



"Responsabilidad con pensamiento positivo"

UNIVERSIDAD TECNOLÓGICA ISRAEL

**TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:
INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES**

TEMA:

**SISTEMA AUTOMÁTICO PARA PREVENIR LA EXPLOSIÓN POR FUGA DE GAS
LICUADO DE PETRÓLEO EN UN DOMICILIO**

AUTOR:

LUIS ALBERTO LLERENA GUAMÁN

TUTORA:

**MSc. MARTÍNEZ MOSQUERA SILVIA DIANA
ING. LUIS HERNÁN MONTOYA LARA
ING. RENÉ ERNESTO CORTIJO LEYVA**

QUITO, ECUADOR

2018

UNIVERSIDAD TECNOLÓGICA ISRAEL

CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación “**SISTEMA AUTOMÁTICO PARA PREVENIR LA EXPLOSIÓN POR FUGA DE GAS LICUADO DE PETRÓLEO EN UN DOMICILIO.**”, presentado por el Sr. Alberto Llerena, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito D.M., 03 Septiembre de 2018

TUTOR

Ing. Martínez Mosquera Silvia Diana, MSc.

C.I. 1718478603

AGRADECIMIENTO

Gracias a Mi Dios y Mi Virgencita por permitirme tener a mi esposa y mis dos hijas, por apoyarme; y permitirme cumplir con excelencia en el desarrollo de este proyecto. Gracias por creer en mí. No ha sido sencillo el camino hasta ahora, pero gracias por su amor, su bondad y por el apoyo. Les agradezco mi hermosa familia.

Le agradezco a la institución y a todos mis profesores por sus esfuerzos para que finalmente me pudiera graduarme como profesional.

DEDICATORIA

A mi Virgencita del Quinche, por la oportunidad de vivir y estar conmigo en cada paso que doy, además de fortalecer mi corazón e iluminar mi mente, poner en mi camino a esas personas que han sido mi soporte durante mi carrera universitaria.

Mi esposa Haydeé, por quererme mucho, creer en mí y porque siempre me apoyaste. Y gracias por darme una carrera para mi futuro, todo esto te lo debo a ti.

Mis suegros por quererme y apoyarme siempre, esto también se lo debo a ustedes.

Mis hijas Naomi y Noelia, para que vean en mí un ejemplo a seguir.

TABLA DE CONTENIDO

PORTADA	I
CERTIFICACIÓN DEL TUTOR	II
AGRADECIMIENTO	III
DEDICATORIA	IV
TABLA DE CONTENIDO.....	V
LISTA DE FIGURAS	VIII
LISTA DE TABLAS	X
RESUMEN	XI
ABSTRACT	XII
INTRODUCCIÓN.....	1
ANTECEDENTES	2
PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA	3
OBJETIVO GENERAL	5
OBJETIVOS ESPECÍFICOS	5
DESCRIPCIÓN DE LOS CAPÍTULOS	5
ALCANCE	6
1 CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
1.1 ANTECEDENTES	7
1.1.1 Gas licuado de petróleo	7
1.1.2 Comercialización en el Ecuador	11
1.2 MARCO TEÓRICO	12
1.2.1 Arduino.....	12
1.2.2 Protocolo Estándar IEEE 802.11 Wi-Fi	14
1.2.3 Módulos Wi-Fi	16
1.2.4 Electroválvulas	20
1.2.5 Sensores de gas.....	24
1.2.6 Norma NTEINEN 2260 (Instalaciones de gases combustibles para uso residencial, comercial e industrial)	26
1.2.7 Android Studio	28

2	CAPÍTULO 2: PROPUESTA	31
2.1	SOLUCIÓN TECNOLÓGICA DEL PROYECTO	32
2.2	DIAGRAMA DE FLUJO DEL SISTEMA	32
2.3	DIAGRAMA DE BLOQUES	34
2.3.1	Diseño del hardware	34
2.3.2	Etapa de entrada	34
2.3.3	Etapa de control	35
2.3.4	Etapa de salida	35
2.3.5	Diseño de software	35
2.3.6	Etapa de entrada	35
2.3.7	Etapa de control	36
2.3.8	Etapa de salida	36
2.4	ANÁLISIS DE COSTOS Y TIEMPO REQUERIDO	36
2.5	VENTAJAS DEL PRODUCTO	37
2.6	CÁLCULOS DE VOLTAJES Y CORRIENTES EN EL CIRCUITO DE ACTIVACIÓN DE LA ELECTROVÁLVULA	37
3	CAPITULO 3: IMPLEMENTACIÓN	40
3.1	DESARROLLO	40
3.1.1	Esquema de proyecto	40
3.1.2	Construcción del prototipo	43
3.1.3	Implementación del hardware	45
3.1.4	Implementación del software	46
3.1.5	Programación en Arduino	46
3.1.6	Implementación total del sistema	64
3.2	PRUEBAS DE FUNCIONAMIENTO	68
3.2.1	Chequeo inicial	68
3.2.2	Pruebas de hardware	69
3.2.3	Pruebas de software	72
3.2.4	Pruebas de GLP	72
3.3	ANÁLISIS DE RESULTADOS	73
	CONCLUSIONES	74
	RECOMENDACIONES	76

REFERENCIAS BIBLIOGRÁFICAS.....	77
ANEXOS	80
ANEXO 1 – CRONOGRAMA	80
ANEXO 2 – DIAGRAMA ELECTRÓNICO COMPLETO	81
ANEXO 3 – MANUAL DE USUARIO	82
ANEXO 4 – MANUAL TÉCNICO.....	89

LISTA DE FIGURAS

Figura 1.1 Ventajas del GLP	9
Figura 1.2 Desventajas del GLP	7
Figura 1.3 Manipulación de cilindros de GLP	8
Figura 1.4 Tipos de comercialización del GLP.	9
Figura 1.5 Arduino Nano ATmega.....	13
Figura 1.6 Características de la placa Arduino Nano.	13
Figura 1.7 Wi-Fi shield 101.....	16
Figura 1.8 Características del Wi-Fi Shield 101	17
Figura 1.9 Arduino MKR1000	17
Figura 1.10 Características Arduino MKR100.....	18
Figura 1.11 Módulo Wi-Fi ESP8266.....	19
Figura 1.12 Características del módulo Wi-Fi ESP8266.....	17
Figura 1.13 Electroválvulas.....	20
Figura 1.14 Detalle de las partes de una electroválvula	19
Figura 1.15 Electroválvulas de acción directa.....	20
Figura 1.16 Electroválvulas de acción indirecta.....	23
Figura 1.17 Electroválvulas de acción mixta.....	21
Figura 1.18 Válvula de tres vías	24
Figura 1.19 Sensor de gas.....	22
Figura 1.20 Clasificación de los gases combustibles.	27
Figura 1.21 Ventajas y desventajas del Android Studio.....	27
Figura 1.22 Android vs iOS.....	28
Figura 2.1 Regulador para cilindros de uso doméstico	30
Figura 2.2 Esquema del proyecto	30
Figura 2.3 Diagrama de flujo del funcionamiento del sistema.....	31
Figura 2.4 Diagrama de bloques del <i>hardware</i> del prototipo.....	32
Figura 2.5 Diagrama de bloques del <i>software</i> del prototipo	33
Figura 2.6 Circuito interno del transistor Darlington TIP122	35
Figura 2.7 Hoja de especificaciones del TIP122	38
Figura 2.8 Electroválvula	38
Figura 3.1 Diagrama esquemático	38

Sistema automático para prevenir la explosión por fuga de gas licuado de petróleo en un domicilio

Figura 3.2 Conexión entre el módulo Wi-Fi y el módulo Arduino Nano.	39
Figura 3.3 Conexión entre el sensor MQ-2 y módulo Arduino Nano	40
Figura 3.4 Fotografía de la tarjeta electrónica elaborada	41
Figura 3.5 Componentes usados.....	42
Figura 3.6 PCB del prototipo completo.....	42
Figura 3.7 Tarjeta del prototipo Completa	43
Figura 3.8 Códigos de programación en Arduino	44
Figura 3.9 Sección de programación del Arduino Nano	45
Figura 3.10 Inicialización del LCD	45
Figura 3.11 Árbol de jerarquías de Android Studio	46
Figura 3.12 Archivos por defecto del área de trabajo de Android Studio	47
Figura 3.13 Componentes del archivo content_main.xml.....	48
Figura 3.14 Menú design y opciones de configuración de la aplicación.....	49
Figura 3.15 Aspecto de la aplicación.....	49
Figura 3.16 Icono de la aplicación en Android Studio	50
Figura 3.17 Circuito total del prototipo	62
Figura 3.18 Visualización de información	63
Figura 3.19 Instalación de sensor en maqueta diseñada.....	64
Figura 3.20 Instalación de sensor MQ-2	65
Figura 3.21 Esquema del prototipo	65
Figura 3.22 Esquema de la alarma.....	66
Figura 3.23 Vista externa del prototipo	66
Figura 3.24 App sistema detector	67

LISTA DE TABLAS

Tabla 1.1 Comparación de gas natural vs GLP.....	6
Tabla 1.2 Estándares IEEE 802.11 (velocidades y bandas de frecuencia).....	13
Tabla 2.1 Presupuesto del proyecto.....	34
Tabla 3.1 Pruebas chequeo inicial.....	68
Tabla 3.2 Pruebas medición de gases.....	68
Tabla 3.3 Pruebas activar/desactivar.....	69
Tabla 3.4 Pruebas diario/registro.....	70
Tabla 3.5 Pruebas android.....	70
Tabla 3.6 Pruebas electroválvula.....	71
Tabla 3.7 Pruebas GLP.....	72

RESUMEN

En la actualidad, en el mercado local del Ecuador no se cuenta con un dispositivo capaz de medir fugas de gas licuado de petróleo a nivel residencial, únicamente existen dispositivos que se enfocan en el GLP centralizado, a diferencia de la gran mayoría de hogares que utilizan el cilindro de GLP y que únicamente poseen válvulas de apertura y cierre para el uso en electrodomésticos. La finalidad de este proyecto tiene por objeto lograr que un dispositivo de control se acople fácilmente a los cilindros de gas, un desafío bastante importante, y que se logró mediante la utilización de circuitos ligeros con un alto grado de resistividad, para que de esta manera se puedan prevenir una gran cantidad de accidentes. Con la creciente utilización de teléfonos móviles, se enfocó en este detalle, puesto que el dispositivo se enlaza mediante Wi-Fi directamente a un celular inteligente, el cual contiene una aplicación propia del sistema y que facilita al usuario poder trabajar con este dispositivo de manera directa y amigable, proyectando así su rápida aceptación en el hogar.

Palabras Clave: Android, Arduino, Detector de gas, Wi-Fi, Tableta

ABSTRACT

At present, in the local market of Ecuador there is no device capable of measuring liquefied petroleum gas emissions at the residential level, the existing devices focus on centralized LPG, a difference of the vast majority of households that use the LPG cylinder and that may have opening and closing valves for use in household appliances. The purpose of this project is that the control devices can be easily used, and that it can be used with a high degree of resistance, so that in this way a large number of accidents can be prevented. With the growing use of mobile phones, focused on this detail, the device was connected via Wi-Fi directly to an intelligent cell phone, which contains an application of the system itself and which facilitates the user to work with this device directly and Friendly, which allows a quick acceptance at home.

Keywords: Android, Arduino, Gas detector, Wi-Fi, Tablet

INTRODUCCIÓN

En muchas ocasiones se han escuchado y visto noticias sobre pérdidas humanas y materiales que una explosión por fuga de gas ha causado, no solo en nuestra ciudad, sino en todo el Ecuador. Por esta razón, diversas entidades relacionadas con el tema han propuesto ciertas pautas para evitar que este tipo de desastres sucedan en los hogares, específicamente el Cuerpo de Bomberos de Quito da las siguientes recomendaciones: verificación del cilindro, manguera y válvula, instalación adecuada con anillos de seguridad, entre otros (Castellanos, 2015), donde mayormente se usa este combustible para realizar diversas actividades, que por lo general se centran en la cocina y/o lavandería.

Actualmente, se estima que alrededor del 90% de los hogares ecuatorianos utilizan gas licuado de petróleo (GLP) para la cocción de alimentos; pese a ello no existe una adecuada normativa que regule el manejo de estos cilindros dentro del hogar o existan sistemas adecuados de seguridad.

En tal sentido, el presente proyecto, cuyo fin es construir un sistema automático para prevenir explosiones de gas licuado de petróleo en un domicilio, aprovechando el avance tecnológico que se da en estos días y la gigantesca aceptación que tiene en la sociedad, en parte gracias al apoyo estatal a proyectos investigativos a nivel universitario y/o científico, propone una opción preventiva compuesta principalmente por componentes electrónicos, como es el caso del presente proyecto que contiene un sensor de gas en el cilindro de gas, un microcontrolador Arduino para control, y por último un teléfono celular con una aplicación elaborada en Android Studio, lo que permitirá monitorear de manera permanente y en línea los eventos que pueden suceder con respecto al comportamiento de dicho dispositivo.

La propuesta ayudará a controlar cualquier fuga de gas en el lugar donde el dispositivo sea implementado de manera inalámbrica, evitando de esta manera un evento mayor que puede causar daños y pérdidas irreparables, aportando así, directamente al beneficio de la comunidad, quienes en un futuro pueden verse favorecidos al obtener dicho dispositivo, si este llegase a verse comercializado.

Asimismo, el proyecto permitirá no solo desarrollar los conocimientos del autor como tal, sino también servirá de base para futuros proyectos investigativos en esta rama, tanto

para estudiante como para profesionales que deseen aportar sustancialmente con proyectos que contribuyan a la comunidad.

Antecedentes

En el país, en el año 2015 se registró un volumen de 1.476 millones de kilogramos despachados de GLP, mientras que en el 2016, fueron despachados 1.503 millones de kilogramos, debido a una menor demanda de los sectores agroindustrial e industrial; sin embargo, se observó una mayor demanda de GLP para uso doméstico (Agencia de Regulación y Control Hidrocarburífero, 2016, pág. 50).

