



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA INGENIERÍA EN TELECOMUNICACIONES

**“EVALUACIÓN DE LA TRANSMISIÓN DE DATOS CON
TECNOLOGÍA IOT DE UN PROTOTIPO DE SISTEMA DE
SEGURIDAD EN COCINAS A GAS PARA LA PREVENCIÓN DE
INCENDIOS DOMÉSTICOS.”**

Trabajo de Integración Curricular

Tipo: PROYECTO TÉCNICO

Presentado para optar al grado académico de:

**INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES**

AUTOR: JONATHAN JACOBO COLLANTES ALCIVAR

DIRECTOR: Ing. MARCO VINICIO RAMOS VALENCIA M.Sc.

Riobamba – Ecuador

2021

© 2021, Jonathan Jacobo Collantes Alcivar




Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA INGENIERÍA EN TELECOMUNICACIONES

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación; tipo: Proyecto Técnico, **“EVALUACIÓN DE LA TRANSMISIÓN DE DATOS CON TECNOLOGÍA IOT DE UN PROTOTIPO DE SISTEMA DE SEGURIDAD EN COCINAS A GAS PARA LA PREVENCIÓN DE INCENDIOS DOMÉSTICOS.”**, realizado por el señor: **Jonathan Jacobo Collantes Alcivar**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Jorge Yuquilema PRESIDENTE DEL TRIBUNAL		23-05-04
Ing. Marco Ramos MsC. DIRECTOR DE TRABAJO DE TITULACIÓN		23-05-04
Ing. Jefferson Ribadeneira PhD. MIEMBRO DEL TRIBUNAL		23-05-04

Yo, Jonathan Jacobo Collantes Alcivar, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 04 de mayo de 2023



JONATHAN JACOBO COLLANTES ALCIVAR
180494350-2

DEDICATORIA

La calidad humana y el valor de un hombre se mide por su constancia, por su perseverancia en la obra que ha emprendido. El éxito es el resultado inmediato del esfuerzo, del valor, del optimismo, del vigor, de la tenacidad en la conquista de este objetivo académico. El crisol en donde se ha fraguado la consecución de esta travesía radica en mi madre, mi padre, mis hermanos y amigos, a quienes dedico este logro alcanzado. Cada uno de ellos ha intervenido en mí con apoyo y consejos dando como resultado valiosas lecciones de vida, lecciones en el campo profesional y laboral. A la prestigiosa Escuela Superior Politécnica de Chimborazo que fue la morada de diversos días y noches en los que brindó sus instalaciones, sus profesores y sus espacios para consolidar la moral y la ética que todo profesional debe llevar junto con la preparación académica que dará pie a nuevos emprendimientos, valiosos aportes a la sociedad y el mundo.

Jonathan

AGRADECIMIENTO

Mi agradecimiento a la Escuela Superior Politécnica de Chimborazo por brindar cúmulo de conocimientos a lo largo de mi paso por la institución. Un efusivo e infinito agradecimiento mi familia y en especial a mis padres que, con sacrificado esfuerzo y apoyo, permitieron que mi formación académica lograra llegar a conseguir un título universitario. A mis tutores, Ing. Vinicio Ramos y al Ing. Jefferson Ribadeneira, quienes me dirigieron con sus conocimientos durante mi recorrido formativo; mi más profundo agradecimiento por ser un gran sustento durante el transcurso del desarrollo de este proyecto.

Jonathan

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE ILUSTRACIONES.....	xi
ÍNDICE DE ABREVIATURAS.....	xiii
ÍNDICE DE ANEXOS.....	xv
RESUMEN.....	xvi
SUMMARY.....	xvii
CAPÍTULO I.....	1
1. DIAGNÓSTICO DEL PROBLEMA	1
1.1 ANTECEDENTES	1
1.2 FORMULACIÓN DEL PROBLEMA	2
1.3 JUSTIFICACIÓN DEL TRABAJO DE TITULACIÓN	3
1.4 OBJETIVOS	5
CAPÍTULO II.....	7
2. MARCO TEÓRICO	7
2.1 Introducción	7
2.2 Internet De Las Cosas	7
2.3 Redes M2M	8
2.4 LP-WAN	8
2.5 Tecnologías de transmisión de datos	9
2.5.1 <i>Bluetooth</i>	9
2.5.2 <i>Wifi</i>	9
2.5.3 <i>ZigBee</i>	9
2.5.4 <i>LoRa</i>	10
2.5.5 <i>Z-Wave</i>	10
2.6.1 <i>Tráfico Elástico</i>	10
2.6.2 <i>Tráfico Inelástico</i>	11
2.6.3 <i>Tráfico en Tiempo Real</i>	11
2.7 Protocolos IoT	11

2.7.1	<i>MQTT (Message Queuing Telemetry Transport Protocol)</i>	11
2.7.2	<i>CoAP (Constrained Application Protocol)</i>	12
2.7.3	<i>AMQP (Advanced Message Queuing Protocol)</i>	12
2.7.4	<i>HTTP (Protocolo de transporte de hipertexto)</i>	12
2.8	Modelos de comunicación de IoT	13
2.8.1	<i>Modelo de comunicación Dispositivo a Dispositivo</i>	13
2.8.2	<i>Modelo de comunicación Dispositivo a Nube</i>	13
2.8.3	<i>Modelo de comunicación Dispositivo a Gateway</i>	14
2.8.4	<i>Modelo de intercambio de datos mediante Back-End</i>	15
2.9	Tarjetas de Desarrollo	16
2.9.1	<i>Arduino</i>	16
2.9.2	<i>Raspberry PI</i>	16
2.9.3	<i>Node MCU</i>	16
2.9.4	<i>Galileo</i>	16
2.10	Incendios	17
2.10.1	<i>Clasificación de Incendios</i>	17
2.10.2	<i>Causas y breves cifras de incendios en los hogares ecuatorianos</i>	18
2.10.3	<i>Fases de un Incendio</i>	18
2.11	Sistemas Contra Incendios Estructurales	19
2.11.1	<i>Tipos de Seguridades</i>	19
2.11.1.1	<i>Seguridad Pasiva</i>	19
2.11.1.2	<i>Seguridad Activa</i>	19
2.11.2	<i>Métodos de Extinción frente un Incendio</i>	20
2.11.2.1	<i>Enfriamiento</i>	20
2.11.2.2	<i>Sofocación</i>	20
2.11.2.3	<i>Eliminación del Combustible</i>	20
2.11.2.4	<i>Inhibición</i>	20
2.11.3	<i>Dispositivos Iniciadores</i>	20
2.11.3.1	<i>Detector de Gases de Combustión o Iónicos</i>	21
2.11.3.2	<i>Detector Óptico de Humos</i>	21
2.11.3.3	<i>Detector Temperatura</i>	21
2.11.3.4	<i>Detector de Llamas</i>	21
CAPÍTULO III		22

3.	MARCO METODOLÓGICO	22
3.1	Indicadores del sistema de seguridad	22
3.2	Requerimientos del Sistema	23
<i>3.2.1</i>	<i>Requerimientos del usuario</i>	23
<i>3.2.2</i>	<i>Requerimientos de software</i>	23
<i>3.2.3</i>	<i>Requerimiento de hardware</i>	23
3.3	Plataforma en la Nube	24
<i>3.3.1</i>	<i>Arduino IoT Cloud</i>	24
3.4	Diseño del sistema	25
<i>3.4.1</i>	<i>Situación actual de cocinas en entornos residenciales</i>	25
<i>3.4.1.1</i>	<i>Zona circundante a la cocina</i>	26
<i>3.4.1.2</i>	<i>Zona de la cocina</i>	26
<i>3.4.2</i>	<i>Selección de tecnología de comunicación</i>	26
<i>3.4.2.1</i>	<i>Análisis</i>	27
<i>3.4.3</i>	<i>Selección de tarjeta de desarrollo</i>	29
<i>3.4.3.1</i>	<i>Análisis</i>	30
<i>3.4.4</i>	<i>Selección de sensor para detección de fugas de gas</i>	31
<i>2.4.4.1</i>	<i>Análisis</i>	32
<i>3.4.5</i>	<i>Selección de sensor para detección de temperatura en el ambiente</i>	33
<i>3.4.5.1</i>	<i>Análisis</i>	33
<i>3.4.6</i>	<i>Esquema de Sistema de Cocina IoT</i>	35
<i>3.4.7</i>	<i>Esquema del circuito controlador de cocina</i>	36
<i>3.4.8</i>	<i>Análisis y recopilación de medidas de una cocina tradicional</i>	37
<i>3.4.9</i>	<i>Ensamblaje de sensores y actuadores a la estructura</i>	39
<i>3.4.10</i>	<i>Creación de cuenta y configuración del sistema en Arduino IoT Cloud</i>	43
<i>3.4.11</i>	<i>Servicio MQTT provistos por la plataforma Arduino IoT Cloud</i>	52
<i>3.4.11.1</i>	<i>Arquitectura</i>	52
<i>3.4.11.2</i>	<i>Características</i>	53
CAPÍTULO IV		54
4.	ANÁLISIS DE RESULTADOS	54
4.1	Evaluación del sistema de seguridad	54
<i>4.1.1</i>	<i>Conectividad</i>	<i>55</i>
<i>4.1.1.1</i>	<i>Conectividad entre la tarjeta de desarrollo y la red wifi</i>	<i>56</i>

4.1.1.2	<i>Conectividad entre la tarjeta de desarrollo y Arduino IoT Cloud</i>	57
4.1.1.3	<i>Envío de paquetes a la plataforma de Arduino IoT Cloud</i>	57
4.1.1.4	<i>Recepción de paquetes en la aplicación móvil IoT Remote</i>	58
4.1.2	<i>Estabilidad</i>	59
4.1.2.1	<i>Estabilidad en condiciones normales</i>	59
4.1.2.2	<i>Estabilidad bajo condiciones de trabajo con períodos de abandono de la cocina</i>	60
4.1.2.3	<i>Estabilidad bajo condiciones de alta peligrosidad</i>	61
4.1.3	<i>Nivel de error</i>	62
4.1.4	<i>Recurrencia bajo condiciones de riesgo</i>	63
4.1.5	<i>Notificaciones en caso de uso emergencia</i>	64
4.1.6	<i>Análisis económico del sistema</i>	66
CONCLUSIONES		67
RECOMENDACIONES		68
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-3: Comparativa entre Bluetooth, Wifi, ZigBee, Lora, Z-Wave	26
Tabla 2-3: Niveles de eficiencia para la ponderación y evaluación de características.....	27
Tabla 3-3: Niveles de eficiencia para la ponderación y evaluación de características para las tecnologías de comunicación	28
Tabla 4-3: Comparativa entre Arduino, Raspberry Pi, Node MCU, Intel Galileo.....	29
Tabla 5-3: Niveles de eficiencia para la ponderación y evaluación de características para las tarjetas de desarrollo	30
Tabla 6-3: Comparativa entre MQ-2, MQ-5, MQ-6, MQ-9.	31
Tabla 7-3: Niveles de eficiencia para la ponderación y evaluación de características de los sensores para la detección de fugas de gas.....	32
Tabla 8-3: Comparativa entre KY-026, DHT11, DHT21, DHT22, DS18B20.	33
Tabla 9-3: Niveles de eficiencia para la ponderación y evaluación de características de los sensores para la detección del encendido de hornillas.	34
Tabla 1-4: Recolección de datos de variables virtuales en condiciones normales	59
Tabla 2-4: Recolección de datos de variables virtuales en condiciones de abandono	60
Tabla 3-4: Recolección de datos de variables virtuales en condiciones de alta peligrosidad ...	61
Tabla 4-4: Descripción de elementos del prototipo con su respectivo costo en dólares	62
Tabla 5-4: Descripción de elementos del prototipo con su respectivo costo en dólares	66

ÍNDICE DE ILUSTRACIONES

Figura 1-2:	Arquitectura de Red LP-WAN.....	8
Figura 2-2:	Modelo de Comunicación Dispositivo a Dispositivo	13
Figura 3-2:	Modelo de Comunicación Dispositivo a Nube	14
Figura 4-2:	Modelo de Comunicación Dispositivo a Gateway.....	15
Figura 5-2:	Modelo de Intercambio de Datos Mediante Back-End.....	15
Figura 1-3:	Esquema de Sistema de Cocina IoT.....	35
Figura 3-3:	Esquema de circuito controlador de cocina	36
Figura 4-3:	Parte frontal de una cocina tradicional.....	37
Figura 5-3:	Parte superior de una cocina tradicional	38
Figura 6-3:	Parte inferior interior de una cocina tradicional.....	38
Figura 7-3:	Parte fontal interior de una cocina tradicional	39
Figura 8-3:	Parte fontal con la incorporación de motores y engranajes.....	39
Figura 9-3:	Engrane tipo piñón para cada hornilla	40
Figura 10-3:	Engrane tipo piñón para cada motor	40
Figura 11-3:	Sensores y actuadores de la parte interna de la cocina.....	41
Figura 12-3:	Sensores de la parte superior de la cocina.....	41
Figura 13-3:	Etapas de control y transmisión, parte posterior de la cocina.....	42
Figura 14-3:	Pantalla inicial de acceso a la plataforma de Arduino IoT Cloud.....	43
Figura 15-3:	Sección de dispositivos en la plataforma Arduino IoT Cloud	43
Figura 16-3:	Selección de la categoría de dispositivo	44
Figura 17-3:	Selección de dispositivo a utilizar con Arduino IoT Cloud	44
Figura 18-3:	Asignación de nombre al dispositivo	45
Figura 19-3:	Asignación de credenciales de acceso	45
Figura 20-3:	Tarjeta de desarrollo asignada satisfactoriamente con Arduino IoT Cloud	46
Figura 21-3:	Panel para creación de variables virtuales	46
Figura 22-3:	Panel para configuración de variables virtuales, dispositivos y red.....	46
Figura 23-3:	Enlace de variables virtuales con tarjeta de desarrollo	47
Figura 24-3:	Enlace de variables virtuales con tarjeta de desarrollo	47
Figura 25-3:	Configuración individual de variables	48
Figura 26-3:	Configuración completa de la sección de variables	49
Figura 27-3:	Configuración panel de dashboards	49
Figura 28-3:	Asignación de elementos a panel de dashboards	49
Figura 29-3:	Configuración de Widgets	50
Figura 30-3:	Conjunto de Widgets destinados al prototipo de sistema de seguridad	50
Figura 31-3:	Editor de código y monitor serial.....	51

Figura 32-3: Broker MQTT	52
Figura 1-4: Monitor serial y prueba de conectividad.....	56
Figura 2-4: Escaneo de dispositivos en la red wifi	56
Figura 3-4: Verificación de estatus del dispositivo en la plataforma.....	57
Figura 4-4: Envío de paquetes visualizado en Serial Port Monitor	57
Figura 5-4: Recepción de paquetes visualizado en la plataforma Arduino IoT Cloud	58
Figura 6-4: Recepción de paquetes visualizado en la aplicación móvil IoT Remote	59
Figura 7-4: Recurrencia de datos en condiciones de riesgo – 30/07/2022 por la mañana	64
Figura 8-4: Recurrencia de datos en condiciones de riesgo – 30/07/2022 por la tarde	64
Figura 9-4: Notificación vía correo electrónico.....	65
Figura 10-4: Notificación vía mensaje de texto	65

ÍNDICE DE ABREVIATURAS

GLP	Gas Licuado del Petróleo
IoT	Internet of Things
WiFi	Wireless Fidelity
M2M	Machine to Machine
WSN	Redes de sensores inalámbricos
LPWAN	Low Power Wide Area Network
RF	Radio Frequency
GHz	Gigahertz
WPAN	Wireless Personal Area Network
WLAN	Wireless Local Area Network
Gbps	Gigabits por Segundo
IEEE	Institute of Electrical and Electronics Engineers
MHz	Megahertz
Kbps	Kilobits por Segundo
Bps	bits por segundo
Khz	Kilohertz
TCP	Transmission Control Protocol
UDP	User Data Protocol
IP	Internet Protocol
QoS	Quality of Service
MQTT	Message Queuing Telemetry Transport Protocol
CoAP	Constrained Application Protocol
DTLS	Datagram Transport Layer Security
AMQP	Advanced Message Queuing Protocol
SASL	Simple Authentication and Security Layer
SSL	Secure Sockets Layer
TLS	Transport Layer Security
HTTP	Protocolo de transporte de hipertexto
HTTPS	Hyper Text Transfer Protocol Secure
API	Interfaz de Programación de Aplicaciones

ICSP	In Circuit Serial Programming
SGR	Servicio Nacional de Gestión de Riesgos y Emergencias
OTA	Over the Air
M	Metros
Km	Kilómetros
dBm	Decibelio-mili vatio
FDMA	Frequency Division Multiple Access
TDMA	Time Division Multiple Access
TDD	Time Division Duplexing
FHSS	Frequency Hopping Spread Spectrum
DSSS	Direct Sequence Spread Spectrum
CSMA/CA	Carrier sense multiple access with collision avoidance
BPSK	Binary Phase Shift Keying
QPSK	Quadrature Phase-Shift Keying
QAM	Quadrature Amplitude Modulation
OQPSK	Offset Quadrature Phase Shift Keying
FSK	Frequency Shift Keying
GFSK	Gaussian Frequency Shift Keying
V	Voltios
mW	Milliwatts
KΩ	Kilohms
°C	Grados Centígrados
ppm	Partículas por millón

ÍNDICE DE ANEXOS

Anexo A: Características de la tarjeta de desarrollo NodeMCU ESP32S

Anexo B: Características del sensor DS18B20

Anexo C: Características del sensor MQ-2

Anexo D: Características del sensor HC-SR501

Anexo E: Características del servomotor MG996R

Anexo E: Módulo SIM800L

Anexo F: Código de programación

RESUMEN

Se implementó un prototipo de seguridad para cocinas a gas tradicionales con monitoreo remoto en tiempo real del estado de una cocina y su entorno, ello con miras a contribuir con medidas de prevención de incendios a través de sensores y actuadores ubicados estratégicamente. Un análisis comparativo determinó que la tecnología de comunicación que mejor se adapta al prototipo es wifi, las velocidades de datos en sus distintas versiones proponen un puente para el tráfico masivo de información, su cobertura, canalización y prestaciones de servicios, dio lugar a que el sistema pueda con el tiempo incluir mejoras para robustecer la prevención y por ende generar escalabilidad. Los dispositivos empleados fueron escogidos en función de los datos de los elementos con mayor utilidad en el mercado; de las ponderaciones de cada dispositivo, de los porcentajes que el método de Likert arrojó y en conformidad con las categorías establecidas en dicha escala se estableció que, los dispositivos con mayores prestaciones son: Node MCU ESP-32S, MQ-2, DS18B20, alcanzando una eficacia de 78.46%, 78.46% y 88.57% respectivamente. Se incorporó MG996R como actuadores y SIM800L junto con un zumbador como etapa de notificación y alerta. La plataforma Arduino IoT Cloud brindó herramientas convirtiendo al prototipo en un conjunto de instrumentos con la capacidad de configuración, programación y monitoreo en tiempo real orientados al propósito de esta investigación. Bajo condiciones controladas y de un conjunto de muestras, se logró alcanzar un nivel de error de $\pm 0,1375$ °C en el sensor que se tomó como referencia, alcanzando un umbral de precisión altamente aceptable. Partiendo de las necesidades establecidas, el sistema reunió un conjunto de medidas encaminadas a la prevención, contemplando un costo relativamente bajo para su implementación y operación, incorporando una forma de trabajo manual y automática, dispositivos eficaces y continuos, y una interfaz interactiva.

Palabras clave: <IOT>, <TELECOMUNICACIONES>, <SENSORES>, <ARDUINO>, <PREVENCIÓN DE INCENDIOS>, <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <COCINA>, <MONITOREO EN TIEMPO REAL>.



SUMMARY

A safety prototype for traditional gas stoves was implemented with real-time remote monitoring of the state of a kitchen and its surroundings, with the aim of contributing to fire prevention measures through strategically located sensors and actuators. A comparative analysis determined that the communication technology that best suits the prototype is Wi-Fi, the data speeds in its different versions propose a bridge for the massive traffic of information, its coverage, channeling and provision of services, led to the fact that the system may over time include improvements to strengthen prevention and therefore generate scalability. The devices used were chosen based on the data of the most useful elements in the market; of the weights of each device, of the percentages that the Likert method yielded and in accordance with the categories established in said scale, it was established that the devices with the highest performance are: Node MCU ESP-32S, MQ-2, DS18B20, reaching an efficacy of 78.46%, 78.46% and 88.57% respectively. MG996R was incorporated as actuators and SIM800L together with a buzzer as notification and alert stage. The Arduino IoT Cloud platform provided tools, turning the prototype into a set of instruments with the ability to configure, program, and monitor in real time oriented to the purpose of this research. Under controlled conditions and a set of samples, it was possible to reach an error level of ± 0.1375 °C in the sensor that was taken as reference, reaching a highly acceptable precision threshold. Based on the established needs, the system brought together a set of measures aimed at prevention, contemplating a relatively low cost for its implementation and operation, incorporating a manual and automatic way of working, effective and continuous devices, and an interactive interface.

Keywords: <IOT>, <TELECOMMUNICATIONS>, <SENSORS>, <ARDUINO>, <FIRE PREVENTION>, <TECHNOLOGY AND ENGINEERING SCIENCES>, <KITCHEN>, <REAL TIME MONITORING>.

A handwritten signature in blue ink, enclosed in a blue oval. The signature is stylized and appears to read 'Wilson G. Rojas'.

MsC. Wilson G. Rojas
C.I. 0602361842

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1 ANTECEDENTES

En la actualidad la tecnología se encuentra fuera del ámbito del escritorio tradicional, el internet de las cosas es una realidad revolucionaria que permite de forma fácil y amigable trascender aparatos eléctricos y electrónicos en “objetos inteligentes”. Gran parte de los objetos que se encuentran a nuestro alrededor están en la red de una forma u otra, apoyándose de elementos adicionales como las redes de sensores que forman parte de este nuevo desafío en el que todas las cosas que nos cubren estén conectadas transmitiendo y recibiendo información de manera imperceptible en el entorno que nos rodea, haciendo así que la vida sea más fácil, nuestras actividades más eficaces y generando seguridad. (Gubbi et al. 2013, pp. 1-2)

El uso indebido, no intencional e incluso deliberado del GLP puede generar una fuga en algún área y atentar contra la vida, es importante que en los sitios en los que se interactúe con GLP, exista una alerta temprana que permita notificar contra esta clase de catástrofes a los individuos. Por ello, la revista International Journal of Scientific & Engineering Research publicó en el año 2015 un sistema denominado “DEVELOPMENT OF WIRELESS SENSOR NETWORK SYSTEM FOR LPG GAS LEAKAGE DETECTION SYSTEM”, en el que se propuso la detección de fugas de gas basada en una red de sensores que detecta la concentración de gas en el aire, el sistema reúne los datos obtenidos conjuntamente con la localización de la fuga para ser monitoreados por medio de un módulo XBee. A partir de cualquier dispositivo celular se controla el sistema desarrollado mediante SMS. (Mujawar et al. 2015, pp. 1-2)

En la publicación “AUTOMATIC GAS COOKER CONTROL SYSTEM”, realizada por la misma revista mantiene firme el llamado a que la seguridad es primordial a la hora de escoger productos como cocinas de gas. Si bien es cierto, las cocinas de gas brindan eficacia, pero también conllevan peligros de manera directa relacionados con la utilización de GLP. El manejo erróneo podría acarrear graves resultados que involucran hasta la muerte. En el sistema propuesto en dicha publicación se consigue automatizar válvulas para proporcionar apertura y cierre en el flujo de gas solo en el caso de existir utensilios de cocina apoyados sobre algún quemador encendido. Cuando la cocción termina, el sistema procede el cierre de la válvula. (Alkali et al. 2017, p. 2)

En el trabajo de titulación elaborado por Edison Noboa “IMPLEMENTACIÓN DE UNA RED INALÁMBRICA CON NODOS INTELIGENTES PARA LA SUPERVISIÓN EN TIEMPO REAL Y PREVENCIÓN DE INCENDIOS EN LA PLANTA DE INDUFRICK.”, realizado en la Escuela Superior Politécnica de Chimborazo de la ciudad de Riobamba implementó un sistema en tiempo real para la supervisión de nodos en una red inalámbrica montados sobre la empresa Indufrick para la prevención de incendios. Inicialmente estructuró la red mediante un previo análisis entre qué dispositivos cumplen con la compatibilidad necesaria con ZigBee, cumpliendo así con requerimientos de seguridad, además de proveer al usuario de portabilidad y un control manual como automático. (Noboa Carrasco 2019, pp. 19-22)

1.2 FORMULACIÓN DEL PROBLEMA

¿Cuál tecnología de transmisión de datos en conjunto con IoT permitirá al prototipo de sistema de seguridad en cocinas a gas el envío adecuado de información para la prevención de incendios domésticos?

SISTEMATIZACIÓN DEL PROBLEMA

Para dar solución al problema planteado, es necesario ir respondiendo durante el desarrollo del trabajo las siguientes interrogantes:

- ¿Cuáles de las diferentes tecnologías de transmisión de datos permitirá el envío óptimo de información del prototipo de seguridad hacia el usuario final a través de IoT?
- ¿Qué requerimientos serán necesarios para que el prototipo de sistema de seguridad de cocción de alimentos se establezca como una prevención de incendios?
- ¿Cuáles van a ser los elementos hardware y software que se requieren en el diseño del sistema de seguridad en cocinas a gas?
- ¿Qué tipo de aplicación va a permitir realizar la toma de decisiones sobre el prototipo?

1.3 JUSTIFICACIÓN DEL TRABAJO DE TITULACIÓN

JUSTIFICACIÓN TEÓRICA

En términos del internet de las cosas, cada uno de los objetos pertenecientes al mundo físico que tienen la capacidad de ser integrados e identificados al interior de las redes de comunicación, tendrán eventualmente información conexas proveniente de su entorno que se podrá monitorear, gestionar y hasta manipular. Es imprescindible considerar la compatibilidad entre cada uno de los elementos que conformen las interfaces dado que el Gateway permite la conexión de dispositivos mediante múltiples tecnologías, ya sean estas cableadas o inalámbricas.

Existen consideraciones que deben tomarse en cuenta cuando se establecen las capacidades que pueda suministrar el Gateway, en primero lugar a causa de los diferentes protocolos que se puede tener a nivel de la capa de dispositivo para su comunicación como Bluetooth, ZigBee, Wifi, entre otras. En segundo lugar, cuando las comunicaciones involucran no solo la capa de dispositivo, sino también la capa de red en su comunicación, por ejemplo, el protocolo de la tecnología Bluetooth en la capa del dispositivo y un protocolo de la tecnología 4G en la capa de red.

En el hogar la preocupación que persiste al momento de integrar nuevos equipos es la seguridad, en especial de los que están proveídos de elementos de potencial riesgo como las cocinas que emplean GLP. Por esta razón, el temor de los usuarios demanda de características en sus equipos que puedan garantizar una seguridad avanzada, esto con miras a salvaguardar la vida y dicho sea de paso también a proteger su propiedad.

En el año 2015 se levantaron cifras estadísticas llevadas a cabo por parte de la Unidad de Investigación de Incendios del Cuerpo de Bomberos del Distrito Metropolitano de Quito, dando como resultado del análisis de 195 incendios la clasificaron tres categorías: estructural (49.74%), forestal (49.23%) y vehicular (1.03%). Basado en estas cifras, se distingue que la primera causa de incendios es de tipo estructural. A inicios del año 2020 en España la revista Vanguardia publicó que, la principal causa de incendios en el hogar es dejar un sartén olvidado en el fuego (74%), seguido del horno (13%), el gas/butano (7%) o la freidora (6%). Actualmente el 91% de los hogares ecuatorianos utilizan el GLP para el uso doméstico en la cocción de alimentos.

Al emplear una comunicación óptima para la transmisión de datos en una infraestructura IoT sobre sistemas que proporcionen seguridad permite favorecer con una acción concreta como parte de un procedimiento para reducir significativamente el número de incidentes en lo que a incendios

se refiere. Dándole al usuario servicios sencillos y de pronta atención gracias a que gran parte de la población cuenta con dispositivos móviles al alcance de su mano.

JUSTIFICACIÓN APLICATIVA

El presente proyecto se encuentra orientado en el implementar de un sistema de seguridad en cocinas a gas mediante el diseño de un prototipo que provea al usuario final de un control manual y automático tanto de la parte física como de la parte lógica. Para dicho propósito, hay que tener en cuenta la tecnología de comunicación utilizada por los dispositivos para enviar de forma eficaz y confiable los datos a la nube, por ejemplo: Wifi, Bluetooth, ZigBee, LoRa, Z-Wave, entre otras tecnologías. En este sentido, cobran importancia las redes de sensores con la tecnología que permite la escalabilidad del IoT y con la funcionalidad suficiente para proporcionar su integración con la arquitectura actual de Internet.

