termopar a cualquier microcontrolador que admita la interfaz SPI.

Dentro del MAX6675 se incluye la electrónica necesaria para amplificar, compensar y convertir la señal de voltaje generada por el termopar en una señal digital, lo que simplifica de manera significativa la tarea de conectar este tipo de sensor a un microcontrolador. La única restricción es que este circuito se encuentra disponible únicamente en un encapsulado SOIC, lo que presenta desafíos en cuanto a su uso en un protoboard. No obstante, este módulo contiene tanto el MAX6675 como todos los componentes electrónicos esenciales y terminales adecuadas, lo cual simplifica la incorporación en prototipos.

Las características de este módulo se detallan a continuación:

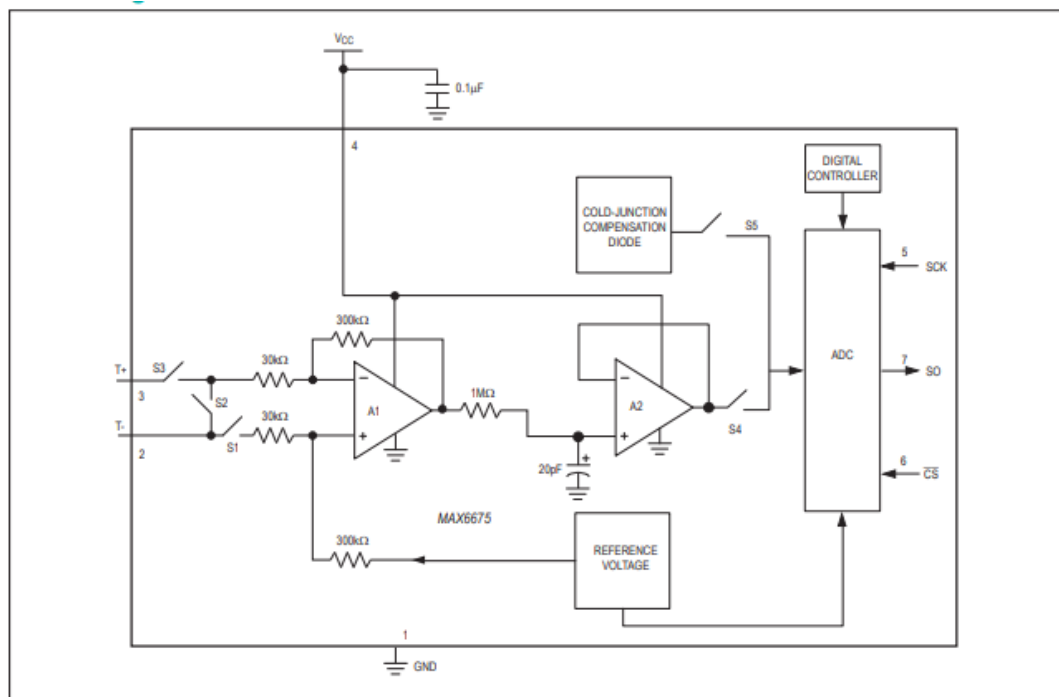
- Elaborador: Genérico
- Interfaz: Digital
- Estándar de comunicación: SPI (solamente en modo de lectura)
- Tensión de suministro: 3.0 a 5.5 VCC
- Corriente máxima en funcionamiento: 1.5 mA
- Corriente típica en reposo: 0.7 mA
- Precisión: 12 bits (equivalente a 0.25 °C)

- Rango máximo de temperatura soportada: 1024 °C
- Frecuencia del reloj para la comunicación SPI: 4.3 MHz
- Tiempo de conversión máximo: 0.22 segundos
- Compensación para la temperatura de la unión fría
- Detección de termopar abierto o desconectado
- El módulo MAX6675 cuenta con terminales de diseño con tornillo
- Incluye un termopar de tipo K

El siguiente circuito detalla el funcionamiento del módulo MAX6675:

Figura 8

Circuito Interno del módulo MAX 6675

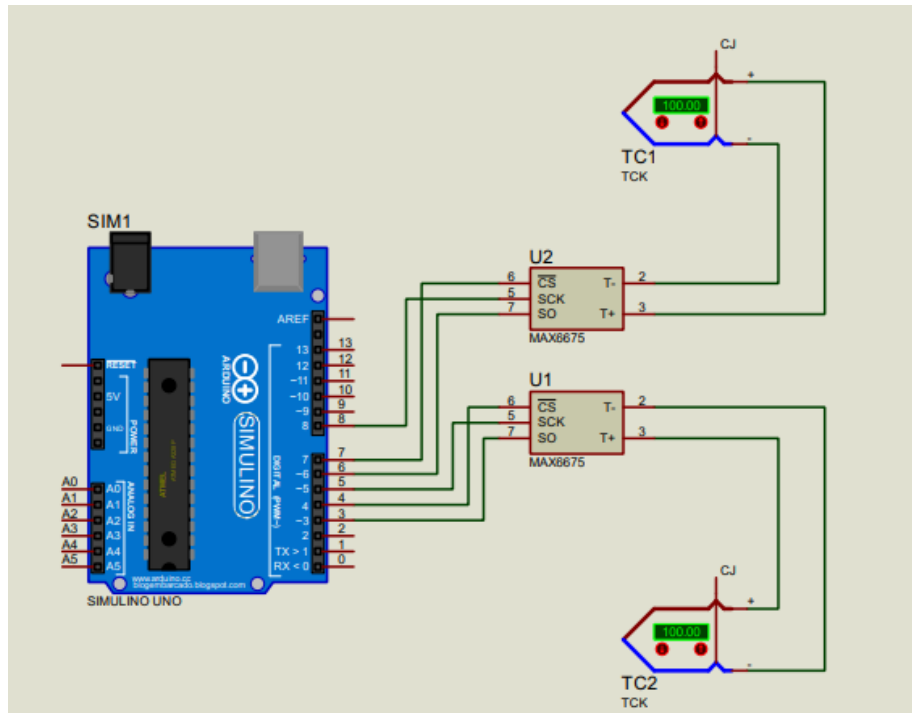


Nota: (Products, 2021)

La conexión del termopar y su módulo con el Arduino se muestra en el siguiente diagrama:

Figura 9

Conexión de termopares a la placa Arduino



Elaboración propia

3.6.2. Control on/off del sistema de cocción

El control On/Off, también conocido como control de encendido/apagado o control binario, es un tipo de control simple y ampliamente utilizado en la automatización y sistemas de control. El objetivo del control On/Off es mantener una variable o proceso dentro de ciertos límites establecidos, mediante la activación o desactivación de un dispositivo de control cuando la variable alcanza ciertos valores de referencia.



En el caso específico del presente estudio, el control On/Off se implementa para activar o desactivar un ventilador (fan) en función del valor promedio de temperatura ($temp_prom$) obtenido de dos sensores de termopar ($ktc1$ y $ktc2$). La lógica de control On/Off en el código es la siguiente: Si el valor promedio de temperatura ($temp_prom$) es mayor que 48 pero menor que 50 grados, no se hace nada (el soplador permanece en su estado actual). Si el valor es menor a 48 grados, el soplador se encenderá y subirá la temperatura, manteniendo la temperatura constante.

Por ello, este enfoque de control On/Off es adecuado para situaciones en las que se necesita mantener una temperatura cerca de un valor objetivo. Sin embargo, tiene algunas limitaciones, como la oscilación alrededor del punto de referencia y el hecho de que el dispositivo de control (el ventilador en este caso) está funcionando completamente encendido o completamente apagado, lo que puede provocar ciclos frecuentes de encendido y apagado en situaciones cercanas al punto de referencia.

Como parte del equipo tenemos un "blower" o soplador el cual es un término utilizado para referirse a un tipo específico de ventilador que es capaz de mover grandes volúmenes de aire a través de un sistema o equipo. Los blowers se utilizan en diversas aplicaciones industriales, HVAC (calefacción, ventilación y aire acondicionado), equipos electrónicos, entre otros. Además, están diseñados para proporcionar un flujo de aire más alto que los ventiladores convencionales y, a menudo, están optimizados para aplicaciones que requieren un alto rendimiento de ventilación y presión estática. Tienen diferentes diseños y tamaños según la aplicación y los requisitos específicos. Los blowers pueden diseñarse para funcionar con diferentes voltajes, como 12V, 24V, 48V, etc. y los parámetros de corriente pueden variar según el modelo y fabricante,



por lo que es esencial revisar las especificaciones proporcionadas por el fabricante antes de su uso.