De tal forma que, para abastecer la demanda interna de GLP en el año 2016 “se vendió un volumen total de 1.097,01 millones de kilogramos con una variación negativa de 1% en comparación al año 201” (Agencia de Regulación y Control Hidrocarburífero, 2016, pág. 54). Concretamente, sobre los despachos de GLP destinados al uso doméstico mantienen su tendencia creciente registrada en los últimos años, esto a pesar de la puesta en marcha del “Programa de Eficiencia Energética para cocción por inducción y calentamiento de agua con electricidad (Agencia de Regulación y Control Hidrocarburífero, 2016, pág. 54).

Es decir, la mayor parte de viviendas usan gas doméstico sin tomar importancia sobre el riesgo que esto trae a su familia y vivienda, por incendios que se puedan dar, o debido a su inhalación en tiempos prolongados. Así, considerando que la Facultad de Ingeniería en Electrónica Digital y Telecomunicaciones de la Universidad Tecnológica Israel, dota a los estudiantes de conocimientos teórico-prácticos relacionados con la carrera, gracias a lo cual, y al continuo aprendizaje y avance tecnológico, se da cabida a la necesidad de investigar nuevos procesos o herramientas, como es caso de un sistema automático para prevenir la explosión y/o intoxicación por fuga de gas licuado de petróleo en residencias.

Gracias a las nuevas tecnologías con elementos electrónicos cada vez más pequeños y potentes, permitirá el diseño un dispositivo compacto con las exigencias de seguridad y confiabilidad que se necesitan para la detección de este tipo de gases. Paralelamente se realizará una comunicación vía Wi-Fi entre el dispositivo detector y un teléfono inteligente el cual tendrá instalado una aplicación que facilitará el manejo del sistema completo al usuario de una manera amigable y simple, y que permitirá controlar los niveles mínimos de gas en el ambiente y a su vez en el caso de una fuga tener una notificación de que el sistema

realizó el cierre automático del gas en el ambiente y ventiló el área dejándola libre de GLP, y en el caso de ser necesario, realizar la apertura para el paso de gas nuevamente en el electrodoméstico que use este dispositivo.

En el ámbito académico, cabe destacar que a nivel nacional no existen proyectos similares al planteado; a nivel regional sobresale el trabajo de titulación “Diseño e Implementación de un prototipo para detectar fugas de gas para uso residencial” de la Universidad Buenaventura, el objetivo de este trabajo es implementar un detector de fuga de gas fusionando conceptos básicos en programación del chip Motorola y electrónica digital, que conjuntamente con el elemento electrónico adecuado permitió un producto de calidad y eficiencia y de fácil acceso a los integrantes del sector residencial (Colmenares, Ordoñez, Vertel, & Walteros, 2007, pág. X). Adicionalmente, existen otros trabajos pero con un enfoque industrial, que se limitan a las necesidades para las cuales fueron desarrollados.

Planteamiento y Justificación del Problema

El gas licuado de petróleo (GLP) se ha convertido en la fuente de energía de mayor consumo en el Ecuador, básicamente debido al reemplazo de artefactos eléctricos por su equivalente a GLP en el sector doméstico; el incremento del uso de GLP para restaurantes, hoteles y comercio en general; y en el sector industrial, el uso más frecuente se da en avícolas y piladoras de arroz (Briones, 2011, pág. 14).

Lamentablemente, de acuerdo a estadísticas presentadas por el Cuerpo de Bomberos, en el año 2015 se registró un promedio de 1 una explosión de gas cada 82 horas en la ciudad de Quito (Jácome, 2015). Tan solo en el mes de Enero de dicho año, fueron registradas alrededor 10 explosiones de GLP en el país, dejando cuatro fallecidos (Andes, 2015).

Por lo antes expuesto, es evidente que al incrementar el consumo de GLP, también aumentan los incidentes o eventos infortunados relacionados con el manejo de este tipo de combustible, principalmente a causa de la mala instalación de las mangueras, problemas en las válvulas utilizadas a presión, y el uso de cilindros deteriorados.

Por lo tanto, y aprovechando el acelerado avance tecnológico en lo relacionado al desarrollo de aplicaciones móviles, con diferentes finalidades en teléfonos celulares, hace

que su uso sea práctico y oportuno, y para este caso, permite desarrollar prototipos que colaboren con la prevención y seguridad de quienes usan este gas en su vida diaria.

En tal sentido, con el sistema a proponer se pretende prevenir los inconvenientes ocasionados por la fuga del gas licuado de petróleo (GLP), ya que dicho sistema estará permanentemente monitoreando, y le informará al usuario vía Internet de cualquier novedad al respecto, pero sobre todo actuará oportunamente, evitando así pérdidas de materiales y/o humanas.

Actualmente, existen diferentes clases de detectores, los cuales están enfocados para el sector comercial e industrial, pero en lo que respecta al sector domiciliario, no hay opciones parecidas al aquí planteado; por tal razón, en primera instancia se plantea el diseño y creación de un dispositivo de gran calidad, el cual posteriormente podría ser comercializado.

Por consiguiente, y tomando en cuenta que no existe un dispositivo que sea capaz de frenar de alguna manera la fuga de este gas, sumado a que la población lo toma de manera superficial, asumiendo que la bombona no necesita mayores cuidados y por ende los accesorios son siempre seguros, además de la poca capacidad que tienen las al reaccionar ante una fuga, siendo el episodio más grave cuando se duerme o no hay nadie en el inmueble, es vital recalcar que se necesita un dispositivo que pueda antelar todas estas situaciones sin generar accidentes, y así colaborar con un ambiente seguro para las familias ecuatorianas que utilicen GLP.

Para lograr lo propuesto, se debe formular la siguiente interrogante:

¿Cómo desarrollar un dispositivo eficiente para el control anti fugas de GLP a nivel residencial?

Finalmente, y a razón de que el dispositivo de detección de fugas de GLP propuesto, nace de la necesidad de prevenir explosiones por GLP en los hogares, es importante indicar que la implementación de dicho dispositivo beneficiará de forma directa no solo al Autor de la presente Tesis, quien cumplirá con los requisitos necesarios para optar por el título de Ingeniero en Electrónica Digital y Telecomunicaciones, sino que además servirá para la

comunidad ecuatoriana, y particularmente a las personas que lo empleen, ya que serán ellas quien residan en un hogar seguro, libre de fugas de GLP.

Objetivo General

Desarrollar un sistema automático para prevenir explosiones de gas licuado de petróleo en un domicilio.

Objetivos Específicos

- Definir los factores del funcionamiento del sistema automático.
- Diseñar un sistema prototipo de prevención de explosiones de gas licuado de petróleo mediante la adaptación de una válvula que detenga de forma automática una fuga cuando reciba la señal de un sensor que medirá la concentración de gas.
- Implementar el sistema de prevención en una maqueta cerrada.
- Incorporar un sistema de ventilación una vez detectada una fuga de gas.
- Desarrollar una aplicación Android que reciba alertas en caso de detectarse una fuga de gas licuado de petróleo.
- Realizar las pruebas de funcionamiento del sistema propuesto.

Descripción de los capítulos

En el primer capítulo, se presenta el marco teórico y conceptual sobre los elementos principales a considerar en el proyecto, como el microcontrolador Arduino Nano, el sensor de gas, la conexión y la configuración de módulo Wi-Fi, entre otros.

En el segundo capítulo, se detalla el diseño realizado, *hardware* y *software* empleado, los diagramas de bloques y diagramas de flujo del funcionamiento del sistema.

En el tercer capítulo se explica el desarrollo del proyecto, la conexión del sensor de gas, las pruebas de funcionamiento del sistema completo. Finalmente, se listan las conclusiones y recomendaciones de la implementación del prototipo, así como también las fuentes bibliográficas y anexos.

Alcance

Con este sistema de control anti fugas de GLP se usará una electroválvula que va estar conectada a un cilindro de gas GLP; se realizará el diseño del circuito que va a estar conectado a un Arduino, que va controlar a la electroválvula para que se active y se desactive.

Se desarrollará la simulación y la programación para la aplicación del sistema Android de este aplicativo, el cual el usuario lo tendrá en su teléfono celular que mediante datos móviles le lleguen las notificaciones cuando haya la fuga y el sistema se active. La comunicación como tal va a ser vía Internet o a su vez vía Wifi.

También existe la opción desconectar un extractor de olores adicional que ayudará a eliminar el gas contaminado que haya en la habitación y lo enviará al exterior del domicilio y liberando el gas contaminado de esa habitación.

CAPÍTULO1: FUNDAMENTACIÓN TEÓRICA

1.1 Antecedentes

1.1.1 Gas licuado de petróleo

El gas licuado de petróleo (GLP), o en inglés *petroleum liquid gas* (LPG), es la combinación de dos hidrocarburos butano y propano. “La palabra licuado viene por el hecho de que a pesar de que los componentes del GLP son gases, sin embargo son fáciles de licuar” (Propanogas, s.f.).

El GLP se encuentra a presión atmosférica normal, en estado gaseoso; en cambio a temperaturas muy bajas o tras un proceso de compresión, en estado líquido. Al liberar la presión, el líquido hierve y forma un vapor; y es este vapor el que se utiliza para suministrar energía a equipos y aparatos (Shell, s.f.). Por lo tanto, éste es el gas licuado que se utiliza como combustible, frecuentemente comercializado al granel o en bombonas.

Los tipos de GLP obedecen a sus componentes, que puede ser solo propano, solo butano o una mezcla de ambos, y en menor proporción también están presentes otros gases (Recope, 2017).

De acuerdo con (Dayre, 2013), el GLP es una fuente de energía respetuosa del medio ambiente, ya que cuando se quema se produce la más limpia de las emisiones.

Se debe resaltar que, el GLP se obtiene tanto del procesamiento del gas natural o mediante un proceso de refinación del petróleo. Así, el GLP de forma natural se obtiene el 60%, y producida el 40%.

Se debe destacar que, cuando se extrae de la tierra gas natural y crudo de petróleo, lo que se consigue es una mezcla compuesta por distintos gases y líquidos, de la que el GLP representa alrededor del 5%. Así, antes de transportar o utilizar el gas natural o el petróleo

como tal, es preciso separar los gases que forman el GLP, los cuales son levemente más pesados (LPG, 2015).

El GLP se utiliza de muchas maneras, algunas personas y/o industrias lo utilizan para aplicaciones muy variadas como generador de energía, para calefacción, para cocinar, en fin en diferentes áreas para infinidad de tareas (LPG, 2015).

No tiene color ni olor, pero para ser detectado se colocan pequeñas cantidades de azufre que le dan su olor característico y así poder detectar cualquier fuga, así sea mínima.

La bombona de gas, comúnmente llamada cilindro, es un magistral representante de las ventajas que el desarrollo del petróleo ha puesto en manifiesto: energía concentrada; fácilmente almacenable, transportable y que puede ser usada de inmediato, comparada con la electricidad, ésta posee una de las características más deseadas, en cuanto a energía se refiere: el fácil almacenamiento (Parra, 2003, pág. 226).

En la Tabla 1.1 se muestran las diferencias entre el gas natural y el GLP. Una de las principales diferencias es el índice de gravedad específica donde el GLP muestra un índice mayor al del gas natural, esto quiere decir que a pesar de las grandes ventajas que posee, si no se utiliza adecuadamente o si no se tienen los cuidados apropiados, el GLP puede ser una latente bomba de tiempo.

PROPIEDAD	GAS NATURAL	GLP
Composición	90% Metano	60% Propano 40% Butano
Fórmula química	CH ₄	C ₄ H ₁₀ C ₃ H ₈
Gravedad específica	0,60	2,05 1.56
Poder calórico	9200 kcal/m ³	22 244 kcal/ m ³ 6 595 kcal/lt 11 739 kcal/Kilo
Presión de Suministro	21 mbar	50 mbar
Estado Físico	Gaseoso sin límite de compresión líquido a -160°C a presión atmosférica	Líquido 20°C con la presión manométrica de 2,5 bar
Color/olor	Incoloro/inodoro	Incoloro/inodoro

Tabla 1.1 Comparación gas natural vs. GLP

Fuente: (Osinergmin, 2018)

Entre las múltiples ventajas que el GLP presenta, se pueden mencionar:

