

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR
ESCUELA DE INFORMÁTICA E INTELIGENCIA ARTIFICIAL**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN**

**APLICACIÓN MÓVIL PARA MONITOREAR LA CANTIDAD DE GAS EN
CILINDROS DOMÉSTICOS MEDIANTE UN PROTOTIPO IOT**

CRISTOPHER JOEL SÁNCHEZ PERUGACHI

TUTOR: STALIN ARCINIEGAS

IBARRA – ECUADOR

JULIO, 2024

Ibarra, 7 de julio de 2024

CERTIFICACIÓN TUTOR

En mi calidad de Tutor del Trabajo de Cristopher Joel Sánchez Perugachi titulado: **APLICACIÓN MÓVIL PARA MONITOREAR LA CANTIDAD DE GAS EN CILINDROS DOMÉSTICOS MEDIANTE UN PROTOTIPO IOT**, presentado por el estudiante Cristopher Joel Sánchez Perugachi con cédula de ciudadanía N° 1004098511, para obtener el Título de Ingeniero en Tecnologías de la Información.

Certifico que el trabajo cumple con todos los parámetros establecidos, mediante el cual el estudiante demuestra el desarrollo de competencias en el campo de conocimiento de su profesión con un nivel de argumentación coherente, para ser sometido a la evaluación por parte de los lectores.

Adicionalmente, se adjunta el certificado de porcentaje de originalidad de TURNITIN.

7/7/24, 19:32

Turnitin - Informe de Originalidad - revisión

Turnitin Informe de Originalidad	
Procesado el: 07-jul.-2024 18:54 -05	
Identificador: 2413588388	
Número de palabras: 15969	
Entregado: 1	
revisión Por STALIN ARCINIEGAS AGUIRRE	
Índice de similitud	Similitud según fuente
9%	Internet Sources: 6% Publicaciones: 3% Trabajos del estudiante: 4%



(f): _____

Mgs. Stalin Marcelo Arciniegas Aguirre

TUTOR DE TRABAJO

C.C.: 1003496815

PÁGINA DE APROBACIÓN DEL TRIBUNAL

El tribunal examinador, aprueba el presente trabajo en nombre de la Pontificia Universidad Católica del Ecuador Ibarra:



(f):

Mgs. Stalin Marcelo Arciniegas Aguirre

C.C.: 1003496815



(f):

Dra. Dulce Milagro Rivero Albarran

C.C.: 1757608961



(f):

Mgs. José Luis Ibarra Estévez

C.C.: 1002640728

ACTA DE CESIÓN DE DERECHOS

Yo, *Cristopher Joel Sánchez Perugachi*, declaro conocer y aceptar la disposición del Art. 165 del Código Orgánico de Economía Social de los Conocimientos, Creatividad e Innovación, que manifiesta textualmente: “Se reconoce facultad de los autores y demás titulares de derechos de disponer de sus derechos o autorizar las utilidades de sus obras o prestaciones a título gratuito y oneroso, según las condiciones que determinen. Esta facultad podrá ejercerse mediante licencias libres, abiertas y otros modelos alternativos de licenciamiento o la renuncia”.

Ibarra, 5 de julio del 2024

(f):  _____

Cristopher Joel Sánchez Perugachi

C.C.: 1004098511

AUTORIA

Yo, *Cristopher Joel Sánchez Perugachi*, portador(a) de la cedula de ciudadanía N° 1004098511, declaro que la presente trabajo de investigación es de total responsabilidad de la autor(a), y eximo expresamente a la Pontificia Universidad Católica del Ecuador Ibarra de posibles reclamos o acciones legales.



(f):.....

Cristopher Joel Sánchez Perugachi

C.C.: 1004098511

DEDICATORIA Y AGRADECIMIENTOS

Dedico esta tesis con todo mi amor y gratitud a mi padre, mi madre y mi hermano. Ellos han sido los tres pilares fundamentales que siempre me han sostenido, brindándome su incondicional apoyo en cada uno de mis logros. A mis padres, mis maestros de vida, cuyo amor y entrega se reflejan en cada paso que he dado. Gracias por su cariño, por enseñarme a soñar, a no rendirme ante cualquier situación y a buscar siempre una solución. Me han enseñado que antes de ser un profesional, es esencial ser una buena persona. También extiendo mi gratitud a toda mi familia, quienes con su apoyo y amor han sido parte esencial de este viaje. A Dios, por darme la sabiduría y la fortaleza para salir adelante, por guiarme en cada paso de este camino y por ser la fuente de mi inspiración y perseverancia.

A mis amigos Nathy, Dany, Naty, Karlita y Diego, ustedes han sido mi compañía y respaldo a lo largo de esta carrera, ayudándome a encontrar momentos de distracción y alegría cuando más los necesitaba. En especial, a Marco, quien es más que un amigo. Compartir este camino desde la escuela hasta nuestra graduación me llena de felicidad y orgullo.

Aprecio a los docentes que han dejado una huella en mí, más allá de lo académico. Sus enseñanzas y apoyo han sido fundamentales en mi desarrollo personal y profesional. Igualmente, expreso mi reconocimiento al personal administrativo que me ha conocido desde pequeño y ha estado presente en mi crecimiento.

Finalmente, reconozco a mi asesor, cuya orientación me motivó a llevar este proyecto fuera del aula y materializarlo. Gracias a su apoyo constante, hemos ido mejorando y perfeccionando el proyecto con cada nueva versión, evitando que quedara en el olvido. Gracias por invitarme a soñar y por hacerme ver que este proyecto puede convertirse en un potencial emprendimiento.

ÍNDICE DE CONTENIDOS

Contenido

Contenido

CERTIFICACIÓN TUTOR.....	2
PÁGINA DE APROBACIÓN DEL TRIBUNAL.....	3
ACTA DE CESIÓN DE DERECHOS.....	4
AUTORIA	5
DEDICATORIA Y AGRADECIMIENTOS	6
ÍNDICE DE CONTENIDOS.....	7
ÍNDICE DE TABLAS	10
ÍNDICE DE FIGURAS.....	11
RESUMEN.....	12
ABSTRACT	13
INTRODUCCIÓN	14
CAPÍTULO 1	17
ESTADO DEL ARTE.....	17
1.1 ANTECEDENTES	17
1.2 MARCO TEORICO	18
1.2.1 Comunicación y Frameworks para el Desarrollo del Proyecto	18
1.2.1.1 Internet de las Cosas	18
1.2.1.2 Protocolos de Comunicación en IoT.....	19
1.2.1.3 Message Queuing Telemetry Transport (MQTT)	19
1.2.1.4 Constrained Application Protocol (CoAP)	20
1.2.1.5 Broker.....	21
1.2.1.6 Mosquitto	22
1.2.1.7 Backend	22
1.2.1.8 Interfaz de Programación de Aplicaciones (API).....	23
1.2.1.9 Framework.....	23
1.2.1.10 NestJS	24
1.2.1.11 NodeJS	24

1.2.1.12 Frontend	25
1.2.1.13 Flutter	25
1.2.1.14 React Native	26
1.2.2 Monitoreo De Gas Mediante IoT	27
1.2.2.1 Microcontrolador	27
1.2.2.2 XIAO Esp32S3	27
1.2.2.3 Sensor	28
1.2.2.4 Sensores de gas semiconductor	28
1.2.2.5 Sensores electroquímicos	28
1.2.2.6 Sensores de infrarrojos	29
1.2.2.7 Sensores de ultrasonido	29
1.2.2.8 L06 Ultrasonic	29
1.2.2.9 Arquitecturas de Monitoreo de Gas con IoT	30
1.2.2.9.1 Arquitectura centralizada	30
1.2.2.9.2 Arquitectura descentralizada	30
1.2.2.10 Desafíos y Consideraciones en el Monitoreo de Gas con IoT.....	30
1.2.2.10.1 Consumo de energía	30
1.2.2.10.2 Seguridad y privacidad de los datos.....	30
1.2.2.10.3 Confiabilidad y tolerancia a fallos	31
1.2.2.10.4 Integración y escalabilidad	31
1.2.3 Aplicaciones Móviles Para Monitoreo De Gas.....	31
1.2.3.1 Diseño de Interfaces de Usuario	31
1.2.3.1.1 Visualización de datos en tiempo real	31
1.2.3.1.2 Alertas y notificaciones	31
1.2.3.1.3 Interacción intuitiva.....	31
1.2.3.2 Integración de Aplicaciones Móviles con Dispositivos IoT	31
1.2.3.2.1 Comunicación bidireccional	32
1.2.3.2.2 Sincronización de datos	32
1.2.3.2.3 Protocolos de comunicación	32
1.2.4 Tendencias Y Desafíos Futuros	32
1.2.4.1 Avances en Tecnologías de Sensores y IoT para Monitoreo de Gas	32
2.1 MATERIALES Y METODOS	33
2.1.2 Técnicas para la recolección de datos.....	34
2.1.3 Metodología	34
2.1.3.1 Desarrollo del Dispositivo IoT (Metodología de Prototipado)	35
2.1.3.2 Desarrollo de la Aplicación Móvil (Metodología de Programación Extrema - XP)	40

2.1.4 Fase 1 (Planificación)	41
2.1.4.1 Historias de Usuario	41
2.1.4.2 Requisitos funcionales	44
2.1.4.4 Módulos.....	46
2.1.4.5 Roles y responsabilidades	47
2.1.4.6 Planificación de actividades	48
2.1.4.7 Iteraciones	50
2.1.4.7.1 Iteración 1: Conexión del microcontrolador a la red (14 días)	50
2.1.4.7.2 Iteración 2: Almacenamiento y gestión de datos (10 días)	50
2.1.4.7.3 Iteración 3: Visualización de Dashboard (40 días)	50
2.1.4.7.4 Iteración 4: Gestión de Notificaciones (14 días)	51
2.1.5 Fase 2 (Diseño)	51
2.1.5.1 Arquitectura orientada a microservicios	51
2.1.5.2 Funcionalidades del Sistema	51
2.1.5.2.1 Casos de Uso	51
2.1.5.2.2 Diagrama de Actividades	52
2.1.5.2.3 Base de Datos	53
2.1.5.2.4 Diseño de la Interfaz	54
2.1.6 Fase 3 (Codificación)	55
2.1.6 Fase 4 (Pruebas)	56
3.1 Resultados Fase 1. Aplicación Móvil	59
3.1 Interfaz de conectividad	59
3.2 Interfaz de Credenciales	59
3.3 Interfaz de Gráficas	61
3.2 Resultados Fase 2. Prototipo Iot	62
3.4 Prototipo Finalizado	62
3.3 Pruebas del sistema	63
3.4 Pruebas del Prototipo Iot	68
CONCLUSIONES	71
RECOMENDACIONES	72
REFERENCIA BIBLIOGRÁFICA	74
ANEXOS	77

ÍNDICE DE TABLAS

Tabla 1: Requisitos Funcionales Prototipo lot	36
Tabla 2: Requisitos No Funcionales Prototipo lot.....	37
Tabla 3: Historias de Usuario	41
Tabla 4: Roles y Responsabilidades	47
Tabla 5: Planificación de Actividades para el proyecto	48
Tabla 6: Pruebas Funcionales Aplicación Móvil	56
Tabla 7: Pruebas Funcionales Prototipo	57
Tabla 8: Prueba de Configuración de Red Wi-Fi	63
Tabla 9: Prueba de Almacenamiento	64
Tabla 10: Prueba de Confirmación de Datos	64
Tabla 11: Prueba de Funcionalidad Dashboard	65
Tabla 12: Prueba de Notificaciones.....	66
Tabla 13: Prueba de Gestión de Permisos	67
Tabla 14: Prueba de Lectura	68
Tabla 15: Prueba de Envío de Datos.....	69
Tabla 16: Prueba de Integración	69
Tabla 17: Prueba de Gestión de energía	70

ÍNDICE DE FIGURAS

Figura 1: Caso de Uso Prototipo lot	37
Figura 2: Diagrama de actividades	38
Figura 3: Ecosistema del proyecto	39
Figura 4: Caso de Uso General del Sistema	51
Figura 5: Caso de Uso Usuario	52
Figura 6: Diagrama de actividades del Usuario	52
Figura 7: Diagrama NoSQL	53
Figura 8: Diseño de la interfaz	54
Figura 9: Distribución de pantalla	54
Figura 10: Interfaz de conectividad	59
Figura 11: Interfaz de credenciales	60
Figura 12: Interfaz de Gráficas	61

RESUMEN

Este informe detalla el desarrollo y la implementación de un sistema integral de monitoreo de GLP, que incluye una aplicación móvil y un dispositivo IoT. Se enfatiza la realización de pruebas unitarias para validar la conectividad, procesamiento de datos en tiempo real, gestión de notificaciones, configuración de red Wi-Fi, almacenamiento en base de datos NoSQL, diseño del dashboard y gestión de permisos. El uso de metodologías ágiles como XP y técnicas de prototipado permitió una entrega iterativa y adaptativa del proyecto. Se recomienda la expansión de la base de datos para integrar identificadores únicos de dispositivos y la implementación de un módulo de autenticación en la aplicación móvil para mejorar la escalabilidad y seguridad del sistema. Además, se sugiere ubicar los sensores en ambientes protegidos de la intemperie y con buena señal de internet para optimizar la precisión y operatividad del monitoreo. Se utilizan protocolos de red como MQTT para la comunicación entre el dispositivo IoT y el broker Mosquitto, aprovechando microcontroladores para la gestión eficiente de datos y señales.

El texto de este informe también ha sido refinado utilizando herramientas de inteligencia artificial para mejorar la redacción y coherencia del contenido, asegurando así una comunicación más clara y efectiva de los conceptos y hallazgos presentados.

Palabras clave: monitoreo de GLP, aplicación móvil, dispositivo IoT, backend, frontend, NoSQL, frameworks, pruebas unitarias, metodologías ágiles, expansión de base de datos, módulo de autenticación, seguridad, protocolos de red, MQTT, Mosquitto, broker, microcontrolador.

ABSTRACT

This report details the development and implementation of a comprehensive LPG monitoring system, including a mobile application and an IoT device. Emphasis is placed on conducting unit tests to validate connectivity, real-time data processing, notification management, Wi-Fi network configuration, NoSQL database storage, dashboard design, and permission management. The use of agile methodologies like XP and prototyping techniques allowed for an iterative and adaptive project delivery. It is recommended to expand the database to integrate unique device identifiers and to implement an authentication module in the mobile application to enhance the system's scalability and security. Additionally, it is suggested to place the sensors in weather-protected environments with good internet signals to optimize the accuracy and operability of the monitoring system. Network protocols such as MQTT are used for communication between the IoT device and the Mosquitto broker, leveraging microcontrollers for efficient data and signal management. The text of this report has also been refined using artificial intelligence tools to improve the writing and coherence of the content, ensuring clearer and more effective communication of the concepts and findings presented.

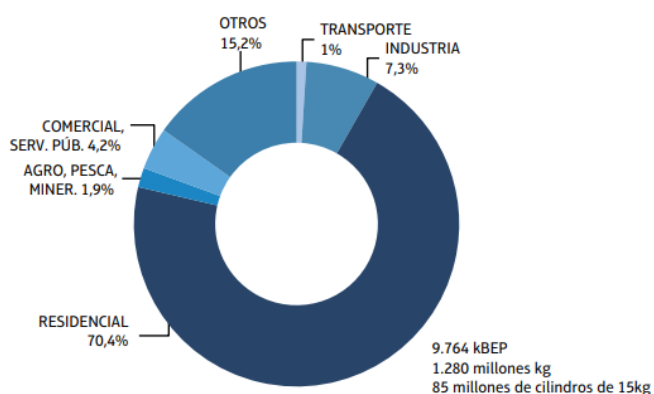
Keywords: LPG monitoring, mobile application, IoT device, backend, frontend, NoSQL, frameworks, unit testing, agile methodologies, database expansion, authentication module, security, network protocols, MQTT, Mosquitto, broker, microcontroller.