El prototipo busca identificar los tiempos requeridos para el envío de los datos en tiempo real y generar un ambiente seguro en ausencia del usuario. El sistema prevé integrar el uso de actuadores que permitan el cierre preciso del flujo de gas en el instante solicitado por los algoritmos de programación implantados en una tarjeta de desarrollo, todo esto apoyado en las condiciones que los sensores (provistos de la sensibilidad requerida) capturen.

Por un lado, la interconexión entre las perillas y le etapa de control enviará la información sobre la condición de las hornillas, es decir, si las hornillas se encuentran encendidas o apagadas. Si llegado el caso, una o todas las hornillas se encuentran apagadas, entrará en funcionamiento un sensor de gas que se encargará de alertar al usuario si existe una fuga de gas o incluso se detectará humo dependiendo de las características del sensor en los alrededores de la cocina al mismo tiempo que se cerrará el flujo de gas en caso de que ése haya sido el motivo de la fuga.

La información de los sensores se almacenará en una base de datos y la conexión desde el prototipo hacia el internet y luego a una aplicación en un dispositivo móvil, se realizará mediante una tarjeta de desarrollo en función de las mejores características de comunicación que se adapten con el prototipo.

Se emitirán alertas al dispositivo móvil del usuario en ausencia del mismo manifestando que “Las válvulas de las hornillas fueron situadas a su mínima capacidad”; el usuario deberá volver a la cocina. Esta alerta podrá ser desactivada con la presencia del usuario en función de los datos recopilados sobre la cocina. En caso de mantener la ausencia del usuario en el perímetro de la cocina, las válvulas cerrarán el flujo de gas definitivamente.

El acceso a internet es fundamental para la implementación de IoT, debido a que proporciona un sistema de enlace que permite a cualquier equipo conectarse a la Red y compartir recursos, esto a través de la puerta de enlace o Gateway. Su propósito es traducir la información del protocolo utilizado en una red inicial, al protocolo usado en la red de destino.

Con los distintos requerimientos del prototipo se hará un análisis comparativo de forma teórica entre las tecnologías más importantes de comunicación sobre IoT que se adapte óptimamente con el sistema de seguridad, ya que a partir de la tecnología de comunicación escogida se procederá a diseñar e implementar el prototipo de sistema de seguridad y que a raíz de tal diseño se realice la evaluación en función de los requerimientos que el presente sistema de prevención de incendios plantea.

Se espera que, a partir de esta evaluación acerca de las comunicaciones que mejor interactúen sobre tecnologías IoT para el prototipo de sistema de seguridad en cocinas a gas, en un futuro próximo las compañías que fabrican cocinas incorporen mecanismos equivalentes en sus productos y así favorecer la seguridad de cada uno de sus clientes.

1.4 OBJETIVOS

OBJETIVO GENERAL

Evaluar la transmisión de datos con tecnología IoT de un prototipo de sistema de seguridad en cocinas a gas para la prevención de incendios domésticos.

OBJETIVOS ESPECÍFICOS

- Estudiar las tecnologías de comunicación Wifi, Bluetooth, ZigBee, LoRa y Z-Wave con las que interactúa IoT, comparándolas entre sí, y que permitan el envío óptimo de información del prototipo de un sistema de seguridad de cocción de alimentos hacia el usuario final para la prevención de incendios.
- Establecer los requerimientos necesarios del prototipo de un sistema de seguridad de cocción de alimentos para la prevención de incendios.
- Analizar los elementos hardware y software que se requieren en el diseño del sistema de seguridad de cocción de alimentos en cocinas a gas.
- Implementar el prototipo de un sistema de seguridad en cocinas a gas basado en tecnología IoT en función de los requerimientos para la prevención de incendios domésticos.

- Evaluar el prototipo en función de las exigencias que cada uno de los requerimientos para prevenir incendios.

CAPÍTULO II

2. MARCO TEÓRICO

2.1 Introducción

En esta primera parte se realiza un compendio del fundamento teórico de las comunicaciones, comparándolas entre sí, y determinando la tecnología idónea para el envío óptimo de información al prototipo, todo desde el punto de vista de redes inteligentes, ahondando en las tecnologías que trabajan bajo estos criterios y los equipos existentes que pueden ser integrados a ella, para construir de manera eficaz y sustentada un sistema de seguridad en cocinas a gas para la prevención de incendios domésticos.

2.2 Internet De Las Cosas

El internet de las cosas o también denominado por sus siglas en inglés IoT, es un desarrollo tecnológico que tiene cada vez más presencia en el mundo cotidiano con una continua revolución que involucra tanto a la informática como a las comunicaciones. Su versatilidad en cuanto a su tamaño o su ubicuidad lo convierten en un telón que deja cubierto gran parte de los avances tecnológicos que sobresalieron anteriormente. Sus esfuerzos se dirigen a interconectar diferentes dispositivos inteligentes a una misma red, permitiendo de esta manera una comunicación fluida entre personas y cosas, y aún más allá como la comunicación entre las cosas en sí mismas. (Stallings, Jelassi, Agboma 2016, pp. 364-369)

El internet en este momento soporta en su estructura innumerables objetos ya sean estos personales, domésticos o industriales, todos ellos interconectados más habitualmente a través de sistemas en la nube. Estos objetos en esencia son los encargados de proporcionar información acerca de sensores, ellos a su vez permiten la gestión de un sistema más grande a partir de los datos recopilados y la posibilidad de actuar sobre su entorno, como una fábrica o inclusive una ciudad. Esto exige una accesibilidad a dichos objetos sin restricciones de tiempo o lugar. (Buyya, Dastjerdi 2016, pp. 3-7)

2.3 Redes M2M

Las redes de máquina a máquina o también llamadas M2M son actualmente una transformación de lo que se conocía como WSN (Redes de sensores inalámbricos) y que son incluidas en la mayoría de soluciones clásicas las cuales contenían tecnologías de corto alcance. (McFarlane, Giannikas, Lu 2016, pp. 2-3)

El valor agregado que dan las redes M2M es el incremento formidable en cuanto a la cantidad de dispositivos que conforma una red se refiere. No solo es importante contemplar el número de dispositivos interconectados, sino también, los beneficios en términos económicos, optimización energética, escalabilidad en la red y la cobertura que pueden brindar.

2.4 LP-WAN

Recientemente ha surgido un nuevo paradigma denominado Redes de área amplia de bajo consumo (LP-WAN), cuyo objetivo es llenar el vacío existente para el despliegue de redes M2M.

La base principal de estos sistemas es la implementación de sistemas altamente escalables, generalmente de forma operada, empleando dispositivos de borde de bajo costo con bajo consumo de batería. La Figura 1-2 presenta la arquitectura típica de un sistema LP-WAN, esencialmente, donde una o una serie de estaciones base proporciona conectividad directa desde los dispositivos de borde a la red de backhaul y, luego, a la nube, donde se almacenan y preparan los datos para ser accedido. En cuanto a la arquitectura de la red de borde, básicamente, en lugar de componer una red local y utilizar una puerta de enlace para enviar los datos recopilados, los nodos finales se conectan directamente a la estación base. Esta configuración permite simplificar la complejidad de la gestión de la red y además reduce el consumo energético dado que se evitan las tareas de enrutamiento. (Xiong et al. 2015, pp. 1-2)

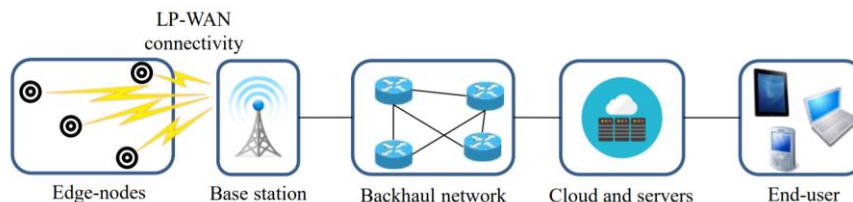


Figura 1-2: Arquitectura de Red LP-WAN

Fuente: (Sánchez R., Cano M., 2016, p. 2)

2.5 Tecnologías de transmisión de datos

Para el para la transmisión y recepción de datos que más se utilizan hoy en día se encuentran, Bluetooth, Wifi, Zigbee, LoRa, Z-Wave, de las cuales se expondrá a continuación.

2.5.1 Bluetooth

Bluetooth es una tecnología perteneciente a las redes de área personal WPAN. Sus características principales engloban una baja potencia lo que significa un bajo consumo de energía, idóneo para pequeños dispositivos que funcionen con baterías, el corto alcance que presenta es esencialmente orientado a un área de operación personal y a una velocidad de transmisión media. Esta tecnología trabaja en la banda RF de 2,4 GHz y dependiendo de la versión, bluetooth puede llegar a velocidades de hasta 48 Mbps. Posee en su esquema 1600 saltos por segundo sobre 79 canales de 1 MHz cada uno, lo que lo hace trabajar correctamente en ambientes en los que exista frecuencias interferentes.

2.5.2 Wifi

Esta tecnología se encuentra enmarcada bajo el estándar IEEE 802.11. Wifi pertenece a las redes de área local WLAN y posee un alcance estimado va desde los 10 hasta los 1000 metros y una velocidad de hasta los 10 Gbps dependiendo de la versión del estándar. Si bien es cierto, wifi cuenta con un mayor alcance, pero no es una solución completamente inalámbrica ya que los puntos de acceso requieren conexión a la red principal de forma cableada. Opera en las bandas de frecuencia de 2,4 y 5 GHz; la banda de 2,4 GHz consta de 14 canales de los cuales solo 3 no presentan solapamiento, mientras que en la banda de 5 GHz se tiene 1, 2, 4, y 8 canales con anchos de banda por canal de 160, 80, 40 y 20 respectivamente.

2.5.3 ZigBee

Es una tecnología que se define en el estándar IEEE 802.15.4 corresponde a las redes de área personal WPAN. Presenta velocidades de transmisión de datos reducida de 20 kbps, 40 kbps y 250 kbps y un alcance limitado aplicadas al control y monitoreo; algo importante que resaltar es que, Zigbee presenta seguridad en todas sus capas y esto se debe a que puede otorgar comunicación encriptada hasta la capa de aplicación, además de constar con una reducida pérdida de datos. Trabaja en las bandas de frecuencia de 868 MHz, 915 MHz y 2.4 GHz, que están disponibles en función de la región.

2.5.4 LoRa

La tecnología LoRa o de largo alcance por sus siglas en inglés, es una evolución reciente a las redes inalámbrica que va ganando popularidad ya que son dispositivos orientado a la conectividad del internet de las cosas. Están orientados a los sistemas integrados que emplean bajo consumo de energía, es decir, aquellos dispositivos que utilizan baterías. Presentan una transmisión de pequeñas cantidades de datos que va desde 300 bps hasta 5 kbps con un ancho de banda de 125 kHz durante pequeños intervalos de tiempo y a largo plazo. (Hersent, Pop, Remond 2016, pp. 5-12)

2.5.5 Z-Wave

Esta tecnología se encuentra orientada a la automatización en entornos residenciales y a la transmisión de datos de mensajes cortos de forma confiable desde una unidad de control a uno o más nodos en la red. Trabaja en las bandas de frecuencia de 868 MHz y 908 MHz con velocidades de 9,6 Kbps y 40 Kbps respectivamente. Dispositivos Z-Wave más recientes incorporan un chip que admite la banda de 2.4 GHz y ofrece velocidades de hasta 200 kbps. Esta tecnología implementa un mecanismo que permite evitar las colisiones a la hora de transmitir una trama en el momento en el que el canal está disponible.

2.6 Tipo de Tráfico de Red y de Internet

En cualquier tipo de red de comunicación existe predeterminadamente un constante tráfico de datos, ya sea en redes locales, empresariales o en internet. Este tráfico está comprendido en dos categorías: elástico e inelástico.

2.6.1 Tráfico Elástico

Es aquel tráfico que satisface toda aquella necesidad que pueda presentar una aplicación ya que puede ajustar sus operaciones a rangos amplios de trabajo, a variaciones en cuanto a retrasos y rendimiento en el internet. Está orientado a redes basadas en TCP / IP y a nivel de la capa de transporte emplea el protocolo TCP como UDP, en función de la disponibilidad y capacidad con la que se generan los datos en una aplicación, así como también la tolerancia máxima de velocidad para aceptar datos en un receptor y a la reducción de velocidad con la que los datos son entregados a una red gracias a que el tráfico es ajustado debido a la congestión que pueda presentarse. Esto presenta beneficios sustanciales ya que los dispositivos de enrutamiento tratan los paquetes de forma equitativa sin distinción de la aplicación que genera dicho paquete o de si el paquete es grande o pequeño. (Stallings, Jelassi, Agboma 2016, pp. 61-62)

2.6.2 Tráfico Inelástico

Este tráfico expone problemas a la hora de adaptarse a las variaciones que una red pueda presentar en internet ya que demanda de requisitos previos para su despliegue. El rendimiento y la latencia deberá ser mínimo dependiendo de la aplicación, el jitter deberá retrasar los paquetes lo suficiente y almacenarlos en un búfer para compensar los tiempos de llegada entre paquetes que es primordial en aplicaciones que manejen datos en tiempo real, la tolerancia en la cantidad paquetes perdidos que una aplicación soporta para ofrecer un razonable QoS. Este tráfico está orientado a aplicaciones multimedia y de simulación interactiva. Esto exige que al interior de la arquitectura de internet exista una especie de protocolo para reservar recursos y de esta manera se puedan presentar requerimientos previos para la preferencia de paquetes cuando los recursos necesarios estén disponibles. (Stallings, Jelassi, Agboma 2016, pp. 62-63)

2.6.3 Tráfico en Tiempo Real

El tráfico en tiempo real está orientado a solventar problema de paquetes perdidos y sincronización. En la mayoría de aplicaciones se maneja el concepto de que la velocidad a la se entregan los datos sea la misma velocidad con la que fueron enviados; de igual manera la disponibilidad de los bloques de datos en función de tiempos preestablecidos o fechas límite para acceder a ellos. Las características anteriores no son necesariamente las que rigen este tipo de tráfico, otro requerimiento especifica que en intervalos fijos de tiempo se generen paquetes de tamaño fijo para reducir así las redundancias que puedan darse, una fuente de encendido / apagado entre los períodos de inactividad y un envío de paquetes en intervalos de tiempo uniformes, pero con un tamaño de paquete variable. (Stallings, Jelassi, Agboma 2016, p. 65)

2.7 Protocolos IoT

El en despliegue de redes IoT se encuentra estrechamente relacionados los protocolos que trabajan sobre ella, indistintamente de la tecnología de radio frecuencia que se empleen. Los datos recopilados por los dispositivos finales, eventualmente tendrán que ser dispuestos a una base de datos local o en la nube, por tal motivo es imprescindible obtener un rendimiento óptimo en la comunicación y unos de los principales factores es el protocolo de mensajería que mejor se adapte a las necesidades que la aplicación de IoT requiera.

2.7.1 MQTT (Message Queuing Telemetry Transport Protocol)

Protocolo de mensajería publicación / suscripción y orientado a comunicaciones ligeras en entornos de redes restringidas. Posee una arquitectura basada en el modelo Cliente / Broker. MQTT es de naturaleza binaria con una cabecera de 2 bytes y con 256 MB como máximo tamaño

de carga útil en cada cabecera. Para el transporte de datos emplea TCP y para su seguridad de apoya en TLS / SSL. En cuanto a la calidad de servicio (QoS) cuenta con 3 niveles lo que hace que MQTT sea cotizado en redes de gran tamaño. (Thangavel et al. 2014, pp. 2-3)

2.7.2 CoAP (Constrained Application Protocol)

Protocolo de mensajería de solicitud / respuesta como de recurso / observación (una variante de publicación / suscripción), se encuentra orientado esencialmente para interoperar con HTTP y web RESTful. Posee una arquitectura basada en el modelo Cliente / Servidor y Cliente / Broker. CoAP es de naturaleza binaria con una cabecera de 4 bytes y con pequeñas cantidades de MB como máximo tamaño de carga útil en cada cabecera debido a que depende del servidor web o de la tecnología de programación. Para el transporte de datos emplea UDP y para su seguridad de apoya en DTLS. En cuanto a la calidad de servicio (QoS) cuenta con 2 niveles y funcionalidades que hace que CoAP sea utilizado en proyectos que requieran de una negociación de contenido. (Ludovici A., Moreno P., Calveras A., 2013)

2.7.3 AMQP (Advanced Message Queuing Protocol)

Protocolo de mensajería corporativa de solicitud / respuesta como de publicación / suscripción, se encuentra orientado a proveer de confiabilidad, seguridad, aprovisionamiento e interoperabilidad. Posee una arquitectura basada en el modelo Cliente / Servidor y Cliente / Broker. AMQP es de naturaleza binaria con una cabecera de 8 bytes y con cantidades de MB en su carga útil de cada cabecera que depende del Broker / Cliente o de la tecnología de programación. Para el transporte de datos emplea TCP y para su seguridad de apoya en TLS / SSL y SASL. En cuanto a la calidad de servicio (QoS) cuenta con 2 niveles. (Ludovici, Moreno, Calveras 2013, pp. 1-2)

2.7.4 HTTP (Protocolo de transporte de hipertexto)

Protocolo de mensajería solicitud / respuesta, basado en texto y orientado a la conexión. Posee una arquitectura establecida en el modelo Cliente / Servidor. HTTP debido a que los datos viajen en forma de texto, el tamaño que le corresponde a cada cabecera y al propio mensaje dependerá exclusivamente del servidor web o de la tecnología de programación. Para el transporte de datos emplea TCP y para su seguridad de apoya en TLS / SSL. En cuanto a la calidad de servicio (QoS) este protocolo no define explícitamente ningún nivel de QoS y requiere adicionalmente de un soporte para prestar ese tipo de servicio. (Han 2015, pp. 7-13)

2.8 Modelos de comunicación de IoT

Para solucionar problemas que se puedan presentar en el diseño de una red IoT, comúnmente se emplean modelos individuales de diseño en las comunicaciones con la capacidad de ser acoplados y reutilizados para de esta manera obtener una reducción en la complejidad e interoperabilidad existente entre las diferentes entidades que dependen de la funcionalidad que se pretenda conseguir. Esto denota un aporte sustancial en los diseños ya que pueden ser aplicados más de un modelo y operar de manera simultáneamente sobre él.

2.8.1 Modelo de comunicación Dispositivo a Dispositivo

Lo que se pretende conseguir con este modelo es que dispositivos de diferentes fabricantes tengan la capacidad de comunicarse directamente sin problemas desde el primer momento. Para que esto pueda ser posible, se requiere que los fabricantes acuerden aspectos de interoperabilidad al interior de su pila de protocolos como: compatibilidad de capa física en función de la tecnología, tecnologías de transición IPv4-IPv6, arquitecturas de comunicación admitidas, mecanismos de descubrimiento de servicios, protocolo de la capa de transporte, seguridad y privacidad. Todas estas características se tienen que contemplar a la hora de un diseño debido a la naturaleza específica de cada dispositivo y de restricciones que puedan tener para adaptarse a un determinado escenario como se presenta en el ejemplo de la Figura 2-2. (Tschofenig et al. 2015, pp. 4-5)

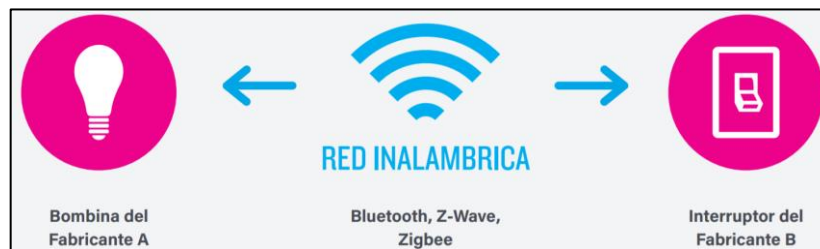


Figura 2-2: Modelo de Comunicación Dispositivo a Dispositivo

Fuente: (Rose, Eldridge, Chapin 2015, p. 20)

2.8.2 Modelo de comunicación Dispositivo a Nube

Este modelo de comunicación se encuentra orientado a cargar en los servicios de aplicaciones que pueda prestar un proveedor los datos obtenidos por sensores. No presenta problemas de interoperabilidad ya que todos los procesos para la comunicación se realizan del lado del proveedor. Permite una comunicación basada en IP de extremo a extremo, en los casos en que los dispositivos no posean maneje esta comunicación, pueden ser fácilmente añadidos mediante la

utilización de un protocolo acorde con el modelo. De igual forma, las tecnologías de radiocomunicación como el conjunto de técnicas de autenticación para el acceso a la red deberán ser reutilizadas debido a que las redes de acceso en las que se encuentre comprendidos los dispositivos no se encuentran comprendidas dentro del control del proveedor de servicios como se plantea en el escenario de la Figura 3-2. (Tschofenig et al. 2015, p. 6)

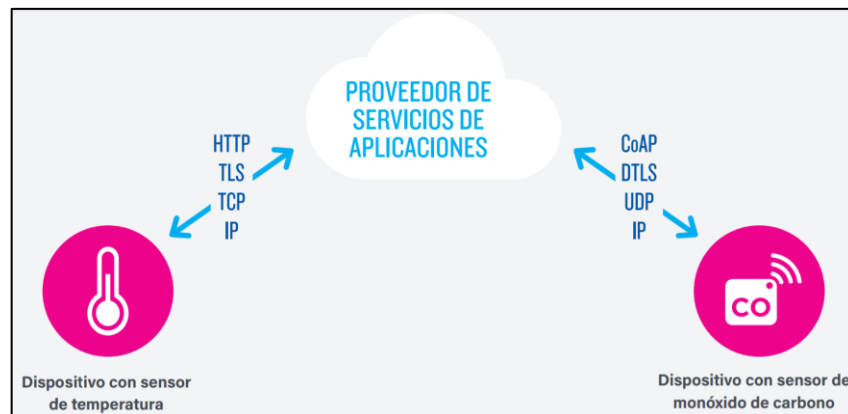


Figura 3-2: Modelo de Comunicación Dispositivo a Nube

Fuente: (Rose, Eldridge, Chapin 2015, p. 21)

2.8.3 Modelo de comunicación Dispositivo a Gateway

Para los escenarios en los que se empleen dispositivos menos disponibles o requieran de una funcionalidad especial en la capa de aplicación o que se necesite incorporar dispositivos heredados que no basan su comunicación en IP, es necesario incorporar un Gateway (puerta de enlace) en la arquitectura de la comunicación que actúe como un puente entre las distintas tecnologías que se encuentren comprendidas en el escenario además de proveer de otras funciones de red y de seguridad al mismo, lo cual visualizamos en la Figura 4-2. Es común ver que los proveedores de estos dispositivos sean quienes también proporcionen estas puertas de enlace, ya sea por el uso de protocolos propietarios, por evitar depender de otros proveedores o incluso por problemas de interoperabilidad. Este modelo se aplica muy frecuentemente en escenarios que requieran de interacción en tiempo real o también en los que necesita acceder de forma remota a configuraciones sobre los dispositivos, operando IPv6 como IPv4 reduciendo los requisitos del sistema y siendo así más flexibles. (Tschofenig et al. 2015, pp. 7-8)

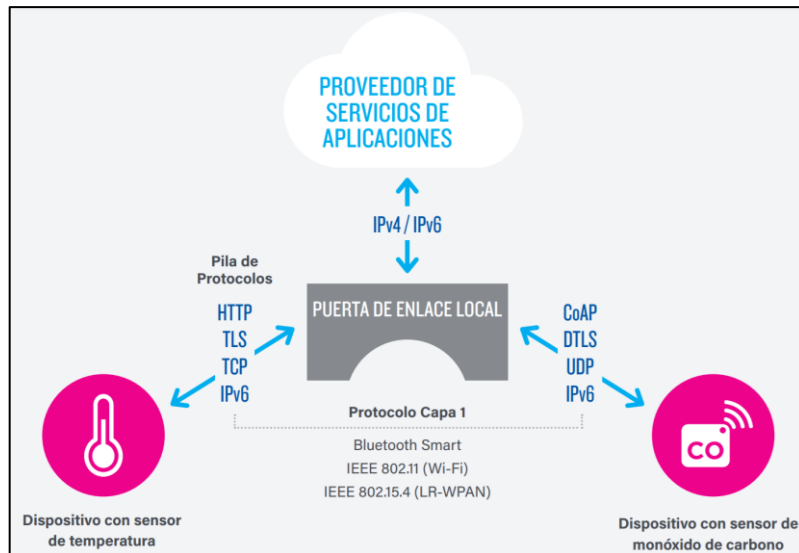


Figura 4-2: Modelo de Comunicación Dispositivo a Gateway

Fuente: (Rose, Eldridge, Chapin 2015, p. 22)

2.8.4 Modelo de intercambio de datos mediante Back-End

Este modelo se enfoca principalmente en que más allá de cargar los datos de sensores a una nube para ser analizados, sean combinados con datos que se puedan cargar de otras fuentes logrando así una interoperabilidad a nivel de los datos y descentralizando a los proveedores de servicios de aplicaciones como se muestra en el ejemplo de la Figura 5-2. Es común que se trabaje con un diseño de API RESTfull en combinación con otras tecnologías para la reutilización a nivel de bloques de construcción, aunque desafortunadamente no se cuenta con una estandarización de la pila de protocolos. (Tschofenig et al. 2015, p. 9)

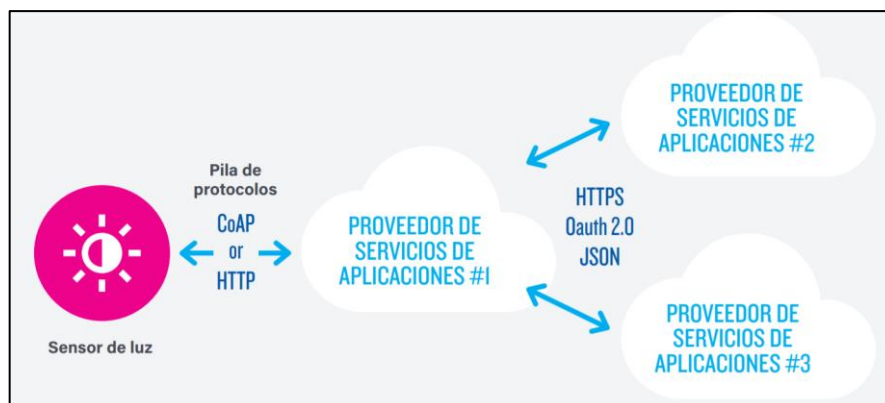


Figura 5-2: Modelo de Intercambio de Datos Mediante Back-End

Fuente: (Rose, Eldridge, Chapin 2015, p. 23)

2.9 Tarjetas de Desarrollo

2.9.1 Arduino

Es una plataforma que comprende tanto hardware como software, emplea código abierto lo que permite al usuario final adaptarla a escenarios de forma independiente para necesidades específicas. Utiliza para su programación el lenguaje propietario de Arduino que se basa en Wiring y el software IDE de igual procedencia que se basa en Processing. Es un entorno multiplataforma ya que se encuentra soportado en los sistemas operativos como MAC, Windows y Linux. Las ventajas que este tipo de dispositivos puede proveer van desde simplificar procesos a lo que trabajar con microcontroladores respecta hasta ser más económicos que otros dispositivos del mercado. Es versátil gracias a que posee en su placa entradas y salidas analógicas y digitales. (Arduino 2018)

2.9.2 Raspberry PI

Presenta características mucho más robustas que sus competidores, es una placa de propósito general asociada generalmente como un mini computador ya que viene integrado un sistema operativo basado en Linux llamado Raspbian. Al igual que Arduino, Raspberry emplea código abierto y es una plataforma que comprende hardware y software. Provee de una conexión a internet de fácil acceso por su puerto ethernet siendo compatible en tiempo real con su placa y con muchas más aplicaciones internas y externas que basen su comunicación mediante internet. Es importante resaltar que estas placas ofrecen interoperabilidad entre sus versiones actuales con sus versiones anteriores. (Raspberry Pi 2019, pp. 2-3)

2.9.3 Node MCU

Es una tarjeta de desarrollo de código abierto y firmware basada en LUA diseñada para aplicaciones orientadas a IoT. Está provisto de un microprocesador Tensilica Xtensa LX106 RISC de alta potencia que trabaja en conjunto con Wi-Fi, Bluetooth y Deep Sleep como funciones integradas en su placa de la cual se desprenden múltiples entradas y salidas. Utiliza para su programación el lenguaje propietario de Arduino y el software IDE de igual procedencia. (Espressif Systems 2015, pp. 6-9)

2.9.4 Galileo

Es una tarjeta Arduino con la arquitectura Intel, cuenta con pines de entrada y salida analógicas y digitales, un puerto serial e ICSP para una programación en serie. Dispone de un procesador de

aplicaciones orientado a IoT con eficiencia energética y un tamaño reducido, un puerto ethernet con capacidad de 100 MB, lector de tarjetas micro SD, ranura Mini PCI Express para conectar módulos Wi-Fi, Bluetooth y tarjeta SIM. Mediante el uso distintos lenguajes de programación basados en Linux, Python, Secure Shell, node.js y OpenCV se puede disponer del acceso a los puertos serie y a los pines de la placa. (Arduino 2015)

2.10 Incendios

Un incendio es toda aquella combustión incontrolada en la que participan materiales que se constituyen un combustible y que forman parte del entorno en el que se desarrolla, liberando energía en forma de calor. (Drysdale 2019, p. 41.2-41.5)

2.10.1 Clasificación de Incendios

Incendio Clase A

Son aquellos que constituyen como combustibles materiales tales como:

- Madera
- Tela
- Papel
- Caucho
- Plásticos

Incendio Clase B

Son aquellos que constituyen como combustibles líquidos tales como:

- Grasas de Petróleo
- Alquitrán
- Aceites
- Pinturas a base de aceite
- Disolventes
- Lacas
- Alcoholes
- Gases Inflamables

Incendios Clase C

Son aquellos que constituyen como combustibles cualquier tipo de aparato eléctrico que se encuentre energizado. (Servicio Ecuatoriano de Normalización 2017, p. 3)

Incendios Clase D

Son aquellos que constituyen como combustibles metales como:

- Magnesio
- Titanio
- Circonio
- Sodio
- Litio
- Potasio

Incendios Clase K

Son aquellos que constituyen como combustibles lo que se utilizan para cocinar tales como:

- Aceites
- Grasas Vegetales o Animales

2.10.2 Causas y breves cifras de incendios en los hogares ecuatorianos

La secretaría de gestión de riesgos en el año 2015 dio a conocer que en 16 provincias de país se suscitaron al menos 81 eventos registrados que involucran el GLP, de los cuales se tuvo como resultado 192 personas directamente afectadas de dichos eventos, entre ellas 83 heridos y 6 fallecidos. Las cifras más altas se concentran en las provincias de Pichincha, Tungurahua y Loja. Manabí por si sola registra 35 personas damnificadas con 54 viviendas afectadas y 16 destruidas en su totalidad. Solo en el mes de enero de ese mismo año estos eventos habrían cobrado las vidas de 4 personas. (Secretaría de Gestión de Riesgos Ecuador 2015)

El reporte de la SGR mediante una campaña de concientización a través de los medios de comunicación indicó que una de las principales causas que surgen de estos incendios es la fuga de gas en los hogares enfatizando en que esto es sinónimo de muertes silenciosas y que de la misma manera una herramienta para dar inicio a un incendio, dando como resultado pérdida de vidas humanas a causa de las cocinas a gas.