Por otro lado, esta “El tubo corrugado” para mejor conducción del aire hacia el carbón el cual es un tipo de tubo flexible utilizado en diversas aplicaciones para mejorar la conducción del aire u otros fluidos. A diferencia de los tubos lisos convencionales, el tubo corrugado presenta una estructura ondulada o corrugada en su pared exterior. Estas corrugaciones crean espacios o canales entre las crestas y valles que permiten ciertas ventajas en la conducción de aire o fluidos. Algunas de las características más destacadas son las siguientes:

- **Flexibilidad:** La principal ventaja del tubo corrugado es su flexibilidad. Las corrugaciones permiten que el tubo se doble y se adapte fácilmente a diferentes trayectorias o curvas, lo que es útil en situaciones donde los tubos rígidos no son prácticos.
- **Resistencia a la compresión:** Las corrugaciones aumentan la resistencia del tubo corrugado a la compresión. Esto es beneficioso en aplicaciones donde el tubo podría estar sujeto a fuerzas externas o al peso de otros objetos sin colapsar.
- **Protección mecánica:** El diseño corrugado proporciona una mayor protección mecánica al tubo, lo que lo hace más resistente a la abrasión y al daño causado por roces o golpes.



- **Reducción de ruido:** En algunos casos, el tubo corrugado puede reducir la transmisión de ruidos y vibraciones, lo que es útil en aplicaciones donde se requiere un funcionamiento silencioso.
- **Aislamiento térmico:** Dependiendo del material utilizado en el tubo corrugado, este puede ofrecer cierto grado de aislamiento térmico para proteger el flujo de aire o fluidos de cambios bruscos de temperatura.
- **Facilidad de instalación:** Gracias a su flexibilidad, el tubo corrugado es fácil de manipular e instalar en diferentes entornos y configuraciones.

Debido a estas características, el tubo corrugado es ampliamente utilizado en diversas industrias y aplicaciones, como sistemas de ventilación y aire acondicionado, sistemas de extracción de humos, protección de cables y alambres, conducción de fluidos en maquinaria, y muchas otras aplicaciones donde la flexibilidad y la protección son importantes.

Así también, como regulador de potencia tenemos un “Dimer”, es un dispositivo utilizado para controlar la cantidad de energía eléctrica que se suministra a una carga eléctrica, como una bombilla o una lámpara. El objetivo principal del dimmer es ajustar la intensidad luminosa o la velocidad de dispositivos eléctricos controlables, lo que permite aumentar o disminuir la intensidad de la luz o la velocidad del motor, por ejemplo. El funcionamiento básico de un dimmer se basa en el principio de modulación del ancho de pulso (PWM, por sus siglas en inglés). En lugar de reducir directamente el voltaje suministrado a la carga, un dimmer rápidamente alterna entre un estado de encendido y apagado. El porcentaje del tiempo que está en el estado de



encendido en relación con el tiempo total se conoce como el ciclo de trabajo. Al cambiar la proporción del ciclo de trabajo, se varía la cantidad de energía entregada a la carga y, por lo tanto, se controla su potencia o velocidad.

Adicionando a todo ello también se tiene un “Relay”, el cual es un componente electromecánico que permite controlar circuitos eléctricos mediante el accionamiento de una bobina, que, a su vez, cambia el estado de los contactos eléctricos para abrir o cerrar un circuito, facilitando así el control de cargas eléctricas de alta potencia o voltaje mediante señales de bajo voltaje.

El sistema de control on/off está integrado con todos los componentes descritos con anterioridad, de tal manera que, si Arduino lee los datos de temperatura y saca el promedio de los dos sensores, este tomará una decisión, que puede ser activar o desactivar el relé. El relé activa un soplador que es regulada en potencia por un dimmer, este tiene el objetivo de mantener la temperatura en no menos de 48°C, esto puede variar según el lugar donde nos encontremos, ya que el Perú cuenta con varias regiones. En este caso nos encontramos en la región Suni, y es por ello que se ha determinado los grados de temperatura de la cocción de alimentos que puede ser carnes y vegetales.

3.6.3. Exportación de datos de temperatura al servidor

En primer lugar, tenemos al servidor web el cual es un software que aloja y entrega páginas web y otros contenidos a través de Internet o una red local. Procesa solicitudes de navegadores web, enviando respuestas con páginas web, imágenes y archivos. Utiliza protocolos HTTP o HTTPS para la comunicación y puede alojar diferentes tipos de sitios web. También ejecuta aplicaciones y sirve contenido dinámico para ofrecer servicios personalizados a los usuarios.



También tenemos una "Shield Ethernet" esta es una placa de expansión que se conecta directamente a una placa Arduino para agregar funcionalidad de red Ethernet a proyectos basados en Arduino. En esencia, es una extensión que permite que un Arduino se comunice con redes locales o a través de Internet utilizando el protocolo Ethernet. La Shield Ethernet generalmente incluye un chip Ethernet, como el Wiznet W5100 o W5500, que maneja la comunicación de red. También puede tener un conector Ethernet RJ45 para conectar el cable de red y una interfaz que permite configurar parámetros de red, como la dirección IP, la máscara de subred y la puerta de enlace.

Además, con una Shield Ethernet, un Arduino puede realizar diversas tareas relacionadas con la comunicación en red, como:

- Establecer una conexión TCP/IP: Puede actuar como un cliente o un servidor TCP/IP y comunicarse con otros dispositivos a través de conexiones TCP (Protocolo de Control de Transmisión) o UDP (Protocolo de Datagramas de Usuario).
- Comunicación con servidores web: Puede enviar solicitudes HTTP y recibir respuestas de servidores web, lo que permite acceder a servicios en línea y obtener información de páginas web.
- Control remoto: Permite el control y monitoreo remoto de un Arduino desde otros dispositivos conectados en la misma red o a través de Internet.
- Comunicación entre Arduinos: Dos o más Arduinos equipados con Shield Ethernet pueden intercambiar datos entre ellos a través de la red local.

La Shield Ethernet es una opción popular y poderosa para proyectos de Internet de las Cosas (IoT), automatización del hogar, monitoreo y control de sistemas en red, y otras aplicaciones que requieran conectividad en red. En este proyecto este módulo que se acopla a Arduino permite el traslado de datos hacia la Raspberry pi.

La raspberry pi usada en este proyecto permite crear una red LAN a través de su puerto USB, y es en esa red LAN donde se conecta el módulo ethernet de Arduino. Raspberry pi envía datos al servidor, es preciso mencionar que este actúa en modo enrutador.

3.6.4. Desarrollo del código en arduino

Teniendo en cuenta que una biblioteca es una colección de partes del programa que hacen cosas comunes y/o especializadas que evitan que el programador necesite programarlas nuevamente al escribir software. Por lo general, consta de funciones para llamar y objetar clases que puede instanciar. Un ejemplo común podría ser funciones que se ocupan de fechas y horas, y uno especializado podría ser un software para manipular un termostato inteligente, para nuestro código de Arduino, se utiliza las librerías siguientes:

Figura 10

Librerías usadas

```
#include <SPI.h>  
#include <Ethernet.h>  
#include "max6675.h"
```

Elaboración propia

Se detalla a continuación:



- SPI.h: Proporciona funciones para trabajar con la comunicación Serial Peripheral Interface (SPI), que se utiliza para comunicarse con dispositivos periféricos como sensores, pantallas, tarjetas SD, etc., a través de un bus de comunicación serie.
- Ethernet.h: Permite habilitar la conectividad Ethernet en Arduino, lo que permite al dispositivo comunicarse a través de una red local o conectarse a Internet.
- max6675.h: Esta es una biblioteca personalizada específica para el sensor de temperatura MAX6675, que proporciona funciones para leer las temperaturas de dicho sensor a través del protocolo SPI.