INTRODUCCIÓN

El proceso de distribución de Gas Licuado de Petróleo (GLP) en Ecuador es una cadena de suministro que involucra varias etapas principales. Comenzando por la producción y refinación en plantas procesadoras o refinerías a lo largo de todo el país, una vez obtenido, se envasa en cilindros de varios tamaños, listos para usarse en distintas acciones de la vida cotidiana de los ciudadanos, estos cilindros se distribuyen desde las plantas de producción hasta los centros de almacenamiento y puntos de venta en las ciudades y regiones del Ecuador mediante camiones cisterna especiales.

De acuerdo con el último balance energético realizado por el Ministerio de Energía y Minas del Ecuador (2021), en la ilustración 1 se puede observar que el sector con más uso del GLP es el sector residencial con un 70,4 % de la producción total generada

GAS LICUADO



La falta de monitoreo del (GLP) plantea un desafío significativo tanto para establecimientos comerciales como "Alpha Food & Drinks", así como para hogares que dependen de este recurso para sus necesidades diarias de cocción de alimentos y calefacción. La ausencia de una supervisión continua de los niveles de gas en los cilindros conlleva a problemas tales como el agotamiento impredecible del GLP y la falta de métodos confiables para verificar su disponibilidad. Estos obstáculos no se circunscriben

a una ubicación específica, sino que afectan a una amplia gama de negocios y hogares que confían en el GLP como su principal fuente de energía.

Esta investigación consideró el desarrollo de una solución innovadora basada en la Internet de las Cosas (IoT). La IoT ofrece la capacidad de conectar de manera inteligente los cilindros de GLP a través de sensores y dispositivos conectados, permitiendo un monitoreo remoto y en tiempo real de los niveles de gas. Esta integración de tecnología IoT una herramienta efectiva para mejorar la gestión del gas y garantizar una mayor eficiencia en el suministro de GLP, tanto para establecimientos comerciales como para hogares, contribuyendo así a una mayor seguridad y comodidad en el uso del GLP como fuente de energía.

En la actualidad, el mundo experimenta una rápida transformación impulsada por los avances tecnológicos, donde la interconexión de dispositivos inteligentes y el acceso móvil a la información desempeñan un papel fundamental en el desarrollo de soluciones innovadoras. En este contexto, la Internet de las Cosas (IoT) emerge como una revolucionaria disciplina que permite la conexión y comunicación entre objetos físicos a través de internet, habilitando una amplia gama de aplicaciones en diversos campos, desde la industria hasta el hogar inteligente.

La Internet de las cosas (IoT) se refiere a una red de objetos físicos equipados con sensores, software y otras tecnologías para comunicarse e intercambiar información con otros dispositivos y sistemas a través de Internet. Estos dispositivos pueden variar desde objetos domésticos simples hasta herramientas industriales avanzadas. Se estima que actualmente hay más de 7 mil millones de dispositivos IoT conectados, y se espera que esta cifra aumente a 10 mil millones para el año 2020 y a 22 mil millones para el año 2025, según los pronósticos de expertos.

En paralelo, el desarrollo de aplicaciones móviles ha experimentado un crecimiento exponencial, convirtiéndose en una herramienta omnipresente en la vida cotidiana. La combinación de la IoT con el desarrollo móvil ofrece un potencial sin precedentes para crear soluciones tecnológicas innovadoras que mejoren la eficiencia, la comodidad y la calidad de vida de las personas.

La interconexión de dispositivos inteligentes y la accesibilidad a través de dispositivos móviles constituyen elementos fundamentales en la evolución tecnológica de la sociedad, en esta intersección entre la IoT y el desarrollo móvil, se observa un vasto potencial para

la creación de soluciones innovadoras que transformen diferentes aspectos de la vida cotidiana. Desde aplicaciones en la industria pública y privada, en empresas de servicios, en los comercios, en instituciones educativas, hospitalarias, etc. Y en los hogares familiares y todas con la intención de mejorar procesos y/o la calidad de vida de las personas o del país.

De este modo el objetivo general del proyecto fue desarrollar una aplicación móvil para monitorear la cantidad de gas en cilindros domésticos mediante un prototipo IoT. Como objetivos específicos para lograr dicha implementación se establecieron los siguientes objetivos específicos:

- Realizar una investigación bibliográfica que fundamenta la investigación, permitiendo una comprensión profunda de las metodologías, tecnologías y mejores prácticas en el monitoreo de GLP mediante IoT.
- Diseñar un dispositivo con capacidad de sensorización y algoritmos eficientes para medir con precisión los niveles de GLP en cilindros domésticos.
- Desarrollar una aplicación móvil que permita a los usuarios visualizar y recibir alertas en tiempo real sobre los niveles de gas.
- Implementar la integración exitosa del dispositivo y la aplicación mediante un protocolo de red seguro y eficiente, sin subirla a tiendas virtuales.

El presente trabajo se encuentra estructurado en tres capítulos. En el primer capítulo, se aborda el estado del arte, en el cual se examina la información más relevante y actualizada de los últimos años en relación con el tema de investigación, el cual sustenta y respalda sólidamente el desarrollo del proyecto. El segundo capítulo detalla la metodología empleada, describiendo las técnicas, instrumentos y procesos involucrados en el desarrollo de la solución tecnológica propuesta. Finalmente, en el tercer capítulo, se exponen los resultados obtenidos, presentando el prototipo armado con la debida aplicación móvil para observar los datos.

CAPÍTULO 1

ESTADO DEL ARTE

Se lleva a cabo un detallado análisis del estado del arte relacionado con el monitoreo de gas en cilindros domésticos a través de prototipos IoT y aplicaciones móviles. Se examinan investigaciones recientes, desarrollos tecnológicos y proyectos relevantes, con el objetivo de establecer un contexto sólido y actualizado para el proyecto en cuestión. Este análisis abarca una amplia gama de fuentes, incluyendo artículos científicos, publicaciones especializadas y avances industriales, con el fin de identificar tendencias emergentes, desafíos actuales y áreas de oportunidad en el campo del monitoreo de recursos domésticos. Al comprender a fondo el estado actual de la tecnología y las investigaciones en este ámbito, se sientan las bases para la investigación posterior, permitiendo identificar posibles brechas de conocimiento y contribuir de manera significativa al avance y desarrollo de soluciones innovadoras en el monitoreo de gas doméstico.

1.1 ANTECEDENTES

En los últimos años, el desarrollo tecnológico ha impulsado la innovación en diversos ámbitos, incluido el monitoreo de recursos mediante la integración del Internet de IoT y el desarrollo de aplicaciones móviles. Esta convergencia ha dado lugar a una serie de proyectos que abordan desafíos específicos, como el seguimiento remoto y en tiempo real de variables en entornos domésticos e industriales.

Uno de estos proyectos relevantes fue llevado a cabo por Ponte (2019), quien se embarcó en un proyecto de monitoreo y control. Su enfoque incorporó cuatro celdas de carga como sensores en la medición de la cantidad de GLP en cilindros. Estas celdas de carga permitieron obtener lecturas precisas y en tiempo real de la masa del gas restante. Para la transmisión de datos, optó por un microcontrolador ESP32. Lo más notable del proyecto fue el uso del protocolo MQTT para transmitir los datos recopilados, además logró una visualización efectiva de los datos al reflejarlos en una tabla alojada en Google Drive. Esta elección permitió un acceso conveniente a la información relacionada con el GLP y ofreció una forma práctica de gestionar y analizar los datos recopilados.

Bastidas Y Murillo (2019) realizaron otro proyecto denominado “Desarrollo de un wearable para gas de uso doméstico en una red IOT y cloud” , enfocado en la utilización de un dispositivo wearable, en lugar de una aplicación móvil, para llevar a cabo el monitoreo. El proyecto incorporó de igual manera como el anterior proyecto realizado por Ponte(2019), celdas de carga HX711 y un sensor de presión HR-800, para obtener mediciones de la cantidad de GLP en cilindros domésticos.

Un aspecto interesante de este proyecto fue la elección del dispositivo wearable como plataforma de visualización y control. Esta decisión permitió que los usuarios accedieran a la información de manera más inmediata y cómoda, ya que podían llevar el dispositivo consigo. Además, la elección de un sensor de presión HR-800 agregó una capa adicional de datos relevantes, lo que proporcionó una visión más completa del estado del GLP.

En cuanto a la recopilación y presentación de datos, el proyecto hizo uso de ThingSpeak, una plataforma de IOT. ThingSpeak permitió recopilar, almacenar y visualizar los datos generados por los sensores de manera eficaz. Esta plataforma brindó una solución conveniente para gestionar y analizar los datos de monitoreo de GLP.

1.2 MARCO TEORICO

1.2.1 Comunicación y Frameworks para el Desarrollo del Proyecto

1.2.1.1 Internet de las Cosas

En la actualidad, la conectividad a Internet ha dado lugar a una proliferación de dispositivos conectados. Desde teléfonos inteligentes y relojes hasta electrodomésticos, vehículos y sensores industriales, la Internet de las Cosas (IoT) ha permitido que una amplia variedad de dispositivos esté en constante comunicación a través de la red.

De acuerdo con el último informe de Statista de 2023 proyecta que hay aproximadamente 15.14 mil millones de dispositivos de Internet de las Cosas (Iot, en inglés) conectados en el mundo, y se calcula que esta cifra se duplique para 2030 (FISICC, 2023).

Para contextualizar más a detalle cómo ha surgido esta tecnología se tiene que:

Kevin Ashton propuso por primera vez el concepto de Internet de las cosas en 1999, y se refirió a la IoT como objetos conectados identificables de forma única con tecnología de

identificación por radiofrecuencia (RFID). Sin embargo, la definición exacta del mismo aún está en proceso de formación y está sujeta a las perspectivas que se adopten.

En general, IoT se definió como "infraestructura de red global dinámica con capacidad de autoconfiguración basada en estándares y protocolos de comunicación" (Pradyumna Gokhale, 2018, pág. 1).

Una de las primeras máquinas que se conectaron a internet fue la de la empresa Coca-Cola, la misma fue manipulada en la Universidad Carnegie Mellon y era capaz de informar sobre su inventario y si dichas bebidas estaban frías o no. Este hito histórico marca el inicio de la era de la conectividad y la comunicación entre dispositivos a través de redes. A medida que la tecnología avanza, surge la necesidad de establecer protocolos y estándares que regulen y faciliten la transferencia de datos entre estos dispositivos interconectados.

1.2.1.2 Protocolos de Comunicación en IoT

Los protocolos de red desempeñan un papel crucial en la comunicación efectiva entre dispositivos en entornos de Internet de las Cosas (IoT). Según Kassab y Darabkh (2020) estos protocolos establecen las reglas y normas para la transmisión de datos, garantizando la interoperabilidad y la fiabilidad de las comunicaciones en redes de dispositivos interconectados. En el contexto de IoT, los protocolos de red son fundamentales para facilitar la transferencia de información entre sensores, actuadores y sistemas de control, permitiendo la recopilación, procesamiento y análisis de datos en tiempo real.

Los distintos protocolos de Transferencia de Hipertexto (HTTP), el Protocolo de Aplicación Constrained (CoAP), el Protocolo de Mensajería Telemetría de Colas (MQTT), entre otros, son esenciales para la comunicación eficiente y segura en aplicaciones IoT, abordando aspectos como la latencia, el consumo de ancho de banda, la seguridad y la elección del desarrollador.

1.2.1.3 Message Queuing Telemetry Transport (MQTT)

En el contexto de este proyecto, se pueden considerar distintos protocolos de comunicación como HTTP y MQTT. Sin embargo, MQTT se destaca como el protocolo más recomendado debido a su eficiencia y orientación hacia dispositivos IoT

Los principios de diseño de este protocolo se centran en minimizar el ancho de banda de la red y los requisitos de recursos de los dispositivos, garantizando una

entrega fiable. Es capaz de transmitir datos a través de redes con poco ancho de banda con un consumo de energía muy bajo (Mishra, Kertesz 2021, pág. 4).

MQTT minimiza la sobrecarga de datos y ofrece una comunicación bidireccional eficiente, su modelo de suscripción y publicación de mensajes simplifica la implementación de sistemas IoT y reduce la carga de la red, lo que se traduce en un menor consumo de energía.

Las principales características que hacen que MQTT sea una excelente opción para el monitoreo de gas en entornos IoT son:

- Bajo consumo de ancho de banda: MQTT minimiza la sobrecarga de datos, lo que lo hace adecuado para redes con ancho de banda limitado.
- Consumo de energía optimizado: El protocolo está diseñado para dispositivos con recursos limitados, como sensores y microcontroladores, lo que permite un bajo consumo de energía.
- Entrega confiable de mensajes: MQTT ofrece diferentes niveles de calidad de servicio (QoS) para garantizar la entrega de los datos, incluyendo confirmación de entrega y reenvío de mensajes.
- Modelo de publicación/suscripción: Esta arquitectura descentralizada facilita la escalabilidad y flexibilidad del sistema de monitoreo, permitiendo la incorporación de nuevos clientes y publicadores de manera sencilla.

El protocolo MQTT (Message Queuing Telemetry Transport) está diseñado específicamente para la comunicación "máquina a máquina". El protocolo MQTT se ejecuta sobre TCP/IP y tiene un tamaño de paquete de datos con una sobrecarga mínima (> 2 bytes), por lo que el consumo de la fuente de alimentación también es lo suficientemente pequeño. Este protocolo es independiente de los datos y puede transmitir datos en diversas formas, como datos binarios, texto, XML o JSON, y utiliza un modelo de publicación/suscripción en lugar de un modelo cliente/servidor además de tener como intermediario un Broker. (R A Atmoko, 2017, pág. 3)

1.2.1.4 Constrained Application Protocol (CoAP)

CoAP es otro protocolo de comunicación relevante en el contexto del monitoreo de gas mediante IoT. Está diseñado específicamente para dispositivos y redes con recursos limitados, como sensores y microcontroladores utilizados en aplicaciones IoT.

Las principales características de CoAP que lo hacen adecuado para el monitoreo de gas son:

- **Simplicidad y ligereza:** CoAP tiene una estructura de mensajes y una lógica de funcionamiento más simplificada que otros protocolos como HTTP, lo que lo hace ideal para dispositivos con recursos limitados.
- **Integración con HTTP:** CoAP puede interoperar con HTTP, permitiendo la integración de dispositivos IoT con aplicaciones web y servicios basados en HTTP.
- **Seguridad y confiabilidad:** CoAP incluye mecanismos de seguridad como autenticación y cifrado, y garantiza la entrega confiable de mensajes.
- **Modelo de solicitud/respuesta:** Al igual que HTTP, CoAP utiliza un modelo de interacción solicitud/respuesta, lo que facilita la integración con aplicaciones y servicios web existentes.

1.2.1.5 Broker

En el contexto del protocolo MQTT, un Broker es un componente fundamental que desempeña un papel crucial en la comunicación eficiente y confiable entre dispositivos en un entorno de IoT.

Un Broker MQTT sirve como centro central en el sistema de mensajería de publicación/suscripción, donde recibe mensajes de los editores y los reenvía a los suscriptores, desempeñando un papel fundamental en la supervisión de la comunicación entre los clientes MQTT y garantizando una entrega confiable de mensajes. (Team HiveMQ, 2023)

Dicho componente cumple el rol como un intermediario que facilita la transferencia de datos entre los dispositivos que publican información (publicadores) y aquellos que la reciben (suscriptores). El Broker recibe los mensajes publicados, los enruta a los suscriptores correspondientes y se garantiza que la información se entregara de manera oportuna y precisa.