2.10.3 Fases de un Incendio

Un incendio se compone de tres fases principales. La fase número 1 o también llamada la fase incipiente, en la cual aún existe la presencia de oxígeno en el ambiente dando como resultado vapor de agua mientras un incendio sigue creciendo, también se encuentra dióxido de carbono, pequeñas concentraciones de monóxido de carbono y dióxido de azufre entre otros gases. (Comisión Nacional De Prevención De Riesgos Y Atención De Emergencias, Costa Rica 2017)

La fase 2 o fase de libre combustión es en la que se expanden los gases calientes hacia la parte superior de las estructuras y así obligando al aire frío que contiene la mayor concentración de oxígeno a que descienda y avive las llamas, debido a este proceso es que todo tipo de material contenido en la infraestructura sea alcanzado por las llamas y elevando rápidamente la temperatura del lugar. (Comisión Nacional De Prevención De Riesgos Y Atención De Emergencias, Costa Rica 2017)

La fase final o fase latente es en la que las llamas pueden cesar debido a la privación de oxígeno indispensable para la combustión. Si bien he cierto el incendio puede haber terminado pero los

riegos siguen latentes ya que se mantienen altas temperaturas acompañadas de humo denso y gases liberados de la presión que se crea al interior de una infraestructura consumida por un incendio. Los gases que intervienen en esta fase son el metano, monóxido de carbono e hidrógeno, los cuales son altamente inflamables y con la capacidad de explotar. (Comisión Nacional De Prevención De Riesgos Y Atención De Emergencias, Costa Rica 2017)

2.11 Sistemas Contra Incendios Estructurales

2.11.1 Tipos de Seguridades

Los tipos de seguridades son aquellas medidas que se toman para actuar frente a un incendio, las cuales se dividen en pasiva y activa.

2.11.1.1 Seguridad Pasiva

La seguridad pasiva involucra todo aquello mecanismo o medio con que debe contar un lugar para advertir de todo posible perjuicio al suscitarse un incendio en su interior sin la intervención del hombre o energía externa durante un tiempo determinado, con el propósito de resguardar tres puntos fundamentales, pérdidas humanas, pérdidas materiales y la expansión del flagelo. La seguridad pasiva contempla varias categorías: 1) Prevención contra incendios, englobando todas aquellas medidas destinadas a que una infraestructura soporte un incendio impidiendo su colapso y asegurando estabilidad en el proceso. 2) Compartimentación, son aquellos sistemas encargados de dividir en módulos un establecimiento para impedir la propagación del fuego. 3) Tratamientos ignífugos, son todas las acciones que se toman para aislar la propagación del fuego de la propia infraestructura, para lo cual se emplean materiales incombustibles con un punto de fusión alto, con un bajo nivel de emisión de humo y capaz de no liberar sustancias tóxicas en el caso de ser consumidos por un incendio. (Chacarrex 2022)

2.11.1.2 Seguridad Activa

La seguridad activa por otro lado son aquellos dispositivos encargados de alertar a quienes se encuentren a interior de un establecimiento y simultáneamente impedir que las llamas se difundan. Esto con el objetivo de controlar directamente un incendio desde el momento en el que éste se genera. La seguridad pasiva agrupa varios tipos de seguridad: 1) Detección, son todos aquellos dispositivos (detectores de calor, humo, llamas) que se mantienen censando su entorno y emitiendo señales que adviertan a los presentes de un posible incendio y faciliten una pronta evacuación. 2) Supresión del fuego, es la parte encargada de hacerle frente directo a un incendio

haciendo uso de equipos como extintores, sistemas de riego contra incendios, entre otros. 3) Ventilación mecánica, son todos aquellos mecanismos instalados en una infraestructura que aseguran mantener las vías de evacuación libres de humo y gases tóxicos. (Chacarrex 2022)

2.11.2 Métodos de Extinción frente un Incendio

Son todas aquellas acciones encaminadas a extinguir un incendio en función del factor al que se esté enfrentando. (FREMAP 2015, p. 12)

2.11.2.1 Enfriamiento

Su función principal radica en intervenir directamente sobre el calor generado por las llamas y eliminarlo por completo.

2.11.2.2 Sofocación

El objetivo primordial de este método es atacar el elemento principal que aviva todo incendio y es el oxígeno, se debe eliminar en su gran mayoría hasta que no constituya un aporte hacia la propagación de las llamas.

2.11.2.3 Eliminación del Combustible

Se enfoca en aislar de cualquier forma los posibles elementos que se constituyan un combustible del camino de las llamas.

2.11.2.4 Inhibición

Son todas aquellas técnicas y elemento de naturaleza química encargados de neutralizar los radicales libres que son los responsables de que un incendio se combustione y propague.

2.11.3 Dispositivos Iniciadores

Son todos aquellos componentes automáticos o manuales distribuidos estratégicamente en función de sus características y encargados de la detección, supresión y extinción de incendios. Estos elementos deberán proveer unos tiempos de respuesta óptimos para emitir una alerta cumpliendo con parámetros mínimos de operación. (Campos 2016, pp. 72-109)

Los dispositivos automáticos o también llamados detectores, son componentes encargados de la detección de los fenómenos propios de un incendio (gases, humo, temperatura o radiación). (Campos 2016, pp. 72-109)

2.11.3.1 Detector de Gases de Combustión o Iónicos

Son aquellos detectores con la capacidad de percibir tanto humos visibles como invisibles provenientes de la combustión de diversos materiales en un incendio. Por otra parte, los detectores de gases están provistos de dos cámaras que entre ellas posee niveles muy bajos de iones de oxígeno y nitrógeno, estos elementos al entrar en contacto con el humo proveniente de un incendio, modifica la corriente que pasa entre las dos cámaras y de esta manera se emitiría una alarma.

2.11.3.2 Detector Óptico de Humos

Son aquellos detectores que basan su funcionamiento en la captación de la luz de forma muy sensible que (hasta el polvo podría entorpecer su funcionamiento) al ser obstruidos por diferentes tipos de humo, emitirá una alerta.

2.11.3.3 Detector Temperatura

Los detectores de temperatura se dividen en termovelocimétricos y de temperatura fija. Los detectores de temperatura fija que se activa al alcanzar un umbral determinado en las características de su diseño a partir de la deformación o fusión de uno de sus elementos. Los detectores termovelocimétricos poseen la función de medir el incremento de la temperatura en función de la velocidad con la que crece y emplean para ello una varilla metálica que gracias a su dilatación permite determinar la temperatura de su entorno.

2.11.3.4 Detector de Llamas

Los detectores de llamas son elementos infrarrojos o ultravioletas con filtros ópticos que captan la radiación emitida por las llamas.

CAPÍTULO III

3. MARCO METODOLÓGICO

Introducción

A continuación, se describe el desarrollo e implementación del diseño del sistema de seguridad en cocinas a gas para generar un entorno protegido a partir de sensores que permitan un acceso confiable del estado de la cocina en tiempo real y de esta manera prevenir los potenciales incendios que puedan darse de acuerdo con los requisitos que la investigación plantea para la correcta selección de componentes tanto de hardware como de software que integren dicho sistema.

Para satisfacer las necesidades que el sistema y la capacidad de brindar seguridad, se establece como metodología de desarrollo un análisis cualitativo identificando a partir de información recolectada, una correcta clasificación e identificación de información que permita establecer un sistema óptimo.

El despliegue de los dispositivos y principalmente las capacidades que cada uno de ellos posea es primordial a la hora de la selección, ya que cada uno varía en función de la sensibilidad, comunicación y procesamientos de los datos recolectados, además de que cuentan con características de escalabilidad, costos y consumo energético diferentes.

3.1 Indicadores del sistema de seguridad

A partir de la información recolectada en el capítulo previo, existen parámetros iniciales que son de carácter obligatorio para arrancar con el diseño y posterior evaluación de desempeño; acorde al entorno en el que se implementará y a las tecnologías que se pretende utilizar.

Entre las condiciones que los dispositivos deben cumplir, está la disponibilidad que los elementos eléctricos, electrónicos y mecánicos deben tener, ya sea de forma manual o automática. Otra consideración es la activación de cada uno de los algoritmos implantados, estos deberán seguir una secuencia previamente fijada y actuar de manera simultánea. Adicionalmente, el sistema debe constar con un indicador que sea sinónimo de una alerta oportuna, ya sea visual y/o sonora además de contar con una notificación remota que permita al usuario estar informado de las condiciones en las que se encuentra al área asegurada.

3.2 Requerimientos del Sistema

En función de los lineamientos que los sistemas contra incendios exigen y con base en las necesidades del usuario final, se instaura los siguientes requerimientos tanto para el sistema como para el usuario.

3.2.1 *Requerimientos del usuario*

El usuario que demande de un sistema de seguridad como el que se plantea en este trabajo, debe contar con cobertura de red móvil en su Smartphone para recibir las notificaciones de alerta transmitidas desde el sistema. Debe contar con un acceso a internet ya sea mediante datos o Wifi para monitorear en tiempo real el estado de la cocina.

3.2.2 *Requerimientos de software*

El software debe constar con la capacidad de recibir los datos que el sistema de seguridad transmita, almacenarlos en la nube, permitir visualizar esa información en una interfaz amigable para el usuario y que provea de igual manera una interpretación sencilla de los datos recolectados por los sensores. Debe ofrecer una configuración simple para un mantenimiento oportuno y para la adhesión de mejoras en lo que a dispositivos y/o lineamientos requiera el sistema. Además, ser compatible con distintos dispositivos y contar con la aplicación o programa que dichos dispositivos soporten.

3.2.3 *Requerimiento de hardware*

Los elementos empleados en el prototipo deben ser interoperables entre ellos, tener un costo reducido, incorporación sencilla a la cocina y al sistema. Los actuadores incorporados deberán ejecutar acciones inmediatas al encontrarse con eventos que puedan desencadenar un incendio mientras que simultáneamente alerten al usuario de los eventos suscitados, todo ello como medidas encaminadas a fortalecer la prevención de incendios. Debe estar provisto de energía eléctrica en todo momento, ya sea de la red eléctrica o de sistemas de alimentación mediante baterías en caso de fallas en el suministro eléctrico. Los dispositivos deben tener la capacidad de comunicación eficaz con el usuario, transmisión de datos con la nube, tiempos de censado y retardos mínimos. Debe tener la capacidad de adaptarse al rendimiento del sistema en función del incremento de elementos y responder a ello.

3.3 Plataforma en la Nube

3.3.1 *Arduino IoT Cloud*

La plataforma con servicios en la nube y con diversas herramientas para sus propios equipos y también para ciertos equipos de otras marcas denominado Arduino IoT Cloud, es un poderoso conjunto de instrumentos con la capacidad de configuración, programación, monitoreo y control de distintos dispositivos inteligentes orientados al internet de las cosas. A su vez presenta un entorno de visualización mediante paneles que convierte los datos brutos en sub paneles que expone dichos datos en una interfaz gráfica amigable para los usuarios.

Entre los requisitos de hardware que Arduino IoT Cloud precisa, se mencionan las tarjetas de desarrollo oficiales de Arduino compatibles con la nube y también con una amplia gama de tarjetas de desarrollo que basan sus operaciones en microcontroladores ESP32/ESP8266. Es propio mencionar que no solo se puede acceder a los servicios en la nube de Arduino mediante Wi-Fi, la plataforma también soporta comunicación mediante LoRaWAN y redes móviles.

Una de las características sobresalientes de esta plataforma es la capacidad de cargar la programación que se desee implantar sobre una de las tarjetas de desarrollo de Arduino, pero de manera inalámbrica mediante la denominada Over-The-Air (OTA), esto hace que no sea necesaria la conexión física de la placa. Esta opción se habilita después de la primera carga cableada, OTA se podrá visualizar como opción en el panel cada vez que se requiera cargar códigos en las placas antes mencionadas y con una conexión a la red de dichas placas.

Otra de las funciones habitualmente utilizadas es la de sincronización de variables, su potencial es grande debido a que permite que las variables se comuniquen entre sí compartiendo sus datos almacenados sin requerir que se digite una línea de código para ello, de esta manera es posible vincular variables que sean del mismo tipo de datos entre diversos dispositivos conectados con la nube de Arduino, todo esto con miras a garantizar que los cambios suscitados sobre este tipo de variables se propaguen entre los paneles que se encuentren vinculados.

Para que las tarjetas de desarrollo puedan ser interpretadas por la plataforma, se deben crear previamente unos elementos llamados “things”, que es un concepto abstracto establecido por los desarrolladores y que comprende todas las variables y ajustes con las que se pretendan trabajar en un proyecto y también recopilar todos los datos ingresados a cada una de las variables en un historial. Cada thing es capaz de almacenar una variable o un conjunto de variables dependiendo de la orientación que se le dé, cada variable al interior de la plataforma es capaz de almacenar datos provenientes de dispositivos finales como por ejemplo sensores, mostrar dichos datos en un

panel que ofrece una interfaz gráfica para ello y acoger los cambios que se pueda dar en el panel o cualquier otro tipo de dispositivo vinculado que dicho panel. Arduino IoT Cloud se compone de mecanismos de comunicación y sincronización no visibles para el usuario haciendo que cada variable se pueda usar directamente al momento de generar una carga de líneas de código.

Entre las variables que se pueden encontrar al interior de la plataforma, se hallan unos tipos de datos especializados con la capacidad de integrar servicios de terceros y permitiendo una integración más inteligente, interoperando con los paneles y los objetos de visualización que posee la plataforma. (Arduino 2022)

3.4 Diseño del sistema

En función de los requerimientos establecidos anteriormente y con base en las características específicas que un entorno de hogar y sobre todo de cocina posee, se establece una estructura de diseño que interconecta una red de sensores, actuadores y módulo de transmisión para así dar cumplimiento a contribuir con la seguridad en entornos de cocinas residenciales establecidos en la investigación del capítulo anterior.

3.4.1 Situación actual de cocinas en entornos residenciales

La manipulación de los mecanismos de apertura o cierre de válvulas que se encarga de cada hornilla hasta el día de hoy se ha manejado directamente por el control manual del usuario final, esto implica también el uso indebido, no intencional e incluso deliberado del gas de uso doméstico y de su accionar.

Si bien he cierto, actualmente existen en el mercado detectores de gas que emiten una alerta sonora al captar una fuga de este elemento, sin embargo, no tienen la capacidad de actuar sobre la fuga ni de alertar de forma remota o mediante otro mecanismo al usuario que corresponda. Por otra parte, también existen dispositivos de apertura y cierre de la válvula que se encuentra directamente sobre el cilindro de gas, estos dispositivos desarrollados recientemente tienen la capacidad de ser accionados de manera remota, pero sin la interpretación de las condiciones de su entorno.

La correcta ubicación y selección de cada uno de los sensores ha sido delimitada con el propósito de asegurar cuatro parámetros fundamentales que se encuentran en una cocina y a su alrededor, para ello se ha considerado dos zonas de riesgo potencial.

3.4.1.1 Zona circundante a la cocina

El área que rodea una cocina en funcionamiento debe constar con la presencia de una persona adulta para su supervisión, tanto de la manipulación de las hornillas como de la presencia de cualquier fuga de gas o de alteraciones anormales en la temperatura de sus alrededores. Por tal motivo, se ha destinado tres sensores para garantizar la supervisión de estos cambios no deseados.

3.4.1.2 Zona de la cocina

Los elementos y los mecanismos con los que trabaja cada cocina no poseen dispositivos que adviertan de su correcto funcionamiento y que alerte de manera eficaz acerca de anomalías en el sistema. Para ello se han destinado tres tipos de sensores con funciones específicas y diferentes con las que se obtenga información de la manipulación de hornillas además de advertir a la etapa de control del sistema sobre la ausencia del usuario en momentos que se requiere de su presencia.

3.4.2 Selección de tecnología de comunicación

Considerando las características más sobresalientes de cada una de las tecnologías de comunicación para la transmisión y recepción de datos, se establece la tabla 1-3.

Tabla 1-3: Comparativa entre Bluetooth, Wifi, ZigBee, Lora, Z-Wave

	Bluetooth	Wifi	ZigBee	LoRa	Z-Wave
Especificaciones IEEE	802.15.1	802.11a 802.11b 802.11g 802.11n 802.11ac 802.11ad 802.11af 802.11ah 802.11ax	802.15.4	802.15.4	ARCOTEL Resolution 12-09-ARCOTEL-2107, Note EQA.45
Banda de Frecuencias	2.4 GHz	2.4 GHz 5 GHz	868/915 MHz 2.4 GHz	433 MHz, 868 MHz 915 MHz	868 MHz 908 MHz 2.4 GHz
Velocidad de Datos	721.2 kbps 3 Mbps 24 Mbps 48 Mbps	54 Mbps – 10 Gbps	250 kbps 40 kbps 20 kbps	300 bps 5 kbps	9,6 kbps 40 Kbps 200 kbps
Cobertura	Clase 1 – 100 m Clase 2 – 10 m Clase 3 – 1m	10 – 1000 m	10 – 100m	> 10 km	< 400 m
Potencia	Clase 1 – 20dBm	< 30 dBm	0 dBm	< 20 dBm	< 30 dBm

	Clase 2 – 4dBm Clase 3 – 0dBm				
Acceso al Medio	TDD FDMA TDMA	DS-SS FH-SS	CSMA-CA	LoRaWAN	
Número de Canales	79	1-14 1/2/4/8	1/10/16	8	
Topología	Piconet Scatternet	BSS	Estrella Árbol	Estrella	Malla
Modulación	GFSK PSK 8 – DQPSK $\pi / 4$ DQPSK	BPSK QPSK QAM 16QAM 64QAM 256QAM 1024QAM	BPSK O-QPSK	Chirp Spread Spectrum (CSS)	FSK GFSK

Realizado por: Jonathan Collantes, 2022

3.4.2.1 Análisis

A partir de las tecnologías de comunicación planteadas para el desarrollo del presente documento y apoyando la investigación en el estudio cualitativo de cada una de ellas, se establece la siguiente tabla basada en el método de escala de Likert que determina los porcentajes de cada una de las categorías que se van a considerar para la selección de la tecnología que mejor se adapte al prototipo y a la prevención de incendios.

Tabla 2-3: Niveles de eficiencia para la ponderación y evaluación de características

1	2	3	4	5
Nada Eficiente	Ineficiente	Poco eficiente	Eficiente	Muy eficiente
0%	1% - 25%	26% - 50 %	51% - 75%	76% - 100%

Fuente: Gusqui Yessenia, 2017

Cada ponderación en función de las características tendrá que ser obtenida a partir de la ecuación (1) descrita a continuación.

$$P = \frac{\text{porcentaje acumulado}}{\text{porcentaje total}} * 100\% = \frac{\sum N_i}{T_i} * 100\% \quad (1)$$

De la fórmula se desprende N_i siendo el nivel de puntaje asignado con base en las características y siendo T_i el puntaje total que corresponde al 100%. A continuación, se procede a determinar las ponderaciones correspondientes a cada una de las tecnologías en función de los requerimientos del prototipo y en conformidad con lo establecido en la escala de Likert que dictamina ponderaciones entre 1 y 5.

Tabla 3-3: Niveles de eficiencia para la ponderación y evaluación de características para las tecnologías de comunicación

Características	Bluetooth	Wifi	ZigBee	LoRa	Z-Wave
Banda de Frecuencias	4	5	4	3	4
Velocidad de Datos	4	5	3	3	3
Cobertura	3	5	4	5	5
Potencia	3	4	5	4	4
Acceso al Medio	4	5	4	3	0
Número de Canales	5	5	4	3	0
Modulación	4	5	3	4	3
Puntaje Acumulado	27	34	27	25	19
Ponderación de eficiencia	77,14%	97,14%	77,14%	71,43%	54,29%

Realizado por: Jonathan Collantes, 2022

Cálculos:

$$P \text{ Acumulado}_{Bluetooth} = \sum N_i = 4 + 4 + 3 + 3 + 4 + 5 + 4 = 27$$

$$P \text{ Acumulado}_{Wifi} = \sum N_i = 5 + 5 + 5 + 4 + 5 + 5 + 5 = 34$$

$$P \text{ Acumulado}_{ZigBee} = \sum N_i = 4 + 3 + 4 + 5 + 4 + 4 + 3 = 27$$

$$P \text{ Acumulado}_{LoRa} = \sum N_i = 3 + 3 + 5 + 4 + 3 + 3 + 4 = 25$$

$$P \text{ Acumulado}_{Z-Wave} = \sum N_i = 4 + 3 + 5 + 4 + 0 + 0 + 3 = 19$$

$$Ponderación \text{ de eficiencia}_{Bluetooth} = \frac{P \text{ Acumulado}_{Bluetooth}}{T_i} * 100\% = \frac{27}{35} * 100\% = 77,14\%$$

$$Ponderación \text{ de eficiencia}_{Wifi} = \frac{P \text{ Acumulado}_{Wifi}}{T_i} * 100\% = \frac{34}{35} * 100\% = 97,14\%$$

$$Ponderación \text{ de eficiencia}_{ZigBee} = \frac{P \text{ Acumulado}_{ZigBee}}{T_i} * 100\% = \frac{27}{35} * 100\% = 77,14\%$$

$$Ponderación \text{ de eficiencia}_{LoRa} = \frac{P \text{ Acumulado}_{LoRa}}{T_i} * 100\% = \frac{25}{35} * 100\% = 71,43\%$$

$$Ponderación \text{ de eficiencia}_{Z-Wave} = \frac{P \text{ Acumulado}_{Z-Wave}}{T_i} * 100\% = \frac{19}{35} * 100\% = 54,29\%$$

Los cálculos realizados y resumidos en la Tabla 3-3 arrojan como resultado que la tecnología Wifi es la más adecuada para el prototipo que se desea implementar alcanzando un 97,14% a partir de la descripción de las características de cada una de las tecnologías y de las prestaciones que cada una tiene para satisfacer las necesidades del prototipo, se establece que la tecnología WiFi es la que mejor se adapta al entorno de trabajo del sistema de seguridad por varias razones. La

velocidad de datos con la que opera en sus distintas versiones propone un puente para el tráfico masivo de información lo que da lugar a que el sistema pueda con el tiempo incluir mejoras para robustecer la prevención de incendios y por ende generar escalabilidad; esto satisface también la parte económica ya que no requiere incorporar dispositivos adicionales al entorno de trabajo para establecer una comunicación entre el sistema y la nube debido a que la mayoría de hogares cuentan con este servicio. En cuanto a la cobertura, existen versiones de wifi e incluso nodos repetidores comúnmente empleados y bastante eficientes para residencias que disponen de un área de cobertura bastante extensa. La canalización tanto en la frecuencia de 2.4 GHz como en la frecuencia de 5 GHz asegura que se pueda establecer mejores prestaciones en el servicio para evitar al máximo posibles interferencias.

3.4.3 Selección de tarjeta de desarrollo

A continuación, se presenta una comparativa en la Tabla 4-3 entre de las tarjetas de desarrollo acompañadas de sus características principales y parámetros técnicos.

Tabla 4-3: Comparativa entre Arduino, Raspberry Pi, Node MCU, Intel Galileo.

	Arduino	Raspberry Pi	Node MCU ESP-32S	Intel Galileo
Procesador	ATMega 328	Broadcom BCM2836 ARM Cortex-A7	Tensilica Xtensa LX106 RISC	SoC Quark X100
Memoria RAM	2 KB	1 GB	520 KB	512 KB
Voltaje de entrada	7 – 12 V	5 V	7 – 12 V	5 V
Voltaje de Operación	5 V	3.3 V / 5 V	5 V	3.3 V / 5 V
E/S Analógicas	6	X	18	6
E/S Digitales	14	8	24	14
Memoria Flash	32 KB	X	4 MB	8 MB
USB	1	4	1	2
Ethernet	No contiene	10/100	No contiene	10/100
Wifi	No contiene	No contiene	Si contiene	No contiene
Precio	\$30	\$50	\$17	\$90
Entorno de Desarrollo	Arduino IDE	Linux, IDLE, Open Embedded, QEMU, Scratch box, Eclipse, Windows.	Arduino IDE	Arduino IDE
Velocidad	16 MHz	900 MHz	80 MHz a 240 MHz	400 MHz
Sistema Operativo	Ninguno	Distribuciones de Linux, Windows 10	Ninguno	Linux, Windows

Realizado por: Jonathan Collantes, 2022

3.4.3.1 Análisis

En función de los datos recolectados de las tarjetas de desarrollo con mayor utilidad en el mercado y sustentadas con los datos recopilados de cada una de ellas, se establece las ponderaciones de cada tarjeta en función de los porcentajes del método de escala de Likert señalados en la tabla 2-3 y sus categorías en conformidad con lo establecido en la escala, que dictamina ponderaciones entre 1 y 5.