En Arduino, `#include` es una directiva del preprocesador que se utiliza para incluir bibliotecas en un programa. Cuando utilizas `#include` seguido del nombre de una biblioteca, estás indicando al compilador que agregue el contenido de esa biblioteca al código fuente antes de compilar el programa completo.

La declaración de variables es una parte fundamental en cualquier lenguaje de programación, incluido Arduino. En términos comunes, una declaración de variable le indica al programa qué tipo de dato va a almacenar esa variable y cuál será su nombre. Al declarar una variable, se reserva un espacio en la memoria del sistema para almacenar el valor de esa variable durante la ejecución del programa.

Como en muchos lenguajes de programación, las variables pueden ser de diferentes tipos. Los tipos más comunes son:

- `int`: Representa un número entero (positivo o negativo) sin parte decimal. Por ejemplo: 1, -10, 100, etc.
- `float`: Representa un número decimal de punto flotante, es decir, un número con parte decimal. Por ejemplo: 3.14, -0.5, 10.75, etc.
- `char`: Representa un solo carácter (letra, número o símbolo) y se representa entre comillas simples. Por ejemplo: 'A', 'b', '1', '&', etc.
- `bool`: Representa un valor lógico, que puede ser verdadero (`true`) o falso (`false`).

Existen otros tipos avanzados como `double`, `byte`, `unsigned int`, etc., que proporcionan más precisión o control sobre el rango de valores permitidos.

Figura 11

Variables

```
int ktc1SO = 3;  
int ktc1CS = 4;  
int ktc1CLK = 5;  
int ktc2SO = 6;  
int ktc2CS = 7;  
int ktc2CLK = 8;  
float temp_prom;  
int fan;  
float temp1=0;  
float temp2=0;
```

Elaboración propia

Se declaran varias variables de tipo `int` (entero) con nombres `ktc1SO`, `ktc1CS`, `ktc1CLK`, `ktc2SO`, `ktc2CS` y `ktc2CLK`. Cada una de estas variables se inicializa con un valor numérico, el cual indica el número de pin que se usará para la conexión con el módulo del sensor. Adicionalmente se declaran variables flotantes, en ellos se almacenarán valores con parte decimal, tenemos a “`temp_prom`”, en esta variable

almacenaremos el promedio de los valores de temperatura obtenidos de los termopares y finalmente tenemos a temp1 y temp2 que inicializamos en el valor numérico 0.

La inicialización de objetos de la clase MAX6675 se utilizan para interactuar con sensores de temperatura tipo termopar MAX6675 en un microcontrolador, como Arduino.

Figura 12

Objetos

```
MAX6675 ktc1(ktc1CLK, ktc1CS, ktc1S0);  
MAX6675 ktc2(ktc2CLK, ktc2CS, ktc2S0);
```

Elaboración propia

Para su mejor entendimiento:

El código inicializa dos objetos ktc1 y ktc2 de la clase MAX6675, que se utilizan para interactuar con sensores de temperatura tipo termopar MAX6675 en un microcontrolador como Arduino. Cada objeto representa un sensor MAX6675 conectado físicamente a pines específicos del microcontrolador.

Con esta configuración, podemos utilizar los objetos ktc1 y ktc2 para leer los datos de temperatura de los termopares conectados y realizar acciones basadas en esas lecturas, como mostrar la temperatura en una pantalla, activar un soplador como en nuestro caso o tomar decisiones en un sistema de control de temperatura.

Para conectar Arduino a un servidor a través de Ethernet, usamos un módulo Ethernet compatible, y una conexión a Internet activa en la red local. Para comenzar, debemos configurar la dirección IP, la puerta de enlace y la máscara de subred de Arduino utilizando las variables ip, gateway y subnet, respectivamente, como se



muestra en el código. Además, es necesario asignar una dirección MAC única a Arduino utilizando la variable `mac`. Esto permitirá que Arduino se comunique en la red Ethernet.

Una vez configurada la red, se usa el objeto `EthernetClient` `client`; para establecer una conexión TCP con el servidor remoto. Para esto, debes especificar la dirección IP del servidor y el puerto al que deseas conectarte. Luego, puedes enviar y recibir datos a través de esta conexión usando métodos como `client.connect()`, `client.write()` y `client.read()`.

Figura 13

Configuración para conexión a internet

```
MAX6675 ktc1(ktc1CLK, ktc1CS, ktc1S0);
MAX6675 ktc2(ktc2CLK, ktc2CS, ktc2S0);

byte mac[] = { 0x54, 0xAB, 0x3A, 0x5B, 0xAD, 0x05 };
IPAddress server(108,62,123,70);
IPAddress ip(172,16,100,133);
byte gateway[] = { 172, 16, 100, 1 };
byte subnet[] = { 255, 255, 255, 0 };
byte myDns[] = { 8, 8, 8, 8 };
EthernetClient client;
bool printWebData = true;
char sensor_id[] = "DIAB-01";
```

Elaboración propia

Para la conexión con el servidor definimos los parámetros siguientes:

- **byte mac [] = {0x54, 0xAB, 0x3A, 0x5B, 0xAD, 0x05}:** Aquí se define la dirección MAC (Media Access Control) del dispositivo Ethernet. Cada dispositivo en una red Ethernet tiene una dirección MAC única.



- **IPAddress server (108,62,123,70):** Se establece la dirección IP del servidor al que el dispositivo se conectará. Esta dirección IP representa el destino de las solicitudes o datos que el dispositivo enviará o recibirá.
- **IPAddress ip (172,16,100,133):** Se define la dirección IP que se asignará al dispositivo Ethernet. Esta es la dirección IP local utilizada por el dispositivo en la red local.
- **byte gateway [] = {172, 16, 100, 1}:** La dirección IP de la puerta de enlace. La puerta de enlace es el punto de acceso para enviar datos fuera de la red local.
- **byte subnet [] = {255, 255, 255, 0}:** La máscara de subred determina qué parte de la dirección IP es la dirección de red y qué parte es la dirección de host.
- **byte myDns [] = {8, 8, 8, 8}:** La dirección IP del servidor DNS que el dispositivo utilizará para resolver nombres de dominio en direcciones IP.
- **EthernetClient client:** Se crea un objeto client de la clase EthernetClient, que se utilizará para establecer una conexión con el servidor remoto y enviar/recibir datos a través de esa conexión.
- **bool printWebData = true:** Esta variable booleana determina si el programa imprimirá datos obtenidos del servidor en la consola serie (por ejemplo, para propósitos de depuración). Si es true, imprimirá los datos; si es false, no lo hará.
- **char sensor_id [] = "DIAB-01":** Aquí se define un arreglo de caracteres que representa una etiqueta o identificador del sensor. Puede ser utilizado para

etiquetar los datos enviados al servidor, para que el servidor pueda identificar la fuente de los datos.

Esta parte de código configura la conexión de red Ethernet para el dispositivo Arduino, estableciendo la dirección MAC, la dirección IP, la puerta de enlace, la máscara de subred y la dirección del servidor DNS. Luego, crea un objeto cliente para comunicarse con un servidor remoto cuya dirección IP está definida en `server`. Además, establece una variable booleana para determinar si se imprimirán los datos obtenidos del servidor y define un identificador del sensor (`sensor_id`). Con esta configuración, el dispositivo puede conectarse a Internet a través de Ethernet y enviar/recibir datos al servidor remoto.

La función `setup()` se ejecuta una vez al inicio del programa, y es donde generalmente se realizan las configuraciones iniciales necesarias para el funcionamiento del dispositivo.

Figura 14

Setup del código

```
void setup(){  
  Ethernet.begin(mac, ip, myDns, gateway, subnet);  
  delay(500);  
  pinMode(2,OUTPUT);  
}
```

Elaboración propia

Veamos qué hace este código:

- **Ethernet.begin(mac, ip, myDns, gateway, subnet):** Esta línea es la parte principal del código. Aquí se inicia la conexión Ethernet utilizando los valores



de configuración previamente definidos para la dirección MAC, dirección IP, servidor DNS, puerta de enlace y máscara de subred. Estos valores se pasan como argumentos a la función `Ethernet.begin()`. Con esta llamada, Arduino intentará establecer una conexión a la red utilizando los parámetros proporcionados. Si todo va bien, Arduino quedará conectado a la red Ethernet y podrá enviar y recibir datos a través de la conexión.