En la implementación de sistemas de adquisición de datos en tiempo real que lo llevo a cabo R A Atmoko (2017), el Broker MQTT en conjunto con Mosquitto fueron de suma importancia para gestionar la comunicación entre dispositivos como el Wemos D1 Mini,

aplicaciones móviles y web. Al actuar como un punto central de intercambio de datos, el Broker permitió una arquitectura de comunicación descentralizada y escalable en la que los dispositivos podían enviar y recibir información de manera eficiente sin necesidad de conocer directamente a los destinatarios de los datos.

1.2.1.6 Mosquitto

Al momento de seleccionar un Broker para la implementación de sistemas IoT, existen diversas opciones disponibles. Una de estas alternativas es optar por servicios de Brokers alojados en la web, sin embargo, ello podría plantear preocupaciones sobre la integridad, disponibilidad y confidencialidad de los datos, dado que se trata de software proporcionado por terceros. Otra opción viable es la instalación y configuración de un Broker local, lo que garantiza un mayor control sobre la seguridad y la gestión de datos. En este contexto, para el presente proyecto se elige la solución de levantar un Broker MQTT local, y uno de los software más utilizados para este propósito es Mosquitto.

Según Light (2017) Mosquitto es una implementación de servidor y cliente MQTT para mensajería ligera en dispositivos con recursos limitados. Desarrollado por Roger A. Light, forma parte de la Fundación Eclipse y consta de servidor Mosquitto. Destaca por su modelo de publicación/suscripción eficiente. Se utiliza en investigaciones académicas para evaluar el rendimiento en IoT, ciudades inteligentes y monitoreo ambiental. Se integra en proyectos de código abierto y comerciales, demostrando su versatilidad en la comunicación IoT.

Lo que distingue a Mosquitto es su ligereza y eficiencia. Está diseñado para ser liviano, siendo apto incluso para dispositivos con recursos limitados, como microcontroladores y dispositivos IoT de baja potencia.

1.2.1.7 Backend

El Backend es la parte de una aplicación o sistema que se encarga de procesar la lógica de negocio, interactuar con bases de datos, manejar la autenticación y autorización, y proporcionar una API para que los clientes (como aplicaciones móviles o Frontend web) puedan comunicarse con él. Es la capa que se ejecuta en el servidor y es responsable de la gestión de datos, la validación de entradas, la aplicación de reglas de negocio y la generación de respuestas.

Se denomina Backend a la capa de acceso a los datos de un software que no es accesible para el usuario final. Además, esta capa contiene toda la lógica de la

aplicación que maneja los datos. Cabe destacar que los datos de una aplicación se encuentran almacenados en una base de datos dentro de un servidor. El responsable del Backend es la persona que trabaja del lado del servidor. (Ibarra et al., 2021, pag. 2)

1.2.1.8 Interfaz de Programación de Aplicaciones (API)

Una API es un conjunto de reglas y protocolos que define cómo se pueden comunicar diferentes componentes de software entre sí. Las APIs permiten que diferentes aplicaciones o servicios intercambien datos y funcionalidades de manera estandarizada y controlada. Actúan como una capa intermedia que facilita la interacción entre sistemas, ocultando la complejidad interna y permitiendo la reutilización de funcionalidades.

Tras la investigación de (Renzel et al., 2013) las define que desempeñan un papel fundamental como "objetos frontera" que facilitan la colaboración entre diversas comunidades en el entorno del Internet de los Servicios. Estos elementos de intermediación permiten traducir y negociar entre los diferentes lenguajes, prácticas y perspectivas que convergen en el ecosistema de servicios en línea. Las APIs actúan como puentes que conectan a desarrolladores, proveedores de servicios y usuarios finales, habilitando la integración de sistemas y la comunicación eficiente entre los distintos actores

1.2.1.9 Framework

Un framework es una estructura conceptual y tecnológica que se utiliza como base para el desarrollo de aplicaciones software. Proporciona un conjunto de herramientas, bibliotecas, directrices y patrones de diseño preestablecidos que facilitan y agilizan el proceso de desarrollo. Los frameworks proporcionan una base sólida y un conjunto de funcionalidades reutilizables que los desarrolladores pueden aprovechar, en lugar de tener que construir todo desde cero.

Generalmente, los frameworks son usados por programadores porque permiten acelerar el trabajo y favorecer que este sea colaborativo, reducir errores y obtener un resultado de más calidad. Pero no pienses que estas estructuras son cosa solo del departamento IT, qué va, en el mundo online hay frameworks para prácticamente todo: para definir el viaje de compra de un cliente, para aplicar mejoras a un producto digital y tener más conversiones. Y es que, hoy en día, la

automatización es la clave para conseguir que cualquier proceso de trabajo sea rápido y eficaz sin perder un ápice de calidad. (Unir, 2023)

1.2.1.10 NestJS

NestJS es un framework progresivo para Node.js que se basa en TypeScript y está diseñado para construir aplicaciones escalables y eficientes del lado del servidor. Utiliza una arquitectura modular basada en principios de diseño y patrones de programación robustos, como la inyección de dependencias y la inversión de control. NestJS facilita el desarrollo de aplicaciones empresariales complejas, microservicios y APIs RESTful, al tiempo que promueve una estructura de código limpia y mantenible.

Nest (NestJS) es un framework para crear aplicaciones del lado del servidor Node.js escalables y eficientes. Utiliza JavaScript progresivo, está construido con TypeScript y es totalmente compatible (aunque aún permite a los desarrolladores codificar en JavaScript puro) y combina elementos de POO (Programación orientada a objetos), FP (Programación funcional) y FRP (Programación reactiva funcional). (NestJS, 2024)

1.2.1.11 NodeJS

Node.js es un entorno de ejecución de JavaScript construido sobre el motor V8 de Google Chrome. Permite ejecutar código JavaScript del lado del servidor, lo que facilita el desarrollo de aplicaciones web escalables y de alto rendimiento. Node.js utiliza un modelo de entrada/salida (E/S) no bloqueante y orientado a eventos, lo que lo hace particularmente eficiente para aplicaciones en tiempo real y con una gran cantidad de conexiones simultáneas. Además, cuenta con un amplio ecosistema de paquetes y módulos reutilizables, lo que promueve la productividad y la modularidad en el desarrollo.

Según el análisis de Basumatary and Agnihotri (2022) Node.js, el entorno de ejecución para JavaScript, es altamente popular entre los desarrolladores Full Stack que buscan crear aplicaciones web. Permite a los desarrolladores escribir código del lado del servidor en JavaScript y gestionar tanto el lado del cliente como el del servidor. Node.js garantiza un rendimiento excepcional y aplica un paradigma asíncronico y sin bloqueo controlado por eventos para la entrada y salida.

1.2.1.12 Frontend

El Frontend, también conocido como "lado del cliente", se refiere a la parte de una aplicación web o móvil que interactúa directamente con el usuario final. Es la capa visible y con la que los usuarios interactúan, y su objetivo principal es proporcionar una experiencia de usuario intuitiva y atractiva. El frontend se encarga de la presentación de datos, la captura de entradas del usuario, la navegación y la interacción con el Backend a través de APIs.

Frontend se encarga de estilizar la página de tal manera que la misma pueda presentar la información de forma agradable para el usuario. El responsable del Frontend, debe de conocer las técnicas de experiencia de usuario para brindar una mejor interacción entre la persona y la página que visita, así mismo debe tener conocimientos de diseño de Interacción para colocar los elementos de tal manera que el usuario las pueda ubicar de forma rápida y cómoda. (Ibarra et al., 2021, pag. 3-4)

1.2.1.13 Flutter

Flutter es un framework de código abierto desarrollado por Google para la creación de aplicaciones nativas multiplataforma para dispositivos móviles, web y escritorio. Utiliza el lenguaje de programación Dart y un enfoque de renderizado basado en widgets, lo que permite una experiencia de desarrollo rápida y eficiente. Flutter ofrece una gran variedad de widgets personalizables y herramientas para el diseño de interfaces de usuario atractivas y fluidas. Además, su capacidad de "hot reload" permite ver los cambios en tiempo real, acelerando el ciclo de desarrollo y pruebas.

Según Tashildar et al. (2020) es conocido por su dependencia de los widgets OEM del dispositivo, en lugar de vistas web, y utiliza un motor de renderizado de alto rendimiento para cada componente de vista. Flutter ha sido elegido por Google como marco a nivel de aplicación para su sistema operativo de próxima generación. Algunas de las ventajas en el contexto del presente trabajo de investigación son:

- Desarrollo multiplataforma: Flutter permite crear aplicaciones para iOS y Android a partir de una misma base de código, reduciendo el esfuerzo de desarrollo.
- Renderizado de alto rendimiento: El motor de renderizado de Flutter basado en Skia proporciona una experiencia de usuario fluida y de alta calidad.

- Integración con IoT: Flutter ofrece facilidades para integrar aplicaciones móviles con dispositivos IoT, como sensores de gas, a través de APIs y protocolos de comunicación como MQTT.
- Visualización de datos: Flutter cuenta con un amplio conjunto de widgets personalizables que facilitan la creación de interfaces de usuario para la visualización de datos en tiempo real, generación de gráficos y alertas.

En Flutter, las aplicaciones se escriben utilizando Dart, un lenguaje de programación desarrollado por Google. Dart se diseñó para sustituir a JavaScript y tiene características como las palabras clave "async" y "await". Flutter es un SDK de código abierto que permite a los desarrolladores crear aplicaciones móviles fiables y de alto rendimiento para diversos sistemas operativos.

1.2.1.14 React Native

React Native es otro framework popular para el desarrollo de aplicaciones móviles multiplataforma. Desarrollado por Facebook, se basa en el ecosistema de React y permite crear aplicaciones nativas utilizando JavaScript y TypeScript. Algunas características relevantes de React Native en el contexto del monitoreo de gas son:

- Integración con IoT: Al igual que Flutter, React Native proporciona herramientas y bibliotecas para conectar aplicaciones móviles con dispositivos IoT y protocolos de comunicación como MQTT.
- Experiencia de usuario nativa: React Native utiliza los componentes nativos de cada plataforma, lo que garantiza una experiencia de usuario fluida y consistente con las convenciones de iOS y Android.
- Ecosistema de bibliotecas: La gran comunidad de React Native ofrece una amplia variedad de bibliotecas y paquetes que facilitan tareas como la visualización de datos, notificaciones y gestión de estados.

Para Danielsson (2016) se utiliza para construir aplicaciones móviles multiplataforma. Permite a los desarrolladores crear aplicaciones móviles utilizando JavaScript y React, y luego compilarlas en código nativo para iOS y Android. React Native se centra en la eficiencia y la reutilización del código, lo que permite a los desarrolladores escribir una vez y ejecutar en múltiples plataformas.

Estas plataformas y frameworks de desarrollo móvil, junto con los protocolos de comunicación IoT como MQTT y CoAP, proporcionan a los desarrolladores las herramientas necesarias para crear aplicaciones móviles robustas y eficientes para el monitoreo de gas en entornos domésticos e industriales.

1.2.2 Monitoreo De Gas Mediante IoT

1.2.2.1 Microcontrolador

Los microcontroladores son dispositivos fundamentales en la implementación de sistemas electrónicos y sistemas embebidos. Desde electrodomésticos hasta vehículos y dispositivos médicos, los microcontroladores desempeñan un papel crucial en la automatización y el control de sistemas en tiempo real. Su versatilidad, eficiencia y capacidad de procesamiento los convierten en elementos esenciales en el desarrollo de tecnologías innovadoras en numerosos campos.

Según Khalifeh et al. (2022) desempeñan un papel crucial en la operación eficiente de redes de sensores inalámbricos (WSNs). Estos dispositivos compactos integran en un solo chip un procesador, memoria y periféricos, lo que los hace ideales para aplicaciones de bajo consumo de energía y tamaño reducido. La arquitectura de un microcontrolador típicamente incluye memoria flash para almacenar el código del programa, RAM para datos temporales y EEPROM para almacenamiento no volátil. La programación de estos nodos se realiza comúnmente en lenguajes estándar como C/C++, con sistemas operativos como TinyOS y Contiki siendo ampliamente utilizados.

1.2.2.2 XIAO Esp32S3

Dentro de los distintos tipos de microcontroladores que existen se hará uso del Esp32S3 de la línea de XIAO el motivo de esta elección es que, su arquitectura RISC-V ofrece un rendimiento eficiente y una amplia compatibilidad con diferentes sistemas operativos y entornos de programación.

Dicho esto, para Espressif Systems (2021) menciona que el ESP32-S3 es un System-on-Chip (SoC) que destaca por su arquitectura de bajo consumo energético y su CPU de un solo núcleo basada en RISC-V. Con capacidades de Wi-Fi de 2.4 GHz y Bluetooth 5 (LE), este SoC ofrece un equilibrio óptimo entre rendimiento y eficiencia energética para aplicaciones IoT y dispositivos conectados. Su diseño compacto, memoria interna y funciones de seguridad avanzadas, como encriptación y arranque seguro, lo convierten en

una opción atractiva para proyectos que requieren conectividad inalámbrica confiable y protección de datos.

Gracias a su amplia gama de periféricos y su capacidad para funcionar en modo de bajo consumo, el ESP32-S3 se posiciona como una solución versátil y robusta para diversas aplicaciones electrónicas.

1.2.2.3 Sensor

Los sensores son dispositivos esenciales en una amplia gama de aplicaciones tecnológicas, ya que permiten la adquisición y medición de diversas magnitudes físicas o químicas del entorno. Estos dispositivos desempeñan un papel crucial en la recopilación de datos precisos y confiables, lo cual es fundamental para el monitoreo, control y toma de decisiones en sistemas automatizados e inteligentes.

Para Torres y Fernández (2023) la definición de un sensor es un dispositivo que detecta o mide una magnitud física y la convierte en una señal que puede ser interpretada por un observador o por un instrumento. Algunos ejemplos de tipos de sensores son sensores de proximidad con contacto y sensores de proximidad sin contacto, como los sensores ópticos de barrera.

Un sensor también puede definirse como un dispositivo que convierte una señal de entrada, ya sea física, química o biológica, en una señal eléctrica procesable. Esta señal eléctrica resultante, ya sea analógica o digital, puede ser interpretada y utilizada por sistemas de control, procesamiento de datos o interfaces de usuario. Técnicamente hablando son “Dispositivo eléctrico/mecánico que convierte magnitudes físicas a valores medibles de dicha magnitud. Generalmente, los valores medibles son señales eléctricas codificadas en analógico o digital”. (Torres & Fernández, 2023, pag. 3)

1.2.2.4 Sensores de gas semiconductor

Estos sensores se basan en la variación de la resistencia eléctrica de un material semiconductor en presencia de gas. Son ampliamente utilizados debido a su bajo costo, tamaño compacto y bajo consumo de energía. Pueden detectar una amplia gama de gases, incluyendo gas natural, GLP, monóxido de carbono, entre otros.

1.2.2.5 Sensores electroquímicos

Los sensores electroquímicos utilizan una reacción química para detectar la presencia de un gas específico. Cuentan con una alta selectividad y sensibilidad, y se destacan por su precisión en la medición de concentraciones de gas.

1.2.2.6 Sensores de infrarrojos

Estos sensores se basan en la absorción de radiación infrarroja por parte de las moléculas de gas. Son especialmente eficaces para la detección de gases como metano, dióxido de carbono y propano. Ofrecen una respuesta rápida y pueden medir concentraciones en tiempo real.