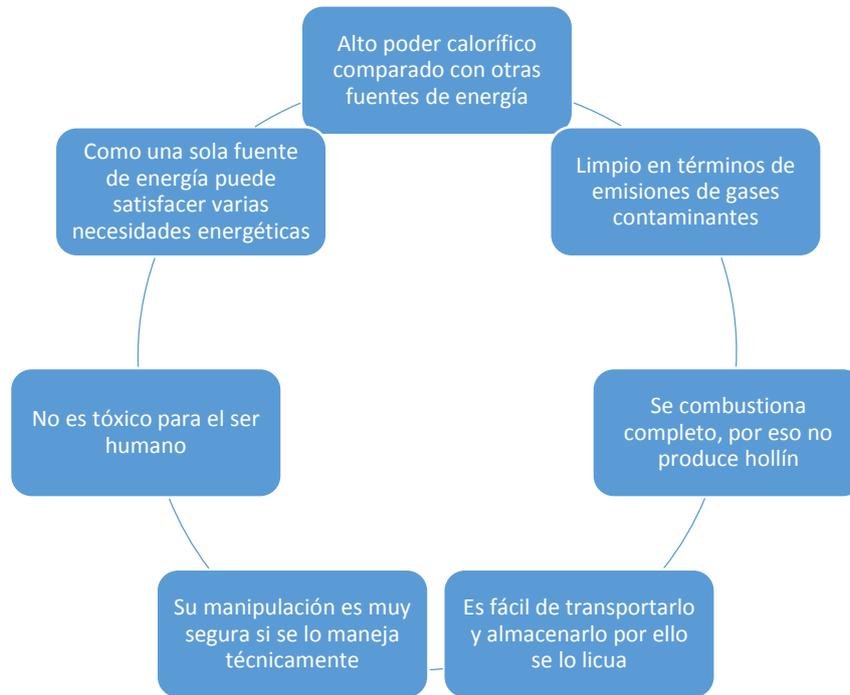
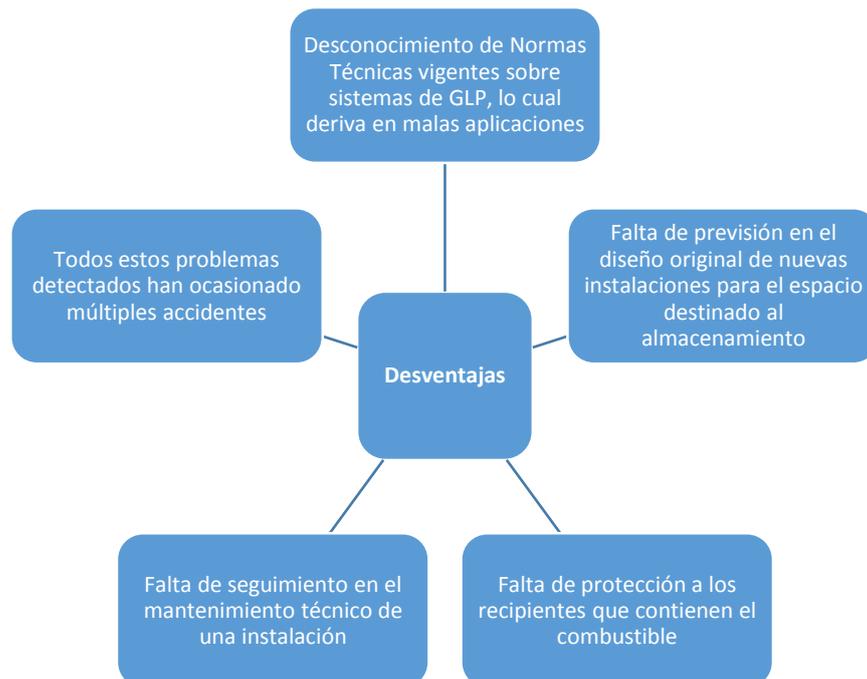


Figura 1.1 Ventajas del GLP

Fuente: (Venegas, Ayabaca, Celi, & Rocha, 2018, pág. 20)

En la Figura 1.1 se presentan las ventajas que presenta el GLP para el uso doméstico, pero así mismo se pueden deducir las consecuencias, si dicho gas no es manejado adecuadamente, las cuales se presentan a detalle en la Figura 1.2



Así, se señalan vario: **Figura 1.2 Desventajas del GLP** Fuente: (Venegas, Ayabaca, Celi, & Rocha, 2018, pág. 20) ejo en las instalaciones cuando no se toman las precauciones debidas, o cuando un usuario desconoce la manipulación y mantenimiento del mismo.

Al respecto, para el manejo e instalación adecuada de los tanques de gas hay una serie de recomendaciones básicas que es necesario aplicar, las cuales se exponen en la Figura 1.3



Figura 1.3 Manipulación de cilindros de GLP

Fuente: (El Nuevo Diario, 2015)

A continuación, se detallan algunos cuidados principales para el uso del cilindro de GLP de uso doméstico, recomendados por el Ministerio de Electricidad y Energía Renovable del Ecuador:

- No rodarlo.
- No golpearlo.
- No invertirlo de posición.

- No colocarlo en posición horizontal.
- No exponer el cilindro al calor directo.
- Verificar que las válvulas estén bien cerradas.
- Chequear periódicamente mangueras y regulador.
- Tener alejado el cilindro de sustancias inflamables.
- Ventilar el lugar donde está el cilindro.

1.1.2 Comercialización en el Ecuador

El GLP proporciona energía a artefactos, equipos y maquinarias para su funcionamiento, y en el Ecuador es usado en aplicaciones domiciliarias, así como comerciales e industriales (Venegas, Ayabaca, Celi, & Rocha, 2018, pág. 19). Además estos autores mencionan que su manejo y utilización, genera riesgos hacia las personas y edificaciones que se encuentran en sus cercanías, los cuales pueden ser disminuidos, si se lo manipula de forma adecuada.

En Ecuador el uso de GLP empezó en el año de 1955, y actualmente existen 16 envasadoras de este gas, ubicadas en Pichincha, Guayas, Esmeraldas, Azuay y Loja (Líderes, s.f.). La comercialización, almacenamiento, y distribución de GLP, se lo realiza en varias presentaciones:

- GLP envasado (se utiliza como combustible, para la cocina, calefacción, etc.).
- GLP almacenado y depósito fijo (se utiliza en el sector comercial e industrial).

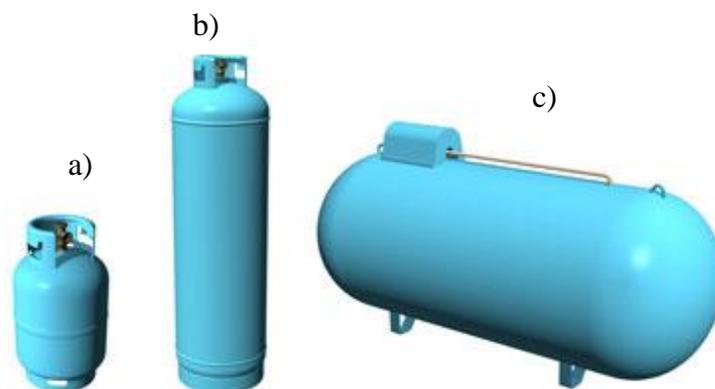


Figura 1.4 Tipos de comercialización del GLP.
Fuente: (Dubois, 2006, pág. 181)

Como se muestra en la figura 1.4, para la comercialización de GLP, se usa las bombonas pequeñas para el uso doméstico, las medianas para uso comercial y/o industrial, y finalmente se tienen el gas para uso centralizado.

En el Ecuador se comercializa el GLP figura a) doméstico en cilindros de 3 kg, 5 kg, 10 kg, 15 kg, el GLP figura b, c) industrial en cilindros de 15 kg y 45 kg, el doméstico e industrial también se venden al granel. El GLP agroindustrial únicamente al granel.

1.2 Marco teórico

En esta sección se detallan los componentes que se requieren para el desarrollo del proyecto, además de otras conceptualizaciones propias de dicho proyecto.

1.2.1 Arduino

Arduino, es una plataforma de código abierto a nivel de *hardware* como *software*, la cual toma la información del ambiente a través de sus pines de entrada mediante una gran gama de sensores que permiten interactuar con todo aquello en su entorno; para programar el microcontrolador que poseen estas placas integrales, se utiliza un lenguaje basado en la plataforma *Wiring* y el entorno de desarrollo basado en *Processing*, toda esta programación se carga en un procesador *AT mega*. (Realpe, 2017, pág. 4).

Existen diferentes tipos de placas de Arduino, pero de acuerdo con las necesidades del proyecto, se enfocará en el Arduino Nano. Esta placa, ilustrada en la Figura 1.6, se basa en el procesador *ATmega328*; carece de un adaptador de corriente, ya que se alimenta a través de uno de sus pines mediante una fuente con o sin regulación o también con un cable mini USB (Bus Universal en Serie), que a su vez sirve de comunicación y entrada para su programación; además para la comunicación con otro microcontrolador u ordenador posee un puerto UART (Transmisor-Receptor Asíncrono Universal,) (Realpe, 2017, pág. 5).

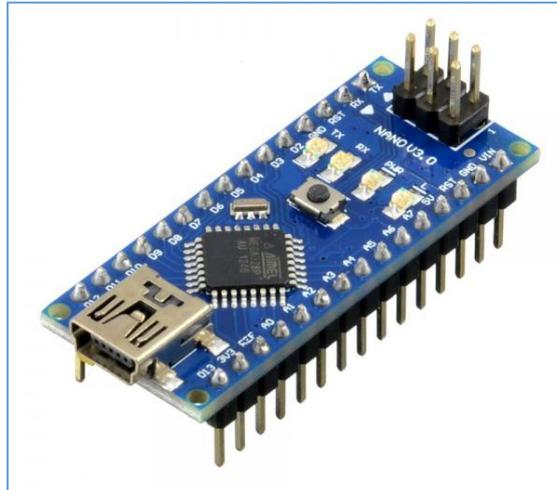


Figura 1.5 Arduino Nano ATmega
Fuente: (Ardudino, 2018)

Las características principales de esta placa son las siguientes:



Figura 1.6 Características de la placa Arduino Nano.
Fuente: (Ardudino, 2018)

En la Figura 1.6 se indica el voltaje de funcionamiento que es de 5v. El voltaje de entrada es entre 7v y 12v, el consumo de corriente es de 40mA, se tienen 14 pines donde están las entradas y salidas, de los cuales 8 pines son de entrada analógicas; adicional a esto, se tiene una memoria de 32 KB.

En los siguientes apartados, se presentará información detallada acerca de los protocolos de comunicación inalámbrica, utilizados en el dispositivo detector de fugas de GLP.

1.2.2 Protocolo Estándar IEEE 802.11 Wi-Fi

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), se dedica a la promoción de estándares internacionales para las áreas de las telecomunicaciones, la tecnología de información y la generación de energía; tiene alrededor de 900 estándares activos y 400 en desarrollo. Entre los productos más conocidos de este Instituto, sobresale el grupo de estándares IEEE 802 para redes, el cual incluye: Ethernet IEEE 802.3 y IEEE 802.11- redes inalámbricas (Pascual, 2007, pág. 5).

IEEE 802.11 (Redes de Área Local Inalámbricas)

El estándar IEEE 802.11, para redes LAN inalámbricas incluye una serie de enmiendas aplicadas al estándar original (2 velocidades de transmisión 1 y 2 Mbps transmitidas por infrarrojas en la banda ISM¹). Éstas abarcan esencialmente las técnicas de modulación, gama de frecuencia y la calidad del servicio. El IEEE 802.11 cubre las primeras dos capas del modelo de OSI (Interconexión de Sistemas Abiertos), es decir la capa física y la capa de enlace (Pascual, 2007, pág. 6).

Entre las enmiendas más reconocidas de esta familia, se tienen: b, a, y g, además de las [c-f], [h-j], n y s que son correcciones, actualizaciones o extensiones de las anteriores (Realpe, 2017, pág. 9).

IEEE 802.11b

El estándar IEEE 802.11b además de incluir perfeccionamientos al estándar original para el soporte de tasas de transmisión más elevadas (5,5 y 11 Mbps), usa el mismo método de acceso y la misma técnica DSSS² del estándar inicial. El IEEE 802.11b permite transmitir

¹ Industrial, Científico y Médico.

² El espectro ensanchado por secuencia directa (DSSS), también conocido en comunicaciones móviles como DS-CDMA (acceso múltiple por división de código en secuencia directa), es uno de los métodos de

hasta en 11 Mbps, reduciendo dicha tasa cuando el receptor comienza a detectar errores, bajando a 5,5 Mbps, luego a 2, y después a 1 Mbps (Vincze, 2014, pág. 34).

Se utilizó este estándar de comunicación en el proyecto es la que más se acerca a mi proyecto por su velocidad de transmisión de datos.

IEEE 802.11g

Es como una extensión de 802.11b. Aumenta la velocidad de transmisión de hasta 54 Mb/s usando el mismo ancho de banda de 2,4 GHz que se utiliza en el estándar 802.11b.

IEEE 802.11n

Es la generación de tecnología inalámbrica que entrega espectaculares mejoras en confiabilidad, velocidad. Cuenta con una velocidad de modulación cerca de seis veces más rápida y una tasa de transferencia de datos de 2 a 5 veces que una antena WiFi 802.11 a/g, mejoras sustanciales en cobertura y calidad de conexión. (Vincze, 2014, pág. 34).

La Tabla 1.2 expone los estándares IEEE 802.11 más comunes, con sus respectivas velocidades de transmisión y bandas de frecuencia en las que operan; así por ejemplo, el estándar 802.11a tiene un flujo de datos máximo, cinco veces mayor que el estándar 802.11b.

Estándar IEEE	Velocidad	Banda de frecuencia
802.11	1 Mbps – 2 Mbps	2.4 GHz
802.11a	Hasta 54 Mbps	5 GHz
802.11b	5.5 Mbps – 11 Mbps	2.4 GHz
802.11g	Hasta 54 Mbps	2.4 GHz
802.11n	100 Mbps	2.4 GHz – 5GHz

Tabla 1.2 Estándares IEEE 802.11 (velocidades y bandas de frecuencia)

Fuente: (Realpe, 2017, pág. 10)

1.2.3 Módulos Wi-Fi

codificación de canal (previa a la modulación) en espectro ensanchado para transmisión de señales digitales sobre ondas radiofónicas que más se utilizan (Villegas, 2012).

El WiFi es un mecanismo que permite, de forma inalámbrica, el acceso a Internet de distintos dispositivos al conectarse a una red determinada. Esta tecnología, al tiempo que ofrece la entrada a la gran red de redes, vincula diferentes equipos entre sí sin la necesidad de cables. Dicha conexión inalámbrica es posible gracias al uso de radiofrecuencias e infrarrojos, empleados para la transmisión de información (ValorTop, 2017).

Wi-Fi Shield 101

Arduino Wifi Shield 101 es un *shield*³ potente para aplicaciones IoT con autenticación criptográfica, desarrollado con ATMEL, éste conecta la placa Arduino a Internet inalámbricamente, mediante una conexión simple, sin configuración adicional, claro está a más del SSID⁴ y la contraseña. Trae una biblioteca de fácil uso que permite conectar la placa a Internet con pocos procedimientos. Se basa en el módulo Atmel SmartConnect-WINC1500, compatible con la norma IEEE 802.11 b/g/n (Aprendiendo Arduino, 2018).