- **delay (500):** Esta línea agrega un retraso de 500 milisegundos después de que se inicia la conexión Ethernet. Esto puede ser útil en algunas situaciones para asegurarse de que la conexión se establezca correctamente antes de continuar con otras operaciones.
- **pinMode (2, OUTPUT):** Esta línea establece el pin 2 como una salida digital. Esto significa que el pin 2 se utilizará para enviar una señal de voltaje (alta o baja, 5V o 0V) en lugar de leer una señal.

Lo siguiente en el código es un bucle que se ejecuta continuamente en un microcontrolador o dispositivo similar. Explicación paso a paso lo que realiza el código:

Figura 15

Lectura de temperatura de las termocuplas

```
void loop(){
  delay(1000);
  temp1=ktc1.readCelsius();
  temp2=ktc2.readCelsius();
  temp_prom=(temp1 + temp2)/2;
  if (temp_prom<=50){
    if (temp_prom<=48 ){
      digitalWrite(2,HIGH);
      fan=1; }
    else{
      }
  }
  else {
    digitalWrite(2,LOW);
    fan=0;
  }
}
```

Elaboración propia

- void loop(): Esta es la función principal que se ejecuta en un bucle infinito.
- delay(1000): Aquí se introduce un retraso de 1000 milisegundos (1 segundo) antes de continuar con las siguientes instrucciones del bucle. Esto significa que todo el contenido dentro del bucle se ejecutará una vez cada segundo.
- temp1=ktc1.readCelsius(): Se lee el valor de la temperatura del primer sensor y se almacena en la variable temp1.
- temp2=ktc2.readCelsius(): Se lee el valor de la temperatura del segundo sensor y se almacena en la variable temp2.
- temp_prom = (temp1 + temp2) /2: Se calcula el promedio de las dos temperaturas leídas y se almacena en la variable temp_prom.

A continuación, hay una serie de comprobaciones para controlar el ventilador en función del valor de temp_prom:

- Si temp_prom es menor o igual a 50, se realiza otra comprobación interna:
- Si temp_prom es menor o igual a 48, se establece el pin 2 como alto (HIGH) y se asigna el valor 1 a la variable fan.
- Si temp_prom está entre 48 y 50, no se hace nada en esta parte del código.
- Si temp_prom es mayor que 50, se establece el pin 2 como bajo (LOW) y se asigna el valor 0 a la variable fan.

Luego, el código intenta conectarse a un servidor (definido por la variable server) en el puerto 80. Utiliza una conexión HTTP para enviar datos al servidor.

Figura 16

Código para conexión a Internet

```
if (client.connect(server, 80)) {
    client.print("GET /TEMPFAN/formaction_tempfan.php?temp_prom=");
    client.print(temp_prom);
    client.print("&fan=");
    client.print(fan);
    client.print("&sensor_id=");
    client.print(sensor_id);
    client.println(" HTTP/1.1");
    client.print("Host: ");
    client.println(server);
    client.println("Connection: close");
    client.println();
} else {
}
while (client.connected()) {
    int len = client.available();
    if (len > 0) {
        byte buffer[80];
        if (len > 80) len = 80;
        client.read(buffer, len);
        if (printWebData) {
        }
    }
}
if (!client.connected()) {
    client.stop();
    delay(28358);
}
}
```

Elaboración propia

Si la conexión se realiza con éxito (`client.connect(server, 80)`), el código envía una solicitud GET al servidor con los valores de `temp_prom`, `fan` y `sensor_id`. Después de enviar la solicitud GET, el código espera a recibir una respuesta del servidor. Sin embargo, parece que el código no hace nada con la respuesta (`buffer`) que recibe del servidor, ya que está comentado dentro del bloque `if (printWebData)`. Si no se logra establecer la conexión con el servidor, el código se queda vacío en el bloque `else`.

Finalmente, si la conexión con el servidor está cerrada o no se pudo conectar, el código llama a `client.stop()` para asegurarse de que la conexión esté cerrada y espera aproximadamente 28.358 segundos (aproximadamente 28 segundos) antes de continuar con el siguiente ciclo del bucle (`delay(28358)`).

3.6.5. Implementación física del prototipo

La implementación física de un prototipo es una etapa esencial en el desarrollo de productos o sistemas antes de su producción. Proporciona una representación tangible y funcional del diseño conceptual, lo que permite probar, evaluar y mejorar diferentes aspectos del producto en el mundo real. Las pruebas pueden ayudar a:

- **Identificar problemas y mejorar el diseño:** Al tener un prototipo físico, los diseñadores y desarrolladores pueden detectar problemas, deficiencias o áreas de mejora que pueden no haber sido evidentes en el diseño en papel o en modelos digitales. Esto permite realizar modificaciones y ajustes antes de llegar a la producción en masa, lo que ayuda a ahorrar tiempo y costos en el proceso.
- **Evaluación de la funcionalidad:** Un prototipo físico permite probar la funcionalidad del producto en condiciones reales. Los usuarios o expertos pueden interactuar con el prototipo para evaluar su rendimiento, usabilidad y

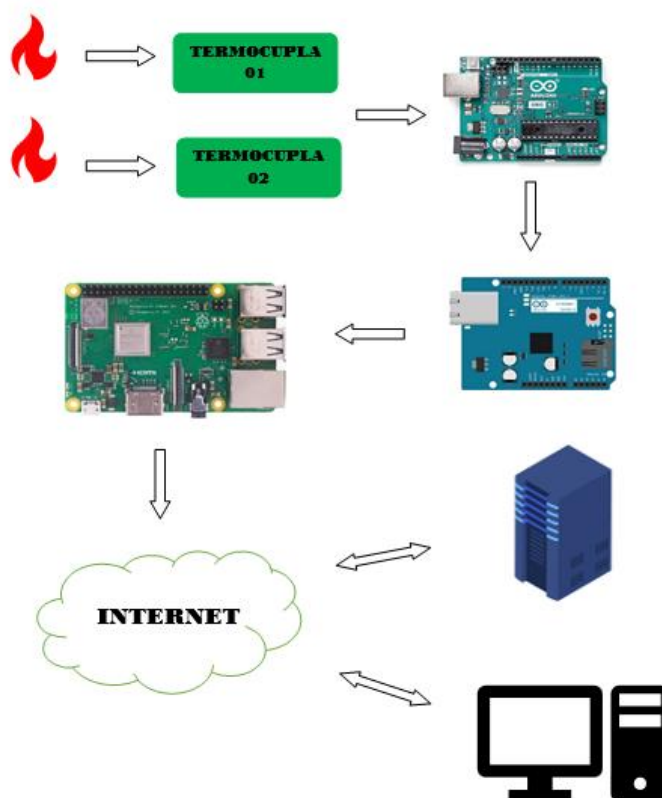
cumplimiento de los requisitos específicos. Esto garantiza que el producto final cumpla con las expectativas y necesidades de los usuarios.

- **Validación de costos y presupuestos:** La implementación física de un prototipo puede ayudar a evaluar los costos de producción y verificar si el presupuesto previsto es adecuado. Esto permite realizar ajustes en el diseño para optimizar los costos sin sacrificar la calidad o la funcionalidad.