1.2.2.7 Sensores de ultrasonido

Los sensores de ultrasonido aprovechan la propagación del sonido a través del gas para determinar la distancia y el nivel de llenado de los cilindros. Estos sensores son útiles para monitorear el nivel de gas en tanques y cilindros sin necesidad de contacto directo.

La selección del tipo de sensor dependerá de factores como el tipo de gas a detectar, la precisión requerida, el consumo de energía y el entorno de aplicación. En el desarrollo de soluciones de monitoreo de gas basadas en IoT, es importante evaluar las características de los diferentes sensores y seleccionar los más adecuados para satisfacer los requisitos específicos del proyecto.

1.2.2.8 L06 Ultrasonic

El sensor ultrasónico L06 es un dispositivo ampliamente utilizado en aplicaciones que requieren mediciones de distancia no invasivas y precisas. Este sensor funciona utilizando ondas ultrasónicas para determinar la distancia entre el sensor y un objeto. Consiste en un transductor ultrasónico que emite pulsos de alta frecuencia y un receptor que detecta los ecos reflejados por el objeto.

La operación del sensor L06 ultrasónico es relativamente simple. El transductor emite una serie de pulsos ultrasónicos hacia el objeto que se desea medir. Estos pulsos viajan a través del aire a una velocidad constante y, al alcanzar el objeto, se reflejan de vuelta hacia el sensor. El sensor calcula la distancia al objeto midiendo el tiempo que tardan los pulsos ultrasónicos en viajar de ida y vuelta. Esta información se utiliza luego para determinar la distancia entre el sensor y el objeto con una precisión notable.

Este tipo de sensor se utiliza en una variedad de aplicaciones, que van desde sistemas de estacionamiento automáticos y sistemas de navegación para vehículos hasta sistemas de control de nivel de líquidos en tanques industriales. Su precisión y confiabilidad lo convierten en una opción popular para aplicaciones donde se requieren mediciones de distancia precisas y no invasivas.

1.2.2.9 Arquitecturas de Monitoreo de Gas con IoT

En el contexto del monitoreo de gas mediante IoT, existen diferentes arquitecturas que pueden ser adoptadas. Estas arquitecturas varían en términos de centralización, distribución de la inteligencia y procesamiento de datos.

1.2.2.9.1 Arquitectura centralizada

En una arquitectura centralizada, los dispositivos IoT (como sensores de gas) envían los datos recopilados a un servidor central, generalmente alojado en la nube. Este servidor se encarga de procesar, almacenar y analizar los datos, y proporcionar una interfaz de visualización y control a los usuarios. Esta arquitectura se caracteriza por una mayor capacidad de procesamiento y almacenamiento en el servidor central, pero también puede presentar desafíos de escalabilidad y latencia.

1.2.2.9.2 Arquitectura descentralizada

En una arquitectura descentralizada, los dispositivos IoT tienen una mayor capacidad de procesamiento y toma de decisiones localmente. Estos dispositivos pueden procesar y analizar los datos de forma autónoma, y solo enviar información relevante a un servidor central o a la nube. Esta arquitectura distribuye la inteligencia y reduce la carga en el servidor central, mejorando la escalabilidad y la respuesta en tiempo real.

Ambas arquitecturas tienen sus ventajas y desafíos. La elección de una u otra dependerá de factores como el volumen de datos, los requisitos de latencia, la disponibilidad de recursos en los dispositivos IoT y las consideraciones de seguridad y privacidad de los datos.

1.2.2.10 Desafíos y Consideraciones en el Monitoreo de Gas con IoT

1.2.2.10.1 Consumo de energía

Los dispositivos IoT utilizados para el monitoreo de gas, como sensores y microcontroladores, deben tener un consumo de energía optimizado para garantizar una operación prolongada y autónoma. Esto implica la selección de componentes de bajo consumo, la implementación de estrategias de gestión de energía y la optimización del código y las comunicaciones.

1.2.2.10.2 Seguridad y privacidad de los datos

La información recopilada por los sistemas de monitoreo de gas, como los niveles de llenado y la ubicación de los cilindros, puede ser sensible. Es crucial implementar medidas de seguridad adecuadas, como cifrado de datos, autenticación de dispositivos y

control de acceso, para proteger la privacidad de los usuarios y evitar el acceso no autorizado a la información.

1.2.2.10.3 Confiabilidad y tolerancia a fallos

Los sistemas de monitoreo de gas basados en IoT deben ser confiables y capaces de tolerar fallos parciales sin interrumpir el servicio. Esto implica el uso de sensores y dispositivos redundantes, mecanismos de detección y recuperación de errores, y planes de mantenimiento preventivo.

1.2.2.10.4 Integración y escalabilidad

La capacidad de integrar los sistemas de monitoreo de gas con otros sistemas y aplicaciones, como plataformas de gestión empresarial o aplicaciones móviles, es fundamental. Además, el sistema debe ser escalable para adaptarse a un mayor número de dispositivos, usuarios y requisitos en el futuro.

Abordar estos desafíos y consideraciones de manera efectiva es crucial para el desarrollo de soluciones de monitoreo de gas basadas en IoT que sean robustas, seguras y escalables.

1.2.3 Aplicaciones Móviles Para Monitoreo De Gas

1.2.3.1 Diseño de Interfaces de Usuario

En el desarrollo de aplicaciones móviles para el monitoreo de gas, el diseño de la interfaz de usuario (UI) juega un papel fundamental. Algunas características clave a considerar son:

1.2.3.1.1 Visualización de datos en tiempo real

Las aplicaciones deben presentar los niveles de gas en tiempo real, utilizando gráficos, indicadores y notificaciones que permitan a los usuarios comprender fácilmente la información.

1.2.3.1.2 Alertas y notificaciones

El sistema debe generar alertas cuando se detecten niveles de gas críticos o cambios significativos, informando a los usuarios de manera oportuna para que puedan tomar las acciones necesarias.

1.2.3.1.3 Interacción intuitiva

La navegación y las funcionalidades de la aplicación deben ser intuitivas y fáciles de usar, minimizando la curva de aprendizaje para los usuarios.

1.2.3.2 Integración de Aplicaciones Móviles con Dispositivos IoT

La integración de las aplicaciones móviles con los dispositivos IoT utilizados para el monitoreo de gas es crucial para garantizar una experiencia fluida y en tiempo real.

1.2.3.2.1 Comunicación bidireccional

Las aplicaciones móviles deben poder enviar comandos y configuraciones a los dispositivos IoT, mientras que estos últimos deben transmitir datos de forma continua a la aplicación.

1.2.3.2.2 Sincronización de datos

Los datos recopilados por los dispositivos IoT deben sincronizarse de manera eficiente con la aplicación móvil, permitiendo a los usuarios acceder a la información más actualizada.

1.2.3.2.3 Protocolos de comunicación

Protocolos como MQTT y CoAP, discutidos anteriormente, facilitan la comunicación entre las aplicaciones móviles y los dispositivos IoT, garantizando la entrega confiable de mensajes y un consumo eficiente de recursos.

1.2.4 Tendencias Y Desafíos Futuros

1.2.4.1 Avances en Tecnologías de Sensores y IoT para Monitoreo de Gas

Las tecnologías de sensores y IoT están en constante evolución, lo que genera oportunidades para mejorar los sistemas de monitoreo de gas. Algunos avances recientes incluyen:

- Sensores de gas más precisos y selectivos, con mayor sensibilidad y rangos de detección más amplios.
- Sensores de gas con mayor eficiencia energética, permitiendo una operación prolongada de los dispositivos IoT.
- Integración de inteligencia artificial y aprendizaje automático en los sensores y dispositivos IoT para el análisis predictivo

CAPÍTULO II

MATERIALES Y METODOS

En el presente capítulo, se detalla la metodología adoptada para el desarrollo de la propuesta tecnológica, enfocada en el monitoreo de gas en cilindros domésticos mediante un prototipo IoT y una aplicación móvil. Se inicia con la definición de los requisitos funcionales y no funcionales del sistema, los cuales sientan las bases para el diseño y la implementación de la solución. Después, se usa la técnica de historias de usuario para capturar de forma clara y concisa las necesidades y expectativas de los usuarios finales. Las historias de usuario es un insumo para la planificación del proyecto, dividiendo el trabajo en iteraciones o *Sprints*, propio de los desarrollos ágiles, orientado a entregar valor de manera incremental. Adicionalmente, se describen los aspectos técnicos y arquitectónicos que conforman la solución propuesta, incluyendo la selección de tecnologías, *frameworks* y herramientas utilizadas.

2.1 MATERIALES Y METODOS

Este apartado describe los aspectos generales que enmarcan la investigación sobre el monitoreo de GLP a través de una aplicación móvil. La investigación se enmarca en el ámbito aplicado, donde se aplican los conocimientos obtenidos durante la formación como ingeniero en Tecnologías de la Información y que se complementaron con aprendizajes de forma autodidacta. Esta combinación de conocimientos fue fundamental para el desarrollo del presente trabajo de investigación.

En cuanto al enfoque metodológico, se ha optado por uno cualitativo, ya que permite una comprensión profunda de las necesidades, preferencias y percepciones de los usuarios finales potenciales de la aplicación. Este enfoque es crucial durante las etapas iniciales de diseño para asegurar que el sistema sea intuitivo y efectivo. Para recopilar información relevante sobre cómo los usuarios interactúan con el GLP, sus preocupaciones, expectativas y problemas comunes, se utilizarán técnicas cualitativas como entrevistas en profundidad y grupos focales. Esta información cualitativa ayudará a identificar las características y funcionalidades más importantes para los usuarios.

Además, la elección del enfoque cualitativo se justifica porque permite explorar en detalle las experiencias y contextos de uso del GLP en diferentes escenarios, lo que no sería posible con métodos cuantitativos. Este enfoque facilita la obtención de datos ricos y

matizados, esenciales para diseñar una solución tecnológica que realmente responda a las necesidades de los usuarios. Al comprender mejor el contexto y las particularidades de los usuarios, se pueden tomar decisiones de diseño más informadas y precisas, lo que contribuye a los objetivos del proyecto de desarrollar una herramienta efectiva y bien recibida por los usuarios finales.

El objetivo principal de esta investigación aplicada y el enfoque cualitativo es generar conocimiento y ofrecer soluciones prácticas que contribuyan a mejorar el monitoreo de GLP. La integración de datos cualitativos asegura que el sistema no solo sea técnicamente sólido, sino también altamente funcional y adaptado a las necesidades reales de los usuarios.

2.1.2 Técnicas para la recolección de datos

Para obtener información valiosa y comprender las necesidades de los usuarios finales, se ha optado por utilizar la técnica de entrevista no estructurada. Esta metodología permite mantener una conversación abierta y flexible con los participantes, sin seguir un guion predeterminado. En este contexto, se llevó a cabo una entrevista con el Señor Jhony Sánchez, dueño del establecimiento donde se implementó el proyecto de investigación. Durante la entrevista, se exploraron diversos aspectos relacionados con el uso y las expectativas respecto al sistema de monitoreo de GLP. Esta entrevista permitió obtener retroalimentación valiosa y capturar los requerimientos de manera natural, garantizando así la relevancia y pertinencia de la solución propuesta. Además, se complementó con la observación directa del entorno y el uso del sistema por parte de los usuarios para obtener una perspectiva más completa de las necesidades y desafíos. La transcripción de esta entrevista se encuentra en los anexos.

2.1.3 Metodología

En este proyecto, se emplean dos metodologías ágiles ampliamente reconocidas: Prototipado y Extreme Programming (XP). Estas metodologías se han seleccionado para abordar los desafíos específicos del proyecto y aprovechar al máximo las ventajas de un enfoque ágil. El prototipado proporciona una manera efectiva de construir el sensor mediante iteraciones y pruebas, permitiendo ajustes continuos según las necesidades y retroalimentaciones de los usuarios.

Por otro lado, XP es una metodología ágil de desarrollo de software que se enfoca en mejorar la calidad del software y la capacidad de respuesta a las necesidades cambiantes

de los clientes. XP promueve el desarrollo iterativo, la comunicación continua con el cliente y la entrega frecuente de software funcional. Sus prácticas clave incluyen la programación en parejas, donde dos desarrolladores trabajan juntos en una misma estación de trabajo para revisar el código en tiempo real y mejorar la calidad del software; pruebas unitarias, que garantizan que cada componente del software funcione correctamente de manera individual; y la integración continua, que asegura que las adiciones y cambios en el código se integren y prueben regularmente en el sistema completo.

Además, XP fomenta la retroalimentación constante del cliente, lo que permite ajustar y adaptar el proyecto según sus necesidades y expectativas. Esta metodología también enfatiza la simplicidad en el diseño, desarrollando únicamente lo necesario para cumplir con los requisitos actuales, y la propiedad colectiva del código, donde todos los miembros del equipo pueden trabajar en cualquier parte del código base. En el contexto de este proyecto, XP se utiliza para asegurar que el software del sistema de monitoreo de GLP sea robusto, eficiente y alineado con las expectativas de los usuarios, permitiendo realizar ajustes rápidos y efectivos durante el ciclo de desarrollo.

2.1.3.1 Desarrollo del Dispositivo IoT (Metodología de Prototipado)

La metodología de prototipado ha sido ampliamente utilizada y desarrollada en el campo de la ingeniería de software y la gestión de proyectos. Esta metodología implica crear un modelo o prototipo funcional del sistema IoT, que se refina iterativamente según las retroalimentaciones y requerimientos de los usuarios. El enfoque de prototipado permite una visualización temprana del producto final, facilitando la identificación y corrección de errores en etapas tempranas del desarrollo. Además, fomenta la participación de los usuarios en el proceso, asegurando que el prototipo final cumpla con los requerimientos establecidos.

Para el desarrollo del dispositivo IoT, se siguieron varias fases:

1. **Recolección de Requisitos:** A través de entrevistas con el gerente del establecimiento, se recopilaron los requisitos funcionales y no funcionales del prototipo IoT. Estos requisitos se listan en la Tabla 1 y la Tabla 2, respectivamente.

2. **Diseño del Prototipo:** Se elaboró un diseño inicial del dispositivo IoT basado en los requisitos recopilados. Este diseño incluyó la selección de componentes, la arquitectura del sistema y los diagramas de flujo de datos.
3. **Desarrollo y Pruebas Iniciales:** Se construyó un prototipo inicial que fue sometido a pruebas funcionales para verificar su capacidad de medir el nivel de gas y comunicarse con la aplicación móvil.
4. **Iteración y Refinamiento:** Basado en los resultados de las pruebas y la retroalimentación de los usuarios, se realizaron ajustes y mejoras en el prototipo. Este ciclo de desarrollo y pruebas se repitió varias veces hasta alcanzar un prototipo satisfactorio.
5. **Implementación Final:** Una vez finalizado el refinamiento, se implementó el prototipo en un entorno real para realizar pruebas finales y asegurar su correcto funcionamiento.

A través de entrevistas con el gerente del establecimiento, se recopilaron los requisitos funcionales y no funcionales del prototipo IoT. Estos requisitos se listan en la Tabla 1.

Tabla 1: Requisitos Funcionales Prototipo IoT

Requisitos Funcionales	
Código	Descripción
RF-01	El prototipo debe medir el nivel de gas en cilindros domésticos utilizando un sensor ultrasónico.
RF-02	El prototipo debe procesar los datos del nivel de gas de manera periódica y almacenar dicha información
RF-03	El prototipo debe funcionar en conjunto con una aplicación móvil para la visualización y monitoreo remoto de los datos.

Igualmente, se listan los requisitos no funcionales, según se observa en la Tabla 2.