Tabla 5-3: Niveles de eficiencia para la ponderación y evaluación de características para las tarjetas de desarrollo

Características	Arduino	Raspberry Pi	Node MCU ESP-32S	Intel Galileo
Procesador	4	5	5	4
Memoria RAM	2	5	4	2
Voltaje de entrada	5	3	5	3
Voltaje de Operación	4	5	4	5
E/S Analógicas	3	0	5	3
E/S Digitales	3	2	5	3
Memoria Flash	2	0	4	5
USB	2	4	2	3
Ethernet	0	3	0	3
Wifi	0	0	5	0
Precio	3	2	5	1
Entorno de Desarrollo	4	5	4	4
Velocidad	2	4	3	4
Puntaje acumulado	34	38	51	40
Ponderación de eficiencia	52,31%	58,46%	78,46%	61,54%

Realizado por: Jonathan Collantes, 2022

Cálculos:

$$P_{Acumulado}_{Arduino} = \sum N_i = 4 + 2 + 5 + 4 + 3 + 3 + 2 + 2 + 0 + 0 + 3 + 4 + 2 = 34$$

$$P_{Acumulado}_{Raspberry Pi} = \sum N_i = 5 + 5 + 3 + 5 + 0 + 2 + 0 + 4 + 3 + 0 + 2 + 5 + 4 = 38$$

$$P_{Acumulado}_{Node MCU} = \sum N_i = 5 + 4 + 5 + 4 + 5 + 5 + 4 + 2 + 0 + 5 + 5 + 4 + 3 = 51$$

$$P_{Acumulado}_{Intel Galileo} = \sum N_i = 4 + 2 + 3 + 5 + 3 + 3 + 5 + 3 + 3 + 0 + 1 + 4 + 4 = 40$$

$$Ponderación\ de\ eficiencia_{Arduino} = \frac{P_{Acumulado}_{Arduino}}{T_i} * 100\% = \frac{34}{65} * 100\% = 52,31\%$$

$$\text{Ponderación de eficiencia}_{\text{Raspberry Pi}} = \frac{P_{\text{Acumulado}_{\text{Raspberry Pi}}}}{T_i} * 100\% = \frac{38}{65} * 100\% = 58,46\%$$

$$\text{Ponderación de eficiencia}_{\text{Node MCU}} = \frac{P_{\text{Acumulado}_{\text{Node MCU}}}}{T_i} * 100\% = \frac{51}{65} * 100\% = 78,46\%$$

$$\text{Ponderación de eficiencia}_{\text{Intel Galileo}} = \frac{P_{\text{Acumulado}_{\text{Intel Galileo}}}}{T_i} * 100\% = \frac{40}{65} * 100\% = 61,54\%$$

Los cálculos realizados y resumidos en la Tabla 5-3 arrojan como resultado que la tarjeta de desarrollo que mejor se adapta al prototipo que se pretende desarrollar en este documento es la tarjeta Node MCU ESP-32S alcanzando una ponderación de eficiencia de 78,46%. En primer lugar, influye de sobremanera dos puntos primordiales en cuanto a la parte física y lógica se refiere, la capacidad suficiente de pines tanto analógicos como digitales, así como también las funciones que cada uno de los pines desempeña son imprescindibles para que el prototipo genere escalabilidad y también la integración con la tecnología wifi en su estructura, ya que gracias a ellos proveemos de movilidad al dispositivo en caso de remodelaciones del entorno de trabajo. Por otro lado, el aspecto económico y las capacidades operacionales (voltajes de entrada, entorno de desarrollo, procesamiento, velocidad de transmisión en función de la tecnología de comunicación empleada) se adaptan perfectamente a las necesidades y medidas de prevención que se desean alcanzar.

3.4.4 Selección de sensor para detección de fugas de gas

A continuación, se presenta una comparativa en la Tabla 6-3 entre de los sensores capaces de detectar una fuga de gas GLP acompañadas de sus características principales y parámetros técnicos.

Tabla 6-3: Comparativa entre MQ-2, MQ-5, MQ-6, MQ-9.

	MQ-2	MQ-5	MQ-6	MQ-9
Consumo de Energía	menos de 800mW	menos de 800mW	menos de 750mW	menos de 340mW
Resistencia de Carga	ajustable	ajustable	20KΩ	ajustable
Tensión de Calentamiento	5V±0.1	5V±0.1	5V±0.1	5V±0.1
Temperatura de Operación	-20°C-50°C	-10°C-50°C	-10°C-50°C	-20°C-50°C
Concentración de Detección	200ppm-5000ppm	200-10000ppm	200-10000ppm	500ppm-10000ppm
Costo	\$4	\$4	\$4.5	\$4.5

Realizado por: Jonathan Collantes, 2022

2.4.4.1 Análisis

Con base en la información recolectada de cada uno de los sensores de gas que mejor se podían adaptar al prototipo y con mayor utilidad en el mercado, se establece las ponderaciones de cada sensor en función de los porcentajes del método de escala de Likert señalados en la tabla 2-3 y sus categorías en conformidad con lo establecido en la escala, que dictamina ponderaciones entre 1 y 5.

Tabla 7-3: Niveles de eficiencia para la ponderación y evaluación de características de los sensores para la detección de fugas de gas.

	MQ-2	MQ-5	MQ-6	MQ-9
Consumo de Energía	3	3	3	4
Resistencia de Carga	4	4	3	4
Tensión de Calentamiento	4	4	4	4
Temperatura de Operación	5	4	5	4
Concentración de Detección	5	5	5	3
Costo	5	5	4	4
Puntaje acumulado	26	25	24	23
Ponderación de eficacia	83,67%	86,33%	80,00%	76,67%

Realizado por: Jonathan Collantes, 2022

Cálculos:

$$P_{Acumulado_{MQ-2}} = \sum N_i = 3 + 4 + 4 + 4 + 5 + 5 = 26$$

$$P_{Acumulado_{MQ-5}} = \sum N_i = 3 + 4 + 4 + 5 + 5 + 5 = 25$$

$$P_{Acumulado_{MQ-6}} = \sum N_i = 3 + 3 + 4 + 5 + 5 + 4 = 24$$

$$P_{Acumulado_{MQ-9}} = \sum N_i = 4 + 4 + 4 + 4 + 3 + 4 = 23$$

$$Ponderación\ de\ eficiencia_{MQ-2} = \frac{P_{Acumulado_{MQ-2}}}{T_i} * 100\% = \frac{26}{30} * 100\% = 83,67\%$$

$$Ponderación\ de\ eficiencia_{MQ-5} = \frac{P_{Acumulado_{MQ-5}}}{T_i} * 100\% = \frac{25}{30} * 100\% = 86,33\%$$

$$Ponderación\ de\ eficiencia_{MQ-6} = \frac{P_{Acumulado_{MQ-6}}}{T_i} * 100\% = \frac{24}{30} * 100\% = 80,00\%$$

$$Ponderación\ de\ eficiencia_{MQ-9} = \frac{P_{Acumulado_{MQ-9}}}{T_i} * 100\% = \frac{23}{30} * 100\% = 76,67\%$$

Los cálculos realizados y resumidos en la Tabla 7-3 arrojan como resultado que el sensor para la detección de fuga de gas que mejor se adapta al prototipo que se pretende desarrollar en este

documento es el sensor MQ-2 alcanzando una ponderación de eficiencia de 78,46% en la escala de Likert. Es importante resaltar las características más sobresalientes que presenta, entre ellas y como característica fundamental es la alta sensibilidad al GLP y Gas natural, iniciando su estado de detección en 200 ppm. Otra característica que se resalta es la resistencia de carga ajustable que contiene en su circuito de acondicionamiento y amplificación, esto permite que se adapte a las necesidades en función de los requerimientos del sistema.

3.4.5 Selección de sensor para detección de temperatura en el ambiente

Continuando con la selección de sensores, se presenta una comparativa en la Tabla 8-3 de los sensores capaces de detectar la temperatura en el entorno circundante a la cocina acompañadas de sus características principales y parámetros técnicos.

Tabla 8-3: Comparativa entre KY-026, DHT11, DHT21, DHT22, DS18B20.

	KY-026	DHT11	DHT21	DHT22	DS18B20
Voltaje de operación	3.3V a 5V	3.3V a 5V	3.3V a 5V	3.3V a 5V	3.3V a 5V
Rango de medición	760nm a 1110nm	0°C a 50°C	-40°C a 80°C	-40°C a 150°C	-55°C a 125°C
PIN de salida	Analógico y Digital	Digital	Digital	Digital	Digital
Sensibilidad	-	+/- 2°C	+/- 0.1°C	+/- 0.5°C	+/- 0.5°C
Velocidad de muestreo	-	Cada segundo	Cada 2 segundos	Cada 2 segundos	Cada 750 mili segundos
Carcasa	No	No	Si	No	SI
Costo	\$3	\$3.5	\$7	\$7	\$4

Realizado por: Jonathan Collantes, 2022

3.4.5.1 Análisis

Con base en la información recolectada de cada uno de los sensores para la detección de la temperatura del ambiente de trabajo de una cocina que mejor se podían adaptar al prototipo y con mayor utilidad en el mercado, se establece las ponderaciones de cada sensor en función de los porcentajes del método de escala de Likert señalados en la tabla 2-3 y sus categorías en conformidad con lo establecido en la escala, que dictamina ponderaciones entre 1 y 5.

Tabla 9-3: Niveles de eficiencia para la ponderación y evaluación de características de los sensores para la detección del encendido de hornillas.

	KY-026	DHT11	DHT21	DHT22	DS18B20
Voltaje de operación	5	5	5	5	5
Rango de medición	1	2	3	4	4
PIN de salida	5	4	4	4	4
Sensibilidad	0	3	5	4	4
Velocidad de muestreo	0	5	4	4	5
Carcasa	0	3	5	3	5
Costo	5	5	3	3	4
Puntaje acumulado	16	27	30	27	31
Ponderación de eficacia	45,71%	77,14%	85,71%	77,14%	88,57%

Realizado por: Jonathan Collantes, 2022

Cálculos:

$$P \text{ Acumulado}_{KY-026} = \sum N_i = 5 + 1 + 5 + 0 + 0 + 0 + 5 = 16$$

$$P \text{ Acumulado}_{DHT11} = \sum N_i = 5 + 2 + 4 + 3 + 5 + 3 + 5 = 27$$

$$P \text{ Acumulado}_{DHT21} = \sum N_i = 5 + 4 + 4 + 5 + 4 + 5 + 3 = 30$$

$$P \text{ Acumulado}_{DHT12} = \sum N_i = 5 + 4 + 4 + 4 + 4 + 3 + 3 = 27$$

$$P \text{ Acumulado}_{DS18B20} = \sum N_i = 5 + 4 + 4 + 4 + 4 + 3 + 3 = 31$$

$$Ponderación \text{ de } eficiencia_{KY-026} = \frac{P \text{ Acumulado}_{KY-026}}{T_i} * 100\% = \frac{16}{35} * 100\% = 45,71\%$$

$$Ponderación \text{ de } eficiencia_{DHT11} = \frac{P \text{ Acumulado}_{DHT11}}{T_i} * 100\% = \frac{27}{35} * 100\% = 77,14\%$$

$$Ponderación \text{ de } eficiencia_{DHT21} = \frac{P \text{ Acumulado}_{DHT21}}{T_i} * 100\% = \frac{30}{35} * 100\% = 85,71\%$$

$$Ponderación \text{ de } eficiencia_{DHT22} = \frac{P \text{ Acumulado}_{DHT22}}{T_i} * 100\% = \frac{27}{35} * 100\% = 77,14\%$$

$$Ponderación \text{ de } eficiencia_{DS18B20} = \frac{P \text{ Acumulado}_{DS18B20}}{T_i} * 100\% = \frac{31}{35} * 100\% = 88,57\%$$

Los valores obtenidos a partir de los cálculos realizados y resumidos en la Tabla 9-3 indican que el sensor que mejor se adapta al prototipo para realizar la tarea de la detección de temperatura ambiente en las cercanías de la cocina es el sensor DS18B20 alcanzando una ponderación de eficacia del 88,57% en la escala de Likert. Una característica que es importante de resaltar del

sensor DS18B20 es su sensibilidad de $\pm 0.5^{\circ}\text{C}$ y sobre todo el tiempo de muestreo que es de 750 mili segundos, superior al de los dispositivos bajo estudio. Adicionalmente, posee una carcasa mucho más robusta para trabajar en entornos que requiere de una exposición permanente a diferentes tipos de elementos.

3.4.6 Esquema de Sistema de Cocina IoT

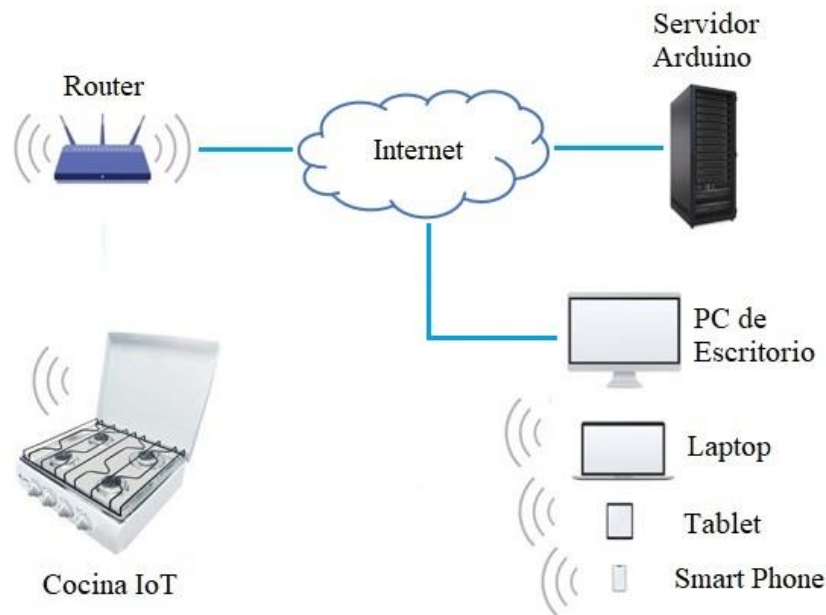


Figura 1-3: Esquema de Sistema de Cocina IoT

Realizado por: Jonathan, Collantes; 2022

La accesibilidad a los datos mediante internet, la necesidad de conocer y controlar el estado de nuestro entorno, orilla a que las tecnologías se adapten a la transmisión y recepción de datos de todo tipo, sobre los objetos interconectados, ya sea de uso cotidiano o de uso industrial, mediante diferentes tecnologías y los servicios con los que constan los dispositivos móviles inteligentes de hoy en día. La amplia gama de sensores y actuadores que se pueden integrar a nuestro alrededor permiten concentrar nuestro hábitat en un solo dispositivo con la finalidad de brindar un valor agregado a los servicios que los objetos o cosas cotidianas pueden ofrecer.

Con base en lo expuesto sobre los beneficios y el alcance potencial que tienen estos sistemas, se presenta una solución a los problemas que se originan en los alrededores de una cocina cotidiana y aportar como medida de prevención de incendios.

El diagrama de bloques ilustrado en la Figura 2-3 expone el proceso de comunicación y de transmisión de los datos desde los dispositivos de censado hasta el entorno de la nube en el que se publica la información de forma amigable con el usuario, haciendo uso de la plataforma Arduino IoT Cloud mediante un smartphone. La tarjeta de desarrollo Node MCU ESP-32S es la encargada de la etapa de control, monitoreo y también de la comunicación inalámbrica que el sistema va a tener con la nube a través de internet, todo esto apoyado en los datos que los sensores MQ-2, HC-SR501 y DS18B20 extraigan del medio en el que serán implantados.

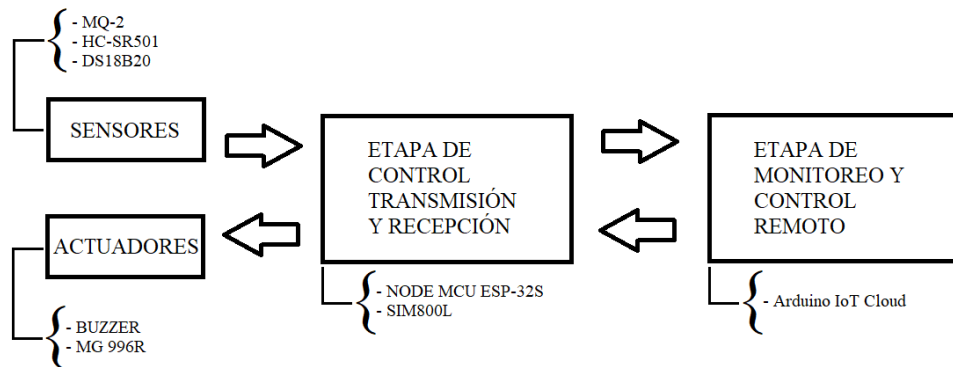


Figura 2-3: Esquema de Bloques del Sistema de Cocina IoT

Realizado por: Jonathan, Collantes; 2022

3.4.7 Esquema del circuito controlador de cocina

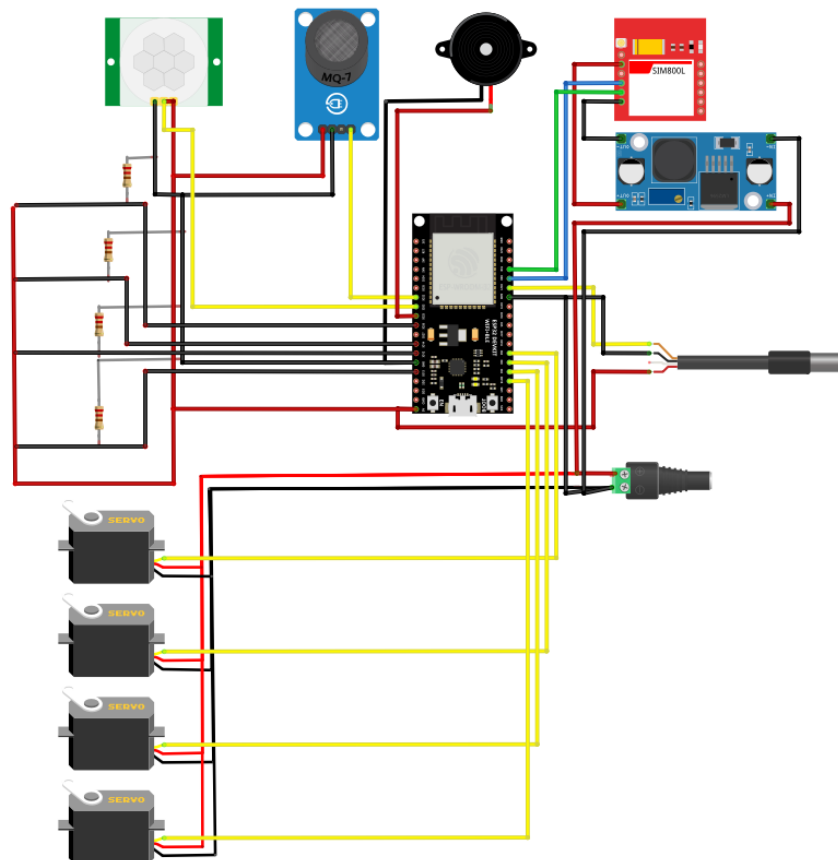


Figura 3-3: Esquema de circuito controlador de cocina

Realizado por: Jonathan, Collantes; 2022

En la Figura 3-3 se expone el diseño esquemático del prototipo para cocinas a gas encargado de la prevención de incendios en entornos domésticos. El diseño consta de una tarjeta de desarrollo ESP-32S encargada de procesar los datos obtenidos a partir de los sensores de gas MQ-2, sensor de proximidad HC-SR501 y el sensor de temperatura DS18B20 para contemplar la temperatura del entorno. La tarjeta de desarrollo también es la encargada de establecer la comunicación con la nube por medio de la tecnología wifi incorporada en su estructura, adicionalmente se encargará de la toma de decisiones para dar paso a los actuadores MG996R para la disminución o cierre de las hornillas y de enviar las alertas mediante correo electrónico como de enviar mensajes de texto a un número configurado previamente.

3.4.8 *Análisis y recopilación de medidas de una cocina tradicional*

A continuación, en la Figura 4-3 se muestra el modelo de cocina que se ha escogido para implementar el prototipo de sistema de seguridad para cocinas a gas. En la Figura 5-3, Figura 6-3 y Figura 7-3 se procede al análisis y desmontaje para la distribución de los distintos sensores y actuadores que se pretende incorporar a un modelo tradicional de cocina y de esta manera convertir un objeto cotidiano en un objeto inteligente.



Figura 4-3: Parte frontal de una cocina tradicional

Realizado por: Jonathan, Collantes; 2022



Figura 5-3: Parte superior de una cocina tradicional

Realizado por: Jonathan, Collantes; 2022



Figura 6-3: Parte inferior interior de una cocina tradicional

Realizado por: Jonathan, Collantes; 2022



Figura 7-3: Parte frontal interior de una cocina tradicional

Realizado por: Jonathan, Collantes; 2022

3.4.9 Ensamblaje de sensores y actuadores a la estructura

En función de las características y medidas de tuberías de gas como de la propia estructura de la cocina, en la Figura 8-3 se procede a la incorporación de una placa para el soporte de los actuadores que en este modelo serán servomotores de la serie MG966R los cuales poseen un torque de 11kg por centímetro, asegurando así que las perillas van a girar correctamente cuando el sistema de control considere conveniente. Dicha placa va a ir de forma oculta detrás de la parte frontal de la cocina, de este modo se facilitará la minimización o cierre de las hornillas en caso de ausencia del usuario.

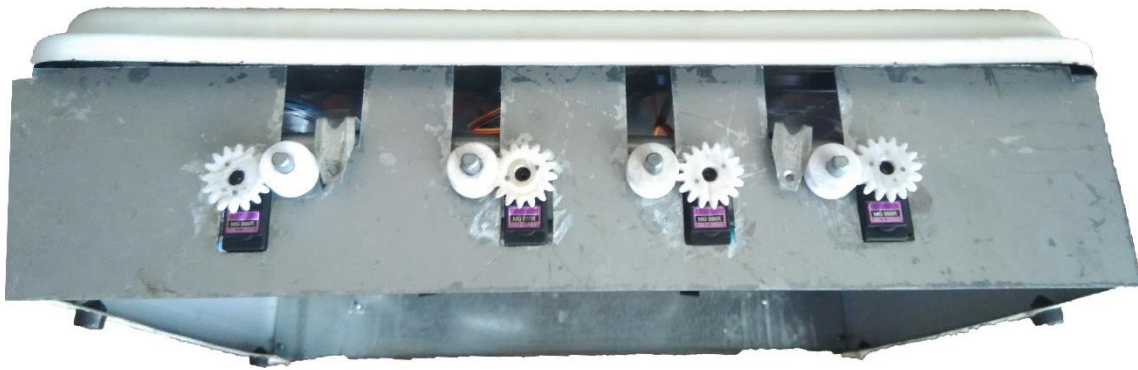


Figura 8-3: Parte frontal con la incorporación de motores y engranajes

Realizado por: Jonathan, Collantes; 2022

Hay que mencionar que se requirió del diseño e impresión 3D de engranajes tipo piñón descritos en la Figura 9-3 y Figura 10-3 que permitieran la transmisión de giro desde los motores hacia las perillas o válvulas que, al girar sobre su propio eje, permiten la apertura y control del paso de gas que va hacia cada hornilla.

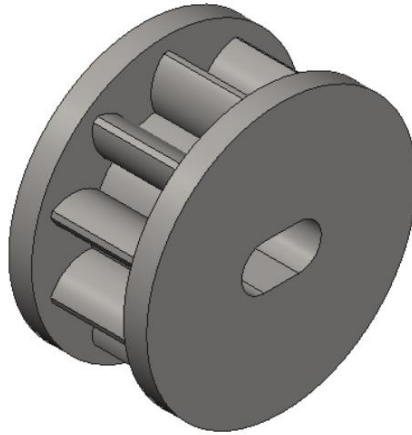


Figura 9-3: Engrane tipo piñón para cada hornilla

Realizado por: Jonathan, Collantes; 2022

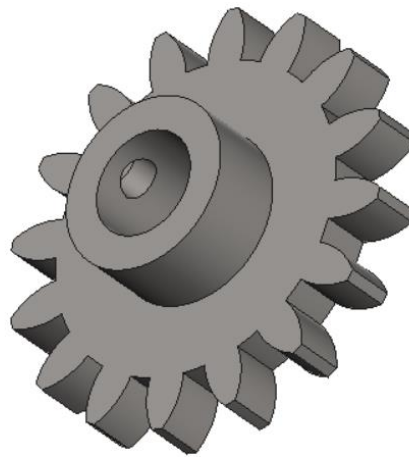


Figura 10-3: Engrane tipo piñón para cada motor

Realizado por: Jonathan, Collantes; 2022

La distribución de los sensores y actuadores de la Figura 11-3 se estableció con base en la distribución de los quemadores o también llamadas hornillas; el sensor de temperatura deberá ir ubicado estratégicamente en la parte superior de la habitación que es en donde se concentra la mayor cantidad de calor en caso de un conato de incendio. Los servomotores por otra parte, fueron ubicados en función de la distancia que se estableció al momento de realizar la inspección previa de la estructura de la cocina, esto arrojó como resultado que 3 de los motores fueran ubicados del lado derecho de la perilla correspondiente y 1 motor del lado izquierdo a la perrilla que le correspondía, todo esto sin interferir con los ángulos de giro con los que se pretende trabajar. El sensor de gas fue remitido al centro de la estructura con el fin de quedar equidistante de cada una de las hornillas para mayor absorción y por ende mejor lectura de su entorno en caso de fuga de gas.

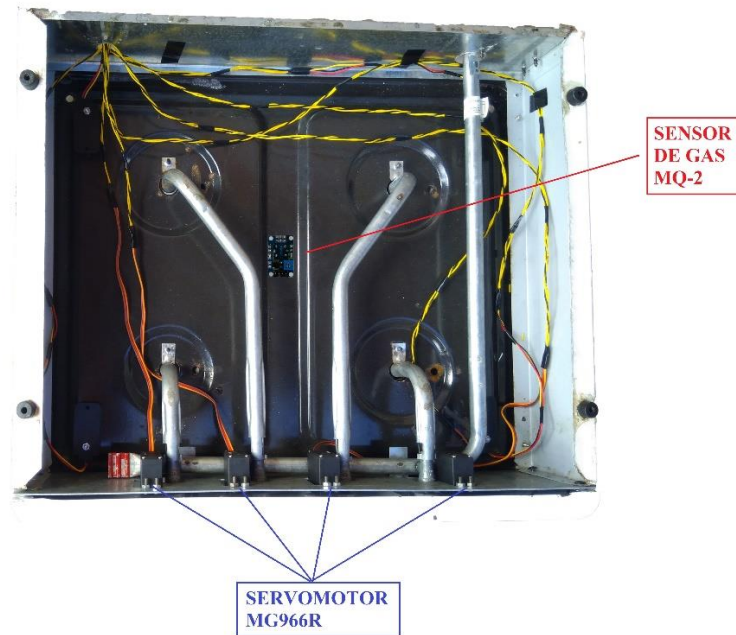


Figura 11-3: Sensores y actuadores de la parte interna de la cocina

Realizado por: Jonathan, Collantes; 2022

Se consideró conveniente que el sensor que va a absorber la información de movimiento sea ubicado en la parte superior de la cocina ilustrado en la Figura 12-3, de esta manera se consigue un mejor ángulo de captura de los datos para cronometrar los tiempos de respuesta de los actuadores en conjunto con las alertas emitidas en caso de que exista un escenario en el que la programación preestablecida considere necesaria.



Figura 12-3: Sensores de la parte superior de la cocina

Realizado por: Jonathan, Collantes; 2022

En la parte posterior de la cocina se logra visualizar la etapa de control y transmisión de los datos. Esta sección involucra un buzzer o zumbador que va a emitir una alerta auditiva en caso de que la etapa de control por medio del sensor MQ-2 hayan detectado una fuga de gas en las zonas circundantes a la cocina. Adicionalmente, la etapa de control disparará dos alarmas o notificaciones hacia el usuario, la primera será un correo con carácter de urgente emitido desde la dirección de correo electrónico que posee previamente la cocina hacia la dirección de correo electrónico que el usuario maneje en su dispositivo móvil, todo esto mediante un enlace vía wifi con la red de la residencia donde se pretenda implementar el sistema y apoyado del servicio que la propia tarjeta de desarrollo Node MCU ESP-32S dispone en su estructura. La segunda alarma o notificación será emitida desde el módulo SIM800L que se encargará de enviar un mensaje de texto desde el número de teléfono asociado al chip que poseerá la cocina hacia el número de teléfono del usuario. Con estas medidas se fortalece la prevención de incendios que se pueda dar en consecuencia de dejar sin supervisión una cocina en pleno funcionamiento.

Hay que considerar que el prototipo de la Figura 13-3 también consta de un sensor encargado de recolectar información de temperatura del ambiente, dicho dispositivo será ubicado de forma conveniente en la ubicación más elevada de la habitación con el fin de informar al usuario final de temperaturas ajenas que un entorno de cocina maneja y así prevenir cualquier siniestro que se pueda presentar.

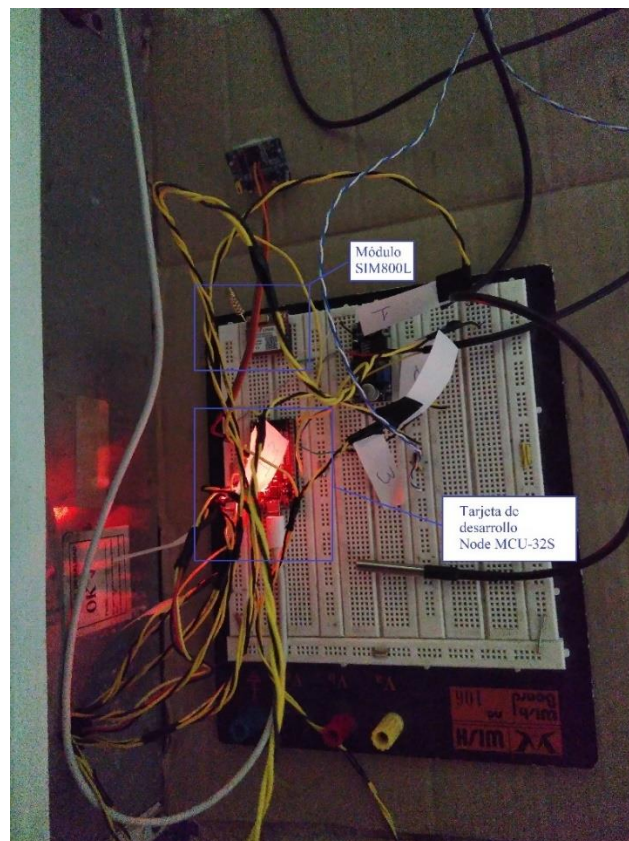


Figura 13-3: Etapa de control y transmisión, parte posterior de la cocina

Realizado por: Jonathan, Collantes; 2022

3.4.10 Creación de cuenta y configuración del sistema en Arduino IoT Cloud

En esta parte, la Figura 14-3 muestra el sistema de Arduino IoT Cloud tal como cualquier otro sistema en la nube, requiere de un correo electrónico con el que se tendrá acceso a la cuenta con la se va a manejar el sistema, así como también de una contraseña para dicho acceso. Es importante mencionar que con esta cuenta se tiene acceso desde un computador para cualquier modificación o configuración y de la misma manera esa cuenta dará acceso al sistema de monitoreo desde la aplicación que se va a instalar al dispositivo móvil.