La parte de conexión a internet en esta implementación del prototipo en esta basada en el siguiente diagrama:

Figura 17

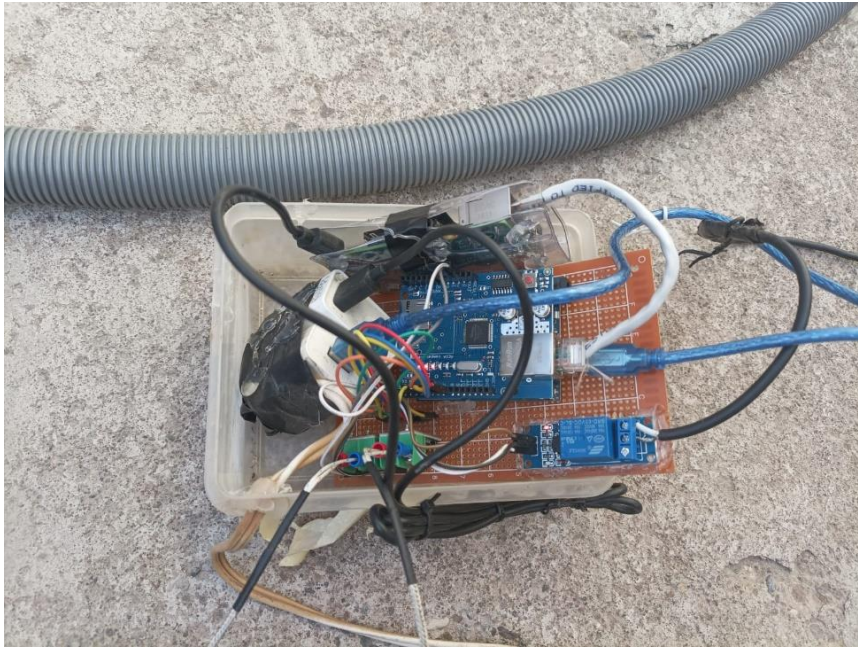
Diagrama del sistema de conexión a Internet



Elaboración propia

Figura 18

Conexión de módulos en el sistema de control



Elaboración propia

La termocupla está colocada lo más próximo al carbón, para que este lea la temperatura, en la siguiente imagen podemos lo detallado:

Figura 19

Termocupla ubicada lo más próximo al carbón



Elaboración propia

El Arduino lee los datos de temperatura con el sensor y este se activa o se desactiva de acuerdo al rango establecido, que en este caso si la temperatura es menor a 48° se activa el soplador.

Figura 20

Soplador acoplado al tubo corrugado



Elaboración propia

El flujo de viento proporcionado por el soplador tiene que regularse debido a que mucha potencia hace que las cenizas del carbón se dispersen, para regular esto es necesario un dimmer. La perilla del dimmer permite regular la potencia del soplador.

Figura 21

Dimmer que regula la potencia del soplador



Elaboración propia

El tubo corrugado permite una mejor conducción del aire y en el en esta investigación se presenta como en la imagen siguiente:

Figura 22

Tubo corrugado para conducción de aire



Elaboración propia



CAPITULO IV

RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS DE COCCIÓN DE LOS ALIMENTOS

La cocina al carbón puede proporcionar ventajas en términos de salud cuando se realiza de manera adecuada y equilibrada. Los resultados que se obtienen en la cocción es lo siguiente:

- **Menor adición de grasas:** La cocina al carbón tiende a requerir menos aceite o grasas adicionales, ya que la carne se cocina directamente sobre el fuego sin necesidad de sumergirla en aceite. Esto puede ayudar a reducir la ingesta de grasas, lo que puede ser beneficioso para aquellos que buscan controlar su consumo calórico o reducir la grasa saturada.
- **Menor formación de compuestos nocivos:** Si se utiliza correctamente, la cocina al carbón puede generar menos compuestos nocivos en comparación con algunos métodos de cocción, como freír a altas temperaturas. Sin embargo, es importante evitar que las llamas entren en contacto directo con la carne, ya que esto puede producir hidrocarburos aromáticos policíclicos (HAP) y aminas heterocíclicas (AHC), sustancias potencialmente dañinas para la salud.
- **Mayor retención de nutrientes:** En comparación con hervir, algunos estudios sugieren que la parrilla puede preservar mejor ciertos nutrientes en la carne, como las vitaminas B y C, al tiempo que reduce la pérdida de agua y minerales.



- **Inclusión de alimentos más saludables:** La cocina al carbón no se limita a carne; también puedes asar verduras, frutas y pescado en la parrilla. Incluir estos alimentos en tu dieta puede proporcionar una mayor variedad de nutrientes y antioxidantes.
- **Menor uso de salsas y condimentos ricos en calorías:** La carne asada a la parrilla puede tener un sabor delicioso por sí sola, lo que puede reducir la necesidad de agregar salsas o condimentos altos en calorías y azúcares.

Es importante destacar que, aunque hay algunas ventajas en términos de salud con la cocina al carbón, también existen posibles riesgos para la salud asociados con la exposición al humo y compuestos tóxicos que se generan cuando se cocinan ciertos alimentos a altas temperaturas. Para evitar riesgos hemos tomado en cuenta:

- No cocinar en exceso la carne y evitar que se queme, ya que esto puede aumentar la producción de sustancias nocivas.
- Marinar la carne antes de cocinarla puede ayudar a reducir la formación de compuestos dañinos.
- Cocinar en una parrilla bien ventilada al aire libre o usar una parrilla de gas para minimizar la exposición al humo.

En general, el prototipo realizado en esta investigación permite la mejora de la cocción uniforme de las carnes y otros alimentos. En la imagen siguiente vemos el prototipo en funcionamiento.

Figura 23

Resultados de cocción de alimentos



Elaboración propia

4.2. RESULTADOS DEL COMPORTAMIENTO DEL PROTOTIPO

Para ver el resultado del comportamiento del prototipo desarrollado se necesitó tener los materiales tales como: la parrilla, la carne, y lo más importante el carbón. Antes de las 4 de la tarde de un día cotidiano, se prende el carbón de forma tradicional, usando unas velas para que inicie el encendido del carbón.

Figura 24

Carbón



Elaboración propia

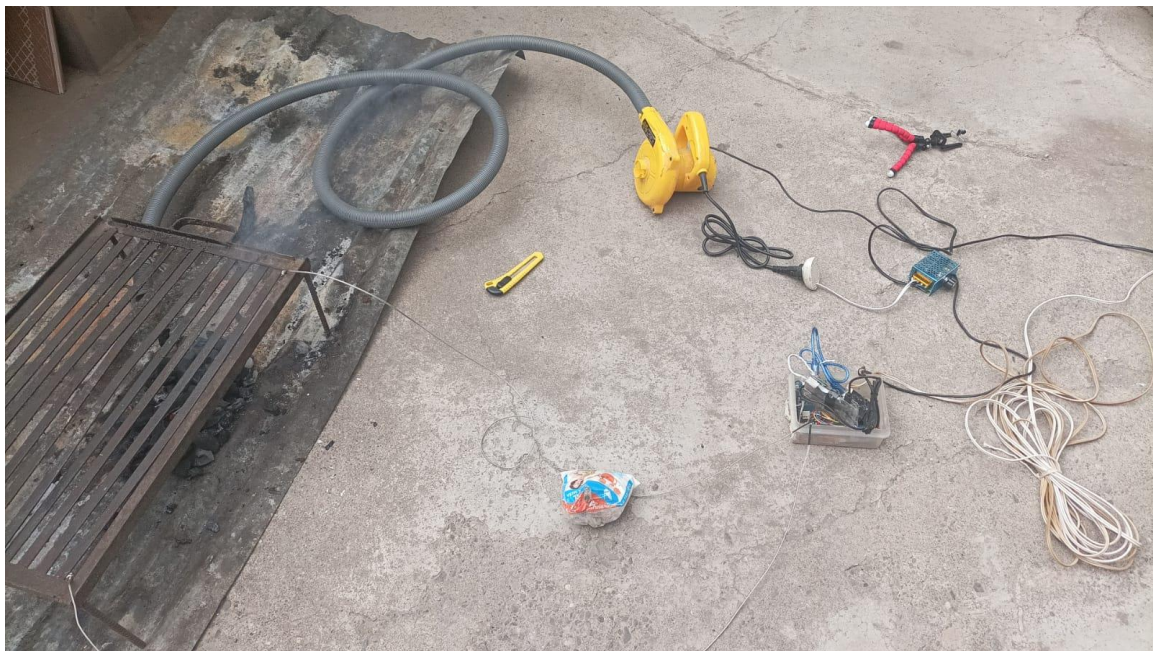
Al tener encendido el carbón, se procede a prender el sistema, este está comprendido por:

- Raspberry pi
- Arduino uno
- Shield Ethernet
- Dimmer
- Soplador

La alimentación de Raspberry como del Arduino es con 5V.