Figura 1.7 Wi-Fi shield 101
Fuente: (Aprendiendo Arduino, 2018)

El circuito de la figura 1.7 trabaja con voltajes de 3.3v y 5v además tiene conexión basada en la norma IEEE 802.11 b/g/n para la comunicación inalámbrica.

³*Shield*: Placa que se apila o conecta encima de la placa Arduino para extender sus capacidades.

⁴SSID significa “Identificador de conjunto de servicios”. Según el estándar de red inalámbrica IEEE 802.11, un “conjunto de servicios” se refiere a una colección de dispositivos de red inalámbricos con los mismos parámetros. Entonces, el SSID es el identificador (nombre) que le dice a qué conjunto de servicios (o red) se debe unir (Pastor, 2017).

Las características principales del WiFi shield 101 son las siguientes:

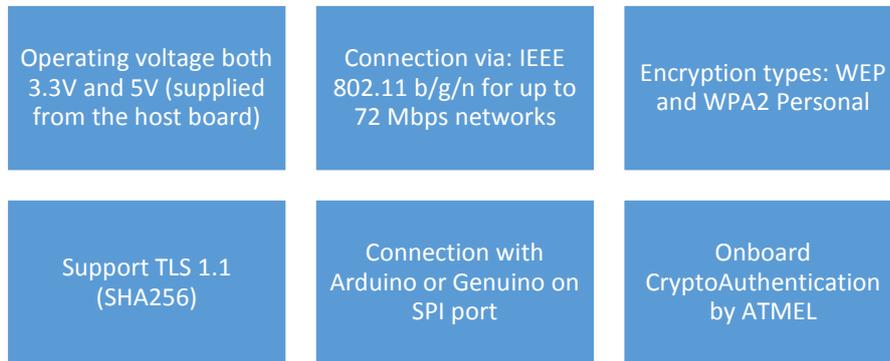


Figura 1.8 Características del Wi-Fi Shield 101

Fuente: (Aprendiendo Arduino, 2018)

En la Figura 1.8 se muestra que este circuito opera con voltajes de 3.3v y 5v, además tiene conexión basada en la norma IEEE 802.11 b/g/n para la conexión del Arduino; la encriptación es de tipo WEP y WPA2 personal que sirve para la contraseña para poder acceder a la conexión wifi.

Arduino MKR1000

Es un nuevo Arduino que ofrece una conectividad Wi-Fi (práctica y económica) para personas con mínima experiencia en redes, gracias a su microcontrolador de compacto tamaño, como se observa en la Figura 1.10 es muy versátil para implementarlo en proyectos que necesitan dimensiones reducidas.



Figura 1.9 Arduino MKR1000

Fuente: (Aprendiendo Arduino, 2018)

En el circuito de la figura 1.9 tiene un microcontrolador que es de bajo consumo de energía que utiliza un procesador ARM CORTEX de 32 bits que van de 32 a 64 pines con una memoria flash 256 KB y una memoria SRAM de 32 KB.

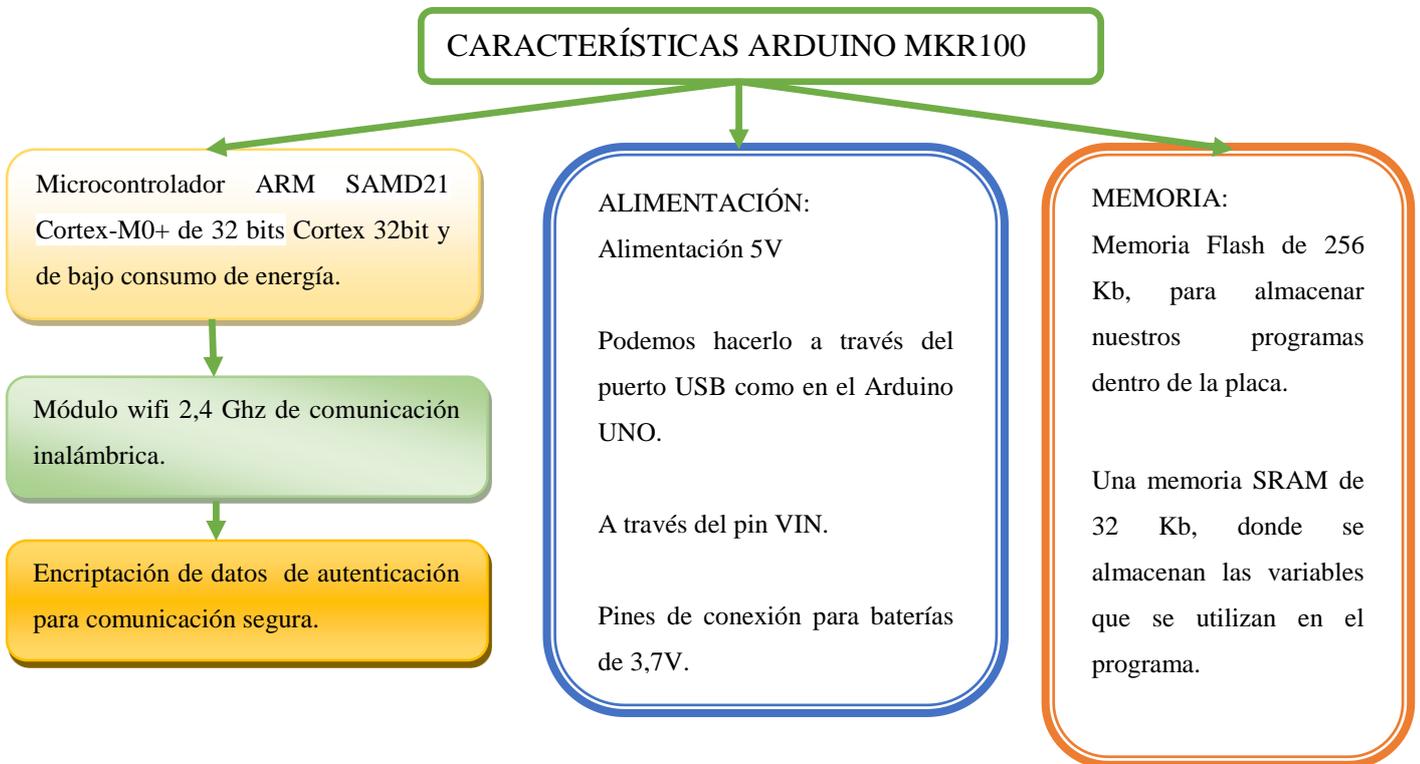


Figura 1.10 Características Arduino MKR100
Fuente: (Aprendiendo Arduino, 2018)

La Figura 1.10 explica las características del Arduino MKR100, donde se indica los voltajes de funcionamiento.

Módulo Wi-Fi ESP8266

Como se aprecia en la Figura 1.11, el módulo ESP 8266, es un chip altamente integrado y diseñado, siendo una solución completa y autónoma en las redes Wi-Fi⁵, lo que le permite ser parte inteligente de una aplicación, así como descargar las grandes posibilidades de una red inalámbrica para comunicarse desde otro procesador de aplicaciones (AG Electrónica, 2016).

⁵El WiFi es un mecanismo que permite, de forma inalámbrica, el acceso a Internet de distintos dispositivos al conectarse a una red determinada. Esta tecnología, al tiempo que ofrece la entrada a la gran red de redes, vincula diferentes equipos entre sí sin la necesidad de cables. Dicha conexión inalámbrica es posible gracias al uso de radiofrecuencias e infrarrojos, empleados para la transmisión de información (ValorTop, 2017).

Para (Electronilab, 2018), este módulo tiene capacidades tanto de procesamiento como de almacenamiento que permiten la integración con sensores y dispositivos específicos, además posee un diseño de antena, lo que le permite tener mayor alcance.

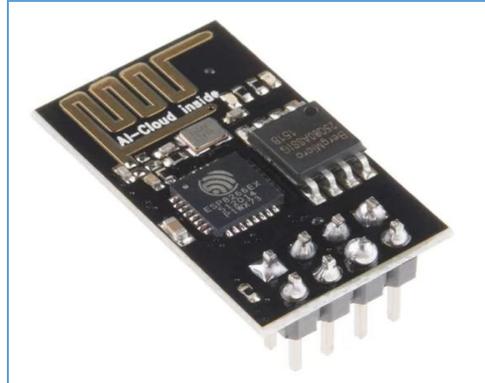


Figura 1.11 Módulo Wi-Fi ESP8266

Fuente: (Electronilab, 2018)

El módulo soporta el protocolo estándar IEEE 802.11 b/g/n, permitiendo programar desde simples sistemas de monitorización hasta completos servidores con soporte web. Su ventaja con otros dispositivos similares es que se puede usar de manera independiente o bien con Arduino para hacer un complemento y aumentar las capacidades que el diseño lo requiera (Realpe, 2017, pág. 11).

Entre sus principales características se tienen:

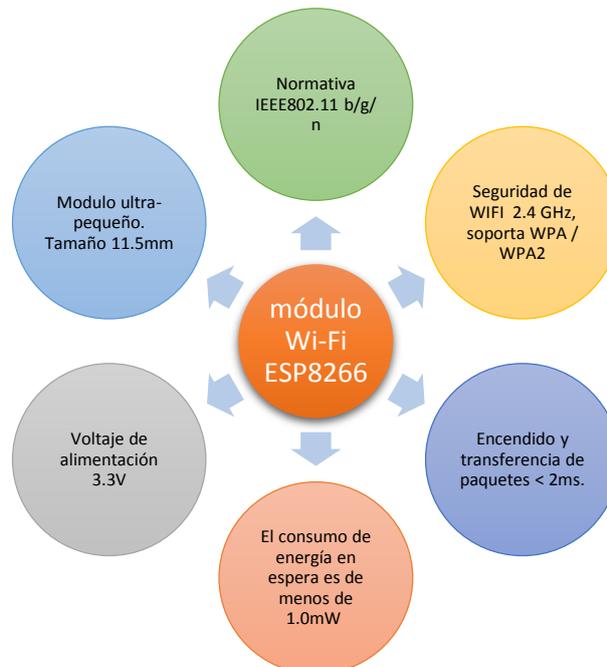


Figura 1.12 Características del módulo Wi-Fi ESP8266

Fuente: (AG Electrónica, 2018)

En la Figura 1.12 se muestran las características del módulo inalámbrico que se utiliza en el proyecto, el protocolo de comunicación inalámbrica, los voltajes de operación, la frecuencia de comunicación, la transferencia de paquetes es a menos de 2ms, consumo de energía menos de 1.0 mW, y sus dimensiones.

En los siguientes apartados, se presenta el detalle de la electroválvula, un componente esencial en el manejo del flujo del gas.

1.2.4 Electroválvulas

La válvula solenoide de uso general o concretamente electroválvula, es una obturadora que abre o cierra el paso de un líquido en un circuito, esto se efectúa mediante un campo magnético generado por una bobina en una base asegurada que atrae el émbolo (Altec, 2018).

Una electroválvula es una válvula controlada eléctricamente, la cual sirve para manejar un fluido, tanto para cerrarlo como para direccionarlo. No es lo mismo electroválvula que válvulas motorizadas, aquellas en las que un motor opera el cuerpo de la válvula. Debido a su reconocida solidez y fiabilidad se pueden adaptar a diferentes aplicaciones industriales como el control de agua, aire, gas, vapor, químicos, gas combustible, entre otros (Distritec, 2013).



Figura 1.13 Electroválvulas
Fuente: (Altec, 2018)

En la Figura 1.13 se muestran las diferentes electroválvulas que hay en el mercado, las cuales se pueden utilizar en diversas aplicaciones industriales que se necesite.

Una electroválvula está compuesta por dos partes elementales una cabeza electromagnética (electroimán) y su núcleo móvil (émbolo) y una válvula en la que el orificio está cubierto por una aguja. (Asconumatics, s.f.).

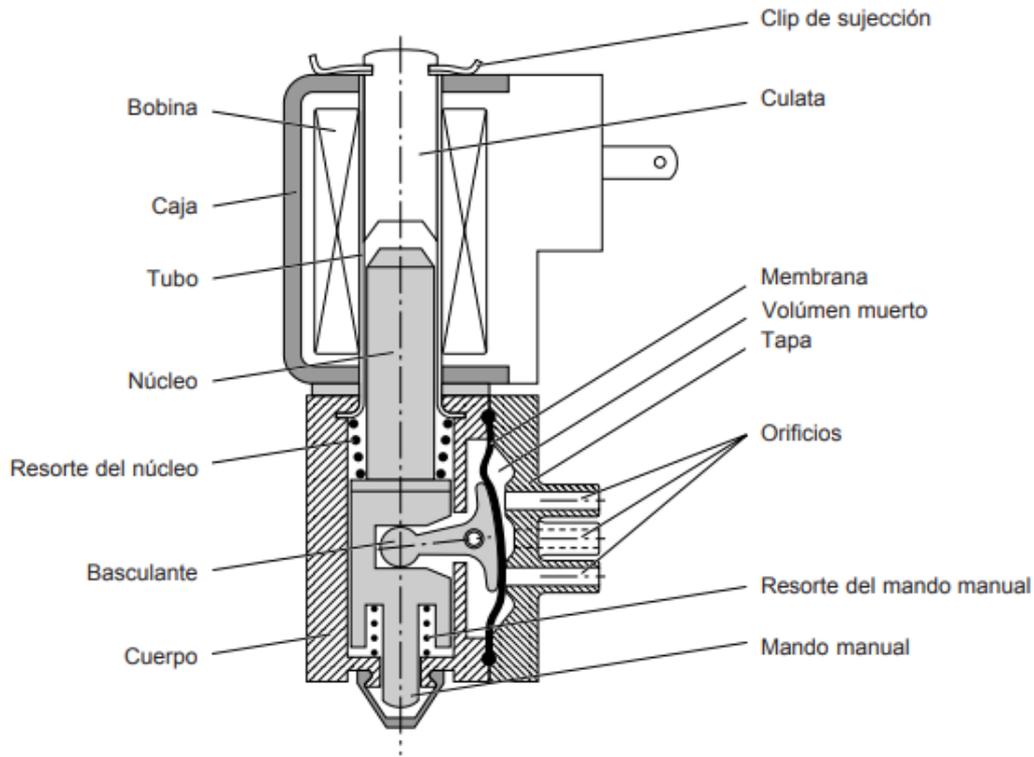


Figura 1.14 Detalle de las partes de una electroválvula

Fuente: (Asconumatics, s.f.)