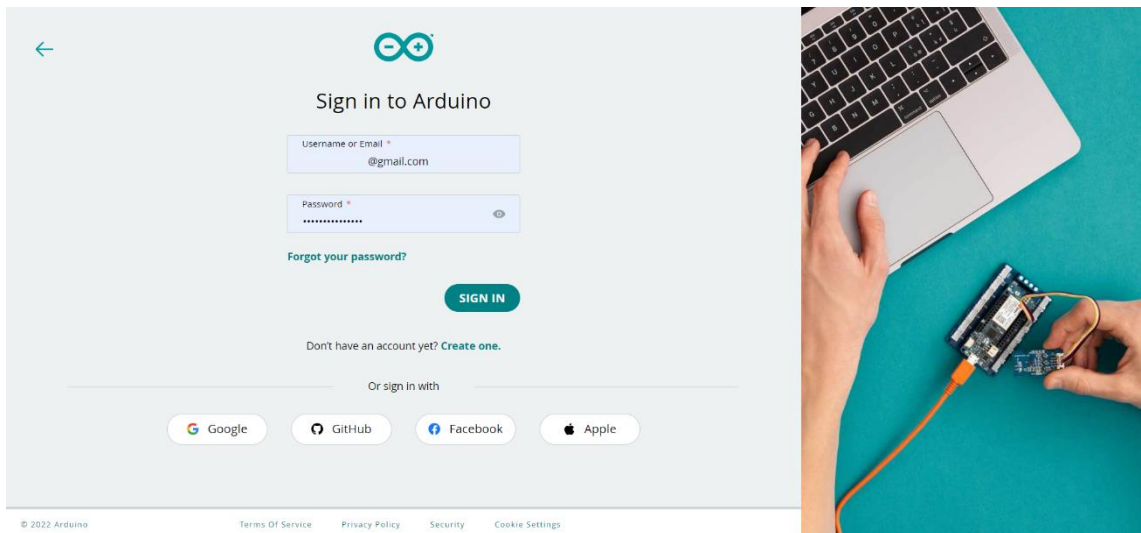


Figura 14-3: Pantalla inicial de acceso a la plataforma de Arduino IoT Cloud

Realizado por: Jonathan, Collantes; 2022

Después de haber creado y autenticado la cuenta se prosigue con los primeros pasos de configuración del sistema descritos en la Figura 15-3. Se dirige a la sección de “*Devices*” y se da clic en el botón llamado “*Add*”, esto va a permitir agregar la tarjeta de desarrollo a panel de la plataforma.

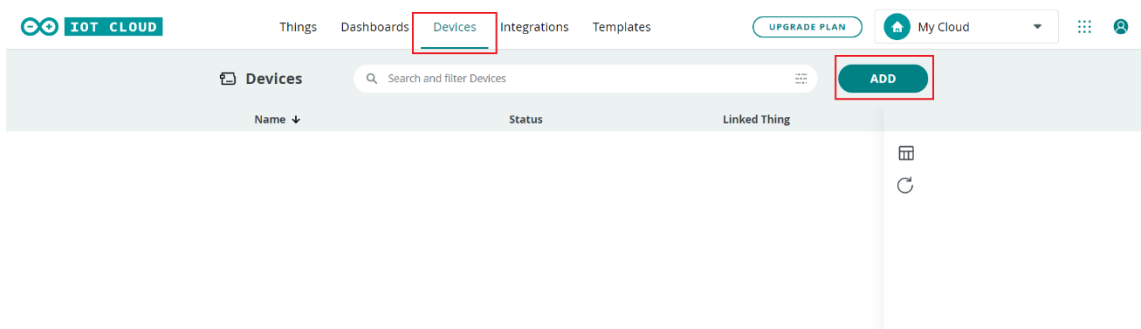


Figura 15-3: Sección de dispositivos en la plataforma Arduino IoT Cloud

Realizado por: Jonathan, Collantes; 2022

En la Figura 16-3 el siguiente paso al interior de la ventana emergente que se despliega es seleccionar la opción “*Set up a 3erd Party device*”, esto permite agregar la tarjeta de desarrollo

de fabricantes que no sean Arduino propiamente y que sean compatibles con los requerimientos que la plataforma demanda.

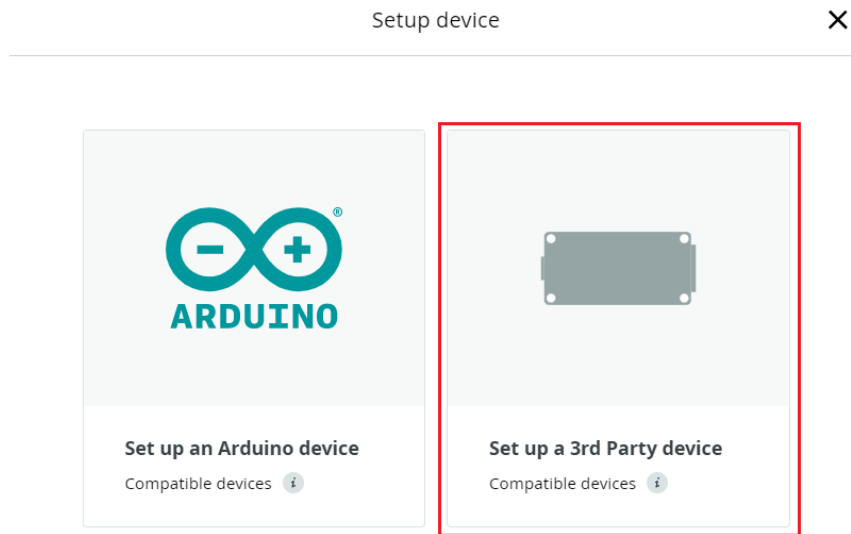


Figura 16-3: Selección de la categoría de dispositivo

Realizado por: Jonathan, Collantes; 2022

A continuación, en la Figura 17-3 sobre la misma pantalla emergente permite elegir entre varias opciones de dispositivos distintos a los de la propia marca de Arduino y que son compatibles con la plataforma Arduino IoT Cloud. Se selecciona la opción de “ESP32” y en el sub menú de la parte inferior se despliega una lista de dispositivos de esa serie, entre los cuales se escoge “NodeMCU-32S”, que es la tarjeta de desarrollo que se ha escogido para la realización de este prototipo; luego se da clic en “CONTINUE” para avanzar con las configuraciones.

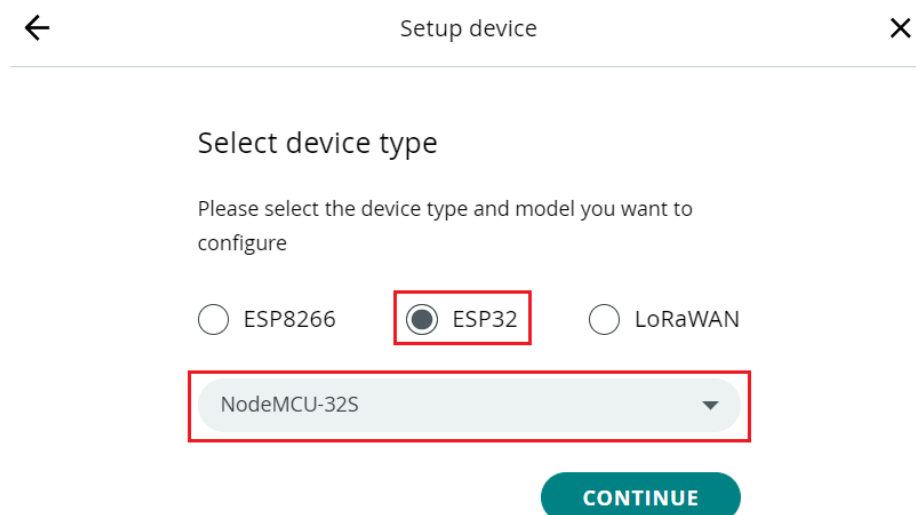


Figura 17-3: Selección de dispositivo a utilizar con Arduino IoT Cloud

Realizado por: Jonathan, Collantes; 2022

Siguiendo con los pasos de configuración se asigna un nombre al dispositivo, este nombre va a ser la identificación de la tarjeta de memoria al interior de la plataforma, en este caso llamaremos a la tarjeta por su nombre “*NodeMCU32-S*” ilustrado en la Figura 18-3, paso siguiente dar clic en “*NEXT*”.

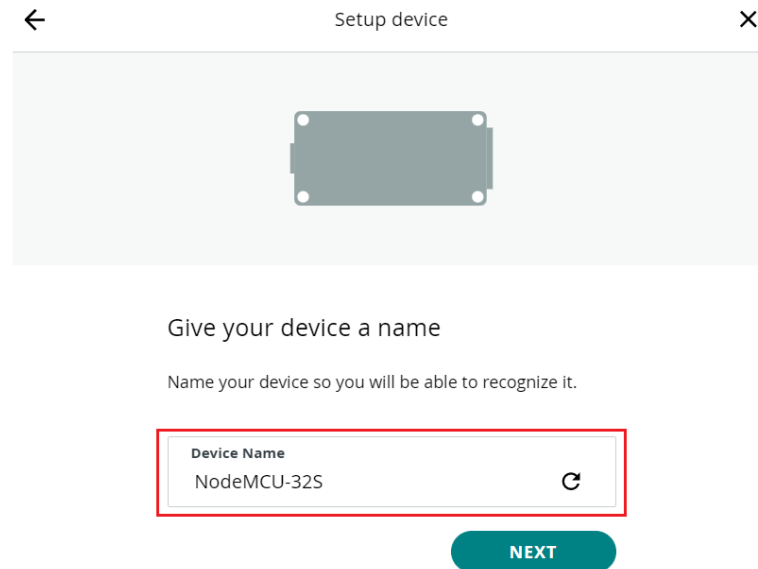


Figura 18-3: Asignación de nombre al dispositivo

Realizado por: Jonathan, Collantes; 2022

Como último paso en la sección de dispositivo de la Figura 19-3, se prepara la tarjeta de desarrollo con la que se va a trabajar para interactuar con el entorno IoT. Para tener acceso desde el dispositivo Node MCU-32S, se requiere un ID de dispositivo y una clave secreta provista en este último paso y que se requerirá más adelante. Paso seguido dar clic en “*NEXT*”.

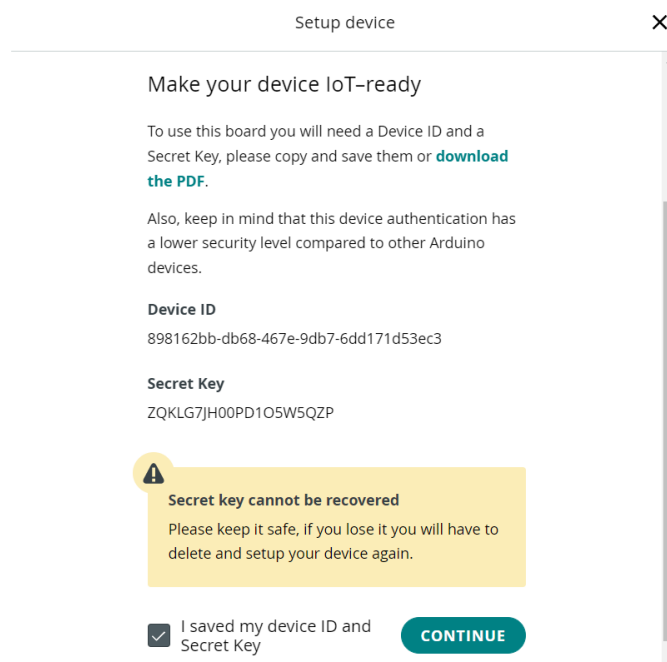


Figura 19-3: Asignación de credenciales de acceso

Realizado por: Jonathan, Collantes; 2022

A continuación, en la Figura 20-3 aparecerá el dispositivo ya creado y con información de que la tarjeta de desarrollo se encuentra fuera de línea debido a que aún no se le ha cargado en código correspondiente.

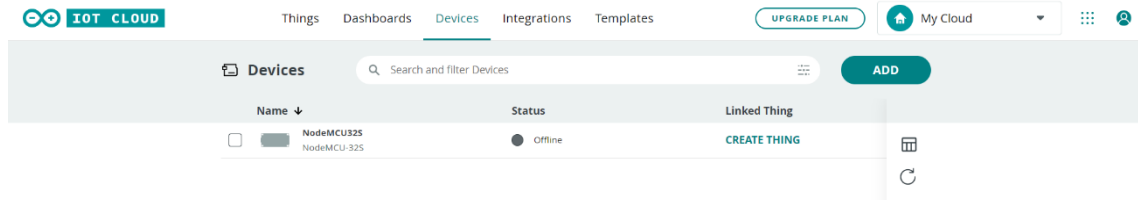


Figura 20-3: Tarjeta de desarrollo asignada satisfactoriamente con Arduino IoT Cloud

Realizado por: Jonathan, Collantes; 2022

En la siguiente parte, dentro de la Figura 21-3 hay que dirigirse a la sección “Things” y dar clic en “CREATE”, esto permitirá crear todas las variables virtuales en función de los requerimientos del sistema.



Figura 21-3: Panel para creación de variables virtuales

Realizado por: Jonathan, Collantes; 2022

En la Figura 22-3 se despliega varios bonotes que se deben configurar. La primera parte es asignar un nombre a este panel el cual se designa como “Variables del Sistema”. El botón de “Select Device” permite conectar las variables virtuales que se pretende configurar después, con la tarjeta de desarrollo que se configuró previamente en la plataforma. Adicionalmente, en esta sección existe el botón para configurar la red a la cual se va a conectar la tarjeta de desarrollo.

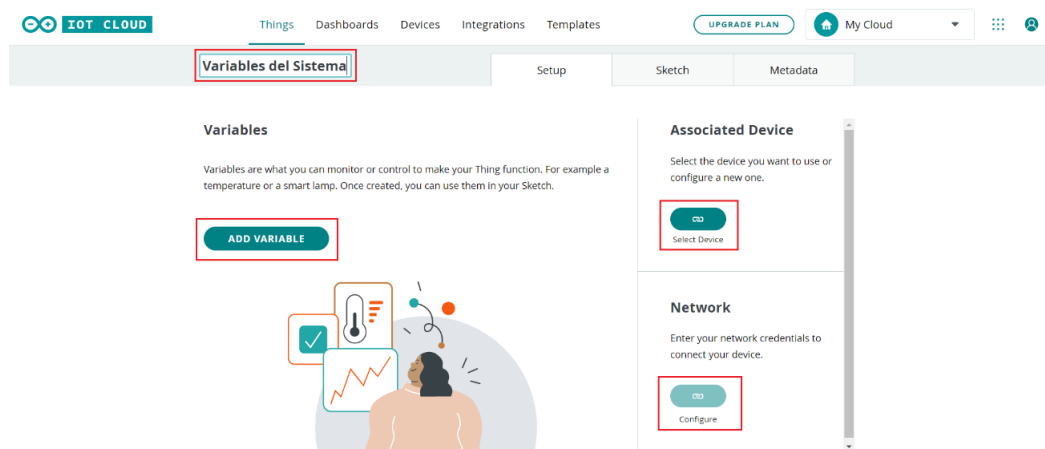


Figura 22-3: Panel para configuración de variables virtuales, dispositivos y red

Realizado por: Jonathan, Collantes; 2022

Al ir al botón de “*Select Device*” de la Figura 23-3, se presenta la siguiente pantalla en la que se asocian las variables virtuales que se va a crear con la tarjeta de desarrollo. Se da clic en “*ASSOCIATE*” y automáticamente quedan enlazados.

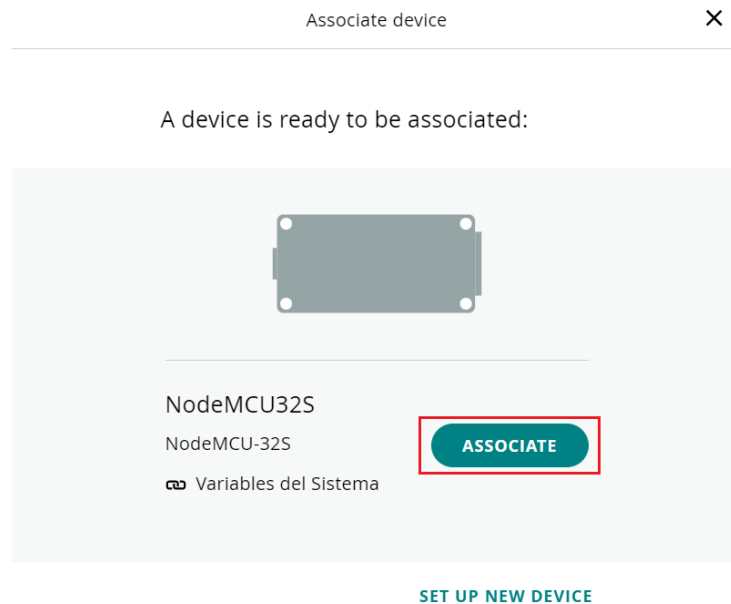


Figura 23-3: Enlace de variables virtuales con tarjeta de desarrollo

Realizado por: Jonathan, Collantes; 2022

En el botón “*Configure*” del apartado “*NETWORK*” en la Figura 24-3, se tiene que ingresar el nombre de la red wifi a la cual se va a conectar el dispositivo, la contraseña de dicha red y también el “*Secret Key*” que es la información que previamente se obtuvo al configurar la tarjeta de desarrollo en el entorno de la plataforma. Se da clic en “*SAVE*” y quedará configurada.

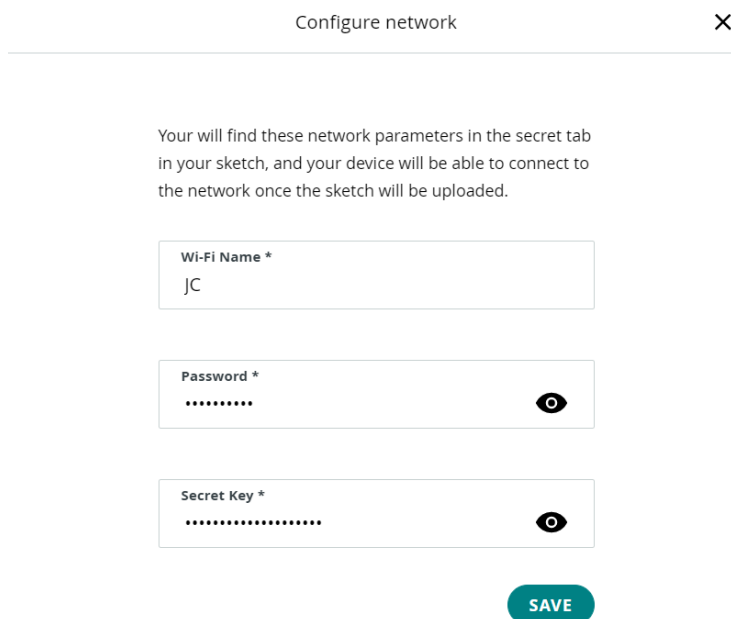
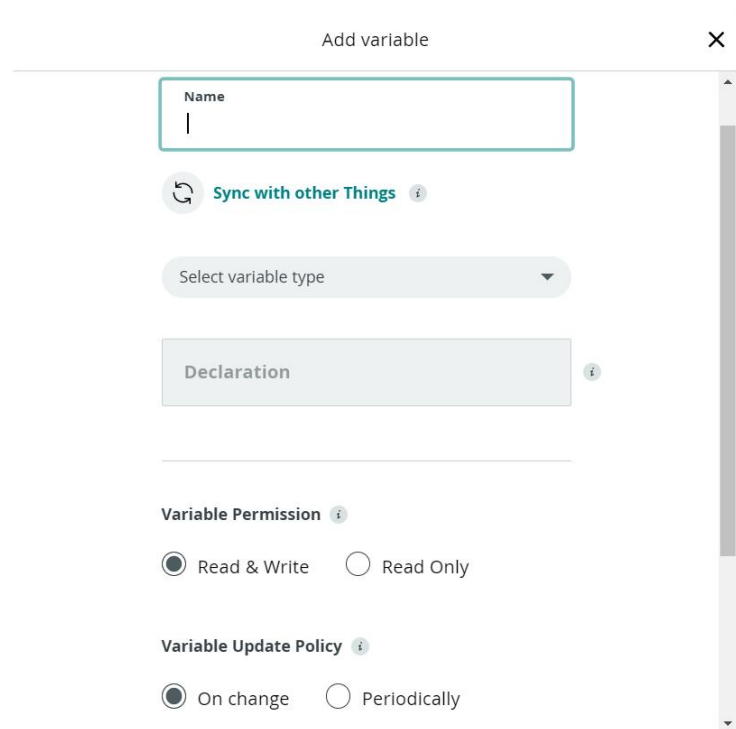


Figura 24-3: Enlace de variables virtuales con tarjeta de desarrollo

Realizado por: Jonathan, Collantes; 2022

En el boton “*ADD VARIABLE*” de la Figura 25-3, es en donde se creó cada una de la variables virtuales que el sistema de seguridad va a requerir para su correcto funcionamiento. Cada variable tiene características específicas que deben ser seleccionadas, entre ellas el nombre, el tipo de variable que se pretende ingresar, si la variable va a ser de escritura y lectura o solo lectura, así como también si en la variable se van a actualizar sus datos de forma periódica o cuando se registre un cambio, de ser el caso de periodicidad se deberá escoger el tiempo de muestreo. Al finalizar se añade la variable y se procede a la configuración de la siguiente variable.



The image shows a web-based configuration form titled "Add variable" with a close button (X) in the top right corner. The form contains several sections:

- Name:** A text input field with a vertical cursor.
- Sync with other Things:** A button with a circular refresh icon and an information icon (i).
- Select variable type:** A dropdown menu.
- Declaration:** A large text area with an information icon (i).
- Variable Permission:** Two radio button options: "Read & Write" (selected) and "Read Only".
- Variable Update Policy:** Two radio button options: "On change" (selected) and "Periodically".

Figura 25-3: Configuración individual de variables

Realizado por: Jonathan, Collantes; 2022

Al terminar el proceso de configuración en la sección de las variables, se obtiene el siguiente entorno con el que se va a trabajar ilustrado en la Figura 26-3. La variable “*G*” se encargará de recolectar la información proveniente de cualquier fuga de gas suscitada, la variable “*H*” se encargará de informar cuando se ha dejado desatendida la cocina en pleno funcionamiento, y la variable “*T*” es aquella que recolecta la temperatura ambiente.

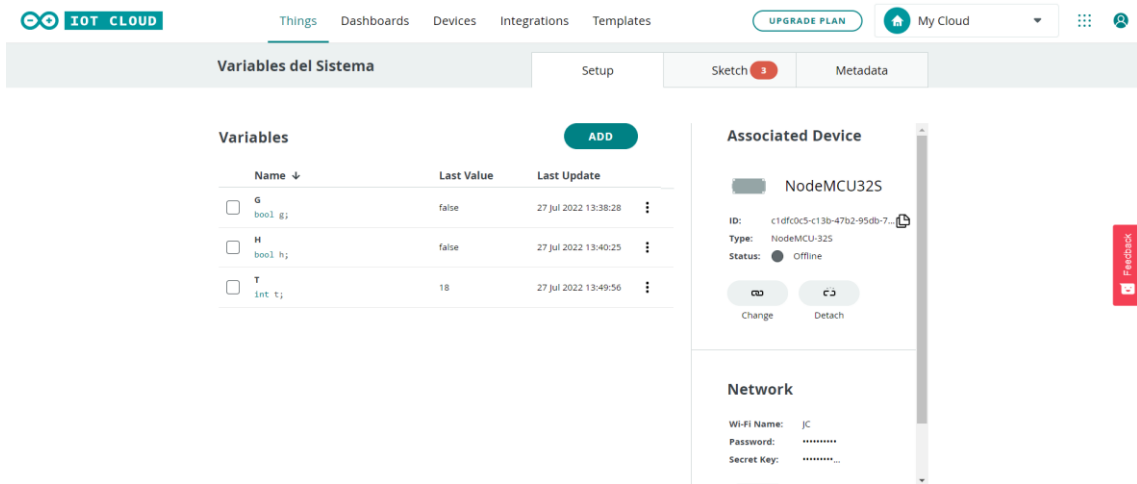


Figura 26-3: Configuración completa de la sección de variables

Realizado por: Jonathan, Collantes; 2022

En la sección de “Dashboards” de la Figura 27-3, es donde se va a crear una especie de panel el que se encuentran muchas herramientas para visualizar y controlar las variables que sean creadas. Se da clic sobre el botón “CREATE” y se empieza la configuración.

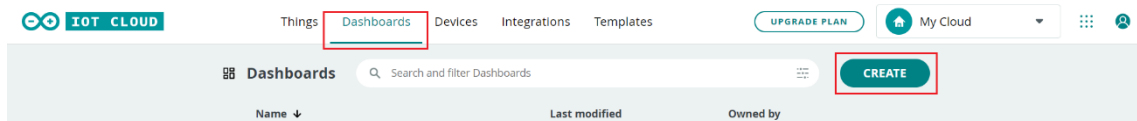


Figura 27-3: Configuración panel de dashboards

Realizado por: Jonathan, Collantes; 2022

Se inicia asignando un nombre al conjunto de elementos que van a permitir visualizar las variables virtuales previamente creadas, en este caso se asigna el nombre de “Sistema de seguridad” ilustrado en la Figura 28-3. A continuación, se da clic en “ADD” para añadir de uno en uno el elemento que se adapte mejor con la variable que se pretende visualizar.

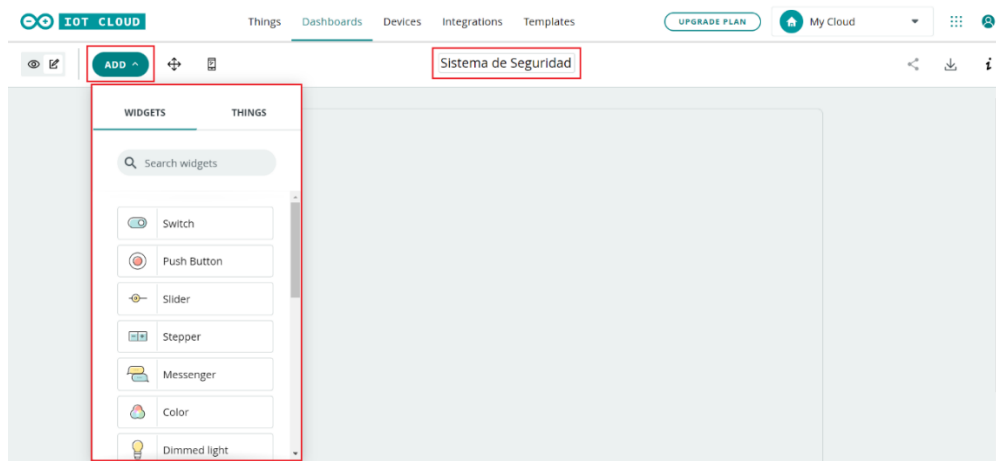


Figura 28-3: Asignación de elementos a panel de dashboards

Realizado por: Jonathan, Collantes; 2022

Al momento de escoger un elemento también llamado “WIDGET”, se ingresará a la configuración del mismo. Esto implica que se tiene que enlazar dicho elemento con la variable que se desea asignar para ser visualizada. En la Figura 29-3 se asigna un nombre para el elemento, se tiene que dar clic en “Link Variable” para el enlace con la variable virtual y como paso final dar clic en “DONE”.