Tabla 2: Requisitos No Funcionales Prototipo lot

Requisitos No Funcionales	
Código	Descripción
USABILIDAD	La interfaz de la aplicación móvil debe ser intuitiva y fácil de usar, permitiendo a los usuarios realizar tareas básicas de monitoreo en menos de 3 minutos sin necesidad de asistencia técnica.
SEGURIDAD	Los datos transmitidos entre el prototipo y la aplicación móvil al momento de ser almacenados debe manejar el uso de Tokens de seguridad
EFICIENCIA	El sistema debe procesar y actualizar los datos del nivel de gas en menos de 15 segundos desde la recolección hasta su visualización en la aplicación móvil. Esta métrica asegura que el sistema responde rápidamente y que los usuarios reciben información en tiempo real.

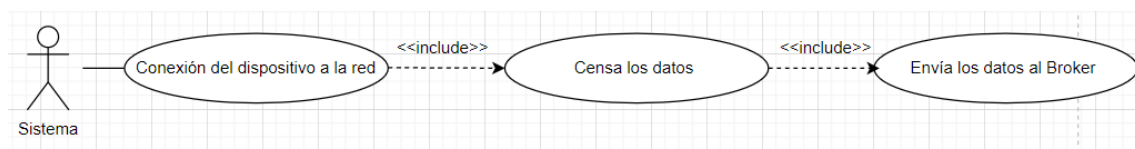
Limitaciones. El prototipo debe monitorear un cilindro de gas a la vez y su rango de medición estará limitado por el alcance del sensor ultrasónico.

Propósito. El objetivo de esta fase es desarrollar un prototipo IoT capaz de medir y monitorear el nivel de gas en cilindros domésticos, brindando información en tiempo real a través de una aplicación móvil.

Diseño rápido. Se realizó un diseño preliminar de la arquitectura y el flujo de funcionamiento del prototipo, incluyendo la selección de componentes clave como el sensor ultrasónico L06 y el microcontrolador ESP32-S3.

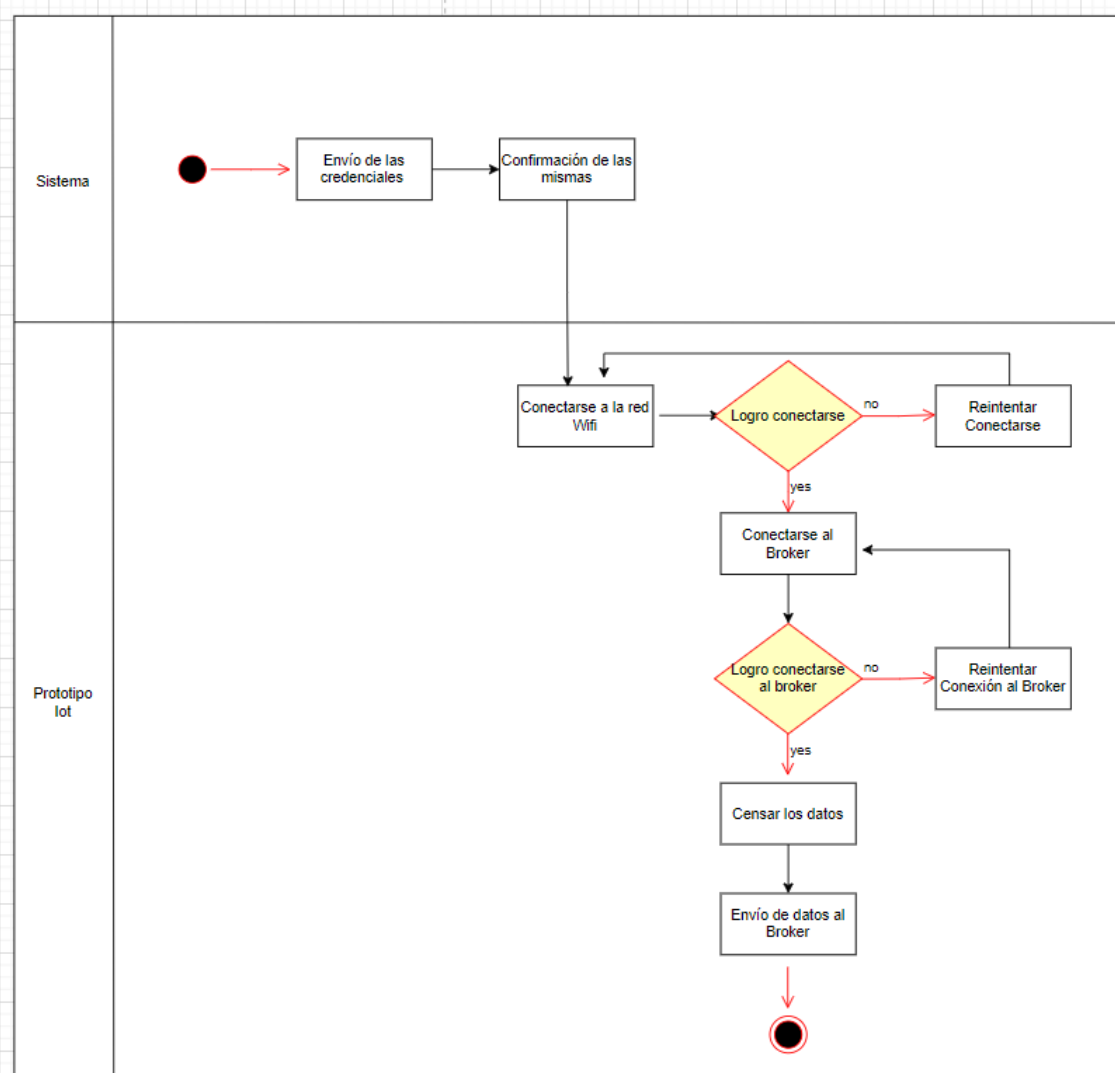
Casos de uso. La Figura 1 ilustra la interacción del sistema con el prototipo IoT para la conexión a la red y envío de datos

Figura 1: Caso de Uso Prototipo lot



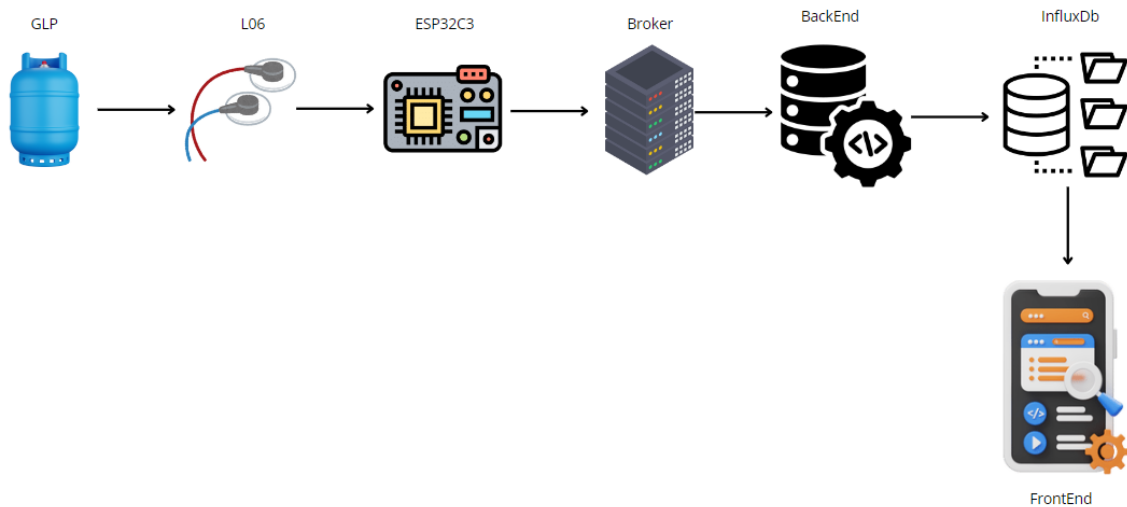
Flujograma del prototipo. La Figura 2 muestra diagrama del flujo operativo detallado del prototipo, incluyendo los procesos de medición, procesamiento de datos y envío a través de Wi-Fi.

Figura 2: Diagrama de actividades



En la figura 3 se muestra un ecosistema de como funcionaría la integración de ambas partes como lo es el hardware y el software

Figura 3: Ecosistema del proyecto



Fases de la Metodología de Prototipado

Fase 1 Construcción. Se realizó el ensamblaje físico del prototipo, conectando el sensor ultrasónico L06 al microcontrolador ESP32-S3 y desarrollando el firmware en C++ para la lectura de mediciones, procesamiento de datos y comunicación Wi-Fi.

Fase 2 Evaluación. Se llevaron a cabo pruebas exhaustivas del prototipo en condiciones reales, evaluando la precisión de las mediciones, la estabilidad de la conexión Wi-Fi y la integración con la aplicación móvil. Se recopiló retroalimentación de usuarios para identificar mejoras y refinamientos necesarios.

Fase 3 Refinamiento e iteración. Basado en la evaluación, se realizaron ajustes en el firmware, se optimizó el consumo energético y se mejoró la interfaz de usuario en la aplicación móvil. Estas iteraciones se repitieron hasta cumplir con los criterios de aceptación establecidos.

2.1.3.2 Desarrollo de la Aplicación Móvil (Metodología de Programación Extrema - XP)

Para el desarrollo de la aplicación móvil destinada al monitoreo de gas, se seleccionó la metodología ágil de XP. Esta metodología promueve el desarrollo iterativo e incremental, la colaboración estrecha con los usuarios y la adaptabilidad ante los cambios. Mediante la implementación de prácticas la integración continua y las pruebas constantes, se busca garantizar la calidad y la entrega oportuna de la aplicación móvil.

2.1.4 Fase 1 (Planificación)

2.1.4.1 Historias de Usuario

Las historias de usuario son una técnica ampliamente utilizada en el desarrollo de software y productos tecnológicos, especialmente en enfoques ágiles como la Programación Extrema (XP). Estas historias representan una forma concisa y comprensible de capturar los requisitos y necesidades de los usuarios finales, sirviendo como base para la planificación y ejecución del proyecto, en la tabla 3 se identificaron las siguientes historias de usuario:

Tabla 3: Historias de Usuario

Aplicación Móvil						
ID	NOMBRE	ESTIMACIÓN ESFUERZO (DÍAS)	IMPORTANCIA	DESCRIPCIÓN	CRITERIOS DE ACEPTACIÓN	DEPENDENCIAS
1	Conexión del microcontrolador a la red y almacenamiento de datos	14	90/100	Poder conectar a la red Wi-Fi mediante la aplicación móvil, con el fin de habilitar la comunicación y el envío de datos del dispositivo IoT al servidor	<ul style="list-style-type: none">• El usuario puede ingresar las credenciales de la red Wi-Fi (SSID y contraseña) desde la aplicación móvil.• La aplicación móvil muestra mensajes informativos sobre el estado de la conexión (conectado, fallido, etc.).	Ninguna

					<ul style="list-style-type: none"> • El microcontrolador establece una conexión segura y estable con la red Wi-Fi. 	
2	Almacenamiento y gestión de datos	10	90/100	Configurar un backend y una base de datos para almacenar y gestionar los datos del nivel de gas recibidos desde el dispositivo IoT, con el fin de mantener un registro	<ul style="list-style-type: none"> • El backend está configurado para recibir y procesar los datos del nivel de gas enviados por el dispositivo IoT. • Se establece una conexión segura y confiable con una base de datos (InfluxDB). • Los datos del nivel de gas se almacenan correctamente en la base de datos, incluyendo información como la fecha 	1
3	Visualización de <i>Dashboard</i>	40	95/100	Visualizar en un <i>dashboard</i> de la aplicación móvil los datos actualizados del nivel de gas en los cilindros domésticos, con el fin de monitorear su estado en tiempo real.	<p>El usuario puede ver las lecturas más recientes del nivel de gas enviadas por el dispositivo IoT.</p> <ul style="list-style-type: none"> • El <i>dashboard</i> presenta la información de manera clara y comprensible, utilizando gráficos y/o indicadores visuales. • Los datos se actualizan automáticamente en el <i>dashboard</i> cada vez que se reciben nuevas lecturas del dispositivo IoT. 	1, 2

4	Gestión de Notificaciones	14	85/100	Recibir notificaciones en el dispositivo móvil cuando los niveles de gas en los cilindros domésticos alcancen un estado crítico o de riesgo	<ul style="list-style-type: none"> ● La aplicación móvil envía notificaciones push en tiempo real cuando se detectan niveles críticos de gas. ● Las notificaciones incluyen información clara y concisa sobre el estado del cilindro de gas. 	1, 2
---	---------------------------	----	--------	---	--	------

2.1.4.2 Requisitos funcionales

A continuación, se detallan los requisitos funcionales del sistema basado en el análisis de las historias de usuario:

RF-01: Conexión del microcontrolador a la red y almacenamiento de datos

- **RF-01.1:** El sistema debe permitir la conexión del microcontrolador a una red Wi-Fi a través de la aplicación móvil.
 - **Criterios de aceptación:**
 - El usuario puede ingresar las credenciales de la red Wi-Fi (SSID y contraseña) desde la aplicación móvil.
 - La aplicación móvil muestra mensajes informativos sobre el estado de la conexión (conectado, fallido, etc.).
 - El microcontrolador establece una conexión segura y estable con la red Wi-Fi.

RF-02: Almacenamiento y gestión de datos

- **RF-02.1:** El sistema debe almacenar los datos del nivel de gas en una base de datos de manera segura y confiable.
 - **Criterios de aceptación:**
 - El backend está configurado para recibir y procesar los datos del nivel de gas enviados por el dispositivo IoT.
 - Se establece una conexión segura y confiable con una base de datos (InfluxDB).
 - Los datos del nivel de gas se almacenan correctamente en la base de datos, incluyendo información como la fecha y la hora de la medición.

RF-03: Visualización de *Dashboard*

- **RF-03.1:** La aplicación móvil debe presentar una interfaz de usuario intuitiva y fácil de usar que permita visualizar los datos del nivel de gas en tiempo real.
 - **Criterios de aceptación:**

- El usuario puede ver las lecturas más recientes del nivel de gas enviadas por el dispositivo IoT.
- El *dashboard* presenta la información de manera clara y comprensible, utilizando gráficos y/o indicadores visuales.
- Los datos se actualizan automáticamente en el dashboard cada vez que se reciben nuevas lecturas del dispositivo IoT.

RF-04: Gestión de notificaciones

- **RF-04.1:** El sistema debe detectar automáticamente niveles críticos de gas y generar notificaciones push instantáneas a la aplicación móvil para alertar al usuario sobre posibles riesgos.
 - **Criterios de aceptación:**
 - La aplicación móvil envía notificaciones push en tiempo real cuando se detectan niveles críticos de gas.
 - Las notificaciones incluyen información clara y concisa sobre el estado del cilindro de gas.

2.1.4.3 Requisitos no funcionales

RNF-01: Usabilidad

- **RNF-01.1:** La interfaz de la aplicación móvil debe ser intuitiva y fácil de usar, permitiendo a los usuarios realizar tareas básicas de monitoreo en menos de 3 minutos sin necesidad de asistencia técnica.

RNF-02: Seguridad

- **RNF-02.1:** Los datos transmitidos entre el prototipo y la aplicación móvil al momento de ser insertadas en la base de datos debe manejar la seguridad en base a Tokens con permisos específicos a la base de datos.

RNF-03: Eficiencia

- **RNF-03.1:** El sistema debe procesar y actualizar los datos del nivel de gas en menos de 15 segundos desde la recolección hasta su visualización en la

aplicación móvil. Esta métrica asegura que el sistema responde rápidamente y que los usuarios reciben información en tiempo real.