Figura 25

Sistema completo



Elaboración propia

Ponemos las termocuplas cerca de la carne, tal como ve en la siguiente imagen. Estos sensores no se ponen directamente al carbón, sino lo más próximo a ella, y si hubiese el caso de ponerlo tendríamos que poner el Setpoint a 100 grados aproximadamente. En la siguiente imagen se puede visualizar la posición de las termocuplas.

Figura 26

Parrilla lista para el funcionamiento



Elaboración propia

A continuación, colocamos el soplador con el tubo corrugado cerca del carbón. Esto se realiza para que el fuego incremente, con el objetivo de subir la temperatura.

Figura 27

Soplador próximo al carbón



Elaboración propia

Al tener lista la temperatura ponemos la carne, tal como se ve en la figura.

Figura 28

Inicia la cocción de los alimentos



Elaboración propia

Finalmente, la cocción de la carne se hace efectiva con el sistema de control, quedando el resultado como la siguiente imagen.

Figura 29

Proceso de cocción de alimentos



Elaboración propia

Al tener todo listo, podemos acceder a la página web de nuestro proyecto poniendo la IP del servidor. Veamos los resultados en nuestro celular.

Figura 30

Datos en la WEB

31.50	1	179.7.224.186	DIAB-01	2023-08-12 16:22:57
31.12	1	179.7.224.186	DIAB-01	2023-08-12 16:22:28
31.75	1	179.7.224.186	DIAB-01	2023-08-12 16:21:58
32.25	1	179.7.224.186	DIAB-01	2023-08-12 16:21:28
32.88	1	179.7.224.186	DIAB-01	2023-08-12 16:20:58
35.00	1	179.7.224.186	DIAB-01	2023-08-12 16:20:28
34.63	1	179.7.224.186	DIAB-01	2023-08-12 16:18:59
33.13	1	179.7.224.186	DIAB-01	2023-08-12 16:17:59
36.38	1	179.7.224.186	DIAB-01	2023-08-12 16:14:56
35.88	1	179.7.224.186	DIAB-01	2023-08-12 16:13:26

Elaboración propia

Vemos claramente como la temperatura va incrementándose a medida que pasa el tiempo. En el programa de Arduino claramente se observa que, si la temperatura alcanza los

50 grados, el soplador debe apagarse, para ello verificamos nuevamente la página y veamos que sucede.

Figura 31

Verificación del encendido y apagado del soplador

Conexión Satisfactoria!

Temperatura Promedio	FAN	IP Cliente	ID Sensor	Fecha Hora
97.62	0	179.7.224.186	DIAB-01	2023-08-12 16:39:13
99.62	0	179.7.224.186	DIAB-01	2023-08-12 16:38:44
59.25	0	179.7.224.186	DIAB-01	2023-08-12 16:37:38
49.50	1	179.7.224.186	DIAB-01	2023-08-12 16:36:38
74.00	0	179.7.224.186	DIAB-01	2023-08-12 16:35:38
70.25	0	179.7.224.186	DIAB-01	2023-08-12 16:35:08
59.25	0	179.7.224.186	DIAB-01	2023-08-12 16:34:38
49.50	1	179.7.224.186	DIAB-01	2023-08-12 16:34:08
34.00	1	179.7.224.186	DIAB-01	2023-08-12 16:33:39
27.75	1	179.7.224.186	DIAB-01	2023-08-12 16:33:09
25.87	1	179.7.224.186	DIAB-01	2023-08-12 16:32:39
29.00	1	179.7.224.186	DIAB-01	2023-08-12 16:32:09
30.75	1	179.7.224.186	DIAB-01	2023-08-12 16:31:39
29.87	1	179.7.224.186	DIAB-01	2023-08-12 16:31:10

Elaboración propia

Vemos claramente que, si la temperatura pasa de los 50 grados, el soplador se apaga con el fin de mantener la temperatura en el rango establecido.

4.3. RESULTADOS DE ENVÍO DE DATOS DE A LA PÁGINA WEB

El envío de datos a Internet se hizo con la raspberry pi, esta pequeña computadora es capaz de crear una red LAN a través de un puerto usb y un módulo, de esta manera tenemos que configurar NAT que significa "Network Address Translation" (traducción de dirección de red), es una técnica utilizada en telecomunicaciones y redes de computadoras para modificar las direcciones IP de los dispositivos en una red. Su objetivo principal es permitir que múltiples dispositivos en una red privada compartan una sola dirección IP pública para comunicarse con otros dispositivos en Internet.



Cuando los datos se transmiten a través de Internet, cada dispositivo conectado a la red debe tener una dirección IP única para ser identificado de manera única en la red global. Sin embargo, las direcciones IP públicas (IPv4) son limitadas en número y, a medida que aumenta la cantidad de dispositivos conectados a Internet, se vuelve necesario utilizar técnicas como NAT para superar esta limitación.

En la página web podemos observar parámetros como:

- Hora y fecha de registro de los datos
- La Temperatura
- La dirección IP
- El encendido o apagado del soplador

Los datos mencionados permiten al chef poder verificar el tiempo de cocción de los alimentos y lo más importante, verificar la temperatura de la parrilla.

En la siguiente imagen se ve la página web completa.

Figura 32

Página web en el servidor

Conexión Satisfactoria!

Temperatura Promedio	FAN	IP Cliente	ID Sensor	Fecha Hora
97.62	0	179.7.224.186	DIAB-01	2023-08-12 16:39:13
99.62	0	179.7.224.186	DIAB-01	2023-08-12 16:38:44
59.25	0	179.7.224.186	DIAB-01	2023-08-12 16:37:38
49.50	1	179.7.224.186	DIAB-01	2023-08-12 16:36:38
74.00	0	179.7.224.186	DIAB-01	2023-08-12 16:35:38
70.25	0	179.7.224.186	DIAB-01	2023-08-12 16:35:08
59.25	0	179.7.224.186	DIAB-01	2023-08-12 16:34:38
49.50	1	179.7.224.186	DIAB-01	2023-08-12 16:34:08
34.00	1	179.7.224.186	DIAB-01	2023-08-12 16:33:39
27.75	1	179.7.224.186	DIAB-01	2023-08-12 16:33:09
25.87	1	179.7.224.186	DIAB-01	2023-08-12 16:32:39
29.00	1	179.7.224.186	DIAB-01	2023-08-12 16:32:09
30.75	1	179.7.224.186	DIAB-01	2023-08-12 16:31:39
29.87	1	179.7.224.186	DIAB-01	2023-08-12 16:31:10
34.63	1	179.7.224.186	DIAB-01	2023-08-12 16:29:53
70.25	0	179.7.224.186	DIAB-01	2023-08-12 16:29:23
59.25	0	179.7.224.186	DIAB-01	2023-08-12 16:28:53
99.62	0	179.7.224.186	DIAB-01	2023-08-12 16:28:24
74.00	0	179.7.224.186	DIAB-01	2023-08-12 16:27:24
200.13	0	179.7.224.186	DIAB-01	2023-08-12 16:25:55
190.13	0	179.7.224.186	DIAB-01	2023-08-12 16:25:25
71.37	0	179.7.224.186	DIAB-01	2023-08-12 16:23:27
31.50	1	179.7.224.186	DIAB-01	2023-08-12 16:22:57
31.12	1	179.7.224.186	DIAB-01	2023-08-12 16:22:28
31.75	1	179.7.224.186	DIAB-01	2023-08-12 16:21:58
32.25	1	179.7.224.186	DIAB-01	2023-08-12 16:21:28
32.88	1	179.7.224.186	DIAB-01	2023-08-12 16:20:58
35.00	1	179.7.224.186	DIAB-01	2023-08-12 16:20:28
34.63	1	179.7.224.186	DIAB-01	2023-08-12 16:18:59

Elaboración propia

4.4. COMPARACIÓN ENTRE LA COCCIÓN NORMAL Y LA COCCIÓN CON EL PROTOTIPO

En una parrilla realizada, se hizo comparaciones y se obtuvo las diferencias entre una cocción estándar y una cocción con el prototipo y se obtuvo las siguientes diferencias.