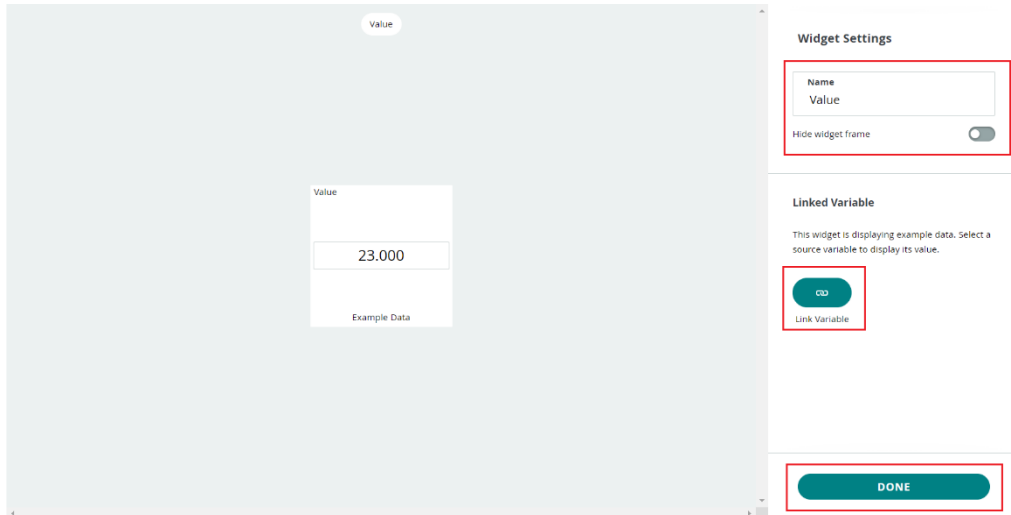


Figura 29-3: Configuración de Widgets

Realizado por: Jonathan, Collantes; 2022

Para el prototipo de sistema de seguridad se plantea el conjunto de widgets de la Figura 30-3 para el monitoreo en tiempo real de las variables virtuales encargadas de visualizar lo que esté pasando con la cocina en todo momento y desde cualquier lugar.

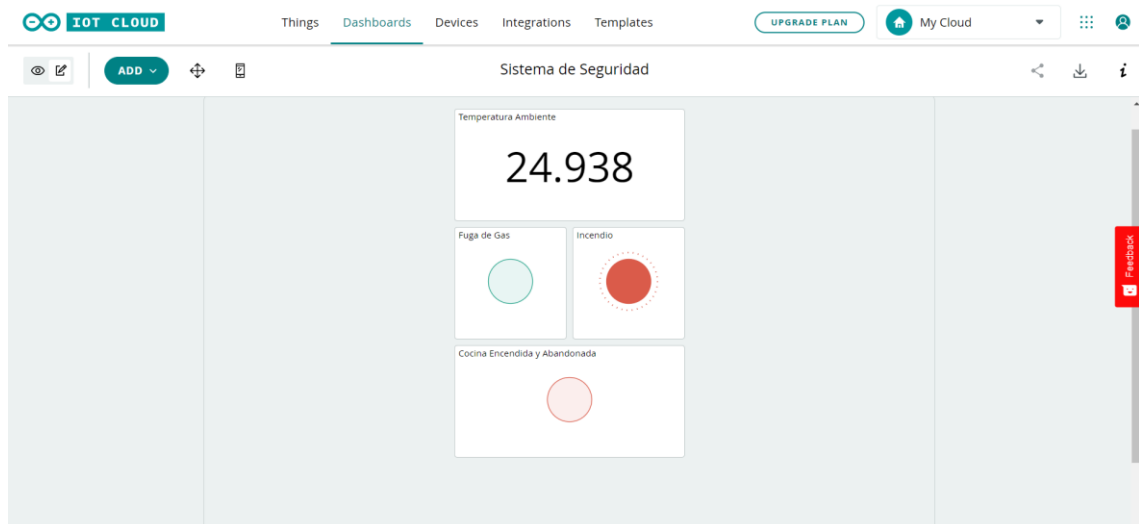


Figura 30-3: Conjunto de Widgets destinados al prototipo de sistema de seguridad

Realizado por: Jonathan, Collantes; 2022

Una de las ventajas predominantes que tiene la plataforma de Arduino IoT Cloud por sobre otras plataformas es que, permite en su misma plataforma modificar el código que se pretende asignar

o cargar a la tarjeta de desarrollo, así como también el monitoreo serial del mismo ilustrado en la Figura 31-3. Este entorno trabaja mediante código abierto basado en lenguaje C++, tal como la versión del software Arduino IDE. Es importante aclarar que las variables virtuales que fueron configuradas anteriormente, ya no necesitan ser declaradas porque el propio entorno de programación las tiene precargadas al abrirlo.

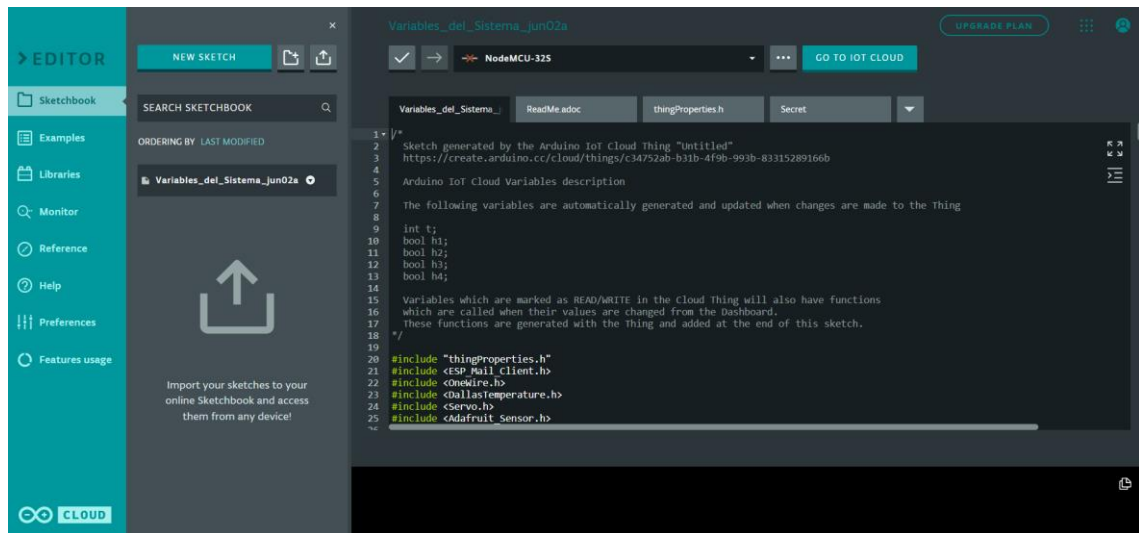


Figura 31-3: Editor de código y monitor serial

Realizado por: Jonathan, Collantes; 2022

El conjunto de librerías que se ha seleccionado para la lectura de los valores de los sensores MQ-2 encargado de obtener la información de concentración de gas en el ambiente, HC-SR501 encargado de obtener la información de la presencia del usuario en las cercanías de la cocina y DS18B20 encargado de obtener la temperatura del ambiente en la habitación; así como las librerías para escritura sobre los actuadores MG996R encargados de la minimización o cierre del flujo de gas mediante la manipulación de las perillas se señalan a continuación:

- `#include "thingProperties.h"`, permite enlazar la tarjeta de desarrollo con las propiedades configuradas en la plataforma de Arduino IoT Cloud
- `#include <ESP_Mail_Client.h>`, es una librería destinada a enviar, leer y recibir notificaciones de correo electrónico entrante y saliente para dispositivos ESP32, ESP8266 y SAMD21.
- `#include <OneWire.h>`, permite el acceso a sensores de temperatura de 1 cable, memoria y otros chips.
- `#include <DallasTemperature.h>`, permite obtener la información compatible con DS18B20, DS18S20, DS1822, DS1820.
- `#include <Servo.h>`, permite que la placa Node MCU32-S controle los servomotores haciendo un uso cuidadoso de los temporizadores.

- #include <Adafruit_Sensor.h>, es una capa de abstracción de sensor unificada utilizada por muchas bibliotecas de sensores de Adafruit y trabaja en conjunto con la librería DHT.

3.4.11 Servicio MQTT provistos por la plataforma Arduino IoT Cloud

Arduino IoT Cloud posee como base para la transferencia de datos entre la tarjeta de desarrollo y la plataforma el protocolo MQTT, dicho protocolo es utilizado muy a menudo por sistema IoT por ser muy ligero y no requerir de un gran ancho de banda. Al contener en su comunicación información con cantidad de datos muy pequeña proveniente del ente publicador (NodeMCU ESP-32S), hace que del lado del suscriptor (Arduino IoT Cloud) dicha información llegue de forma rápida y gracias los identificadores de acceso no solo para la tarjeta de desarrollo sino también para cada una de las variables hace de este sistema forma confiable de interactuar bidireccionalmente.

3.4.11.1 Arquitectura

Cliente. - La tarjeta de desarrollo Node MCU ESP-32S pasa a tomar el rol del cliente la cual se conecta a los servidores de la plataforma que cumplen el papel de agente MQTT en este intercambio de datos. La etapa de control se conecta con la plataforma que actúa de Broker MQTT en este protocolo y de esta manera se publican la información del entorno de la cocina recolectada por los sensores.

Broker. - En este contexto, los servidores en donde se almacena la información recolectada por las cuentas de los suscriptores, radican en el enrutamiento de los datos al destino que corresponda dicha información.

Topic. – No es más que la codificación de caracteres por medio de una cadena UTF-8 lo que permite que los paquetes se puedan filtrar eficientemente a cada cliente.

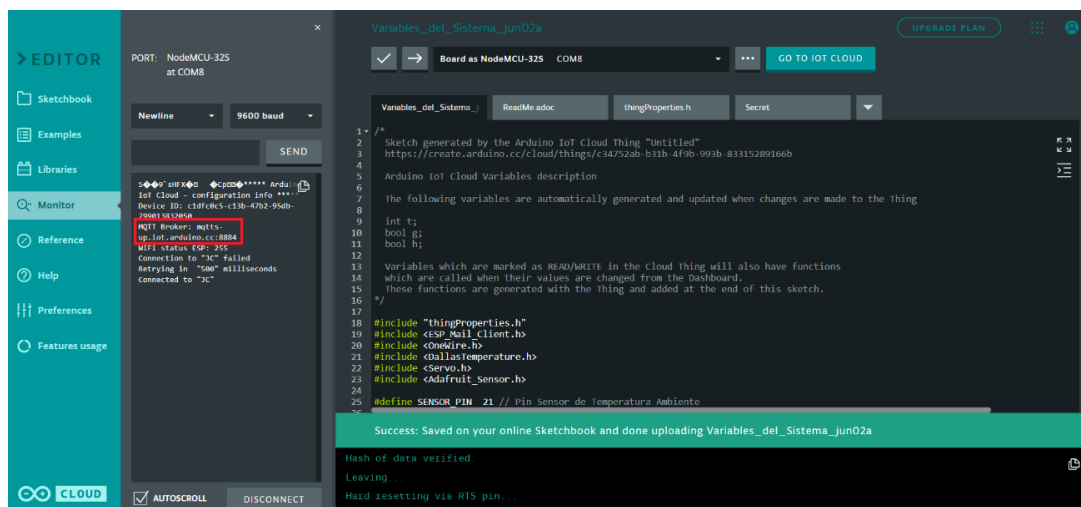


Figura 32-3: Broker MQTT

Realizado por: Jonathan, Collantes; 2022

Se puede visualizar en la Figura 32-3 que los datos publicados por el cliente (Node MCU ESP32-S) apuntan a la dirección del Broker MQTT, es decir, a los servidores de la plataforma de Arduino Cloud. Adicionalmente el cliente asigna un puerto de salida que es el puerto 8884, dicho puerto es utilizado para el protocolo TCP lo que hace que se garantice la entrega de paquetes a la plataforma.

3.4.11.2 Características

Autenticación. – El protocolo MQTT asigna un identificador a cada cliente que desea publicar sus datos en la plataforma, esta información (usuario y contraseña) se requerirá para para cotejar con la información contenida en la base de datos del Broker.

Control de Acceso. – En esta sección el protocolo determina el topic al que le corresponde signar la información del usuario.

QoS. – Mediante los 3 niveles de calidad de servicio que posee ese protocolo, se garantiza que los datos que van a ser publicados lleguen a destino con la seguridad del caso.

Token. – Al publicar la información a la plataforma, el protocolo MQTT instaurado en los servidores, generan tokens únicos de extremo a extremo para cada una de las variables digitales, esto hace que en el caso de existir intrusiones de terceros no pueda acceder a los datos provenientes de sistema de seguridad.

CAPÍTULO IV

4. ANÁLISIS DE RESULTADOS

Como parte complementaria, este capítulo se va a centrar en exponer los resultados obtenidos con base en los valores recolectados de las mediciones que el prototipo de sistema de seguridad en cocinas a gas capturó en un tiempo determinado.

4.1 Evaluación del sistema de seguridad

Al finalizar el montaje de todos los dispositivos tanto sensores como actuadores en la cocina que se tomó como referencia para el desarrollo del presente documento, se procedió en segunda instancia a la estimulación de los parámetros que pueden influir en el inicio de un incendio para capturar las distintas mediciones que los sensores iban a brindar al igual que la respuesta de los actuadores en dicho momento.

Evaluación de Conectividad

Se establece que para este apartado se opte por la utilización de dos servicios que la misma plataforma posee para identificar la conexión de prototipo con la red wifi configurada y también la conexión con la plataforma Arduino IoT Cloud.

Evaluación de Estabilidad

Se plantearon tres escenarios en los que el prototipo debía interactuar eficientemente, el primer caso obteniendo información de un entorno bajo condiciones normales y seguras, el segundo escenario bajo condiciones de trabajo con períodos de abandono de la cocina, y el tercer caso en condiciones de alta peligrosidad.

El primer escenario en el que se va a evaluar el prototipo de sistema de seguridad será bajo condiciones normales y seguras, esto denota que no existirá presencia del usuario en las cercanías de la cocina y que de la misma manera no exista uso de la cocina durante ese período de tiempo. El segundo caso se establece condiciones de trabajo de la cocina con períodos de tiempo en el que el usuario estará presente sujeto a los parámetros establecidos de seguridad y prevención que se debe tener, y con períodos de tiempo en el que el usuario abandone de manera descuidada una cocina en pleno uso y que el sistema alerte al usuario del abandono.

El tercer caso de estudio fue generar una fuga de gas en el entorno de la cocina propiciado por una o varias hornillas, esto con el fin de establecer un posible conato de incendio o explosión y que el sistema reaccione conteniendo la posible fuga y alertando al usuario de lo sucedido.

Evaluación de nivel de error

En esta sección se pretende evaluar el sistema de seguridad en condiciones en las cuales cada uno de los sensores con sus valores obtenidos sean lo más estables y constantes posible para de esta manera delimitar cualquier anomalía en proceso de prevención y contención de posibles escenarios de riesgo que terminen en incendios.

Evaluación de recurrencia bajo condiciones de riesgo

El uso que se tiene de una cocina en un solo día constituye repetidas ocasiones, eso permite que se tomen muestras aleatorias del comportamiento del sistema bajo condiciones de uso normal y deliberado de la cocina.

Evaluación de notificaciones

Es fundamental que el usuario final responsable del monitoreo de la cocina se alertado oportunamente de cualquier anomalía en el entorno de trabajo de la cocina, esto enfocado en la prevención de cualquier conato de incendio. Para ello se establecieron tres tipos de alertas que notifiquen de manera eficaz fugas de gas o abandono de una cocina que ha sido encendida previamente.

Evaluación económica del sistema

Al finalizar de este proceso, es realizar el balance económico que el prototipo generó en su totalidad, contemplando cada uno de los elementos y dispositivos que fueron integrados en la cocina que se tomó como referencia para este sistema de seguridad.

4.1.1 Conectividad

Para evaluar que la tarjeta de desarrollo esté correctamente enlazada no solo con la red wifi pre configurada sino también que dicha placa se encuentre conectada con el entorno en la plataforma, se establecen dos procedimientos a seguir.

4.1.1.1 Conectividad entre la tarjeta de desarrollo y la red wifi

En la Figura 1-4 se establece como requisito para esta prueba la conexión directa de la tarjeta de desarrollo con el computador en el que se esté manejando la configuración. Se instituye una comunicación serial y se levanta el servicio de monitoreo serial que la plataforma Arduino IoT Cloud ofrece.

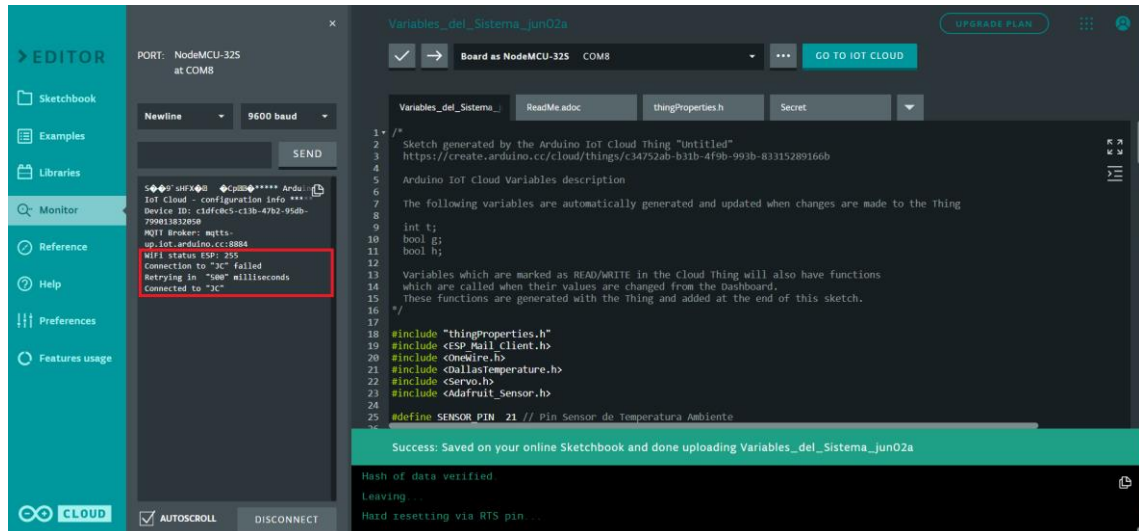


Figura 1-4: Monitor serial y prueba de conectividad

Realizado por: Jonathan, Collantes; 2022

En la Figura 1-4 el primer valor obtenido dentro de la sección de conexión con la red es “*WiFi status ESP: 255*” lo que denota que la red pre configurada a la que se pretende conectar la tarjeta de desarrollo no presenta ningún impedimento para el establecimiento de una conexión. Como segundo código a interpretar es “*Connection to “JC” failed*”, esto indica que a pesar de que la red permite el acceso, la tarjeta no pudo establecer la conexión, pero esto no es problema porque cada 500 mili segundos la tarjeta de desarrollo enviará una solicitud al Router para establecer una comunicación. A continuación, en la captura de pantalla se visualiza “*Connected to “JC”*” lo que efectivamente comprueba que se ha establecida una comunicación exitosa entre la tarjeta de desarrollo y el Router. Esto también se puede fácilmente corroborar realizando un escaneo en la red y verificando que ha sido asignada una dirección IP dentro de la red al dispositivo ESP32S ilustrado en la Figura 2-4.

Dispositivos conectados: 1 de 8

Nombre del dispositivo	Dirección IP	Dirección física (MAC)
esp32-arduino	192.168.137.202	0c:b8:15:c4:86:c0

Figura 2-4: Escaneo de dispositivos en la red wifi

Realizado por: Jonathan, Collantes; 2022

4.1.1.2 Conectividad entre la tarjeta de desarrollo y Arduino IoT Cloud

Como segunda prueba y no menos importante en la Figura 3-4 se verifica que la tarjeta de desarrollo no solo esté conectada a una red wifi, sino que también se encuentra conectada a la plataforma Arduino IoT Cloud. Para acceder a esta información se ingresa a la plataforma y se dirige al apartado denominado “DEVICES”.

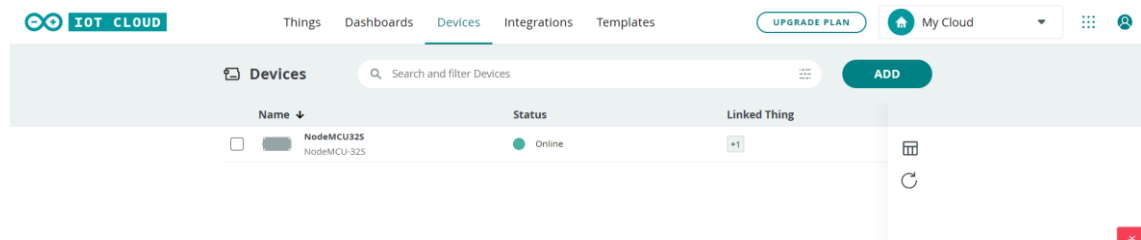


Figura 3-4: Verificación de estatus del dispositivo en la plataforma

Realizado por: Jonathan, Collantes; 2022

En esta sección de la plataforma se puede encontrar el dispositivo que fue configurado previamente (NodeMCU32S) y la información acerca del estado, encontrando como resultado que dicho dispositivo se encuentra conectado satisfactoriamente a la plataforma con las palabras “Online”.

4.1.1.3 Envío de paquetes a la plataforma de Arduino IoT Cloud

Se procede a enviar información hacia la plataforma de Arduino IoT Cloud tomando como referencia el sensor de temperatura y generando valores a través de estímulos artificialmente por motivos de prueba en la variación de temperatura ilustrado en la Figura 5-4.

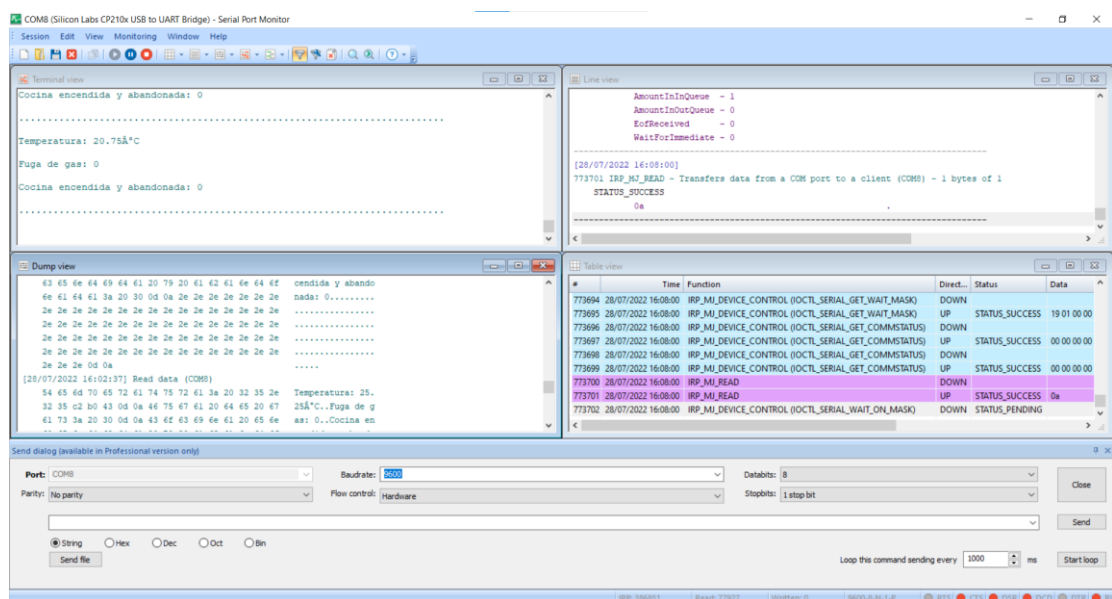


Figura 4-4: Envío de paquetes visualizado en Serial Port Monitor

Realizado por: Jonathan, Collantes; 2022

Se dio seguimiento a valores obtenidos mediante el sensor de temperatura con la ayuda de la herramienta “*Serial Port Monitor*” ilustrado en la Figura 4-4, se logra apreciar exactamente los valores que fueron enviados hacia la plataforma en un período determinado de tiempo. Las variaciones abruptas en la temperatura ambiente fueron valores que se estimularon intencionalmente a modo de prueba y fueron visualizadas exitosamente en una gráfica que se incluyó brevemente para la prueba de recepción de la información en la plataforma desde el sensor, todo esto ilustrado en la Figura 5-4.

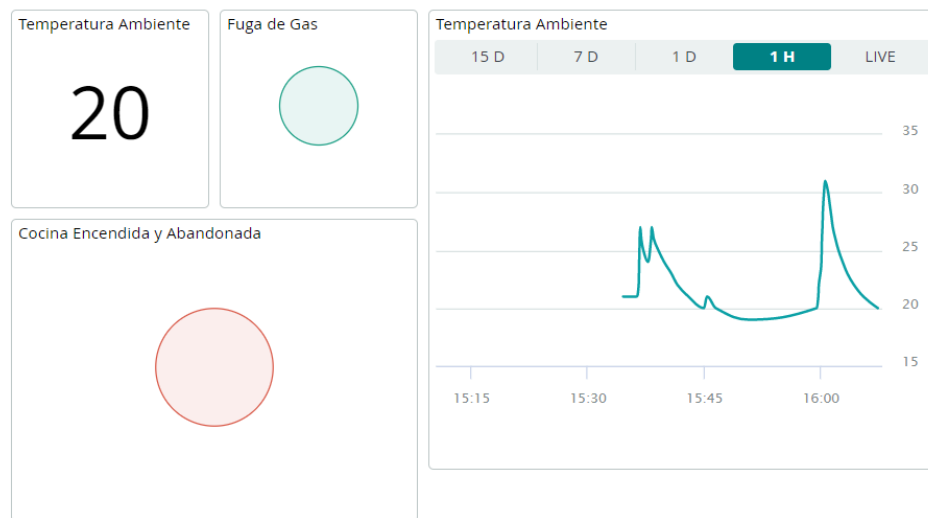


Figura 5-4: Recepción de paquetes visualizado en la plataforma Arduino IoT Cloud

Realizado por: Jonathan, Collantes; 2022

4.1.1.4 Recepción de paquetes en la aplicación móvil IoT Remote

La aplicación IoT Remote es una extensión de la plataforma Arduino IoT Cloud para dispositivos móviles. Esto provee de una herramienta eficaz para el monitoreo en tiempo real de los valores censados por el sistema de seguridad en cocinas a gas. A continuación, se logra apreciar que efectivamente los valores emitidos por la tarjeta de desarrollo que fueron monitoreados desde su envío y que de la misma manera fueron recibidos y visualizados en la plataforma, también fueron recibidos por la aplicación móvil de forma satisfactoria. Se puede observar las variaciones en la temperatura que bajos estímulos fueron captadas por el sensor lo que se aprecia en la Figura 6-4.

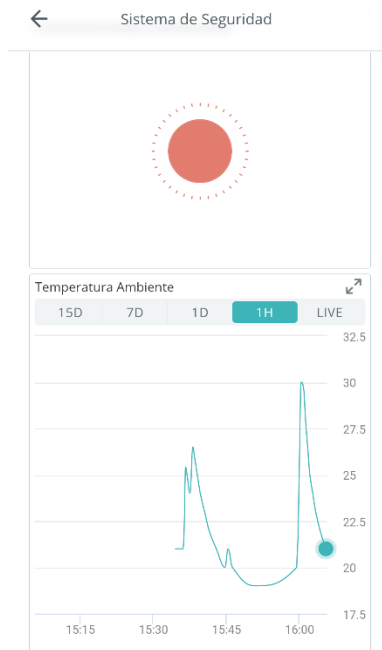


Figura 6-4: Recepción de paquetes visualizado en la aplicación móvil IoT Remote

Realizado por: Jonathan, Collantes; 2022

4.1.2 Estabilidad

Se establecen distintos escenarios en los que el sistema debe comportarse de manera determinada en cada caso. La visualización de estos escenarios se verá reflejados en las cifras recolectadas por los sensores y publicadas en el entorno de monitoreo. Se toma como plan de referencias 20 muestras por cada escenario con intervalos de tiempo de 10 segundos entre cada valor obtenido de las variables virtuales pre establecidas y configuradas.

4.1.2.1 Estabilidad en condiciones normales

Para la primera etapa de recolección, la cocina permaneció en todo momento con las hornillas apagadas y sin la presencia del usuario, todas esas pruebas se describen en la Tabla 1-4. Esto se verifica gracias a que las variables virtuales relacionadas con “fuga de gas” y “Cocina Encendida y Abandonada” solo han generado valores de “false”. Hay que tomar como referencia que en promedio la temperatura ambiente en condiciones normales oscila en los 20°C.