En la figura 1.14 se exponen las partes de una electroválvula.

Bobina: Parte eléctrica de la electroválvula que crea un campo magnético cuando es alimentada y constituida por un cilindro formado por un hilo de cobre enrollado y aislado.

Caja: Cubierta metálica que asegura la protección eléctrica y mecánica de la bobina, y una protección al mismo tiempo contra el agua y el polvo.

Tubo: Tubo de latón o acero inoxidable. Sirve de guía al núcleo que se acciona por la fuerza magnética generada por la bobina que rodea.

Núcleo: Cilindro de acero inoxidable, de bajo magnetismo residual, que se desplaza por la fuerza electromagnética creada por la bobina.

Resorte del núcleo: Resorte que mantiene el núcleo en una posición preestablecida cuando la bobina está sin tensión.

Basculante: Pieza móvil que permite la apertura o el cierre de los orificios para el paso del fluido.

Cuerpo: Parte principal que contiene los orificios y asientos principales.

Clip de sujeción: Clip que asegura la sujeción de la bobina en la culata.

Culata: Masa situada en el fondo del tubo para mejorar el campo magnético cuando la bobina está con tensión.

Membrana: Pieza estanca de aislamiento del fluido con respecto a la parte de mando.

Volumen muerto: Volumen de fluido en las zonas no funcionales de la electroválvula.

Tapa: Fija en el cuerpo e incluye los orificios.

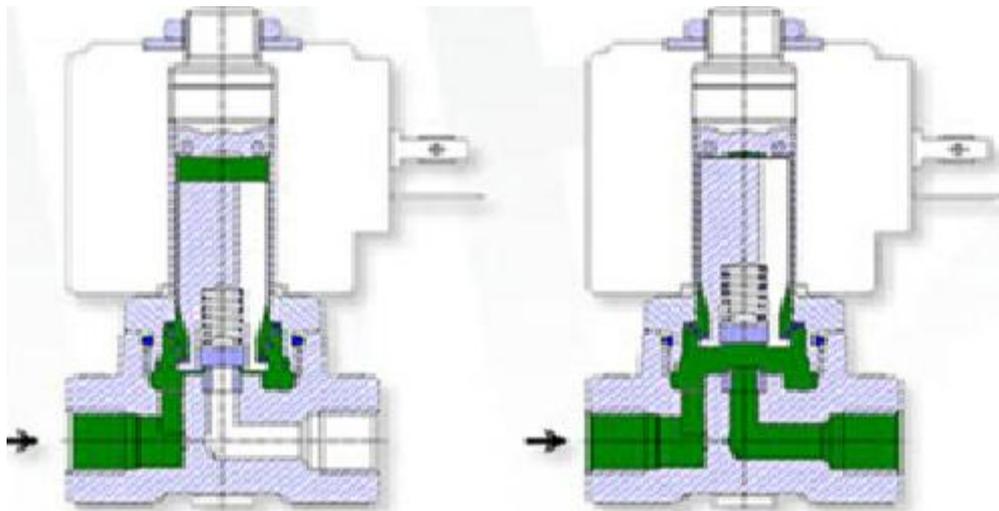
Orificios: Permiten el paso del gas.

Mando manual: Permite accionar manualmente la palanca para abrir o cerrar los orificios.

Resorte del mando manual: Resorte que asegura el retorno a la posición inicial del mando por impulsión.

Entre los tipos de electroválvulas, se distinguen:

- **Acción directa**, donde el flujo electromagnético actúa directamente en el émbolo que cierra o abre el orificio permitiendo que el líquido pase o pare (Altec, 2018).



a) Válvula de solenoide sin energía de la bobina cerrada

b) válvula de solenoide activada por bobina abierta

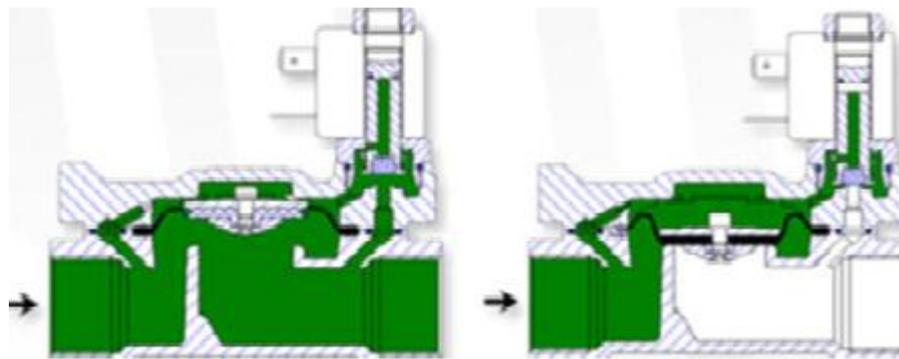
Figura 1.15 Electroválvulas de acción directa

Fuente: (Altec, 2018)

En la Figura 1.15 se ilustran las válvulas de acción directa, lo que permite detectar la presencia del gas GLP, donde la figura a) donde la bobina esta energizada se cierra y no hay paso del GLP.

En la figura b) donde la bobina no está energizada se abre y hay paso del GLP.

Acción indirecta, el orificio principal se abre debido al desequilibrio entre las presiones de las superficies del diafragma superior e inferior; cuando se energiza la bobina, el émbolo abre el orificio de piloto y descarga el compartimiento superior del diafragma (el desequilibrio de la presión mueve el diafragma que abre el orificio principal), como se muestra en la Figura 1.16 (Altec, 2018).



a) Válvula de solenoide activada por bobina abierta

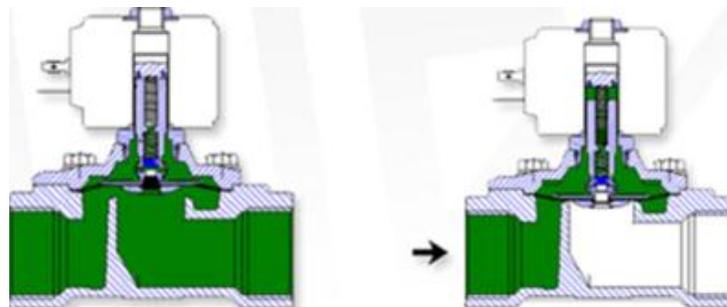
b) Válvula de solenoide sin energía de la bobina cerrada

Figura 1.16 Electroválvulas de acción indirecta

Fuente: (Altec, 2018)

En la Figura 1.16 se indica en la figura a) cuando no recibe un pulso eléctrico se abre la electroválvula. En la figura b) cuando recibe un pulso eléctrico se cierra la electroválvula.

Acción mixta, la abertura del orificio principal se realiza debido al desequilibrio de presiones del cuerpo superior e inferior, combinando con la acción directa del émbolo que está fijo al diafragma mediante un resorte, como se observa en la Figura 1.17 (Altec, 2018).



a) Válvula de solenoide activada por bobina abierta

b) Válvula de solenoide sin energía de la bobina cerrada

Figura 1.17 Electroválvulas de acción mixta

Fuente: (Altec, 2018)

Existen electroválvulas de 2 y 3 vías, las de 2 vías son las válvulas más conocidas ya que tienen una entrada y una salida; en cambio las de 3 vías tienen una entrada, una salida y un escape, tal como se muestra en la siguiente Figura 1.18 (Altec, 2018).

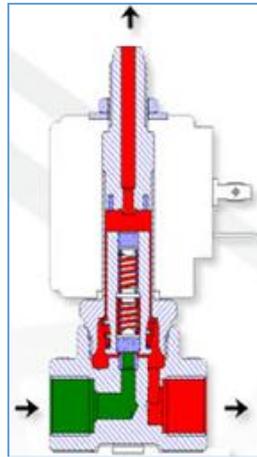


Figura 1.18 Válvula de tres vías
Fuente: (Altec, 2018)

Seguidamente, se muestra información acerca del sensor que se utilizó para medir los niveles de gas en el ambiente.

1.2.5 Sensores de Gas

Los sensores de gas detectan la presencia de algún gas específico, estos dispositivos pueden configurarse para brindar mayor precisión; miden la concentración de gas; son usados para prevenir la exposición a gases combustibles y gases tóxicos (330ohms, 2016).



Figura 1.19 Sensor de gas.
Fuente: (330ohms, 2016)

En la Figura 1.20 se ilustran sensores de gas, los cuales permiten detectar presencia de gas y son precisos al momento de medir la concentración de gas y son usados para diferentes ambientes que se los requieran.

Los sensores de gas posibilitan la detección de mezclas combustibles o tóxicas gas/aire, lo que permite prevenir accidentes que conlleven pérdidas humanas o materiales, tanto en entornos industriales como domésticos (Torres, 2008).

Pueden detectar una gran gama de gases, pudiendo conseguir diferentes características de respuesta variando los materiales, las técnicas del proceso y la temperatura de trabajo. Su simplicidad de construcción resulta en una vida de 10 a 25 años libre de problemas. Son robustos, aguantan choques y vibraciones siendo válido para servicio anti explosión (Torres, 2008).

Hay varios tipos de sensores de gas, y dependiendo de su operación, existen dos grupos de sensores de gas: el primero son los que funcionan por medio de absorción, reacciones químicas y de contacto con el gas; el segundo grupo en cambio, funcionan con emisiones infrarrojas o ultrasónicas. Por otro lado, los sensores se agrupan de acuerdo al tipo de gas que detectan: para gases combustibles generalmente son sensores catalíticos e infrarrojos, y para gases tóxicos se usan sensores electroquímicos y de semiconductores de óxido metal (330ohms, 2016).

A continuación, se mencionan algunos tipos de sensores según su operación:

- **Sensores semiconductores** (semiconductores de óxido metal)
- **Sensores infrarrojos** (emisores y receptores de luz infrarroja).
- **Sensores ultrasónicos** (emisiones ultrasónicas para detectar variación en el ruido de fondo)
- **Sensores electroquímicos** (dos electrodos divididos por una capa de electrolitos-líquida, sólida o gel).
- **Sensores catalíticos** (oxidación del gas vía catalítica) (330ohms, 2016).

Adicionalmente, es necesario destacar que uno de los aspectos esenciales para un funcionamiento adecuado del sensor es su calibración, y de esta manera asegurar que el

sensor detecte efectivamente los gases, ya que puede sufrir contaminación por su uso, desajustarse y no medir la cantidad de gas adecuadamente

En el proyecto se utilizó un sensor catalítico. Estos sensores catalíticos son sensibles y pueden funcionar indeseablemente en presencia de gases inhibidores tales como dióxido de azufre (SO₂), ácido sulfhídrico (H₂S), etc. Además estos sensores de gas son más asequibles para el público en general. (330ohms, 2016).

A continuación, se puntualizan las normas para las instalaciones de GLP.

1.2.6 Norma NTEINEN 2260 (Instalaciones de gases combustibles para uso residencial, comercial e industrial)

Según lo mencionado por (INEN, 2010, pág. 1), la NTEINEN 2260 implanta los requisitos técnicos y las medidas de seguridad mínimas que deben cumplirse al diseñar, construir, ampliar, reformar, revisar y operar las instalaciones receptoras de gases combustibles residencial, comercial e industrial; también expone las exigencias mínimas de los sitios donde se coloquen los artefactos que usan gases combustibles, así como las condiciones técnicas de su conexión, ensayos de comprobación y su puesta en funcionamiento.

Dicha norma, define a los detectores de fugas de gas, objeto de la presente propuesta, como aparato que detecta la presencia de gas en el aire y que a una determinada concentración emite una señal de aviso, el cual puede poner en funcionamiento un sistema automático de corte de gas (INEN, 2010, pág. 3). Asimismo, clasifica a los gases combustibles de la siguiente manera:

Familia	Nombre del Gas	Componente Principal	Observación	Densidad	PCS en volumen kWh/m ³	PCS en masa kWh/kg	Índice de Wobbe MJ/m ³ (n)	Límites de inflamabilidad	Odorizante
1ra. Familia	Gas manufacturado	Metano+ H ₂ +CO	Tóxico, en desuso	< 1	5, 23	-----	19, 13 a 27, 64	6 a 45	
2da. Familia	Gas Natural	Metano	No tóxico	< 1	12, 2	-----	39, 1 a 54, 7	5 a 15	Tetrahi-Drotiofeno
			Inoloro						
3ra. Familia	GLP	Propano	No tóxico	> 1	27, 29	14, 0	72, 9 a 87, 3	2, 4 a 9, 5	Mercaptano
			Inoloro						
		Butano	No tóxico	> 1	36, 0	13, 95		1,8 a 8,4	Mercaptano
			Inoloro						

Figura 1.20 Clasificación de los gases combustibles

Fuente: (INEN, 2010, pág. 10)

En la Figura 1.20 se presenta la clasificación de gases combustibles para el consumo de 3 diferentes familias, donde se tienen los componentes del GLP, también la toxicidad, características y la inflamabilidad de cada uno. Como se puede observar en la figura anterior, el GLP se ubica en la 3ra familia, la cual incluye los GLP: propano y butano, con poder calorífico entre 27,9 y 36 kWh/m³ (n) (INEN, 2010, pág. 10). También clasifica las instalaciones de gas, las cuales se encasillan de acuerdo a la forma de suministro y son de dos clases:

- Suministradas desde una red de distribución (canalizado).
- Suministradas desde tanques estacionarios o semi estacionarios (INEN, 2010, pág. 11).