2.1.4.4 Módulos

- **Módulo de Conexión**

Roles:

- Desarrollador

Actividades:

- Implementar la conexión Wi-Fi en el microcontrolador
- Desarrollar la interfaz de configuración de red en la aplicación móvil
- Establecer la comunicación segura entre el microcontrolador y la aplicación móvil

- **Módulo de Monitoreo y Visualización**

Roles:

- Desarrollador

Actividades:

- Configurar el sensor ultrasónico para la medición de niveles de gas
- Implementar la lógica de procesamiento de datos en el microcontrolador
- Desarrollar el *dashboard* en la aplicación móvil para visualizar los datos de nivel de gas
- Implementar la actualización en tiempo real del *dashboard*
- Pruebas de visualización y usabilidad
- Desarrollar la funcionalidad de notificaciones *push* en la aplicación móvil
- Pruebas de notificaciones y gestión de alertas

2.1.4.5 Roles y responsabilidades

Tabla 4: Roles y Responsabilidades

Nombre	Rol	Responsabilidad
Jhony Sánchez	Gerente del local	Proporcionar los requisitos y necesidades del negocio Validar el cumplimiento de los requisitos Aprobar la implementación final
Joel Sánchez	Desarrollador	Desarrollar el prototipo IoT y la aplicación móvil Implementar la integración entre el dispositivo IoT y la aplicación móvil Realizar pruebas y solucionar problemas Documentar el proyecto

2.1.4.6 Planificación de actividades

Las historias de usuario definidas anteriormente han sido desglosadas en actividades detalladas que representan las tareas específicas necesarias para desarrollar la aplicación móvil de monitoreo de gas en cilindros domésticos. Cada historia de usuario se ha descompuesto en una serie de actividades con sus respectivas duraciones y fechas de inicio y fin planificadas. Este enfoque permite una mejor comprensión y gestión de las acciones requeridas para cumplir con los objetivos establecidos

Tabla 5: Planificación de Actividades para el proyecto

Historia de Usuario	Duración (Días)	Actividades	Duración de las Actividades (Días)	Inicio	Fin
1	14	Desarrollar el código para el microcontrolador que permita la conexión a la red Wi-Fi utilizando las credenciales proporcionadas por el usuario	7	11/03/2024	18/03/2024
		Configurar el microcontrolador para establecer una conexión Wi-Fi segura y estable	3	19/03/2024	22/03/2024
		Instalación del <i>Broker</i> en una máquina virtual	2	23/03/2024	25/03/2024
		Configuración del <i>Broker</i>	2	26/03/2024	28/03/2024
2	10	Configurar un servidor <i>backend</i> para recibir y procesar los datos del nivel de gas enviados por el dispositivo IoT	3	29/03/2024	01/04/2024
		Seleccionar y configurar InfluxDB como la base de datos para almacenar los datos del nivel de gas de manera eficiente y segura.	2	02/04/2024	04/04/2024
		Desarrollar la lógica para recibir y procesar los datos del nivel de gas enviados por el dispositivo IoT.	5	09/04/2024	14/04/2024
3	40	Diseñar la estructura y el diseño del <i>dashboard</i> en la aplicación móvil.	10	15/04/2024	25/04/2024

		Desarrollar la lógica para recuperar los datos del nivel de gas almacenados en la base de datos.	10	26/04/2024	06/05/2024
		Implementar elementos visuales para representar los datos del nivel de gas de forma clara y comprensible.	10	07/05/2024	17/05/2024
		Configurar la actualización automática del <i>dashboard</i> para reflejar los cambios en tiempo real	10	18/05/2024	28/05/2024
4	14	Diseñar la estructura y el formato de las notificaciones push en la aplicación móvil	4	29/05/2024	02/06/2024
		Desarrollar la lógica para generar y enviar notificaciones push cuando se detecten niveles críticos de gas.	5	03/06/2024	08/06/2024
		Configurar las notificaciones push para que interactúe con la aplicación móvil	5	09/06/2024	14/06/2024

2.1.4.7 Iteraciones

Detallas las actividades necesarias para desarrollar la aplicación móvil destinada al monitoreo de gas en cilindros domésticos, se inicia la fase de iteración ejecutándose un conjunto de iteraciones sucesivas.

2.1.4.7.1 Iteración 1: Conexión del microcontrolador a la red (14 días)

Desarrollar el código para el microcontrolador que permita la conexión a la red Wi-Fi utilizando las credenciales proporcionadas por el usuario.

Configurar el microcontrolador para establecer una conexión Wi-Fi segura y estable.

Instalación del Broker en una máquina virtual.

Configuración del Broker.

2.1.4.7.2 Iteración 2: Almacenamiento y gestión de datos (10 días)

Configurar un servidor backend para recibir y procesar los datos del nivel de gas enviados por el dispositivo IoT.

Seleccionar y configurar InfluxDB como la base de datos para almacenar los datos del nivel de gas de manera eficiente y segura. Desarrollar la lógica para recibir y procesar los datos del nivel de gas enviados por el dispositivo IoT.

2.1.4.7.3 Iteración 3: Visualización de Dashboard (40 días)

Diseñar la estructura y el diseño del dashboard en la aplicación móvil.

Desarrollar la lógica para recuperar los datos del nivel de gas almacenados en la base de datos.

Implementar elementos visuales para representar los datos del nivel de gas de forma clara y comprensible.

Configurar la actualización automática del dashboard para reflejar los cambios en tiempo real.

2.1.4.7.4 Iteración 4: Gestión de Notificaciones (14 días)

Diseñar la estructura y el formato de las notificaciones push en la aplicación móvil.

Desarrollar la lógica para generar y enviar notificaciones push cuando se detecten niveles críticos de gas.

Configurar las notificaciones push para que interactúen con la aplicación móvil.

2.1.5 Fase 2 (Diseño)

2.1.5.1 Arquitectura orientada a microservicios

La elección de la arquitectura de microservicios para este proyecto se fundamenta en su capacidad para ofrecer modularidad y escalabilidad. Al descomponer la aplicación en pequeños servicios independientes, cada uno centrado en una función específica, se facilita la gestión y el mantenimiento del sistema en su conjunto. En el contexto de nuestra aplicación móvil para el monitoreo de gas en cilindros domésticos, esta arquitectura nos permite desarrollar y desplegar componentes individualmente, lo que agiliza el proceso de desarrollo y permite la actualización continua de cada servicio sin afectar al resto del sistema. Además, al distribuir la carga de trabajo entre diferentes servicios aquellos que requieran mayor capacidad, sin afectar al rendimiento global de la aplicación.

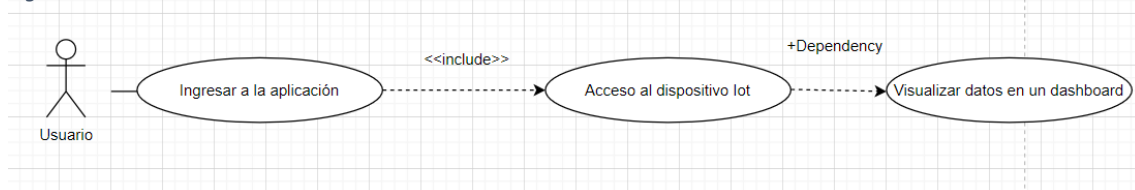
2.1.5.2 Funcionalidades del Sistema

2.1.5.2.1 Casos de Uso

En este apartado se desarrollan los casos de uso que representan las funcionalidades y los actores involucrados en la aplicación móvil del proyecto. Estos casos de uso son fundamentales para definir cómo interactúan los usuarios con la aplicación, estableciendo las acciones principales y los objetivos que se deben cumplir.

En la Figura 4 se presenta el caso de uso general del sistema:

Figura 4: Caso de Uso General del Sistema



Nota. Fuente – Elaboración propia

De otra forma en la figura 5 se detallan los casos de uso específico para el rol de usuario:

Figura 5: Caso de Uso Usuario

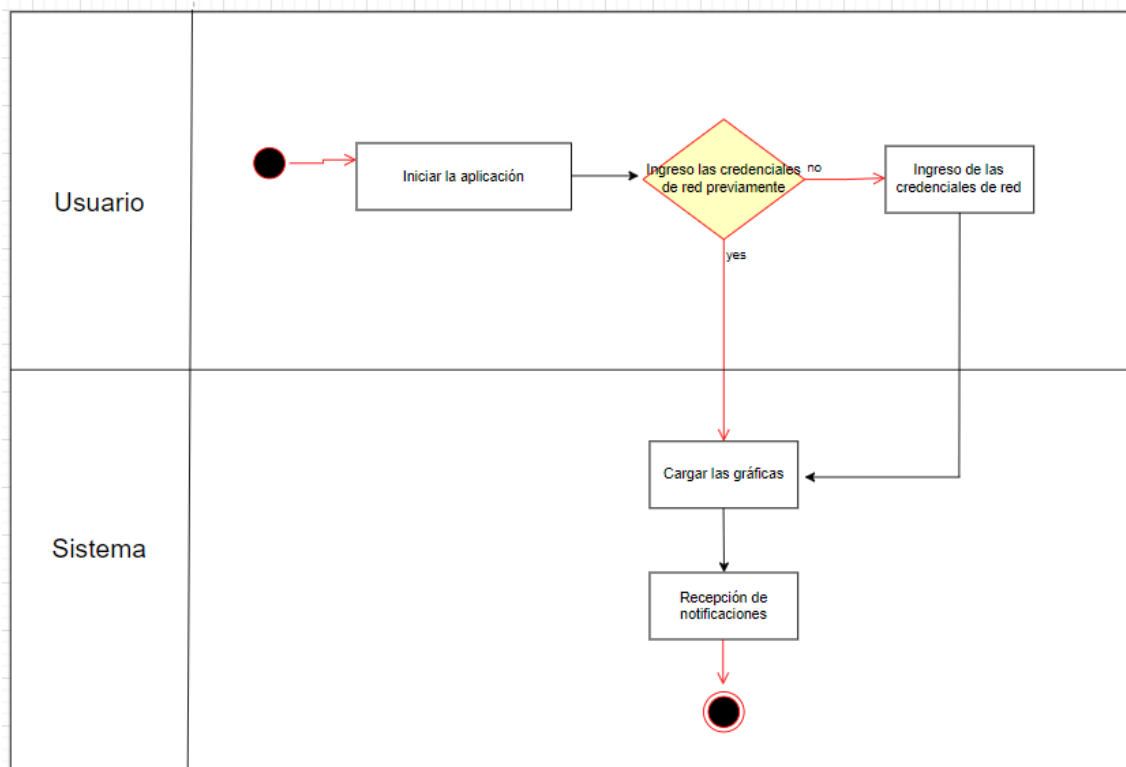


Nota. Fuente – Elaboración propia

2.1.5.2 Diagrama de Actividades

A continuación, se presenta los diagramas con las decisiones y acciones que tendrá el usuario al interactuar con la aplicación. En la Figura 6 se puede observar las actividades correspondientes que se realizan

Figura 6: Diagrama de actividades del Usuario



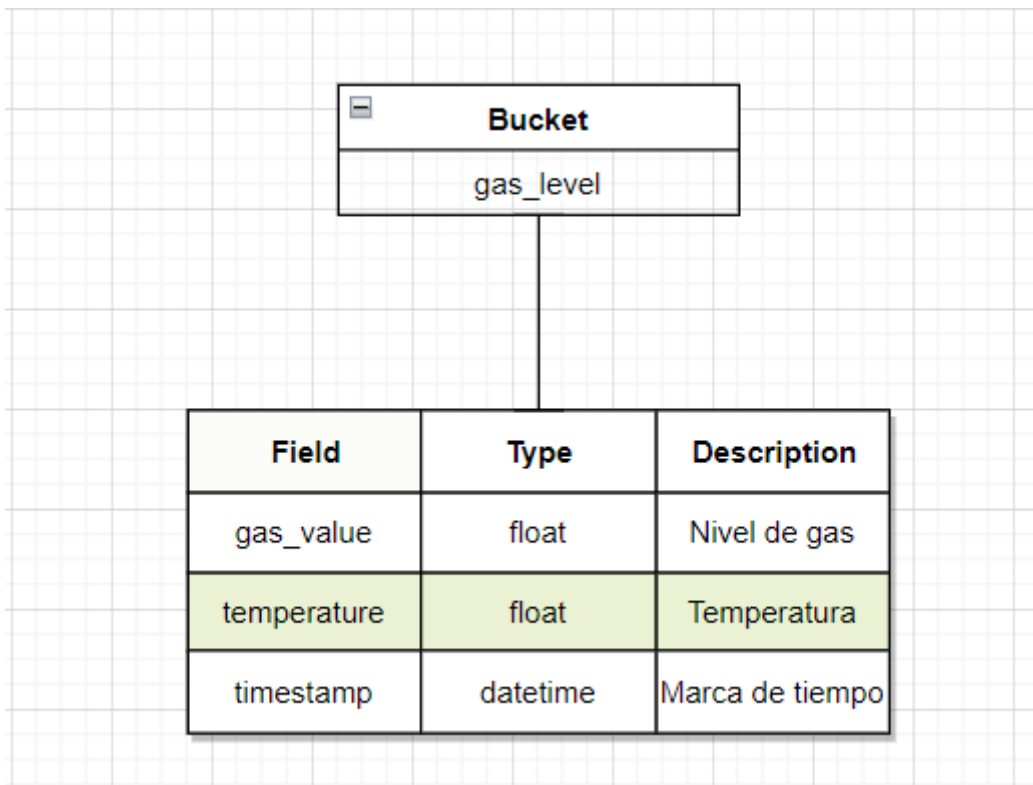
Nota. Fuente – Elaboración propia

2.1.5.2.3 Base de Datos

InfluxDB es la base de datos NoSQL seleccionada para este proyecto debido a sus beneficios en aplicaciones IoT y el monitoreo en tiempo real. InfluxDB está optimizada para el manejo eficiente de series de tiempo y mediciones continuas, lo cual se ajusta perfectamente a las necesidades de almacenar y procesar los datos de nivel de gas enviados por el dispositivo IoT a intervalos frecuentes. A diferencia de las bases de datos SQL tradicionales, InfluxDB no genera errores si falta algún dato en la secuencia temporal, lo que la hace más robusta y tolerante a fallas en el flujo de datos.

La estructura de InfluxDB se basa en mediciones, que en este caso representan las lecturas de nivel de gas, junto con campos para almacenar valores adicionales y etiquetas para metadatos relevantes como la ubicación del cilindro o el identificador del dispositivo. Esta arquitectura flexible y orientada a series de tiempo facilita el almacenamiento, consulta y análisis de los datos de monitoreo de manera eficiente y escalable y está estructurado como se observa en la Figura 7.

Figura 7: Diagrama NoSQL

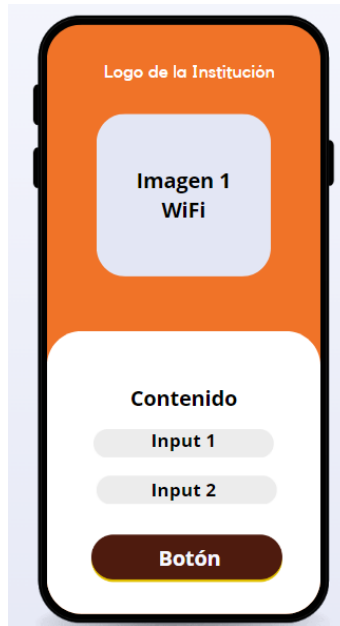


Nota. Fuente – Elaboración propia

2.1.5.2.4 Diseño de la Interfaz

En la Figura 8 se muestra la interfaz gráfica de la aplicación móvil, el cual sirvió como punto de partida para distribuir los elementos que irán asignados a la misma.

Figura 8: Diseño de la interfaz



Nota. Fuente – Elaboración propia

Igualmente, como se observa en la Figura 9, se tiene un esquema de cómo se distribuirá la pantalla.