Tabla 1-4: Recolección de datos de variables virtuales en condiciones normales

Fecha	Hora	Temperatura	Fuga de Gas	Cocina Encendida y Abandonada
29/7/2022	21:34:10.714	20,50	false	false
29/7/2022	21:34:21.748	20,50	false	false
29/7/2022	21:34:32.882	20,50	false	false
29/7/2022	21:34:43.926	20,50	false	false
29/7/2022	21:34:55.063	20,50	false	false

29/7/2022	21:35:06.098	20,50	false	false
29/7/2022	21:35:17.15	20,50	false	false
29/7/2022	21:35:28.277	20,50	false	false
29/7/2022	21:35:39.522	20,50	false	false
29/7/2022	21:35:50.661	20,50	false	false
29/7/2022	21:36:01.698	20,44	false	false
29/7/2022	21:36:12.735	20,44	false	false
29/7/2022	21:36:24.087	20,44	false	false
29/7/2022	21:36:35.32	20,44	false	false
29/7/2022	21:36:46.354	20,44	false	false
29/7/2022	21:36:57.407	20,44	false	false
29/7/2022	21:37:08.831	20,44	false	false
29/7/2022	21:37:20.373	20,44	false	false
29/7/2022	21:37:31.814	20,44	false	false
29/7/2022	21:37:42.255	20,52	false	false

Realizado por: Jonathan, Collantes; 2022

4.1.2.2 Estabilidad bajo condiciones de trabajo con períodos de abandono de la cocina

El siguiente escenario exhibe condiciones en las que la cocina se encuentra en pleno funcionamiento y se estimula negativamente el sensor de proximidad, es decir, se deja sin supervisión las hornillas que quedaron encendidas por el tiempo que el sistema tiene preestablecido. Esto se logra visualizar en el sistema de monitoreo cambiando del estado “false” al estado “true” la variable digital que centra sus intereses en exponer este tipo de negligencias, eso se describe en la Tabla 2-4. Es importante distinguir que la temperatura ambiente pasó a manejar valores promedio entre 22°C y 23°C.

Tabla 2-4: Recolección de datos de variables virtuales en condiciones de abandono

Fecha	Hora	Temperatura	Fuga de Gas	Cocina Encendida y Abandonada
29/7/2022	21:43:06.438	22,25	false	false
29/7/2022	21:43:18.241	22,25	false	false
29/7/2022	21:43:29.468	22,31	false	false
29/7/2022	21:43:40.82	22,38	false	false
29/7/2022	21:43:52.052	22,38	false	false
29/7/2022	21:44:02.08	22,38	false	false
29/7/2022	21:44:12.948	22,45	false	false
29/7/2022	21:44:25.083	22,45	false	false
29/7/2022	21:44:36.419	22,45	false	false
29/7/2022	21:44:48.277	22,45	false	false
29/7/2022	21:44:59.48	22,56	false	false
29/7/2022	21:45:10.921	22,56	false	true
29/7/2022	21:45:22.262	22,72	false	true
29/7/2022	21:45:33.502	22,72	false	true
29/7/2022	21:45:44.858	22,79	false	true

29/7/2022	21:45:56.184	22,94	false	true
29/7/2022	21:46:07.221	22,94	false	true
29/7/2022	21:46:18.477	22,94	false	true
29/7/2022	21:46:29.609	23,08	false	true
29/7/2022	21:46:43.448	23,08	false	true

Realizado por: Jonathan, Collantes; 2022

No hay que dejar de lado en este punto de la recolección de datos que la temperatura ambiente no va a arrojar los mismos valores en distintos lugares en los que se haga la prueba. Esto dependerá directamente de las condiciones climáticas que se manejen en esa época del año, así como también de factores como ventilación del entorno de recolección de datos, estructuras de la habitación en la que se encuentre la cocina, entre otras. Con base en los valores medios obtenidos mensualmente en los últimos 20 años recopilados por DatosMundial, establece que en la región costa se tienen temperaturas máximas de 31.1°C, en la región sierra temperaturas máximas de 24.3°C y en la región oriental temperaturas máximas de 21.6°C, por lo que se agrega 15 °C al valor más alto obtenido, dicho valor corresponde al incremento de la temperatura al momento de que la cocina se encuentra en funcionamiento y de esta manera establecer un umbral que divida un estado seguro de un estado de potencial riesgo.

4.1.2.3 Estabilidad bajo condiciones de alta peligrosidad

En el último escenario se ha planteado condiciones de riesgo que pueden desencadenar un posible incendio o hasta una explosión al contacto con una fuente de ignición externa al entorno de la cocina. Se genera una inactividad del usuario en las cercanías de la cocina, instantes después el sistema de seguridad detecta que ya no solo está abandonada la cocina en funcionamiento, sino que también a causa de eso se ha detectado una fuga que la parte de control se encargará de contener. Uno de los casos típicamente recurrentes es que se deje sin supervisión la cocción de alimentos y que éstos hagan que desborden su contenido, lo que a continuación siempre sucede es que se apague el fuego en la hornilla y genere una fuga de gas, todo ello descrito en la Tabla 3-4.

Tabla 3-4: Recolección de datos de variables virtuales en condiciones de alta peligrosidad

Fecha	Hora	Temperatura	Fuga de Gas	Cocina Encendida y Abandonada
29/7/2022	21:54:02.589	23,19	false	false
29/7/2022	21:54:14.342	23,19	false	false
29/7/2022	21:54:24.678	23,19	false	false
29/7/2022	21:54:35.704	23,19	false	false
29/7/2022	21:54:47.217	23,19	false	false
29/7/2022	21:54:59.807	23,19	false	false
29/7/2022	21:55:38.525	23,19	false	true
29/7/2022	21:56:16.974	23,19	false	true

29/7/2022	21:56:50.307	23,19	false	true
29/7/2022	22:00:19.73	23,19	false	true
29/7/2022	22:00:19.748	23,25	true	true
29/7/2022	22:01:18.727	23,25	true	true
29/7/2022	22:02:06.392	23,25	true	true
29/7/2022	22:02:18.139	23,25	true	true
29/7/2022	22:02:29.79	23,31	true	true
29/7/2022	22:02:41.843	23,31	true	true
29/7/2022	22:02:54.159	23,31	true	true
29/7/2022	22:03:06.501	23,31	true	true
29/7/2022	22:03:18.648	23,31	true	true
29/7/2022	22:03:29.702	23,31	true	true

Realizado por: Jonathan, Collantes; 2022

La temperatura ambiente que se maneja en este punto es relativamente similar a la del anterior escenario, con un valor promedio que oscila alrededor de los 23°C.

4.1.3 Nivel de error

Se toma como referencia los valores de temperatura ambiente debido a que arrojan valores de tipo flotantes a la plataforma, es decir, medidas concretas con decimales. Las otras variables con las que se maneja el sistema de seguridad son de tipo bool lo que imposibilita una correcta evaluación del nivel de error. Se recoge una muestra de 20 mediciones con intervalos de tiempo aproximados de 10 segundos descritos en la Tabla 4-4.

Tabla 4-4: Descripción de elementos del prototipo con su respectivo costo en dólares

Fecha	Hora	Temperatura Ambiente	
		Medición	$ \bar{x} - x_i $
29/7/2022	15:12:03.709	21,75	0,25
29/7/2022	15:12:12.541	21,75	0,25
29/7/2022	15:12:23.82	21,75	0,25
29/7/2022	15:12:45.362	21,63	0,13
29/7/2022	15:12:54.502	21,63	0,13
29/7/2022	15:13:05.934	21,63	0,13
29/7/2022	15:13:10.465	21,63	0,13
29/7/2022	15:13:16.092	21,63	0,13
29/7/2022	15:13:40.481	21,5	0
29/7/2022	15:13:50.317	21,5	0
29/7/2022	15:13:59.098	21,5	0
29/7/2022	15:14:09.041	21,5	0
29/7/2022	15:14:43.809	21,38	0,12
29/7/2022	15:14:50.908	21,38	0,12
29/7/2022	15:15:00.949	21,38	0,12
29/7/2022	15:15:10.292	21,38	0,12
29/7/2022	15:15:20.618	21,38	0,12

29/7/2022	15:15:58.258	21,25	0,25
29/7/2022	15:16:00.774	21,25	0,25
29/7/2022	15:16:10.411	21,25	0,25
		Media	21,5
		Error Absoluto	0,1375

Realizado por: Jonathan, Collantes; 2022

Los niveles de temperatura en un ambiente con condiciones controladas para mantener lo más constante posible el valor de censado, de una muestra de 20 medidas con espacios de tiempo aproximados entre cada medida de 10 segundos, determinaron que el nivel de error que posee este sensor es de $\pm 0,1375$ °C, lo que se traduce en un nivel de certeza o confianza en el monitoreo en caso del abandono descuidado de la cocina va a ser altamente aceptable.

4.1.4 Recurrencia bajo condiciones de riesgo

Para la evaluación de las variables digitales en la etapa de recurrencia se establecen tres graficas de tipo “WIDGET” en la plataforma de monitoreo ilustradas en la Figura 7-4. La primera gráfica representa la temperatura ambiente, para la gráfica de “Fuga de Gas” y de “Cocina Encendida y Abandonada” se cambió la configuración del tipo de variable a tipo “int”, en la programación se asignaron valores de 0 y 1 para la interpretación de los datos. Adicionalmente para esta prueba se cambian los tiempos de permanencia del usuario cuando la cocina esté encendida para fines de evaluación de datos bajo condiciones de riesgo, con tiempo de 20 segundos de abandono.

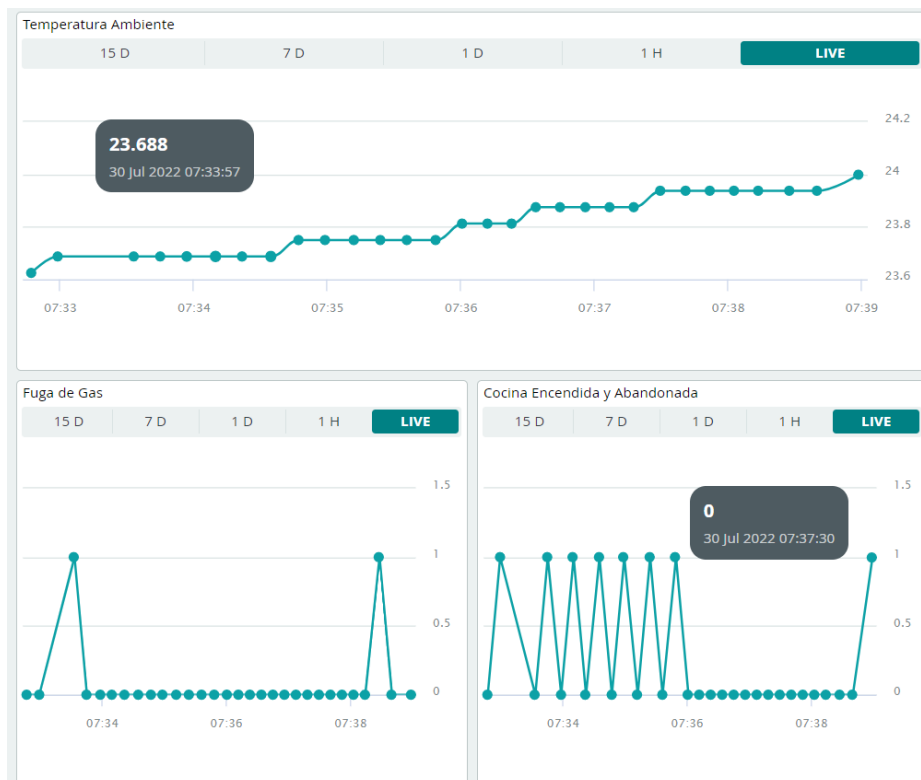


Figura 7-4: Recurrencia de datos en condiciones de riesgo – 30/07/2022 por la mañana

Realizado por: Jonathan, Collantes; 2022

Los valores esperados mediante la estimulación de las variables fueron visualizados satisfactoriamente en las gráficas establecidas para esta prueba. La temperatura ambiente estuvo en aumento dentro de los rangos esperados en °C. La gráfica de fuga de gas logró detectar en dos ocasiones anomalías en su entorno, tal como la gráfica del abandono de la cocina la cual percibió instantes en los que el usuario abandonó la cocina encendida.

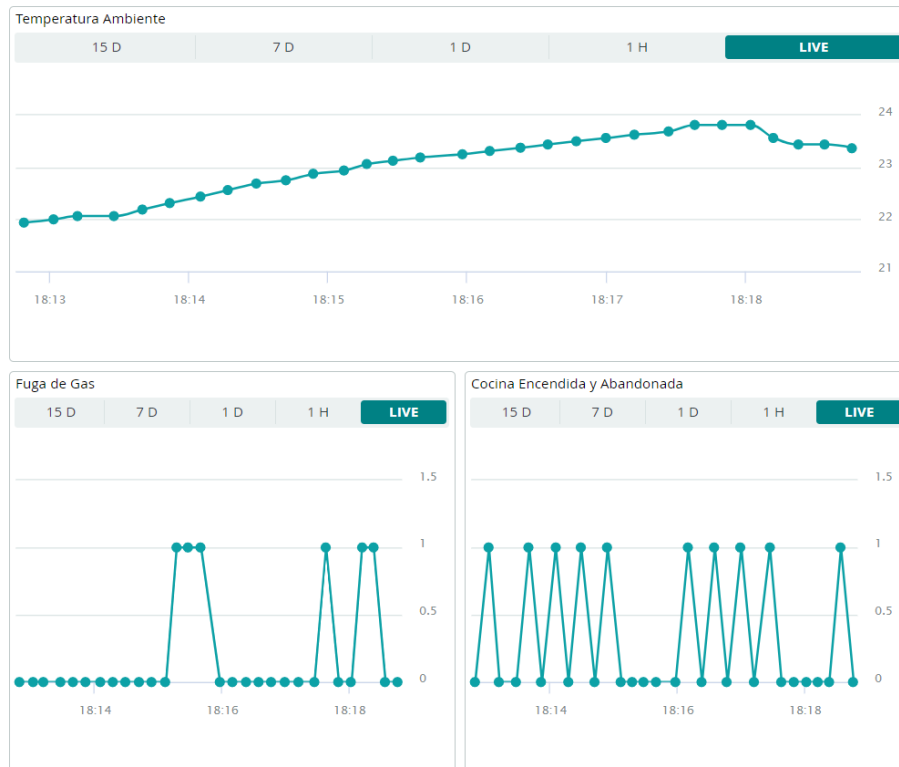


Figura 8-4: Recurrencia de datos en condiciones de riesgo – 30/07/2022 por la tarde

Realizado por: Jonathan, Collantes; 2022

Los valores recolectados e ilustrados en la Figura 8-4 por el sistema son bastante similares a los de la mañana bajo las mismas condiciones, esto denota que el sistema presenta una estabilidad muy favorable para que el prototipo de sistema de seguridad y que las herramientas que comprenden el monitoreo sean eficaces a la hora de encaminar esfuerzos de prevención de incendios en los hogares.

4.1.5 Notificaciones en caso de uso emergencia

El sistema de seguridad implantado en la cocina que se tomó como referencia emite tres diferentes notificaciones de manera oportuna. En orden de programación primero el sistema envía un e-mail desde la dirección de correo electrónico asignada al sistema hacia el correo electrónico del usuario responsable del monitoreo e ilustrado en la Figura 9-4.

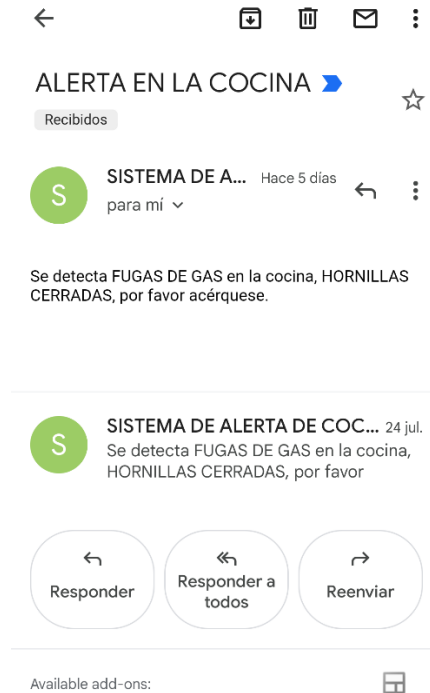


Figura 9-4: Notificación vía correo electrónico

Realizado por: Jonathan, Collantes; 2022

La siguiente notificación ilustrada en la Figura 10-4, es emitida desde el módulo SIM800L provisto de un chip exclusivo para el sistema. Esta notificación será un mensaje de texto con la información del estado de la cocina y las acciones tomadas a en función de lo sucedido.



Figura 10-4: Notificación vía mensaje de texto

Realizado por: Jonathan, Collantes; 2022

Por último, la tercera notificación es una alerta sonora emitida desde un buzzer implantado en el prototipo. Esta alarma quedará permanentemente encendida mientras en las cercanías de la cocina se haya detectado una fuga de gas.

4.1.6 Análisis económico del sistema

El prototipo de sistema de seguridad en cocinas a gas contiene todos los elementos y dispositivos descritos en la Tabla 5-4, especificando la cantidad, el costo individual de los elementos y el costo total del proyecto.

Tabla 5-4: Descripción de elementos del prototipo con su respectivo costo en dólares

Cantidad	Descripción	P. Unitario	P. Total
1	Node MCU ESP-32S	17	17
4	MG996R	12	48
1	MQ-2	4	4
1	DS18B20	5	5
1	HC-SR501	4	4
1	Placa de soporte	15	15
8	Impresiones 3D	2	16
1	LM2596	4	4
1	SIM800L	10	10
1	Cableado	5	5
1	Caja de control	8	8
1	Fuente de alimentación	10	10

Total	146
--------------	-----

Realizado por: Jonathan, Collantes; 2022

CONCLUSIONES

- El prototipo de sistema de seguridad en cocinas a gas consta con una comunicación wifi entre la plataforma Arduino IoT Cloud y la tarjeta de desarrollo, operando en la banda de frecuencia ISM de 2.4 GHz, con base en las funcionalidades, ésta satisface con un 97.14% la eficiencia en cuanto a las prestaciones que dicha tecnología proporciona al sistema de seguridad, consolidada con la transmisión de datos recolectados por los distintos sensores con los que está equipado el sistema para la prevención de incendios.
- Con base en las necesidades que se establecieron para este prototipo, el sistema logra reunir un conjunto de medidas encaminadas a la prevención, contemplando un costo relativamente bajo para su implementación y operación, incorporando una forma de trabajo manual y automática, dispositivos de censado para monitoreo eficaz y continuo del entorno de la cocina y una interfaz amigable con el usuario.
- El análisis presentado arrojó como resultado que, para interpretar correctamente las variaciones en los valores de gas, temperatura y presencia del usuario, se requería un conjunto de sensores constituidos por MQ-2 (con una concentración de detección de 200ppm a 5000ppm), DS18B20 (con un margen de error del sensor de temperatura de $\pm 0,1375^{\circ}\text{C}$), HC-SR501 (con una sensibilidad ajustable de entre 3m a 7m), conjuntamente del módulo SIM800L y un buzzer para emitir alertas oportunas. Todo esto es apoyado de la etapa de control de la tarjeta de desarrollo Node MCU ESP-32S y una plataforma configurada para cubrir todos los aspectos de seguridad.
- La incorporación de la etapa de censado, etapa de control y transmisión fueron mecanismos con los que se soporta eficientemente una seguridad sustentable. La incorporación de servomotores lo suficientemente robustos, fue en gran medida el mecanismo más coherente de mantener un control sobre el flujo de gas con el que se maneja una cocina convencional.
- Los resultados obtenidos cubrieron en su totalidad las necesidades de control y monitoreo en caso de ausencia del usuario, los tiempos de respuesta son oportunos y las alerta junto con el monitoreo conforman una nueva manera de manejar medidas de prevención, abriendo nuevas posibilidades y haciendo que dichas medidas puedan ser escalables para robustecer entornos que involucren una cocina a gas.

RECOMENDACIONES

- Proveer de una conexión continua a internet no solo a la tarjeta de desarrollo sino también a dispositivo móvil encargado del monitoreo, y de la misma manera de saldo en el chip que se vaya a incorporar al prototipo.
- Considerar que para análisis a posteriori, la plataforma Arduino IoT Cloud en su versión gratuita puede contener en sus servidores hasta un día de datos recolectados. La utilización de diferentes herramientas de visualización en la plataforma puede proveer de una mejor visión del sistema con las configuraciones adecuadas.
- Gestionar mantenimientos periódicos de los entornos de trabajo de los sensores y actuadores para no entorpecer su accionar y alargar la vida útil de los mismos.
- En caso de tomar la posta de esta investigación, se recomienda incorporar al sistema medidas de alimentación complementarias en caso de fallos en la red eléctrica.
- Un sistema jamás será 100 % seguro, por lo que no hay que dejar de lado las buenas costumbres a la hora de manejar cocinas que hagan uso de gas, toda medida que sume a una prevención de incendios hará que no existan pérdidas humanas y materiales.

BIBLIOGRAFÍA

ALKALI, A.; & MSHELIA, D E, ISUWA, S y BOLORI, H T, 2017. Automatic Gas Cooker Control System. . julio 2017. Vol. 6, no. 7, pp. 6.

ARDUINO, 2015. Getting Started with Intel® Galileo Gen2. en línea. 2015. [Consulta: 27 diciembre 2021]. Disponible en: <https://docs.arduino.cc/retired/getting-started-guides/IntelGalileoGen2>

ARDUINO, 2018. Introduction. What is Arduino? en línea. 2018. [Consulta: 27 diciembre 2021]. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>

ARDUINO, 2022. Getting Started With the Arduino IoT Cloud. en línea. 2022. [Consulta: 8 mayo 2022]. Disponible en: <https://docs.arduino.cc/arduino-cloud/getting-started/iot-cloud-getting-started>

BUYYA, Rajkumar y **DASTJERDI**, Amir Vahid, 2016. *Internet of Things Principles and Paradigms*. Cambridge USA: Morgan Kaufmann. ISBN 978-0-12-805395-9.

CAMPOS, Carlos Enrique Quirós, 2016. NFPA 72 2016 Español. *NFPA 72 - 2016 Español*. en línea. 1 enero 2016. [Consulta: 25 abril 2022]. Disponible en: https://www.academia.edu/42245935/NFPA_72_2016_Espa%C3%B1ol

CHACARREX, 2022. Protección contra incendios: Diferencia entre seguridad activa y pasiva en edificios. *Chacarrex*. en línea. 2022. [Consulta: 25 abril 2022]. Disponible en: <https://www.chacarrex.com/cuales-son-las-diferencias-entre-la-proteccion-pasiva-y-activa-contraincendios/>

COMISIÓN NACIONAL DE PREVENCIÓN DE RIESGOS Y ATENCIÓN DE EMERGENCIAS, COSTA RICA, 2017. Incendios. en línea. 2017. [Consulta: 25 abril 2022]. Disponible en: http://www.cne.go.cr/reduccion_riesgo/informacion_educativa/recomentaciones_consejos/incendios.aspx

DRYSDALE, Dougal, 2019. La química y la física del fuego. en línea. 2019. [Consulta: 27 diciembre 2021]. Disponible en: <https://www.insst.es/documents/94886/162520/Cap%C3%ADtulo+41.+Incendios>

ESPRESSIF SYSTEMS, 2015. *ESP8266EX Datasheet*. en línea. junio 2015. [Consulta: 27 diciembre 2021]. Recuperado a partir de:

https://components101.com/sites/default/files/component_datasheet/ESP8266-NodeMCU-Datasheet.pdf

FREMAP, 2015. *Guía Básica sobre PREVENCIÓN DE INCENDIOS*. en línea. 2015. Imagen Artes Gráficas, S.A. [Consulta: 25 abril 2022]. Disponible en: https://www.icv.csic.es/prevencion/Documentos/manuales/Guia_basica_sobre_Preencion_de_Incendios.pdf

GUBBI, Jayavardhana, BUYYA, Rajkumar, MARUSIC, Slaven y PALANISWAMI, Marimuthu, 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. septiembre 2013. Vol. 29, no. 7, pp. 1645-1660. DOI 10.1016/j.future.2013.01.010.

HAN, Ngoc Son, 2015. Semantic service provisioning for 6LoWPAN: powering internet of things applications on Web. . octubre 2015. pp. 123.

HERSENT, Olivier, POP, Gabor y REMOND, Guillaume, 2016. LoRa Device Developer Guide. en línea. 2016. [Consulta: 4 noviembre 2021]. Disponible en: <https://developer.orange.com/od-uploads/LoRa-Device-Developer-Guide-Orange.pdf>

LUDOVICI, Alessandro, MORENO, Pol y CALVERAS, Anna, 2013. TinyCoAP: A Novel Constrained Application Protocol (CoAP) Implementation for Embedding RESTful Web Services in Wireless Sensor Networks Based on TinyOS. *Journal of Sensor and Actuator Networks*. junio 2013. Vol. 2, no. 2, pp. 288-315. DOI 10.3390/jsan2020288.

MCFARLANE, Duncan, GIANNIKAS, Vaggelis y LU, Wenrong, 2016. Intelligent logistics: Involving the customer. *Computers in Industry*. 1 septiembre 2016. Vol. 81, pp. 105-115. DOI 10.1016/j.compind.2015.10.002.

MUJAWAR, T.H., BACHUWAR, V.D., KASBE, M.S., SHALIGRAM, A.D. y DESHMUKH, L.P., 2015. Development of wireless sensor network system for LPG gas leakage detection system. en línea. abril 2015. [Consulta: 3 noviembre 2021]. Disponible en: <https://www.ijser.org/paper/Development-of-wireless-sensor-network-system-for-LPG-gas-leakage-detection-system.html>

NOBOA CARRASCO, Edison Javier, 2019. *IMPLEMENTACIÓN DE UNA RED INALÁMBRICA CON NODOS INTELIGENTES PARA LA SUPERVISIÓN EN TIEMPO REAL Y PREVENCIÓN DE INCENDIOS EN LA PLANTA DE INDUFRICK*. en línea. Riobamba Ecuador: Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de

Ingeniería Electrónica, Telecomunicaciones y Redes. [Consulta: 4 noviembre 2021]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/13484/1/98T00254.pdf>

RASPBERRY PI, 2019. *Raspberry Pi 4 Computer*. en línea. junio 2019. Raspberry Pi Trading Ltd. [Consulta: 27 diciembre 2021]. Disponible en: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>

ROSE, Karen, ELDRIDGE, Scott y CHAPIN, Lyman, 2015. LA INTERNET DE LAS COSAS— UNA BREVE RESEÑA. . octubre 2015. pp. 83.

SECRETARÍA DE GESTIÓN DE RIESGOS ECUADOR, 2015. Incendios estructurales en Imbabura, Carchi y Esmeraldas. en línea. 2015. [Consulta: 25 abril 2022]. Disponible en: <https://www.gestionderiesgos.gob.ec/incendios-estructurales-en-imbabura-carchi-y-esmeraldas/>

SERVICIO ECUATORIANO DE NORMALIZACIÓN, 2017. *EXTINTORES PORTÁTILES. SELECCIÓN Y DISTRIBUCIÓN EN EDIFICACIONES*. en línea. 2017. [Consulta: 27 diciembre 2021]. Disponible en: https://www.normalizacion.gob.ec/buzon/normas/nte_inen_802-1.pdf

STALLINGS, William, JELASSI, Sofiene y AGBOMA, Florence, 2016. *Foundations of Modern Networking SDN, NFV, QoE, IoT, and Cloud*. Indianapolis, Indiana. 1. Indiana USA: Pearson Education, Inc. ISBN 978-0-13-417539-3.

THANGAVEL, Dinesh, MA, Xiaoping, VALERA, Alvin, TAN, Hwee-Xian y TAN, Colin Keng-Yan, 2014. Performance evaluation of MQTT and CoAP via a common middleware. En: *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. en línea. abril 2014. pp. 1-6. [Accedido 27 diciembre 2021]. DOI 10.1109/ISSNIP.2014.6827678.

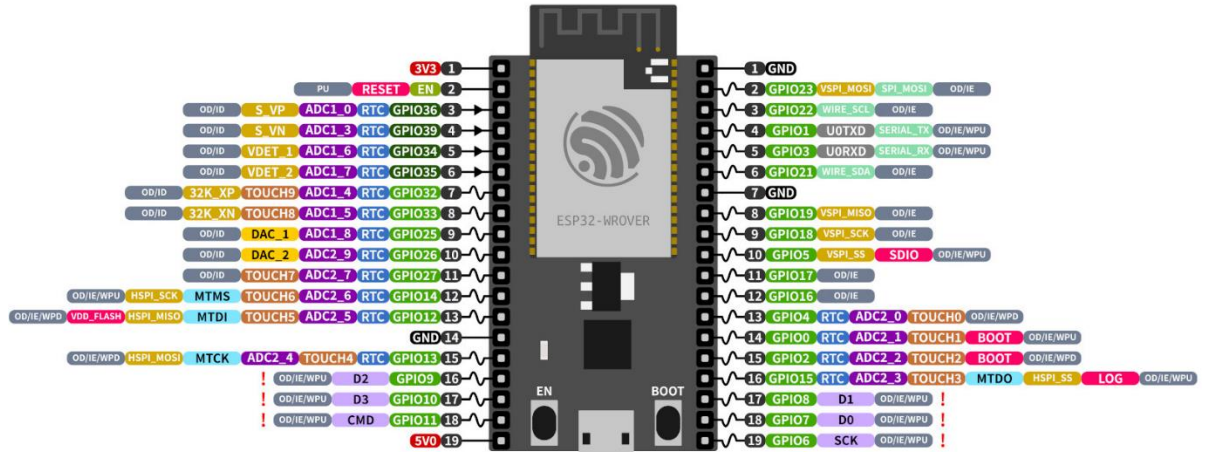
TSCHOFENIG, Hannes, ARKKO, Jari, THALER, Dave y MCPHERSON, Danny R., 2015. *Architectural Considerations in Smart Object Networking* en línea. Internet Engineering Task Force. [Consulta: 27 diciembre 2021].