Finalmente, es importante destacar que las instalaciones para suministro de gases combustibles deben contemplar como mínimo los siguientes aspectos básicos:

- Las instalaciones de gas deben ser construidas de tal forma que todas las partes sean accesibles.
- La máxima presión de operación en tuberías instaladas debe ser de máximo 35 kPa.
- Se prohíbe ubicar y acceder a válvulas, reguladores, tubería vista, etc. en área de escaleras de emergencia (INEN, 2010, pág. 12).

Existen dos tipos de poder calórico:

Poder calorífico superior: El agua producida por la combustión está supuestamente condensada, símbolo, PCS estos parámetros se expresan generalmente en megajulios (MJ/kg) o kilojulios por kg (kJ/kg).

Poder calorífico inferior: El agua producida por la combustión permanece supuestamente en estado de vapor, símbolo. PCI su unidad de medida es megajulios por kg MJ/m³ (n) de gas seco ó MJ/kg.

En función del valor de Ws (Wobbe) se clasifican los gases en tres familias.

La primera familia incluye los gases manufacturados, gas de coquería y mezclas hidrocarburos aire (aire propanado y aire metanado) de bajo poder calorífico entre 4,65 y 5,5 kWh/m³ (n).

La segunda familia incluye los gases naturales, gas natural sintético y las mezclas hidrocarburo aire (aire propanado) de poder calorífico entre 9,3 y 14 kWh/m³ (n).

La tercera familia incluye los gases licuados de petróleo (GLP): propano y butano, con poder calorífico entre 27,9 y 36 kWh/m³ (n).

El índice de Wobbe es una medida del grado de combustibles, el índice de Wobbe de un combustible se calcula dividiendo su índice de calentamiento superior, la cantidad de energía liberada en forma de calor cuando se quema por la raíz cuadrada de la densidad del gas, y megajulios por metro cúbico estándar (MJ / Sm³). (INEN, 2010, pág. 04).

A continuación se describirá la herramienta Android Studio.

1.2.7 Android Studio

Android Studio proporciona las herramientas más rápidas para crear apps en todas las clases de dispositivos Android; esto incluye: edición de códigos -primer nivel-, depuración, herramientas de rendimiento, sistemas de compilación e implementación (Android Studio, s.f.).

Entre las ventajas y desventajas del Android Studio sobresalen las siguientes:

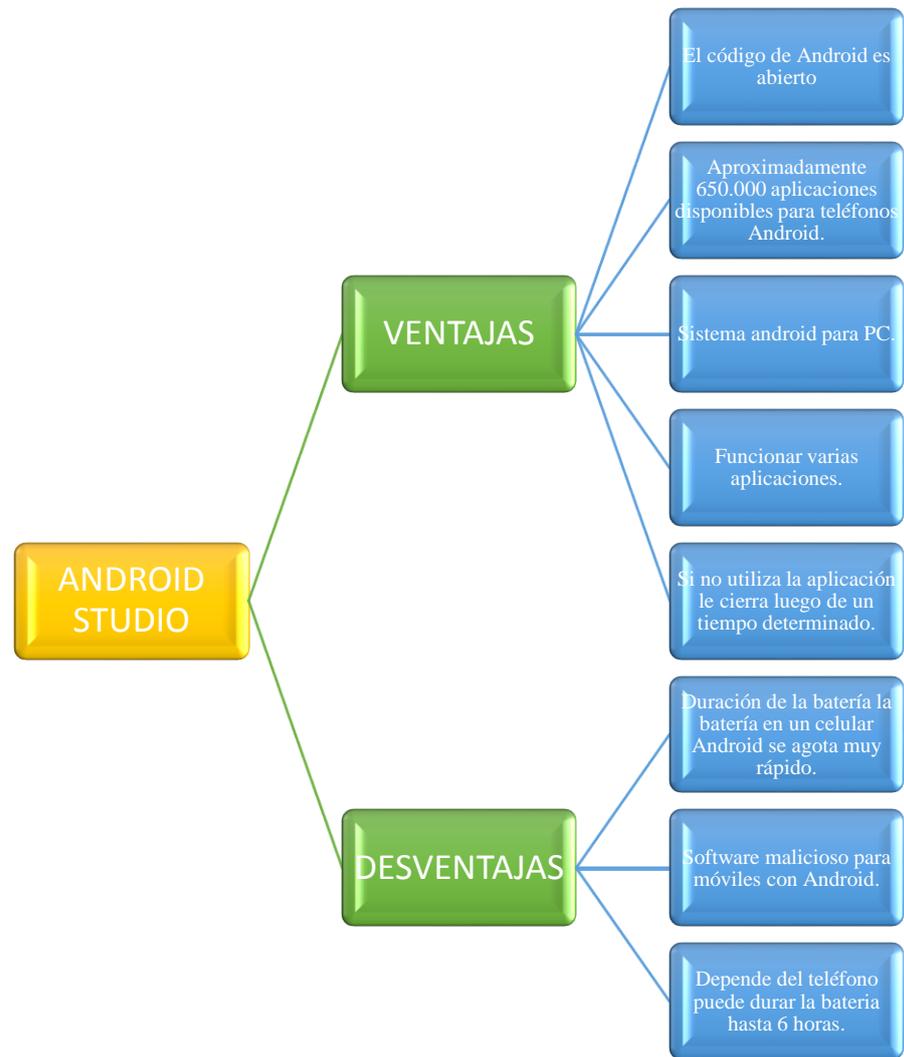


Figura 1.21 Ventajas y desventajas del Android Studio
Fuente: (Android Studio FAQs, 2016).

En la Figura 1.21 se manifiesta que Android Studio es una herramienta muy buena para crear aplicaciones para todo sistema Android, que son sistemas operativos que se encuentran en los dispositivos móviles y ayudan al usuario a manipular de mejor manera sus aplicaciones en dichos dispositivos.

Android VS iOS

Finalmente, es relevante mencionar que el dominio de Android sobre iOS, en la región y específicamente en Ecuador es claro, tal y como se muestra en la Figura 1.22

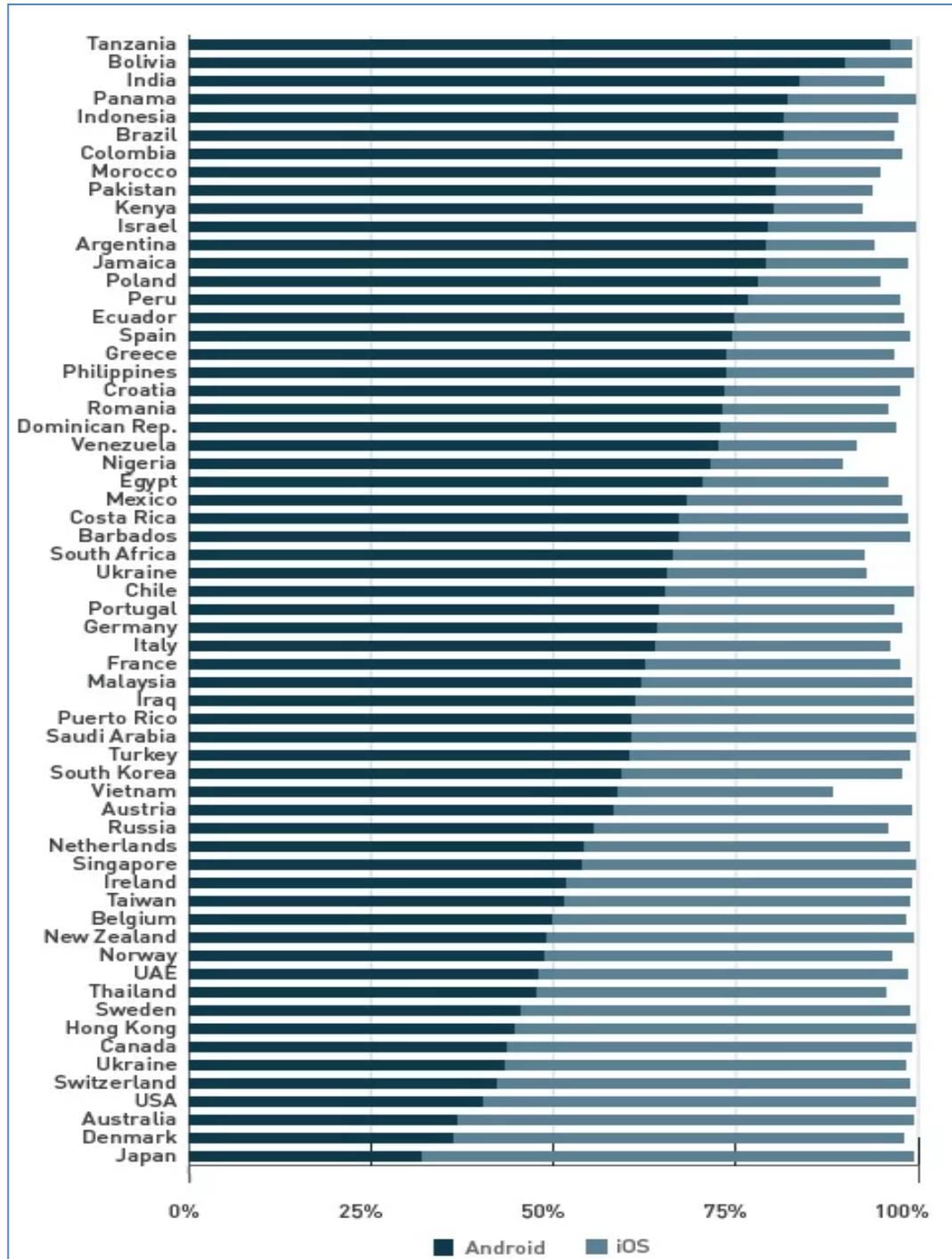


Figura 1.22 Android vs IOS

Fuente: Fuente: (Gisbert, 2017)

Dichos resultados, vigorizan la decisión de usar Android Studio en la presente propuesta.

CAPÍTULO 2: PROPUESTA

Es preciso ser muy cuidadoso a la hora de utilizar el gas doméstico como fuente energética, pues está comprobado que la mayoría de los accidentes están vinculados a descuidos principalmente de las personas perjudicadas (que sufren algún tipo de daño). El uso del gas en el hogar representa un doble riesgo para las personas, ya que es una sustancia inflamable y su inhalación es tóxica; por lo que, las precauciones deben centrarse en evitar que se produzcan fugas en lugares sin ventilación, imposibilitando una explosión o intoxicación. Considerando así que, el almacenamiento del GLP es la parte más crítica, en cuanto a seguridad en un sistema que utilice este combustible. Todo esfuerzo realizado por minimizar los riesgos y precautelar la integridad de las personas y las edificaciones es de vital importancia hacerlo, por ello es necesario ejecutar la inspección y el mantenimiento frecuente en las instalaciones de un combustible (Venegas, Ayabaca, Celi, & Rocha, 2018, pág. 27).

Antes de la implementación de la solución a dicho problema, es necesario conocer los requerimientos necesarios para el desarrollo, con el fin de conocer los lineamientos que se deben seguir, al respecto. Es vital cumplir con las especificaciones de la Norma NTEINEN 2260, la cual trata de las instalaciones de gases combustibles para uso residencial, comercial e industrial, la cual es como se mostró en la sección 1.2.6, se aplican a las instalaciones receptoras que utilizan gases combustibles suministrados desde tanques, cilindros portátiles, redes de distribución, que corresponden a los diferentes tipos de gases cuya presión máxima de servicio sea inferior o igual a 500 kPa⁶, que abarcan los gases de la primera familia (ciudad), la segunda (natural) y la tercera (GLP).

En la actualidad, existe un regulador para cilindros de uso doméstico, que la empresa “ENI Ecuador” ha fabricado, éste cumple con altos estándares de calidad, así como también con la norma INEN 1682:98 “Reguladores de baja presión para gas licuado de Petróleo (GLP)”, este dispositivo controla el flujo de gas y mantiene la presión uniforme y estable, así garantiza al usuario seguridad, eficiencia y confiabilidad en el uso del cilindro doméstico.

⁶ kPa: Kilo Pascal, es un múltiplo del Pascal, que es una unidad de presión atmosférica, que equivale a la Fuerza que ejerce 1 newton sobre la superficie de 1m².



Figura 2.1 Regulador para cilindros de uso doméstico
Fuente: (ENI Ecuador, 2018)

2.1 Solución Tecnológica del proyecto

En la Figura 2.2 se presenta el boceto del presente proyecto, es decir el proceso que se siguió para demostrar el funcionamiento del mismo; desde la conexión del sensor de gas en el cilindro, el proceso de control por parte del microcontrolador Arduino, y por último la interacción con el teléfono celular mediante la aplicación elaborada en Android Studio.