Figura 9: Distribución de pantalla



Nota. Fuente – Elaboración propia

2.1.6 Fase 3 (Codificación)

Para el desarrollo del backend, se ha optado por TypeScript como el lenguaje de programación, un superconjunto de JavaScript que añade tipado estático y otras características modernas. Complementariamente, se utilizó NestJS, un *framework* progresivo de Node.js que promueve el uso de patrones de diseño y arquitecturas escalables.

En el caso del frontend, se ha seleccionado Flutter, un *framework* de código abierto desarrollado por Google especializado en la construcción de aplicaciones nativas multiplataforma. Flutter utiliza Dart como lenguaje de programación, conocido por su alto rendimiento y facilidad de aprendizaje, y compila el código a binarios nativos.

2.1.6 Fase 4 (Pruebas)

Para asegurar el correcto funcionamiento del sistema, se definió un plan de pruebas tanto como la aplicación móvil como se puede observar en la Tabla 6 y posteriormente otro plan de pruebas para el prototipo Iot que se observa en la tabla 7

Tabla 6: Pruebas Funcionales Aplicación Móvil

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado
1	Configuración de Red Wi-Fi desde la Aplicación Móvil	Verificar la configuración correcta de la red Wi-Fi desde la aplicación móvil	Ingreso de credenciales correctas (SSID: WIFI-ALPHAFOOD, Contraseña: XYZ123)	La aplicación muestra un mensaje de conexión exitosa y viaja a la siguiente pantalla
			Ingreso de credenciales incorrectas (SSID: WIFI-ALPHAFOOD, Contraseña: XYZ)	La aplicación muestra un mensaje de error indicando que las credenciales son incorrectas
			Dejar los campos de las credenciales vacías	La aplicación muestra un mensaje de error indicando que las credenciales son incorrectas
2	Almacenamiento y Procesamiento de Datos en el Backend	Verificar que el backend pueda recibir, procesar y almacenar correctamente los datos del nivel de gas	Recepción de datos del nivel de gas	El backend recibe los datos sin errores y muestra un registro de la transacción
			Procesamiento de datos en el backend	Los datos del nivel de gas se procesan correctamente y se almacenan en la base de datos InfluxDB
			Conexión segura y confiable	La conexión entre el dispositivo IoT y el backend es segura y confiable, sin pérdida de datos
3	Confirmación de Datos en Tiempo Real	El sistema debe actualizar los dashboard en tiempo real	Recepción de datos a través de la base de datos	El dashboard se actualiza de una forma dinámica
			No recibe recepción a través de la base de datos	El dashboard se queda con el último dato recuperado
4	Diseño y Funcionalidad del Dashboard	Verificar que el dashboard de la aplicación móvil muestre de manera clara y comprensible los datos del nivel de gas	Acceso al dashboard de la aplicación móvil	El diseño del dashboard facilita la visualización y comprensión de los datos del nivel de gas

5	Notificaciones en Tiempo Real	El sistema debe enviar notificaciones push	Recepción de datos por debajo de los niveles establecidos	Envío de notificaciones push para estar alerta
			Recepción de datos por encima de los niveles establecidos	No se envían notificaciones push
6	Gestión de Permisos	Verificar que la aplicación solicite permisos para activar notificaciones y Bluetooth según sea necesario	Pedir permisos para activar notificaciones	La aplicación muestra un diálogo solicitando permisos para activar notificaciones antes de proceder
			Pedir permisos para activar Bluetooth	La aplicación muestra un diálogo solicitando permisos para activar Bluetooth antes de proceder

De la misma forma en la tabla 7 se realiza el plan de pruebas que dará validez al funcionamiento del prototipo Iot:

Tabla 7: Pruebas Funcionales Prototipo

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado
1	Lectura de Niveles de GLP	El prototipo debe ser capaz de leer los niveles de GLP	Lectura de niveles de GLP con sensor en condiciones normales	Lectura precisa y estable de los niveles de GLP
			Lectura de niveles de GLP con sensor en condiciones de baja batería	Lectura precisa y estable de los niveles de GLP, sin interferencia de la baja batería
2	Envío de Datos al Backend	El prototipo debe ser capaz de enviar los datos al backend	Envío de datos con conexión Wi-Fi estable	Los datos se envían correctamente al backend
			Envío de datos con pérdida momentánea de conexión Wi-Fi	Los datos se envían una vez se restablece la conexión
3	Integración de Sensores Adicionales	Verificar la integración con otros sensores	Integración de sensor L06 Ultrasonic	Lectura precisa de la distancia (nivel de GLP)
4		Verificar el funcionamiento del prototipo en diferentes	Funcionamiento enchufado a un toma corriente	El dispositivo envía los datos sin ninguna complicación

	Gestión de Energía	condiciones de energía		
--	-----------------------	---------------------------	--	--

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

En este capítulo, se presentan los resultados obtenidos durante el desarrollo del sistema de monitoreo de gas en cilindros domésticos mediante un prototipo IoT y una aplicación móvil. Se detallan los hallazgos y observaciones derivados de las pruebas realizadas en condiciones reales, así como la validación del cumplimiento de los requisitos funcionales y no funcionales establecidos inicialmente. Además, se analizan los desafíos y limitaciones encontrados durante el proceso de desarrollo, y se discuten las posibles soluciones y mejoras implementadas para abordarlos

3.1 Resultados Fase 1. Aplicación Móvil

3.1 Interfaz de conectividad

La primera fase del proyecto consistió en el desarrollo de la aplicación móvil. En la Figura 10 se muestra la interfaz de conexión, que es la primera pantalla que el usuario encuentra al abrir la aplicación por primera vez. Esta interfaz permite establecer la conexión entre el dispositivo móvil y el dispositivo IoT a través de Bluetooth. Si se desea volver a configurar las credenciales, se puede hacer a través de un botón flotante ubicado en la pantalla principal. La relevancia de esta interfaz radica en su simplicidad y facilidad de uso, asegurando que los usuarios puedan conectar su dispositivo IoT de manera eficiente.

Figura 10: Interfaz de conectividad



3.2 Interfaz de Credenciales

En la Figura 11 se muestra la interfaz de credenciales, esta permite al usuario ingresar y enviar las credenciales de la red Wi-Fi al dispositivo IoT. Esta interfaz se caracteriza por ser sencilla solicitando al usuario el nombre de la red (SSID) y la contraseña correspondiente. Una vez ingresadas las credenciales, el usuario puede enviarlas al dispositivo IoT, que intentará conectarse a la red Wi-Fi utilizando estas credenciales.

Figura 11: Interfaz de credenciales



The image shows a mobile application interface with an orange background. At the top, there is a status bar with the time 10:45, signal strength, Wi-Fi, and battery icons. Below the status bar is a back arrow and the 'Alpha' logo. The main heading reads 'Dispositivo Conectado' followed by 'ESP32-C3'. In the center is a stylized illustration of an ESP32-C3 microcontroller board. Below this is a white rounded rectangle containing the form. The form has a title 'Ingrese sus credenciales' and two input fields: 'Nombre de la Red:' with a placeholder 'Ingrese el nombre de la red' and 'Contraseña:' with a placeholder 'Ingrese la contraseña de la red'. At the bottom of the form is an orange 'Guardar' button.

3.3 Interfaz de Gráficas

En la Figura 12 se muestra la interfaz gráfica donde el usuario puede monitorear en tiempo real el estado del cilindro de gas. En esta pantalla, se muestran de manera clara y concisa los datos recopilados por el dispositivo IoT, como el peso actual del cilindro de gas y la temperatura ambiente. Además, la interfaz de gráficas proporciona alertas y notificaciones cuando los niveles de gas caen por debajo de un umbral crítico

Figura 12: Interfaz de Gráficas



3.2 Resultados Fase 2. Prototipo Iot

3.4 Prototipo Finalizado

El desarrollo del prototipo IoT para el monitoreo de GLP se llevó a cabo en varias fases. Inicialmente, se utilizó una placa de expansión para conectar el sensor y el microcontrolador, facilitando la integración y prueba de los componentes esenciales del sistema.

Una vez que el prototipo demostró ser funcional, se procedió a soldar el sensor directamente al microcontrolador, eliminando la necesidad de la placa de expansión. Este paso permitió reducir el tamaño del dispositivo y mejorar su robustez.

Finalmente, se integró una batería soldada directamente al prototipo, eliminando así el uso de un cable de alimentación y haciendo el dispositivo completamente autónomo. Además, se diseñó una caja utilizando tecnología de impresión 3D, proporcionando un alojamiento robusto y compacto para todos los componentes electrónicos. Este diseño asegura la protección adecuada del dispositivo mientras facilita su instalación en entornos domésticos.

El prototipo final representa el resultado optimizado del proceso de desarrollo, preparado para proporcionar un monitoreo preciso y continuo de los niveles de GLP en tiempo real.



Figura 13: Prototipo Final

3.3 Pruebas del sistema

Para garantizar el correcto funcionamiento del sistema de monitoreo de GLP, se realizaron una serie de pruebas funcionales para evaluar cada una de sus funcionalidades clave. Estas pruebas fueron diseñadas y ejecutadas para validar que el sistema cumpla con los requisitos establecidos, asegurando que el sistema funcione según lo esperado. Durante este proceso, se cubrieron diversos escenarios posibles, incluyendo conectividad, procesamiento de datos en tiempo real, gestión de notificaciones, configuración de la red Wi-Fi, almacenamiento de datos en el *backend*, diseño del dashboard y gestión de permisos.

En la Tabla 8 se muestran los resultados de las pruebas que tenían como objetivo verificar que el dispositivo IoT se conectara a la red WiFi utilizando las credenciales proporcionadas. Se evaluaron varios casos de pruebas, incluyendo credenciales correctas, incorrectas y campos vacíos, para garantizar que el dispositivo respondiera de forma adecuada ante estas situaciones y enviara alertas a la aplicación móvil en caso de que la conexión a la red no se haya concretado.

Tabla 8: Prueba de Configuración de Red Wi-Fi

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
1	Configuración de Red Wi-Fi desde la Aplicación Móvil	Verificar la configuración correcta de la red Wi-Fi desde la aplicación móvil	Ingreso de credenciales correctas (SSID: WIFI-ALPHAFOOD, Contraseña: XYZ123)	La aplicación muestra un mensaje de conexión exitosa y viaja a la siguiente pantalla	La aplicación muestra un mensaje de conexión exitosa y viaja a la siguiente pantalla
			Ingreso de credenciales incorrectas (SSID: WIFI-ALPHAFOOD, Contraseña: XYZ)	La aplicación muestra un mensaje de error indicando que las credenciales son incorrectas	La aplicación muestra un mensaje de error indicando que las credenciales son incorrectas
			Dejar los campos de las credenciales vacías	La aplicación muestra un mensaje de error indicando que las credenciales son incorrectas	La aplicación muestra un mensaje de error indicando que las credenciales son incorrectas

La Tabla 9 muestra los resultados obtenidos al evaluar la capacidad del backend para recibir, procesar y almacenar los datos del nivel de gas enviados por el dispositivo IoT. Esta prueba funcional busca validar la correcta recepción de los datos, el procesamiento efectivo en la base de datos InfluxDB, y una conexión segura y confiable entre el dispositivo y el backend. Se espera que el backend registre los datos correctamente, asegurando la integridad y disponibilidad de la información en todo momento.

Tabla 9: Prueba de Almacenamiento

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
2	Almacenamiento y Procesamiento de Datos en el Backend	Verificar que el backend pueda recibir, procesar y almacenar correctamente los datos del nivel de gas	Recepción de datos del nivel de gas	El backend recibe los datos sin errores y muestra un registro de la transacción	El backend recibe los datos sin errores y muestra un registro de la transacción
			Procesamiento de datos en el backend	Los datos del nivel de gas se procesan correctamente y se almacenan en la base de datos InfluxDB	Los datos del nivel de gas se procesan correctamente y se almacenan en la base de datos InfluxDB
			Conexión segura y confiable	La conexión entre el dispositivo IoT y el backend es segura y confiable, sin pérdida de datos	La conexión entre el dispositivo IoT y el backend es segura y confiable, sin pérdida de datos

En la Tabla 10 se detallan los resultados obtenidos al evaluar la capacidad del sistema para actualizar los *dashboards* en tiempo real a partir de los datos recibidos. Esta prueba funcional tiene como objetivo verificar que el dashboard de la aplicación móvil se actualice dinámicamente con los datos del nivel de gas, asegurando una visualización precisa y en tiempo real para el usuario. Se evaluaron escenarios donde los datos fueron recibidos correctamente y donde no hubo recepción, garantizando así un dashboard actualizado y funcional. La validación de la actualización en tiempo real se realizó mediante la simulación del envío de datos al *backend* y la observación directa del comportamiento del dashboard.

Tabla 10: Prueba de Confirmación de Datos

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
------------------	--------	-------------	-----------	--------------------	--------------------

3	Confirmación de Datos en Tiempo Real	El sistema debe actualizar los dashboard en tiempo real	Recepción de datos a través de la base de datos	El dashboard se actualiza de una forma dinámica	El dashboard se actualiza de una forma dinámica
			No recibe recepción a través de la base de datos	El dashboard se queda con el último dato recuperado	El dashboard se queda con el último dato recuperado

La Tabla 11 presenta los resultados obtenidos al verificar que el diseño del dashboard en la aplicación móvil muestre de manera clara y comprensible los datos del nivel de gas. Esta prueba funcional se enfoca en asegurar que los usuarios puedan acceder fácilmente al dashboard y entender la información presentada, facilitando así la interpretación de los datos y la toma de decisiones informadas. La prueba incluyó la evaluación de la usabilidad del diseño, verificando que la presentación de los datos sea intuitiva y accesible.

Tabla 11: Prueba de Funcionalidad Dashboard

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
4	Diseño y Funcionalidad del Dashboard	Verificar que el dashboard de la aplicación móvil muestre de manera clara y comprensible los datos del nivel de gas	Acceso al dashboard de la aplicación móvil	El diseño del dashboard facilita la visualización y comprensión de los datos del nivel de gas	El diseño del dashboard facilita la visualización y comprensión de los datos del nivel de gas

En la tabla 12 se muestran los resultados de evaluar la capacidad del sistema para enviar notificaciones push en tiempo real. Esta prueba busca confirmar que se envíen notificaciones cuando los datos del nivel de gas están por debajo o por encima de los límites establecidos, alertando así al usuario de posibles situaciones críticas. Se esperan notificaciones efectivas que mantengan al usuario informado y alerta ante cualquier eventualidad.

Tabla 12: Prueba de Notificaciones

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
5	Notificaciones en Tiempo Real	El sistema debe enviar notificaciones push	Recepción de datos por debajo de los niveles establecidos	Envío de notificaciones push para estar alerta	Envío de notificaciones push para estar alerta
			Recepción de datos por encima de los niveles establecidos	No se envían notificaciones push	No se envían notificaciones push

La Tabla 13 detalla los resultados obtenidos al verificar que la aplicación solicite adecuadamente los permisos necesarios para activar funciones como notificaciones y Bluetooth. Esta prueba funcional tiene como objetivo asegurar que la aplicación solicita permisos de manera clara y oportuna, garantizando una experiencia de usuario transparente y cumpliendo con los estándares de privacidad y seguridad establecidos.

Tabla 13: Prueba de Gestión de Permisos

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
6	Gestión de Permisos	Verificar que la aplicación solicita permisos para activar notificaciones y Bluetooth según sea necesario	Pedir permisos para activar notificaciones	La aplicación muestra un diálogo solicitando permisos para activar notificaciones antes de proceder	La aplicación muestra un diálogo solicitando permisos para activar notificaciones antes de proceder
			Pedir permisos para activar Bluetooth	La aplicación muestra un diálogo solicitando permisos para activar Bluetooth antes de proceder	La aplicación muestra un diálogo solicitando permisos para activar Bluetooth antes de proceder

3.4 Pruebas del Prototipo Iot

Además, se verificó la integración exitosa con sensores adicionales como el sensor L06 Ultrasonic, obteniéndose una mejora significativa en la precisión del monitoreo. Finalmente, se evaluó el desempeño del prototipo en diferentes escenarios de energía, incluyendo el uso de baterías de respaldo.