XIONG, Xiong, ZHENG, Kan, XU, Rongtao, XIANG, Wei y CHATZIMISIOS, Periklis, 2015. Low power wide area machine-to-machine networks: key techniques and prototype. *IEEE Communications Magazine*. septiembre 2015. Vol. 53, no. 9, pp. 64-71. DOI 10.1109/MCOM.2015.7263374.



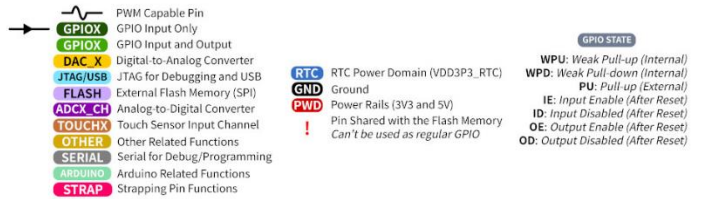
ANEXOS

Anexo A: Características de la tarjeta de desarrollo NodeMCU ESP32S



ESP32 Specs

32-bit Xtensa® dual-core @240MHz
 Wi-Fi IEEE 802.11 b/g/n 2.4GHz
 Bluetooth 4.2 BR/EDR and BLE
 520 KB SRAM (16 KB for cache)
 448 KB ROM
 34 GPIOs, 4x SPI, 3x UART, 2x I2C,
 2x I2S, RMT, LED PWM, 1 host SD/eMMC/SDIO,
 1 slave SDIO/SPI, TWAI, 12-bit ADC, Ethernet



Anexo B: Características del sensor DS18B20



www.maxim-ic.com

DS18B20 Programmable Resolution 1-Wire Digital Thermometer

FEATURES

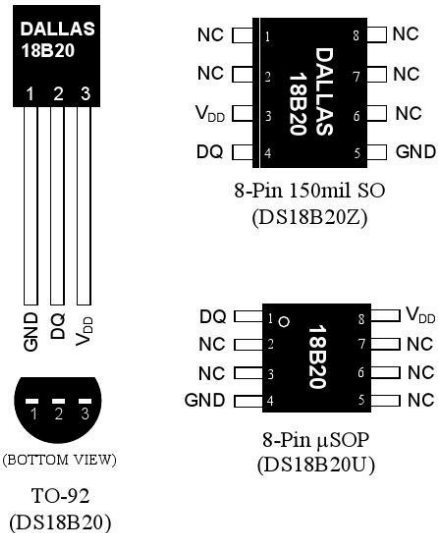
- Unique 1-Wire[®] interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an onboard ROM
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Measures temperatures from -55°C to $+125^{\circ}\text{C}$ (-67°F to $+257^{\circ}\text{F}$)
- $\pm 0.5^{\circ}\text{C}$ accuracy from -10°C to $+85^{\circ}\text{C}$
- Thermometer resolution is user-selectable from 9 to 12 bits
- Converts temperature to 12-bit digital word in 750ms (max.)
- User-definable nonvolatile (NV) alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Available in 8-pin SO (150mil), 8-pin μSOP , and 3-pin TO-92 packages
- Software compatible with the DS1822
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

DESCRIPTION

The DS18B20 Digital Thermometer provides 9 to 12-bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18B20 communicates over a 1-Wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of -55°C to $+125^{\circ}\text{C}$ and is accurate to $\pm 0.5^{\circ}\text{C}$ over the range of -10°C to $+85^{\circ}\text{C}$. In addition, the DS18B20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18B20 has a unique 64-bit serial code, which allows multiple DS18B20s to function on the same 1-wire bus; thus, it is simple to use one microprocessor to control many DS18B20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.

PIN ASSIGNMENT



PIN DESCRIPTION

- GND - Ground
- DQ - Data In/Out
- V_{DD} - Power Supply Voltage
- NC - No Connect

Anexo C: Características del sensor MQ-2

HANWEI ELETRONICS CO.,LTD

MQ-2

<http://www.hwsensor.com>

TECHNICAL DATA

MQ-2 GAS SENSOR

FEATURES

Wide detecting scope
Stable and long life

Fast response and High sensitivity
Simple drive circuit

APPLICATION

They are used in gas leakage detecting equipments in family and industry, are suitable for detecting of LPG, i-butane, propane, methane ,alcohol, Hydrogen, smoke.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
Vc	Circuit voltage	5V±0.1	AC OR DC
VH	Heating voltage	5V±0.1	ACOR DC
R _L	Load resistance	can adjust	
R _H	Heater resistance	33 Ω ± 5%	Room Tem
P _H	Heating consumption	less than 800mw	

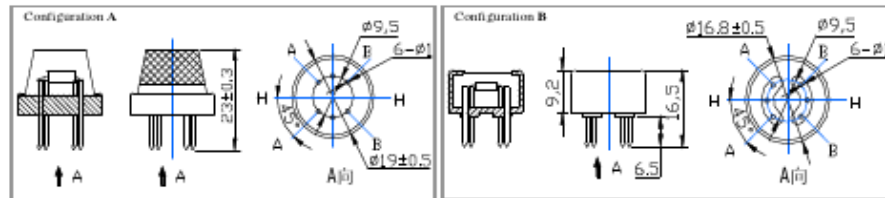
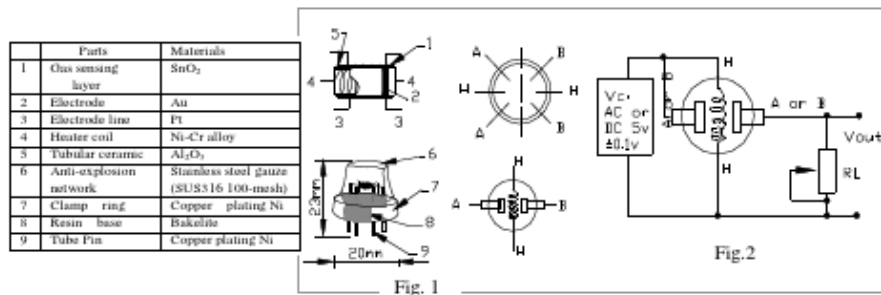
B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
Tao	Using Tem	-20℃-50℃	
Tas	Storage Tem	-20℃-70℃	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remarks
R _s	Sensing Resistance	3K Ω -30K Ω (1000ppm iso-butane)	Detecting concentration scope: 200ppm-5000ppm LPG and propane 300ppm-5000ppm butane 5000ppm-20000ppm methane 300ppm-5000ppm H ₂ 100ppm-2000ppm Alcohol
α (3000/1000) isobutane	Concentration Slope rate	≤0.6	
Standard Detecting Condition	Temp: 20℃ ± 2℃ Humidity: 65% ± 5%	Vc: 5V ± 0.1 Vh: 5V ± 0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit



Structure and configuration of MQ-2 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro AL₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a

TEL: 86-371-67169070 67169080 FAX: 86-371-67169090

E-mail: www.DataSheet4U.com

Anexo D: Características del sensor HC-SR501



Instruction Sheet

SR501-HC-4 Switching Relay, High Current Capacity

102-418

SUPERSEDES: June 1, 2010

EFFECTIVE: June 1, 2012

Plant ID# 9300-2795

Features:

- External Indicator Lights
- Simplified Wiring
- Sealed Relays
- 100% Factory Tested
- Contractor Friendly PC Board Layout
- Universal Thermostat Compatibility
- UL Approved
- Extended 3 Year Warranty

120 VAC Application:

1 zone switching relay with boiler enable or
2 zone without boiler enable.

240 VAC Application:

1 zone switching relay when switching both
of the circulator hot leads (L1 & L2).

Operation:

Connect a thermostat to the R & W1 and/or W2 terminals on the switching relay. When the thermostat(s) call for heat, the relays are energized and power is given to the circulator(s).

Zone Control Power Input:

Connect 120 volt AC power input to terminals N and H. Neutral wire to terminal N. Hot wire to terminal H.

External Diagnostics:

The external lights show full functionality of the switching relay. The green light should always be on, indicating that power is connected. When either thermostat calls for heat, both the appropriate relay and red indicating light are energized.

The Taco Connection:

Combine the reliability of the Taco Zone Valves, Priority Zoning Circulators, Thermostats, and the "00" family of circulators with the advanced features of the Taco Zone Controls to achieve total system integration. No matter the application, Taco now provides the products to maximize system performance while simplifying both installation and service.

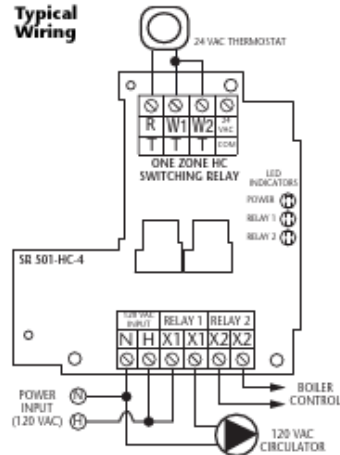
Terminal Description:

R Hot side of transformer. Connect to R on thermostat.
W1 Switched R signal from thermostat #1. Connect to W on thermostat.
W2 Switched R signal from thermostat #2. Connect to W on thermostat.
C Common side of transformer. Connect to COM on thermostat (optional).
N Neutral wire of power input.
H Hot wire of power input.
X1 Dry contacts for relay 1 (W1).
X1 Dry contacts for relay 1 (W1).
X2 Dry contacts for relay 2 (W2).
X2 Dry contacts for relay 2 (W2).

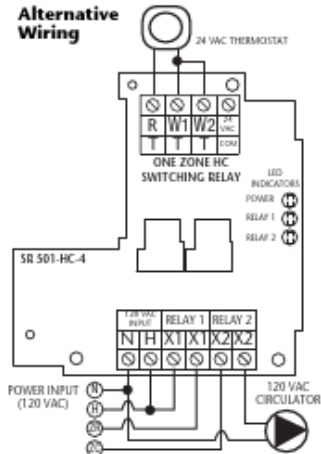
Specifications:

PRODUCT NUMBER	NUMBER OF ZONES	INPUT VOLTAGE	TYPE 1	ENCLOSURE WIDTH	ENCLOSURE HEIGHT	DEPTH
SR501-HC-4	1 or 2	120/240 VAC, 90 mA		4 1/8"	6 3/8"	2 1/4"
RELAY RATING: 1/4 HP (13.8 FLA, 82.8 LRA at 120 VAC) (6.9 FLA, 41.4 LRA at 240 VAC) The thermostat connection supplies a 24 VAC class 2 output.						

Typical Wiring



Alternative Wiring

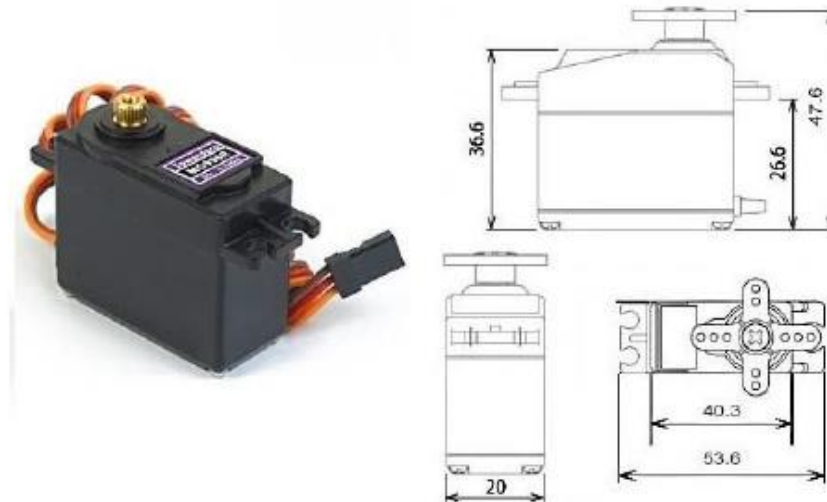


NOTE: When using Alternative Wiring diagram, the boiler operating control's ZC terminal will see the load of the circulator(s).

WARNING: When using Alternative Wiring diagram, wiring instructions must be followed so power originates from the boiler aquastat. Failure to follow these wiring instructions may result in a secondary source of power being connected to the boiler that may activate it under certain circumstances, causing injury or death.

WARNING: Wiring connections must be made in accordance with all applicable electrical codes. Use copper wire only. Failure to follow this instruction can result in personal injury or death and/or property damage. 12-18 gauge wire recommended for 120/240 VAC connections, 14-22 gauge wire for thermostat connections, and 14-22 gauge wire for 24 VAC source connections.

MG996R High Torque Metal Gear Dual Ball Bearing Servo



This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque in a tiny package. The MG996R is essentially an upgraded version of the famous MG995 servo, and features upgraded shock-proofing and a redesigned PCB and IC control system that make it much more accurate than its predecessor. The gearing and motor have also been upgraded to improve dead bandwidth and centering. The unit comes complete with 30cm wire and 3 pin 'S' type female header connector that fits most receivers, including Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum and Hitec.

This high-torque standard servo can rotate approximately 120 degrees (60 in each direction). You can use any servo code, hardware or library to control these servos, so it's great for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. The MG996R Metal Gear Servo also comes with a selection of arms and hardware to get you set up nice and fast!

Specifications

- Weight: 55 g
- Dimension: 40.7 x 19.7 x 42.9 mm approx.
- Stall torque: 9.4 kgf·cm (4.8 V), 11 kgf·cm (6 V)
- Operating speed: 0.17 s/60° (4.8 V), 0.14 s/60° (6 V)



7. Manufacturing

7.1. Top and Bottom View of SIM800L



Figure 53: Top and bottom view of SIM800L.

7.2. Typical Solder Reflow Profile

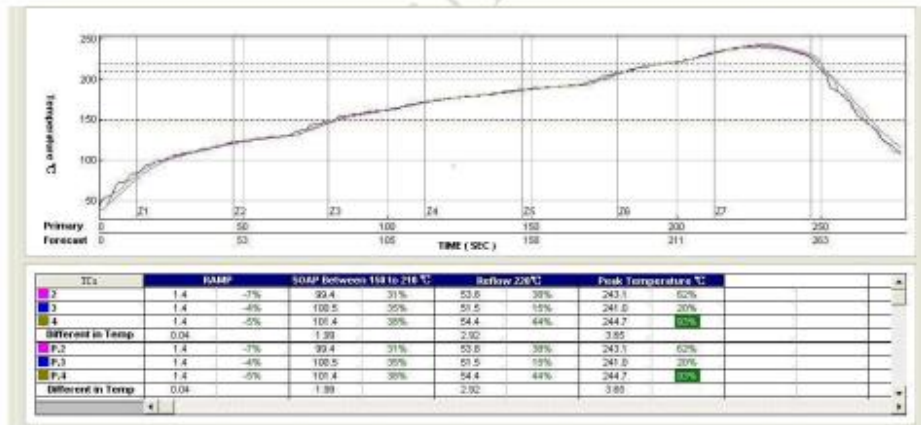


Figure 54: Typical solder reflow profile

7.3. The Moisture Sensitivity Level

The moisture sensitivity level of SIM800L module is 3. The modules should be mounted within 168 hours after unpacking in the environmental conditions of temperature <30°C and relative humidity of <60% (RH). It is

Anexo F: Código de programación

```
/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/c34752ab-b31b-4f9b-993b-83315289166b

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

float t;
int g;
int h;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"
#include <ESP_Mail_Client.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Servo.h>
#include <Adafruit_Sensor.h>

#define SENSOR_PIN 21 // Pin Sensor de Temperatura Ambiente
#define mov 33 // Pin Sensor de Movimiento
#define gas 32 // Pin Sensor de Gas
#define buzzer 15 // Pin Zumbador
#define h1 27 // Pin Hornilla 1
#define h2 14 // Pin Hornilla 2
#define h3 12 // Pin Hornilla 3
#define h4 13 // Pin Hornilla 4

OneWire oneWire(SENSOR_PIN);
DallasTemperature DS18B20(&oneWire);

Servo servo1; // Servomotor de Hornilla #1
Servo servo2; // Servomotor de Hornilla #2
Servo servo3; // Servomotor de Hornilla #3
Servo servo4; // Servomotor de Hornilla #4

int valm = 0; // Variable que detecta el cambio de estado en el movimiento inicializada en 0
int valg = 0; // Variable que detecta el cambio de estado de la concentración de gas inicializada en 0
int acumulador = 0; // Variable destinada a la etapa de temporización
float tempC; // Temperatura Ambiente
int L1 = 0; // Variable que detecta la apertura de la hornilla 1
int L2 = 0; // Variable que detecta la apertura de la hornilla 2
int L3 = 0; // Variable que detecta la apertura de la hornilla 3
int L4 = 0; // Variable que detecta la apertura de la hornilla 4

/***** Nombre del servidor smtp *****/

#define SMTP_HOST "smtp.gmail.com"
#define SMTP_PORT esp_mail_smtp_port_465
```



```

// Declaramos una variable para referenciar la sesión SMTP
/***** Declaramos una variable para referenciar la sesión SMTP *****/
SMTPSession smtp;

/** Prototipo de la función callback para obtener el estado del envío del correo **/
void smtpCallback(SMTP_Status status);

void setup()
{
  Serial.begin(9600);
  //delay(1500); // Este retraso da la posibilidad de esperar un monitor serie sin bloquearse si no se encuentra ninguno.
  initProperties();// Definido en thingProperties.h
  ArduinoCloud.begin(ArduinoIoTPreferredConnection); // Conexión con Arduino Cloud
  setDebugMessageLevel(15);
  ArduinoCloud.printDebugInfo();

  DS18B20.begin(); // Inicializamos Sensor 1 de temperatura ambiente
  pinMode(h1, INPUT);
  pinMode(h2, INPUT);
  pinMode(h3, INPUT);
  pinMode(h4, INPUT);
  pinMode(mov, INPUT);
  pinMode(gas, INPUT);
  pinMode(buzzer, OUTPUT);
}

void loop()
{
  ArduinoCloud.update();

  /***** Detección de Temperatura Ambiente *****/
  DS18B20.requestTemperatures(); // Envía el comando para obtener temperaturas
  tempC = DS18B20.getTempCByIndex(0); // Lectura de temperatura en °C
  t = tempC;

  Serial.print("Temperatura: ");
  Serial.print(tempC); //Temperatura en °C
  Serial.println("°C");

  /***** Detección de Fuga de Gas y Cierre de Hornillas *****/
  valg = analogRead(gas); // Lectura de valores de Gas
  valg = map(valg, 0, 4095, 0, 100); // Lectura de valores de Gas
  Serial.print("Porcentaje de Gas: ");
  Serial.print(valg);
  Serial.println("%");
}

```

```

if (valg > 1)
{
  g = 1;
  digitalWrite(buzzer, HIGH);
  Serial.println("HORNILLAS CERRADAS POR FUGAS DE GAS ");

  servo1.attach(4);
  servo1.write(150);
  delay(1500);
  servo1.detach();

  servo2.attach(23);
  servo2.write(150);
  delay(1500);
  servo2.detach();

  servo3.attach(25);
  servo3.write(150);
  delay(1500);
  servo3.detach();

  servo4.attach(26);
  servo4.write(150);
  delay(1500);
  servo4.detach();

  correofuga();
}
else
{
  g = 0;
  digitalWrite(buzzer, LOW);
}

/***** Detección de Movimiento, Minimización o Cierre de Hornillas *****/

L1 = digitalRead(h1);
L2 = digitalRead(h2);
L3 = digitalRead(h3);
L4 = digitalRead(h4);

int contador = 10; // Variable que determina la duración del temporizador
valm = digitalRead(mov); // Variable que detecta movimiento

if (L1 == HIGH || L2 == HIGH || L3 == HIGH || L4 == HIGH)
{
  h = 0;
}

if (L1 != HIGH)
{
  Serial.println("Hornilla 1 ENCENDIDA");
}

```

```

}
else
{
  Serial.println("Hornilla 1 Apagada");
}
if (L2 != HIGH)
{
  Serial.println("Hornilla 2 ENCENDIDA");
}
else
{
  Serial.println("Hornilla 2 Apagada");
}
if (L3 != HIGH)
{
  Serial.println("Hornilla 3 ENCENDIDA");
}
else
{
  Serial.println("Hornilla 3 Apagada");
}
if (L4 != HIGH)
{
  Serial.println("Hornilla 4 ENCENDIDA");
}
else
{
  Serial.println("Hornilla 4 Apagada");
}

if (valm == HIGH)
{
  h = 0;
  Serial.println("Movimiento Detectado");
  acumulador = 0;
  Serial.print("Acumulador igual a ");
  Serial.println(acumulador);
  delay(500);
}
if ((valm == LOW) && (L1 != HIGH || L2 != HIGH || L3 != HIGH || L4 != HIGH) && (valg < 2))
{
  Serial.println("Movimiento NO detectado");

  while (valm == LOW && contador != 0)
  {
    valm = digitalRead(mov);
    Serial.print("Tiempo igual a ");
    Serial.println(contador);
    contador = contador - 1;
    delay(1000);
  }
  acumulador = acumulador + 1;
  Serial.print("Acumulador igual a ");
  Serial.println(acumulador);
}

```

```
if (L1 != HIGH)
{
  Serial.println("Hornilla 1 ENCENDIDA");
}
else
{
  Serial.println("Hornilla 1 Apagada");
}
if (L2 != HIGH)
{
  Serial.println("Hornilla 2 ENCENDIDA");
}
else
{
  Serial.println("Hornilla 2 Apagada");
}
if (L3 != HIGH)
{
  Serial.println("Hornilla 3 ENCENDIDA");
}
else
{
  Serial.println("Hornilla 3 Apagada");
}
if (L4 != HIGH)
{
  Serial.println("Hornilla 4 ENCENDIDA");
}
else
{
  Serial.println("Hornilla 4 Apagada");
}

if (acumulador == 2)
{
  h = 1;
  if (L1 != HIGH)
  {
    servo1.attach(4);
    servo1.write(60);
    delay(1500);
    servo1.detach();
  }
  if (L2 != HIGH)
  {
    servo2.attach(23);
    servo2.write(60);
    delay(1500);
    servo2.detach();
  }
  if (L3 != HIGH)
  {
    servo3.attach(25);
```

```

        servo3.write(60);
        delay(1500);
        servo3.detach();
    }
    if (L4 != HIGH)
    {
        servo4.attach(26);
        servo4.write(60);
        delay(1500);
        servo4.detach();
    }
    Serial.println("HORNILLAS ABANDONADAS A SU MINIMA CAPACIDAD");
    Serial.println("Iniciando correo!!!");
    correo1();
}

if (acumulador == 4)
{
    h = 1;
    if (L1 != HIGH)
    {
        servo1.attach(4);
        servo1.write(60);
        delay(1500);
        servo1.detach();
    }
    if (L2 != HIGH)
    {
        servo2.attach(23);
        servo2.write(60);
        delay(1500);
        servo2.detach();
    }
    if (L3 != HIGH)
    {
        servo3.attach(25);
        servo3.write(60);
        delay(1500);
        servo3.detach();
    }
    if (L4 != HIGH)
    {
        servo4.attach(26);
        servo4.write(60);
        delay(1500);
        servo4.detach();
    }
    Serial.println("HORNILLAS ABANDONADAS A SU MINIMA CAPACIDAD");
    Serial.println("Iniciando correo!!!");
    correo1();
}

if (acumulador == 6)
{

```

```

h = 1;
if (L1 != HIGH)
{
  servo1.attach(4);
  servo1.write(150);
  delay(1500);
  servo1.detach();
}
if (L2 != HIGH)
{
  servo2.attach(23);
  servo2.write(150);
  delay(1500);
  servo2.detach();
}
if (L3 != HIGH)
{
  servo3.attach(25);
  servo3.write(150);
  delay(1500);
  servo3.detach();
}
if (L4 != HIGH)
{
  servo4.attach(26);
  servo4.write(150);
  delay(1500);
  servo4.detach();
}
Serial.println("HORNILLAS CERRADAS");
acumulador = 0;
Serial.print("Iniciando correo!!!");
correo2();
}
}
Serial.println(".....");
delay(1000);

}

void correo1()
{
  smtp.debug(1);
  ESP_Mail_Session session;
  session.server.host_name = SMTP_HOST;
  session.server.port = SMTP_PORT;
  session.login.email = AUTHOR_EMAIL;
  session.login.password = AUTHOR_PASSWORD;
  SMTP_Message message;
  message.sender.name = "SISTEMA DE ALERTA DE COCINA";
  message.sender.email = AUTHOR_EMAIL;
  message.subject = "ALERTA EN LA COCINA";
  message.addRecipient("Usuario.", "jon.coll12345@gmail.com");
}

```

```

String textMsg = "No se detecta presencia en la cocina, hornillas abandonadas y puestas en modo seguro, por favor acérquese.";
message.text.content = textMsg.c_str();
message.text.charset = "us-ascii";
message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit;
message.priority = esp_mail_smtp_priority::esp_mail_smtp_priority_normal;
if (!smtp.connect(&session))
    return;
if (!MailClient.sendMail(&smtp, &message))
    Serial.println("Error sending Email, " + smtp.errorReason());
ESP_MAIL_PRINTF("Liberar memoria: %d\n", MailClient.getFreeHeap());
}

void correo2()
{
    smtp.debug(1);
    ESP_Mail_Session session;
    session.server.host_name = SMTP_HOST;
    session.server.port = SMTP_PORT;
    session.login.email = AUTHOR_EMAIL;
    session.login.password = AUTHOR_PASSWORD;
    SMTP_Message message;
    message.sender.name = "SISTEMA DE ALERTA DE COCINA";
    message.sender.email = AUTHOR_EMAIL;
    message.subject = "ALERTA EN LA COCINA";
    message.addRecipient("Usuario.", "jon.coll12345@gmail.com");
    String textMsg = "No se detectó presencia en la cocina por tiempo considerable, las hornillas abandonadas fueron cerradas, por favor acérquese.";
    message.text.content = textMsg.c_str();
    message.text.charset = "us-ascii";
    message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit;
    message.priority = esp_mail_smtp_priority::esp_mail_smtp_priority_normal;
    if (!smtp.connect(&session))
        return;
    if (!MailClient.sendMail(&smtp, &message))
        Serial.println("Error sending Email, " + smtp.errorReason());
    ESP_MAIL_PRINTF("Liberar memoria: %d\n", MailClient.getFreeHeap());
}

void correofuga()
{
    smtp.debug(1);
    ESP_Mail_Session session;
    session.server.host_name = SMTP_HOST;
    session.server.port = SMTP_PORT;
    session.login.email = AUTHOR_EMAIL;
    session.login.password = AUTHOR_PASSWORD;
    SMTP_Message message;
    message.sender.name = "SISTEMA DE ALERTA DE COCINA";
    message.sender.email = AUTHOR_EMAIL;
    message.subject = "ALERTA EN LA COCINA";
    message.addRecipient("Usuario.", "jon.coll12345@gmail.com");
    String textMsg = "Se detecta FUGAS DE GAS en la cocina, HORNILLAS CERRADAS, por favor acérquese.";
    message.text.content = textMsg.c_str();
    message.text.charset = "us-ascii";
    message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit;

    message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit;
    message.priority = esp_mail_smtp_priority::esp_mail_smtp_priority_normal;
    if (!smtp.connect(&session))
        return;
    if (!MailClient.sendMail(&smtp, &message))
        Serial.println("Error sending Email, " + smtp.errorReason());
    ESP_MAIL_PRINTF("Liberar memoria: %d\n", MailClient.getFreeHeap());
}

```



ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO

DIRECCIÓN DE BIBLIOTECAS Y RECURSOS DEL
APRENDIZAJE



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 08/05/2023

INFORMACIÓN DEL AUTOR
Nombres – Apellidos: JONATHAN JACOBO COLLANTES ALCIVAR
INFORMACIÓN INSTITUCIONAL
Facultad: FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
Carrera: TELECOMUNICACIONES
Título a optar: INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES
f. Analista de Biblioteca responsable: Ing. Fernanda Arévalo M.



0579-DBRA-UPT-2023