Cocción Tradicional:

- La cocción tradicional implica un mayor tiempo de preparación.
- Es necesario el esfuerzo humano para ventilar manualmente el carbón.

Figura 33

Cocción estándar



Elaboración propia

Cocción con el Prototipo Desarrollado:

- En comparación con el método tradicional, la cocción con el prototipo desarrollado se lleva a cabo en un tiempo significativamente menor.
- La ventaja radica en que el sistema elimina la necesidad de intervención humana, haciendo que el proceso sea prácticamente libre de esfuerzo.

Figura 34

Cocción con el prototipo



Elaboración propia



V. CONCLUSIONES

PRIMERO: El diseño de este sistema de control on/off con IoT para la supervisión de la temperatura en la cocción de alimentos al carbón representa una integración efectiva de tecnologías modernas para mejorar la eficiencia, la comodidad y la calidad en el proceso de preparación de alimentos a la parrilla. El desarrollo integral del proyecto ha demostrado ser eficiente, ya que permite al usuario tener una mejor cocción en comparación a la cocina sin el prototipo.

SEGUNDO: Se ha desarrollado un sistema de Internet de las Cosas para la monitorización de la temperatura en las parrillas, utilizando Arduino para la lectura de datos de las termocuplas. La transmisión de los datos al internet se logra mediante una shield Ethernet y una Raspberry Pi actuando como enrutador. Una vez que los datos están en línea, la verificación se realiza fácilmente accediendo a la página web desde un celular o una PC al introducir la IP en el navegador. En las pruebas realizadas, el sistema ha demostrado ofrecer resultados favorables, brindando a las personas la capacidad de monitorear constantemente la temperatura de la parrilla, un parámetro crucial para lograr una cocción óptima de los alimentos.

TERCERO: El diseño del sistema de control ON/OFF para la temperatura durante el proceso de cocción de alimentos al carbón demuestra una solución efectiva y práctica. La implementación de un programa en Arduino IDE, que activa un soplador cuando la temperatura de la parrilla cae por debajo del setpoint y lo apaga cuando supera dicho umbral, proporciona un control preciso y automático. Este enfoque



garantiza que la temperatura se mantenga dentro del rango establecido,
facilitando una cocción uniforme.



VI. RECOMENDACIONES

Para futuras investigaciones, se recomienda realizar una revisión más detallada y específica de la disposición de los sensores. Es recomendable considerar aspectos como la ubicación precisa en relación con la superficie de cocción y la distribución del calor, con el objetivo de lograr una aproximación más efectiva al seguimiento de la temperatura de los alimentos. Además, se podría explorar la posibilidad de utilizar múltiples sensores estratégicamente ubicados para obtener mediciones más precisas y representativas en diversas áreas de la parrilla. De esta manera la disposición de sensores contribuirá significativamente a mejorar la calidad de las lecturas y optimizar el control del proceso de cocción.

El control ON/OFF puede presentar problemas de histéresis, lo que significa que la respuesta para encender y apagar puede no ser instantánea al alcanzar el setpoint. Esto puede resultar en fluctuaciones de temperatura alrededor del valor deseado. Se recomienda explorar alternativas más avanzadas y precisas, como el control PID (Proporcional, Integral y Derivativo). El control PID es un enfoque más sofisticado que permite gestionar de manera más efectiva la variabilidad y los cambios en la temperatura. Este método combina tres componentes fundamentales: la proporción (P) para ajustar la respuesta a las variaciones inmediatas, la integral (I) para manejar desviaciones a largo plazo, y el derivativo (D) para anticipar y corregir tendencias futuras. Al adoptar un sistema PID, se logra un control más fino y adaptable, mejorando significativamente la capacidad del sistema para mantener una temperatura estable y optimizar así el proceso de cocción de alimentos al carbón.



VII. REFERENCIAS BIBLIOGRÁFICAS

- Auza, A. F., Bello, D. A., & Gomez Rodriguez, G. (2020). *Parrilla Verde*.
- Blum, J. (2019). *Exploring Arduino: Tools and Techniques for Engineering*. Wiley.
- Cahuana Aquino, M. F. (2022). *Sistema domotico con la tecnologñia arduino a través de un aplicativo movil para el control de temperatura del puesto de salud Tingua – 2020*. Chimbote: Universidad Católica los Ángeles.
- Carthern, C., Wilson, W., & Rivera, N. (2021). *Cisco Networks: Engineers' Handbook of Routing, Switching, and Security with IOS, NX-OS, and ASA*. Apress.
- Casares, E. (2022). *Cocina Para la Parrilla de Gas al Aire Libre*. Emigdia Casares.
- Code Quickly . (2020). *Learn C++ Quickly: A Complete Beginner's Guide to Learning C++, Even If You're New to Programming*. Drip Digital.
- Correa Cárdenas, L. E. (2022). *Comportamientos críticos y de histéresis de un ferromagneto de momentos magnéticos semienteros*.
- Debele, G. M., & Xiao, Q. (2020). Sistema de regulación automática de temperatura ambiente utilizando arduino y sensor dht11. *IEEE*, 5.
- Dodd, A. (2019). *Essential Guide to Telecommunications*. Pearson.
- Escobar Gallardo, E., & Villazón, A. (2018). Sistema de monitoreo energético y control domótico basado en tecnología "internet de las cosas". *Investigación y Desarrollo - Universidad Privada Boliviana*, 14.
- Ganazhapa, B. O. (2021). *Arduino e Internet de las Cosas*. RC Libros.



- Greengard, S. (2021). *The Internet of Things*. The MIT Press.
- Hallberg, G. (2020). *Working with Displays*. North Border Tech Training.
- Hernández Velásquez, V. M., & Alvarado Bawab, M. B. (2017). Control on-off de temperatura y potencia para el mejoramiento de las condiciones de procesos asistidos con microondas. *INGE CUC*, 53-59.
- Hernández-Sampieri, R., & Mendoza, C. (2020). *Metodología de la investigación: las rutas cuantitativa, cualitativa y mixta*. McGraw-hill.
- Huisacaina Soto, H. L., Pesantes Rojas, J. M., & Herrera Ortiz, M. (2021). Control Electrónico on/off para el protocolo de temperatura usado en la criopreservación de embriones de ovino. *South Florida*, 14.
- Jimenes Guaman, P. F., & Cabrera Mejia, J. B. (2020). Sistema de monitoreo remoto del consumo energético para hogares en la ciudad de Cuenca, basado en principios de IoT y servicios en la nube. *Polo del Conocimiento*, 443-458.
- Khan, J. Y., & Yuce, M. R. (2019). *Internet of Things (IoT): Systems and Applications*. Jenny Stanford Publishing.
- King, A. (2021). *Programming the Internet of Things*. O'Reilly Media.
- LaCroix, J. (2022). *Mastering Ubuntu Server: Explore the versatile, powerful Linux Server distribution Ubuntu 22.04*. Packt Publishing.
- Malm, A. (2020). *Capital Fósil*. Capitán Swing Libros.
- Natheem, A. (2021). *Arduino Book for Beginners*. Independently published.



- Odom, W. (2019). *CCNA 200-301 Official Cert Guide Library*. Cisco Press.
- Products, M. I. (2021). *Termopar K compensado por unión fría*.
- Purdum, J., & Peter, A. (2020). *Microcontroller Projects*. Amer Radio Relay League.
- Raghunandan, K. (2022). *Introduction to Wireless Communications and Networks*.
Springer .
- Ramos Galarza, C. (2020). Los alcances de una investigación. *CienciAmérica*, 5.
- Sambamurthy, M. (2023). *Test Automation Engineering*. Packt Publishing.
- Sanchez Carlos, L. A., & Cruz Irigoien, D. A. (2018). *Diseño e implementación de un módulo didáctico de control de temperatura para la escuela profesional de ingeniería electrónica de la unprg*. Lambayeque: Universidad Nacional Pedro Ruiz Gallo.
- Sands, N., & Ian, V. (2018). *A Guide to the Automation Body of Knowledge*. International Society of Automation.
- Stuart, N. (2020). *Arduino Programming: Advanced Methods and Strategies to Learn Arduino Programming*. Independently published.
- Szuhay, J. (2020). *Learn C Programming*. Packt Publishing.
- Team Codex. (2022). *C Programing*. Independently published.
- Tsiatsis, V., Karnouskos, S., Holler, J., Boyle, D., & Mulligan, C. (2018). *Internet of Things: Technologies and Applications for a New Age of Intelligence* . Academic Press.