La Tabla 14, corresponde se presentan los resultados obtenidos al evaluar la capacidad del prototipo para leer con precisión los niveles de GLP. Esta prueba busca validar la lectura estable y precisa de los niveles de gas en condiciones normales y bajo baja batería, asegurando así un monitoreo confiable y continuo del GLP.

Tabla 14: Prueba de Lectura

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
1	Lectura de Niveles de GLP	El prototipo debe ser capaz de leer los niveles de GLP	Lectura de niveles de GLP con sensor en condiciones normales	Lectura precisa y estable de los niveles de GLP	Lectura precisa y estable de los niveles de GLP
			Lectura de niveles de GLP con sensor en condiciones de baja batería	Lectura precisa y estable de los niveles de GLP, sin interferencia de la baja batería	Lectura precisa y estable de los niveles de GLP, sin interferencia de la baja batería

En la Tabla 15 se presentan los resultados obtenidos al evaluar la capacidad del prototipo para enviar datos al *backend*, incluso ante pérdidas momentáneas de conexión Wi-Fi. Esta prueba tiene como objetivo garantizar que los datos sean enviados correctamente al *backend*, asegurando la integridad y disponibilidad de la información en todo momento, incluso en condiciones de conectividad inestable.

Tabla 15: Prueba de Envío de Datos

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
2	Envío de Datos al Backend	El prototipo debe ser capaz de enviar los datos al backend	Envío de datos con conexión Wi-Fi estable	Los datos se envían correctamente al backend	Los datos se envían correctamente al backend
			Envío de datos con pérdida momentánea de conexión Wi-Fi	Los datos se envían una vez se restablece la conexión	Los datos se envían una vez se restablece la conexión

En la Tabla 16, correspondiente a la Prueba 3, se detallan los resultados obtenidos al verificar la integración del prototipo con sensores adicionales, como es el sensor L06 Ultrasonic. Esta prueba tiene como objetivo validar que el prototipo pueda integrarse adecuadamente con otros sensores para mejorar la precisión y funcionalidad del monitoreo de GLP.

Tabla 16: Prueba de Integración

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
3	Integración de Sensores Adicionales	Verificar la integración con otros sensores	Integración de sensor L06 Ultrasonic	Lectura precisa de la distancia (nivel de GLP)	Lectura precisa de la distancia (nivel de GLP)

La tabla 17 de la Prueba 4 muestra los resultados obtenidos al evaluar el funcionamiento del prototipo en diferentes condiciones de energía, incluyendo el uso de batería de respaldo. Esta prueba busca asegurar que el prototipo pueda operar de manera continua y efectiva, independientemente de las condiciones de suministro eléctrico, garantizando así un monitoreo ininterrumpido del nivel de GLP.

Tabla 17: Prueba de Gestión de energía

Número de Prueba	Nombre	Descripción	Escenario	Resultado Esperado	Resultado Obtenido
4	Gestión de Energía	Verificar el funcionamiento del prototipo en diferentes condiciones de energía	Funcionamiento enchufado a un toma corriente	El dispositivo envía los datos sin ninguna complicación	El dispositivo envía los datos sin ninguna complicación

CONCLUSIONES

Durante el desarrollo del sistema de monitoreo de GLP, se ha puesto un énfasis significativo en la integración entre la aplicación móvil y el dispositivo IoT. Este enfoque no solo buscó mejorar la seguridad y eficiencia del monitoreo de gas, sino también, garantizar una buena experiencia de usuario. Las pruebas unitarias, junto con la implementación de metodologías ágiles como XP y técnicas de prototipado de dispositivos IoT, han sido fundamentales para garantizar el funcionamiento de cada aspecto del sistema garantizando su funcionalidad integral.

- El desarrollo de la aplicación móvil contribuyó a mejorar el monitoreo del GLP, ofreciendo una solución segura y confiable para los usuarios finales. La implementación de la app permitió una supervisión precisa y continua de los niveles de gas, asegurando así entornos seguros y operativos.
- El uso de *frameworks* fue fundamental para organizar y estructurar de manera eficiente el código del proyecto. Esta estrategia no solo facilitó el desarrollo y la mantenibilidad del software, sino que también optimizó los procesos de implementación y pruebas, asegurando estándares de calidad y rendimiento.
- La aplicación de la metodología XP y el prototipado en el proyecto posibilitó ajustes continuos según la retroalimentación recibida, asegurando que tanto la aplicación móvil como el dispositivo IoT evolucionaron conforme a los requisitos del usuario final.
- La exitosa integración entre la aplicación móvil y el dispositivo IoT estableció un entorno integrado y funcional. Esta colaboración mejoró significativamente la operatividad del sistema de monitoreo de GLP, asegurando una experiencia de usuario mejorada y una operación fluida y efectiva.

RECOMENDACIONES

Con el objetivo de optimizar el rendimiento y garantizar la eficiencia del sistema de monitoreo de GLP, se han identificado una serie de recomendaciones clave. Estas recomendaciones están diseñadas para mejorar la integración de sensores, asegurar la estabilidad operativa y facilitar la gestión escalable de usuarios y dispositivos. Desde la selección de sensores comerciales con documentación robusta hasta la implementación de medidas estratégicas para la protección y ubicación de los dispositivos, cada recomendación busca fortalecer la infraestructura y funcionalidad del sistema.

- **Utilización de Sensores Comerciales y Documentación Disponible**

Se recomienda utilizar sensores comerciales ampliamente documentados y con ejemplos de uso disponibles. Esto garantizará una integración más fluida y facilitará el desarrollo y la comprensión del funcionamiento del sistema. Además, se sugiere realizar pruebas exhaustivas con los sensores antes de la implementación para asegurar su funcionamiento óptimo en el entorno específico de monitoreo de GLP.

- **Ubicación Estratégica de Sensores**

Es crucial colocar los sensores en lugares protegidos donde no estén expuestos a condiciones climáticas severas como lluvia o luz solar directa. Esto ayudará a preservar la vida útil del sensor y mantener la precisión de las mediciones a lo largo del tiempo. Asimismo, se recomienda instalar los sensores en áreas con buena cobertura de señal de internet para asegurar una comunicación estable y continua con el *backend*.

- **Expansión de la Capacidad de la Base de Datos**

Para soportar la creciente demanda de usuarios y dispositivos, se sugiere expandir la capacidad de almacenamiento de la base de datos.

Integrar la dirección MAC del dispositivo IoT como identificador único permitirá una gestión más eficiente y personalizada de cada dispositivo. Esta medida garantizará la consistencia operativa y la escalabilidad del sistema a medida que se expanda a más usuarios.

- **Implementación de un Módulo de Autenticación y Gestión de Usuarios**

Se recomienda implementar un módulo de autenticación y gestión de usuarios en la aplicación móvil. Este módulo facilitará la asociación segura de cada dispositivo IoT con un usuario específico mediante un proceso de creación de cuentas y autenticación robusto. Los usuarios podrán gestionar y monitorear sus dispositivos de manera individualizada, mejorando así la seguridad y el control del sistema global de monitoreo de GLP.

REFERENCIA BIBLIOGRÁFICA

- Bastidas, m. (2020). Desarrollo de un wearable para gas de uso doméstico. Universidad politécnica salesiana sede quito. Obtenido de <http://riut.utfpr.edu.br/jspui/bitstream/1/25878/1/sistemamonitorementoresidencial.pdf>
- Basumatary, B., & Agnihotri, N. (2022, May 30). Benefits and challenges of using NodeJS. <https://acspublisher.com/journals/index.php/ijrcst/article/view/10433>
- Canive, T. (2020, 27 mayo). Metodología XP o programación extrema: ¿Qué es y cómo aplicarla? Gestor de proyectos online. <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>
- Danielsson, W. (2016). *React Native application development*. <https://www.diva-portal.org/smash/get/diva2:998793/FULLTEXT02.pdf>
- Documentation | NestJS - A progressive Node.js framework. (n.d.). Documentation | NestJS - a Progressive Node.js Framework. <https://docs.nestjs.com/>
- Espressif Systems. (2021). ESP32-C3 Series Datasheet v1.7.
- FISICC. (2023, August 18). Telecomunicaciones: ¿Cuántos objetos están conectados en el mundo? - FISICC. FISICC - Facultad de Ingeniería de Sistemas, Informática y Ciencias de la Computación. <https://www.galileo.edu/fisicc/historias-de-exito/telecomunicaciones-cuantos-objetos-estan-conectados-en-el-mundo/>
- Ibarra, S., Quispe, J., Mullicundo, F., & Lamas, D. (2021). HERRAMIENTAS Y TECNOLOGÍAS PARA EL DESARROLLO WEB DESDE EL FRONTEND AL BACKEND. <https://sedici.unlp.edu.ar/bitstream/handle/10915/120476/Ponencia.pdf-PDFA.pdf?sequence=1&isAllowed=y>

- IEEE. (12 de 2019). Internet of Things (IoT): A Survey. Obtenido de <https://ieeexplore.ieee.org/document/9105831>
- Khalifeh, Mazunga, Nechibvute, & Nyambo. (2022). Microcontroller Unit-Based Wireless Sensor Network Nodes: A Review.
- Light. (2017). Mosquitto: server and client implementation of the MQTT protocol.
- Ministerio de Energía y Minas. (2021). Balance Energético Nacional. Obtenido de Balance Energético Nacional: https://www.rekursyenergia.gob.ec/wp-content/uploads/2022/08/Balance_Energético_Nacional_2021-VF_opt.pdf
- Mishra, B., Kertesz, A., & IEEE. (2021). Use of MQTT in M2M and IoT Systems: A Survey.
- ¿Qué es el Internet de las cosas (IoT)? (s. f.). Oracle Argentina.
<https://www.oracle.com/ar/internet-of-things/what-is-iot/>
- Ponte, G. B. (2019). Desarrollo de sistema de monitoreo remoto de consumo de gas envasado utilizando tecnología IOT.
- Pradyumna Gokhale, O. B. (2018). Introduction to IOT. International Advanced Research Journal in Science, Engineering and Technology.
- R A Atmoko, R. R. (2017). IoT real time data acquisition using MQTT protocol. International Conference on Physical Instrumentation and Advanced Materials, 3.
- Red Hat. (20 de 01 de 2023). ¿Qué es el Internet de las cosas (IoT)? Obtenido de ¿Qué es el Internet de las cosas (IoT)?: <https://www.redhat.com/es/topics/internet-of-things/what-is-iot#:~:text=¿Cómo%20funciona%20el%20internet%20de,en%20todo%20tipo%20de%20objetos.>
- Renzel, D., Klamma, R., & Jarke, M. (2013). From minds to machines: APIs as boundary objects in the internet of services. International Journal of Web Engineering and Technology, 8(3), 261-281.
- Sinnaps. (27 de 05 de 2020). METODOLOGÍA XP O PROGRAMACIÓN EXTREMA. Obtenido de METODOLOGÍA XP O PROGRAMACIÓN EXTREMA: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>

Tashildar, Shah, Gala, Giri, & Chavhan. (2020). APPLICATION DEVELOPMENT USING FLUTTER.

https://d1wqtxts1xzle7.cloudfront.net/64302984/APPLICATION_DEVELOPMENT_USING_FLUTTER_-libre.pdf?1598690095=&response-content-disposition=inline%3B+filename%3DAPPLICATION_DEVELOPMENT_USING_FLUTTER.pdf&Expires=1712351999&Signature=SIux8io272sJ77IurGL2nT6cOYPcqsopP5izBBMjismG-r~nbZ7EpEkzD9JsWtUqJCeXaFG-i2TCeZiQR6JA4RVuSDb8rjW7KpMg94JtqM65h8reWK6-zONZkjQhqjzAUAgXhZj9TKyB2uWAJR2h08u8cJDe55TENzrHCiv4Gzt8G4R8y3BHfSj7yqlVJ4YZJupN-i0B0zNkIm00kgLtfv~FahKIuaJ9EvwuislMOVgkyIHrpWybj5JLS7P6KeZcFhTsTb72um3UBIAqGEE~D2P4w675g8BRQPCnwrP8bQpqTEN-4Ao4MOM1e7dzZAAOc1EWcf3WOsqIu7cJjfYQug__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

Team, HiveMQ. (2023, 27 noviembre). A Beginner's Guide to MQTT Brokers. A Beginner's Guide to MQTT Brokers. <https://www.hivemq.com/blog/mqtt-brokers-beginners-guide>

Torres, & Fernández. (2023). Sensores y Actuadores. https://rua.ua.es/dspace/bitstream/10045/18433/1/Tema%202_Sensores%20y%20Detectores.pdf

Unir, V. (2023, September 27). Framework: qué es, para qué sirve y algunos ejemplos. *UNIR FP*. <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>

ANEXOS

Anexo 1. Entrevista

- ¿Actualmente utilizas alguna tecnología para monitorear o gestionar aspectos relacionados con el negocio, como el nivel de gas?
- ¿Cómo se lleva a cabo actualmente el monitoreo del nivel de gas en los cilindros domésticos en tu local?
- ¿Cuáles son las principales necesidades o desafíos que enfrentas en relación con el monitoreo del nivel de gas en los cilindros domésticos?
- ¿Has experimentado alguna vez situaciones de emergencia o riesgo relacionadas con el gas en tu local?
- ¿Qué esperas lograr implementando un sistema de monitoreo del nivel de gas en los cilindros domésticos?
- ¿Cuáles son tus principales objetivos al utilizar esta tecnología en tu establecimiento?
- ¿Cómo te imaginas interactuando con una aplicación móvil para monitorear el nivel de gas en tus cilindros domésticos?
- ¿Qué funcionalidades te gustaría que tuviera esta aplicación para facilitar la gestión y el monitoreo del gas en tu local?
- ¿Consideras importante recibir notificaciones o alertas en tiempo real sobre el nivel de gas en tus cilindros domésticos?
- ¿En qué situaciones te gustaría recibir estas notificaciones y cómo deberían ser estas alertas?
- ¿Qué medidas de seguridad consideras importantes para proteger tanto a tus clientes como a tu personal en relación con el uso de gas en tu local?
- ¿Qué tipo de seguimiento o mantenimiento esperarías después de la implementación del sistema?

Anexo 2. Carta de Aceptación



Ibarra, 07 de julio del 2024

Yo **JHONNY FERNANDO SÁNCHEZ ESPINOSA**, con cédula 1003267661, propietario del restaurante **ALPHA FOOD AND DRINKS**, tengo el agrado de comunicar que el Sr. **CRISTOPHER JOEL SÁNCHEZ PERUGACHI**, con cédula 1004098511, realizó el proyecto de titulación denominado "**APLICACIÓN MÓVIL PARA MONITOREAR LA CANTIDAD DE GAS EN CILINDROS DOMÉSTICOS MEDIANTE UN PROTOTIPO IOT**". Este proyecto fue desarrollado en beneficio de nuestro restaurante, permitiéndonos llevar a cabo el monitoreo del GLP.

Es importante mencionar que el proyecto cumplió con las expectativas y satisfacción de nuestra empresa. Además, el autor, Sr. Christopher Joel Sánchez Perugachi, demostró profesionalidad y compromiso durante todo el desarrollo del mismo.

Atentamente,|

Mgs. Jhonny Sánchez

Propietario

Alpha Food and Drinks