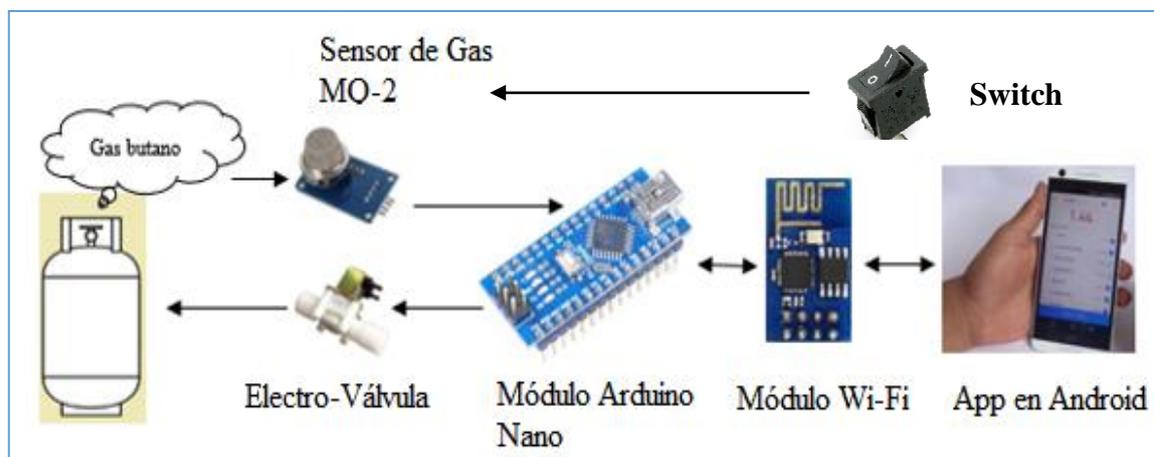


Figura 2.2 Esquema del Proyecto
Elaborado por: Autor

2.2 Diagrama de Flujo del Sistema

El funcionamiento del prototipo, comienza con el encendido del *Switch* que se encuentra instalado, lo que permitirá la conexión del teléfono celular con el módulo Wi-Fi. Mediante el teléfono celular se ingresará a la aplicación creada en Android Studio, para configurar el umbral de sensibilidad del sensor, si fuese necesario, esto va a permitir que el sensor sea más o menos sensible al momento de detectar el gas, dependiendo de la configuración que se realice en el microcontrolador Arduino. El sensor realiza detecciones periódicas y constantes

del ambiente, cuando este se encuentre con un nivel considerado de gas, el sensor va a alarmarse y enviar una señal, dicha alarma va a llegar mediante la aplicación al teléfono celular.

Después de detectar la alarma la electroválvula cerrará el paso de gas y se activará el ventilador para disipar el gas que se encuentre en el lugar, como el sensor va a detectar continuamente el nivel de gas, una vez que el aire esté libre de esta sustancia tóxica la electroválvula se abre y el ventilador se apagará.

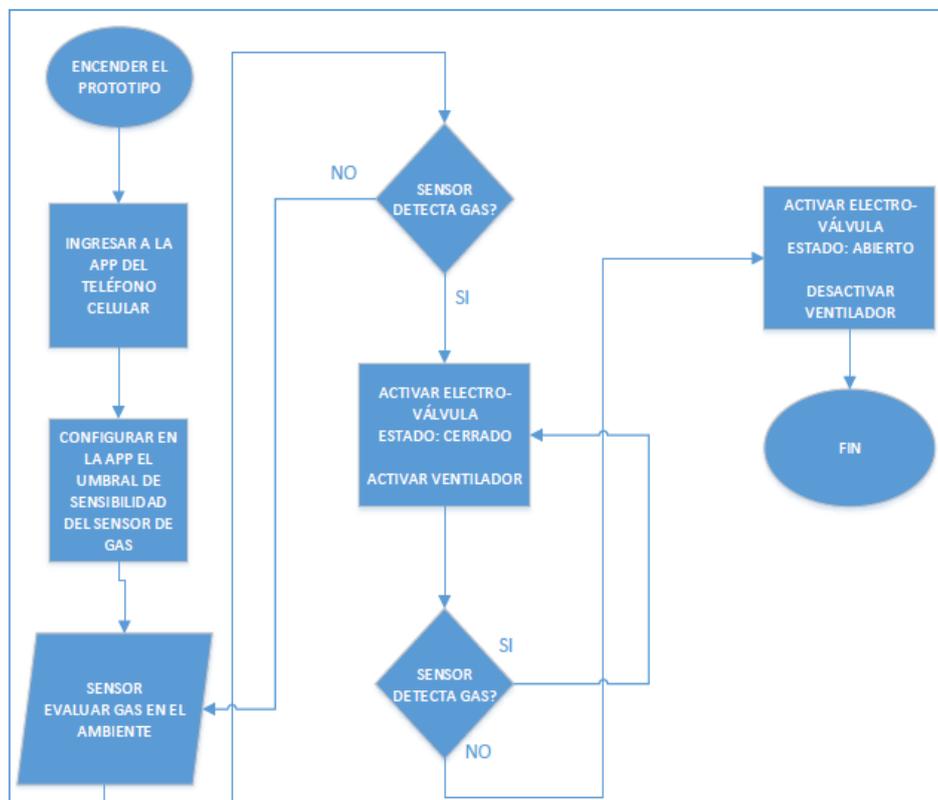


Figura 2.3 Diagrama de flujo del funcionamiento del Sistema

Elaborado por: Autor

En la Figura 2.3 se muestra el funcionamiento del sistema, primero se enciende el prototipo, luego se debe ingresar la aplicación en el dispositivo móvil, en esta aplicación se configurará el umbral de sensibilidad que va a tener el sensor de gas con esto el sensor detectará si en el ambiente habrá presencia de gas GLP. Al detectar gas se va activar la electroválvula en estado cerrado, ahí se activará el ventilador cuando el sensor detecte la presencia de gas GLP. El sensor nuevamente empezará a detectar la presencia de gas, y se activará la electroválvula en estado abierto cuando no detecte presencia de gas GLP y se desactivará el ventilador.

2.3 Diagrama de Bloques

En esta sección se muestran los componentes electrónicos y en base a que fueron seleccionados para la elaboración y construcción del prototipo; para esto, se desarrollan dos aristas, por un lado se presenta el diseño de *hardware* y por el otro, el diseño de *software*.

2.3.1 Diseño del Hardware

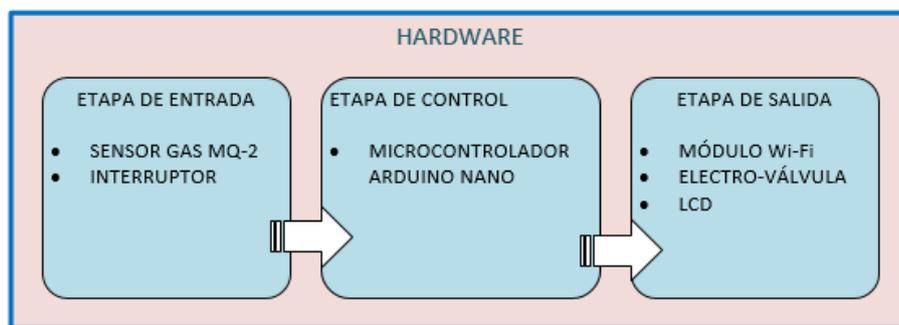


Figura 2.4 Diagrama de bloques del *hardware* del prototipo
Elaborado por: Autor

En la Figura 2.4 se muestran las etapas del prototipo, se tiene primeramente la etapa de entrada, que está conforma por el sensor de gas MQ-2 y el interruptor, en la siguiente etapa se tiene la de control, que está conformada por el microcontrolador y el Arduino nano, y por último la etapa de salida, que está conformada por el módulo Wi-Fi, la electroválvula, y el LCD.

2.3.2 Etapa de Entrada

Como se indicó en el apartado anterior, la Figura 2.4 muestra los componentes del *hardware*, en la etapa de entrada se utiliza el sensor de gas MQ-2, el cual está conectado de manera analógica. Con esta conexión se medirá la cantidad de gas en el aire en ppm (partes por millón). Adicional al sensor también se tiene un interruptor, el cual permitirá el encendido y apagado del prototipo.

2.3.3 Etapa de Control

En esta etapa el módulo Arduino Nano (seleccionado porque su tamaño es reducido, con una memoria de 16 KB, y funciona con un voltaje que puede oscilar entre 5 voltios a 12 voltios) se interconecta con el módulo Wi-Fi ESP-12F (seleccionado por su bajo consumo de energía, por su confiabilidad para el envío de datos hacia Internet, permite configuración de parámetros mediante comandos AT); ambos poseen la programación necesaria para su intercomunicación.

2.3.4 Etapa de Salida

El módulo Arduino Nano, interactúa con el módulo Wi-Fi, y la electroválvula. Permite verificar notificaciones del estado del sensor, y la entrega de esa información al usuario mediante el teléfono celular, y el LCD (display de cristal líquido), el cual exhibe mensajes, por ejemplo: la detección del gas, símbolo de la conexión Wi-Fi, entre otros.

2.3.5 Diseño de Software

En la figura 2.5 se muestran las etapas de funcionamiento que debe proporcionar el *software* del proyecto, las cuales se detallan a continuación, en los apartados consiguientes.

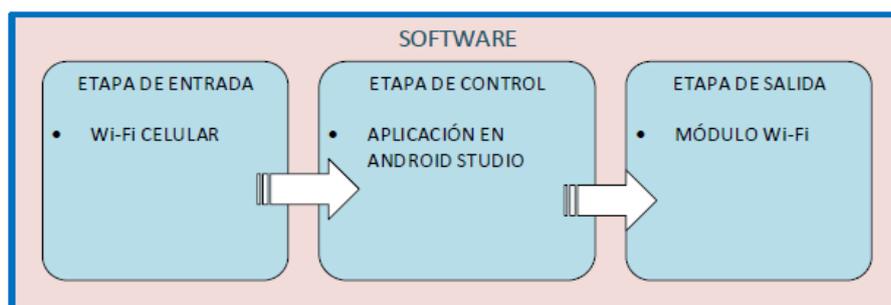


Figura 2.5 Diagrama de bloques del *software* del prototipo

Elaborado por: Autor

2.3.6 Etapa de Entrada

En esta etapa se realiza la activación del hardware del proyecto, mediante la señal inalámbrica del Wi-Fi, que va a transmitir la información hacia la siguiente etapa.

2.3.7 Etapa de Control

La etapa de control consta de la aplicación creada en Android Studio, que permite la configuración y modificación de los niveles de detección del sensor de gas, y que además, recibe las alarmas de notificaciones, en caso de haber una fuga de gas.

2.3.8 Etapa de Salida

La información y configuraciones que se realicen se envían mediante la señal inalámbrica, que envía la información para ser transmitida hacia el módulo Wi-Fi.

2.4 Análisis de Costos y Tiempo requerido

Respecto a los costos incurridos, estos abarcan esencialmente a las piezas que forman el prototipo, dichos valores se detallan a continuación.

Tabla 2.1 Presupuesto del proyecto.

Componente	Valor en USD
Arduino	\$15,00
Sensor de gas	\$20,00
Placa	\$15,00
Válvula solenoide	\$35,00
LCD	\$12,00
Ventilador	\$8,00
Elementos varios (lápiz, esféros, hojas, tinta impresora, transporte)	\$50,00
Subtotal	\$155,00
Imprevistos 10% (impresiones, anillados nuevos a color y blanco y negro, pasajes)	\$15,50
Total	\$170,50

Elaborado por: Autor.

2.5 Ventajas del Producto

La correspondiente propuesta, permitirá reducir el número de incidentes que se producen por la exposición al GLP, ya que mediante la correspondiente alarma al detectar alguna fuga y el control vía inalámbrica, mediante el uso de un teléfono celular, se podrá solucionarla de manera pronta y expedita, y así evitar cualquier tipo de accidente, como una explosión, mala combustión y/o exposición de monóxido de carbono por quienes habitan en el domicilio donde se posea el dispositivo en cuestión.

2.6 Cálculos de voltajes y corrientes en el circuito de activación de la electroválvula

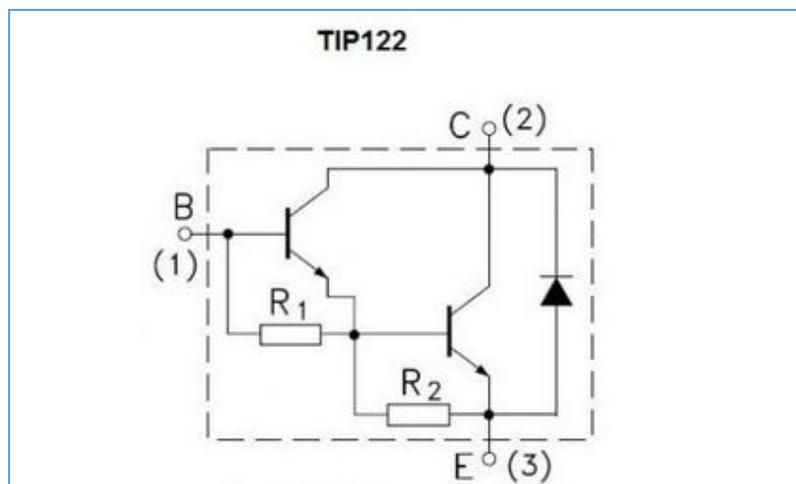


Figura 2.6 Circuito interno del transistor Darlington TIP122

Elaborado por: Autor.

Characteristic	Symbol	Min	Max	Unit
OFF CHARACTERISTICS				
Collector–Emitter Sustaining Voltage (Note 2) ($I_C = 100\text{ mA}$, $I_B = 0$)	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	$V_{CE(sus)}$	60 80 100	– – – Vdo
Collector Cutoff Current ($V_{CE} = 30\text{ Vdo}$, $I_B = 0$) ($V_{CE} = 40\text{ Vdo}$, $I_B = 0$) ($V_{CE} = 50\text{ Vdo}$, $I_B = 0$)	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	I_{CEO}	– – –	0.5 0.5 0.5 mA
Collector Cutoff Current ($V_{CB} = 60\text{ Vdo}$, $I_E = 0$) ($V_{CB} = 80\text{ Vdo}$, $I_E = 0$) ($V_{CB} = 100\text{ Vdo}$, $I_E = 0$)	TIP120, TIP125 TIP121, TIP126 TIP122, TIP127	I_{CBO}	– – –	0.2 0.2 0.2 mA
Emitter Cutoff Current ($V_{BE} = 5.0\text{ Vdo}$, $I_C = 0$)		I_{EBO}	–	2.0 mA
ON CHARACTERISTICS (Note 2)				
DC Current Gain ($I_C = 0.5\text{ A}$, $V_{CE} = 3.0\text{ Vdo}$) ($I_C = 3.0\text{ A}$, $V_{CE} = 3.0\text{ Vdo}$)		h_{FE}	1000 1000	– –
Collector–Emitter Saturation Voltage ($I_C = 3.0\text{ A}$, $I_B = 12\text{ mA}$) ($I_C = 5.0\text{ A}$, $I_B = 20\text{ mA}$)		$V_{CE(sat)}$	– –	2.0 4.0 Vdo
Base–Emitter On Voltage ($I_C = 3.0\text{ A}$, $V_{CE} = 3.0\text{ Vdo}$)		$V_{BE(on)}$	–	2.5 Vdo
DYNAMIC CHARACTERISTICS				
Small–Signal Current Gain ($I_C = 3.0\text{ A}$, $V_{CE} = 4.0\text{ Vdo}$, $f = 1.0\text{ MHz}$)		h_{fe}	4.0	–
Output Capacitance ($V_{CB} = 10\text{ Vdo}$, $I_E = 0$, $f = 0.1\text{ MHz}$)	TIP125, TIP126, TIP127 TIP120, TIP121, TIP122	C_{ob}	– –	300 200 pF
Product parametric performance is indicated in the Electrical Characteristics for the listed test conditions, unless otherwise noted. Product performance may not be indicated by the Electrical Characteristics if operated under different conditions. 2. Pulse Test: Pulse Width $\leq 300\text{ }\mu\text{s}$, Duty Cycle $\leq 2\%$				

Figura 2.7 Hoja de especificaciones del TIP122
Elaborado por: Autor.