Ward, H. H. (2020). *Intermediate C Programming for the PIC Microcontroller: Simplifying Embedded Programming*. Apress.



ANEXOS

Anexo 1: Código Fuente de Arduino UNO

En este anexo se expone el código fuente en lenguaje Arduino para las funciones de control para el sistema de cocción y envío de datos desde el prototipo a un servidor en la nube.

```
#include <SPI.h>
#include <Ethernet.h>
#include "max6675.h"
int ktc1SO = 3;
int ktc1CS = 4;
int ktc1CLK = 5;
int ktc2SO = 6;
int ktc2CS = 7;
int ktc2CLK = 8;
float temp_prom;
int fan;
float temp1=0;
float temp2=0;
MAX6675 ktc1(ktc1CLK, ktc1CS, ktc1SO);
MAX6675 ktc2(ktc2CLK, ktc2CS, ktc2SO);
byte mac[] = { 0x54, 0xAB, 0x3A, 0x5B, 0xAD, 0x05 };
IPAddress server(108,62,123,70);
IPAddress ip(172,16,100,133);
byte gateway[] = { 172, 16, 100, 1 };
byte subnet[] = { 255, 255, 255, 0 };
byte myDns[] = { 8, 8, 8, 8 };
EthernetClient client;
bool printWebData = true;
char sensor_id[] = "DIAB-01";
void setup(){
  Ethernet.begin(mac, ip, myDns, gateway, subnet);
  delay(500);
  pinMode(2,OUTPUT);
}
void loop(){
  delay(1000);
  temp1=ktc1.readCelsius();
  temp2=ktc2.readCelsius();
  temp_prom=(temp1 + temp2)/2;
  if (temp_prom<=50){
    if (temp_prom<=48 ){
```



```
        digitalWrite(2,HIGH);
        fan=1; }
    else{
    }
}
else {
    digitalWrite(2,LOW);
    fan=0;
}
if (client.connect(server, 80)) {
    client.print("GET /TEMPFAN/formacion_tempfan.php?temp_prom=");
    client.print(temp_prom);
    client.print("&fan=");
    client.print(fan);
    client.print("&sensor_id=");
    client.print(sensor_id);
    client.println(" HTTP/1.1");
    client.print("Host: ");
    client.println(server);
    client.println("Connection: close");
    client.println();
} else {
}
while (client.connected()) {
    int len = client.available();
    if (len > 0) {
        byte buffer[80];
        if (len > 80) len = 80;
        client.read(buffer, len);
        if (printWebData) {
        }
    }
}
if (!client.connected()) {
    client.stop();
    delay(28358);
}
}
```



Anexo 2: Declaración jurada de autenticidad de tesis



Universidad Nacional
del Altiplano Puno



Vicerrectorado
de Investigación



Repositorio
Institucional

DECLARACIÓN JURADA DE AUTENTICIDAD DE TESIS

Por el presente documento, Yo RICHARD DALY GARNICA CURASI, identificado con DNI 76157383 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

“DISEÑO DE UN SISTEMA DE CONTROL ON/OFF CON INTERNET DE LAS COSAS PARA LA SUPERVISIÓN DE TEMPERATURA DEL PROCESO DE COCCIÓN DE LOS ALIMENTOS AL CARBÓN”

Es un tema original.


Declaro que el presente trabajo de tesis es elaborado por mi persona y **no existe plagio/copia** de ninguna naturaleza, en especial de otro documento de investigación (tesis, revista, texto, congreso, o similar) presentado por persona natural o jurídica alguna ante instituciones académicas, profesionales, de investigación o similares, en el país o en el extranjero.

Dejo constancia que las citas de otros autores han sido debidamente identificadas en el trabajo de investigación, por lo que no asumiré como tuyas las opiniones vertidas por terceros, ya sea de fuentes encontradas en medios escritos, digitales o Internet.

Asimismo, ratifico que soy plenamente consciente de todo el contenido de la tesis y asumo la responsabilidad de cualquier error u omisión en el documento, así como de las connotaciones éticas y legales involucradas.

En caso de incumplimiento de esta declaración, me someto a las disposiciones legales vigentes y a las sanciones correspondientes de igual forma me someto a las sanciones establecidas en las Directivas y otras normas internas, así como las que me alcancen del Código Civil y Normas Legales conexas por el incumplimiento del presente compromiso

Puno 14 de mayo del 2024


FIRMA (obligatoria)



Huella



Anexo 3: Autorización para el depósito de tesis o trabajo de investigación en el repositorio institucional



Universidad Nacional
del Altiplano Puno



Vicerrectorado
de Investigación



Repositorio
Institucional

AUTORIZACIÓN PARA EL DEPÓSITO DE TESIS O TRABAJO DE INVESTIGACIÓN EN EL REPOSITORIO INSTITUCIONAL

Por el presente documento, Yo RICHARD DALY GARNICA CURASI, identificado con DNI 76157383 en mi condición de egresado de:

Escuela Profesional, Programa de Segunda Especialidad, Programa de Maestría o Doctorado

ESCUELA PROFESIONAL DE INGENIERIA ELECTRONICA, informo que he elaborado el/la Tesis o Trabajo de Investigación denominada:

"DISEÑO DE UN SISTEMA DE CONTROL ON/OFF CON INTERNET DE LAS COSAS PARA LA SUPERVISIÓN DE TEMPERATURA DEL PROCESO DE COCCIÓN DE LÓS ALIMENTOS AL CARBÓN"

para la obtención de Grado, Título Profesional o Segunda Especialidad.

Por medio del presente documento, afirmo y garantizo ser el legítimo, único y exclusivo titular de todos los derechos de propiedad intelectual sobre los documentos arriba mencionados, las obras, los contenidos, los productos y/o las creaciones en general (en adelante, los "Contenidos") que serán incluidos en el repositorio institucional de la Universidad Nacional del Altiplano de Puno.

También, doy seguridad de que los contenidos entregados se encuentran libres de toda contraseña, restricción o medida tecnológica de protección, con la finalidad de permitir que se puedan leer, descargar, reproducir, distribuir, imprimir, buscar y enlazar los textos completos, sin limitación alguna.

Autorizo a la Universidad Nacional del Altiplano de Puno a publicar los Contenidos en el Repositorio Institucional y, en consecuencia, en el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto, sobre la base de lo establecido en la Ley N° 30035, sus normas reglamentarias, modificatorias, sustitutorias y conexas, y de acuerdo con las políticas de acceso abierto que la Universidad aplique en relación con sus Repositorios Institucionales. Autorizo expresamente toda consulta y uso de los Contenidos, por parte de cualquier persona, por el tiempo de duración de los derechos patrimoniales de autor y derechos conexos, a título gratuito y a nivel mundial.

En consecuencia, la Universidad tendrá la posibilidad de divulgar y difundir los Contenidos, de manera total o parcial, sin limitación alguna y sin derecho a pago de contraprestación, remuneración ni regalía alguna a favor mío; en los medios, canales y plataformas que la Universidad y/o el Estado de la República del Perú determinen, a nivel mundial, sin restricción geográfica alguna y de manera indefinida, pudiendo crear y/o extraer los metadatos sobre los Contenidos, e incluir los Contenidos en los índices y buscadores que estimen necesarios para promover su difusión.

Autorizo que los Contenidos sean puestos a disposición del público a través de la siguiente licencia:

Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional. Para ver una copia de esta licencia, visita: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

En señal de conformidad, suscribo el presente documento.

Puno 14 de mayo del 2024

FIRMA (obligatoria)



Huella