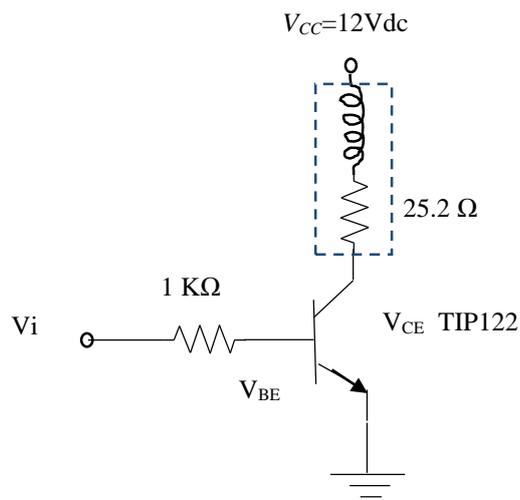


Figura 2.8 Electroválvula
Elaborado por: Autor.

DATOS

VCC= 12V

RB= 1000Ω ó 1KΩ

VBE (on)= 2.5V

VCE (sat)= 2V (on)

R electroválvula = 25.2 (Medido con el óhmetro)

VBE (off)= 5 Vdc

La electroválvula está activada

VBE (on) = 2.5V

VCE (sat) = 2V (on)

CÁLCULOS

Cálculo de la corriente de base:

$$I_B = \frac{V_{BE} (OFF) - V_{BE} (SAT)}{R_B}$$

$$I_B = \frac{5 - 2.5}{1000} = 2.5mA$$

Cálculo de la corriente de colector:

$$I_C = \frac{V_{CC} - V_{CE} (SAT)}{R_{electroválvula}}$$

$$I_C = \frac{12-2}{25.2} = 0.39A$$

Cálculo del voltaje en la electroválvula V_E

$$V_E = V_{CC} - V_{BE}(SAT)$$

$$V_E = 12 - 2.5$$

$$V_E = 9.5V$$

CAPITULO 3: IMPLEMENTACIÓN

3.1 Desarrollo

En el presente capítulo se explica mediante diagramas y gráficos, cómo se llevó a cabo la implementación del proyecto, las pruebas necesarias que se realizaron para probar y comprobar que el diseño se encuentra óptimo para su uso.

3.1.1 Esquema de proyecto

En la Figura 3.1 se muestra el diagrama esquemático del proyecto, donde se visualizan los componentes que son parte del diseño como: el microcontrolador Arduino Nano, el módulo LCD, el sensor de gas, el módulo Wi-Fi, entre otros, que se encuentran en el programa Proteus de simulación.

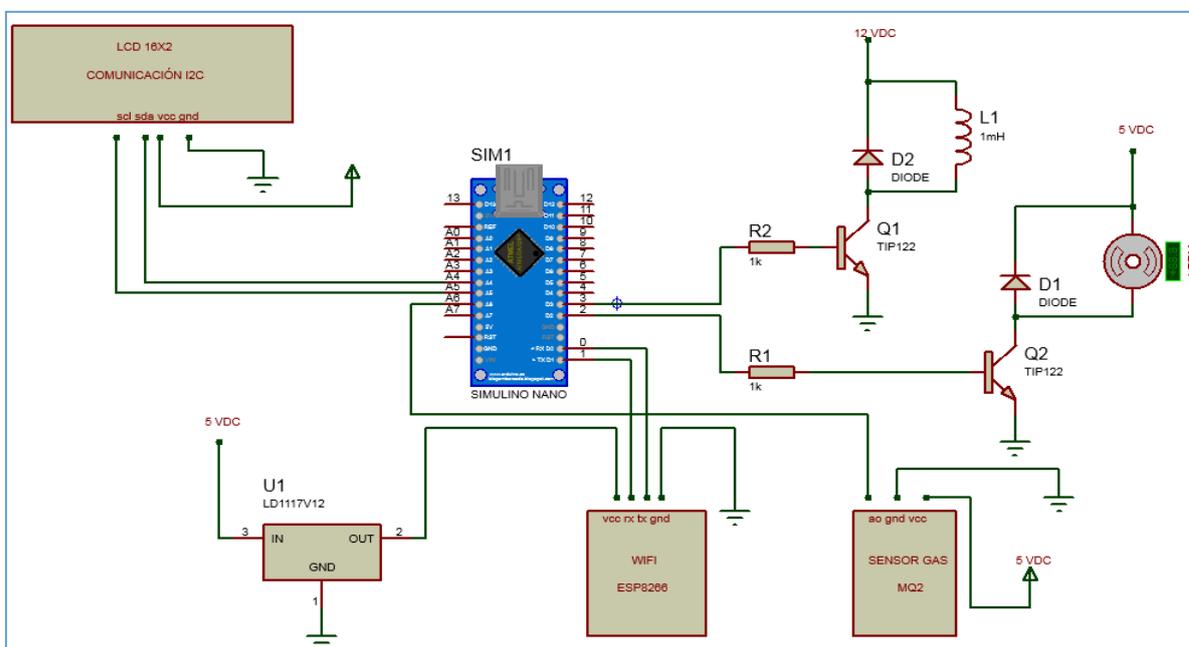


Figura 3.1 Diagrama Esquemático

Elaborado por: Autor

Al microcontrolador Arduino Nano se encuentran conectado los módulos, Wi-Fi, el sensor de gas, el ventilador, la electroválvula, y demás componentes electrónicos, por este

motivo es el encargado de la parte de control del Prototipo, además en el microcontrolador se encuentra la programación necesaria que se va a ejecutar.

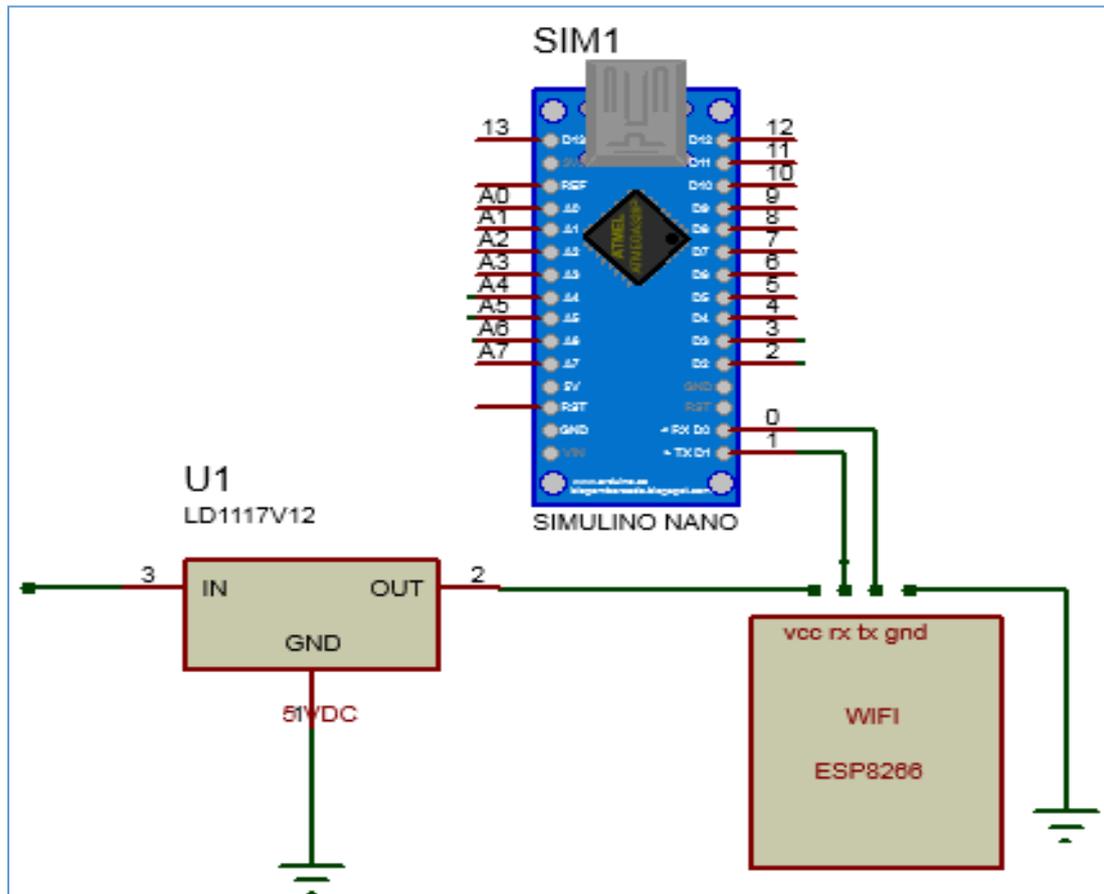


Figura 3.2 Conexión entre el módulo Wi-Fi y el módulo Arduino Nano.
Elaborado por: Autor

La figura 3.2 ilustra la conexión del microcontrolador Arduino Nano y el módulo Wi-Fi ESP8266, dicha comunicación se realiza a través del puerto serie, el módulo Wi-Fi puede configurarse mediante comandos AT la velocidad de transmisión que puede ser: 9600, 19200, 38400, 74880, 115200, 230400, 460800 y 921600, pero por defecto se usa a 9600 baudios, los comandos AT son comandos de texto enviados por Serial.

La conexión se realiza mediante los pines de transmisión y recepción de ambos módulos, el pin de transmisión del módulo Wi-Fi irá conectado al pin de la recepción de módulo Arduino y el pin de recepción del módulo Wi-Fi irá conectado al pin de transmisión del microcontrolador. Además está conectado a un transistor que transforma la energía a los 3.3V que funciona el módulo Wi-Fi.

Estos pines se observa en la figura 3.2 en la parte del Arduino Nano

D0 y A0 = son salidas de datos

D0= salida digital

A0= Salida analógica

Vcc= Voltaje continuo

GND= Tierra

Vcc y GND son los pines para la conexión del voltaje de alimentación y el otro pin es la Tierra.

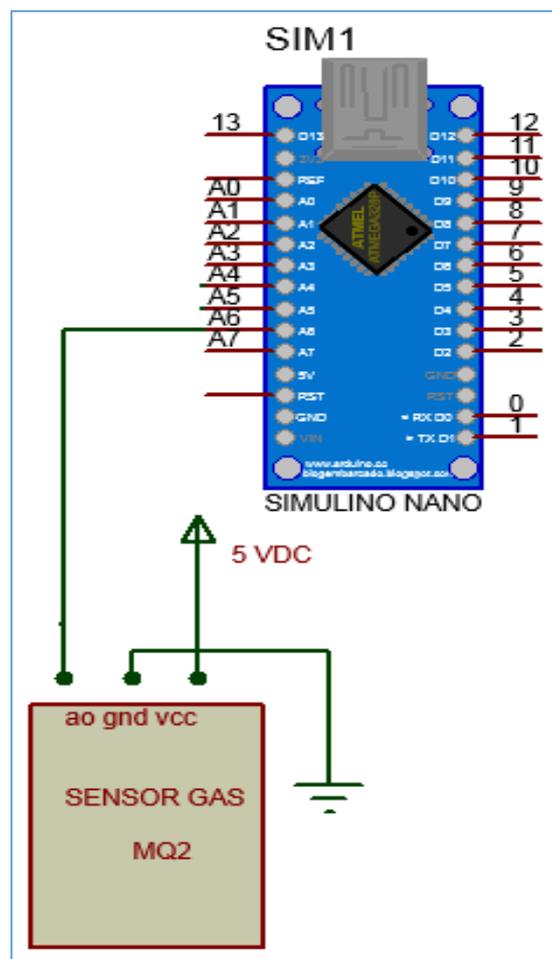


Figura 3.3 Conexión entre el sensor MQ-2 y módulo Arduino Nano
Elaborado por: Autor

En la Figura 3.3 el sensor MQ-2 consta de 4 pines de conexión: V_{CC}, GND, DO y AO. DO y AO son salidas de datos, la primera digital y la segunda analógica, para el proyecto se usa la salida de datos analógica AO, esta se encuentra conectada al pin A6 del microcontrolador Arduino Nano, este pin recibe señales analógicas. La salida analógica amplifica el valor del

voltaje en proporción al nivel de gas detectado. V_{CC} y GND son pines de conexión para el voltaje de alimentación del sensor.

3.1.2 Construcción del Prototipo

Para iniciar la construcción del prototipo primero se realizaron los diagramas de las PCB (Placas de Circuito Impreso), a partir de los diagramas esquemáticos obtenidos, los cuales fueron realizados mediante el *software* de diseño Proteus con la herramienta para la creación de PCB llamada Ares. Con los diagramas se fabricaron las placas de circuito impreso correspondientes, como se muestra en la Figura 3.4

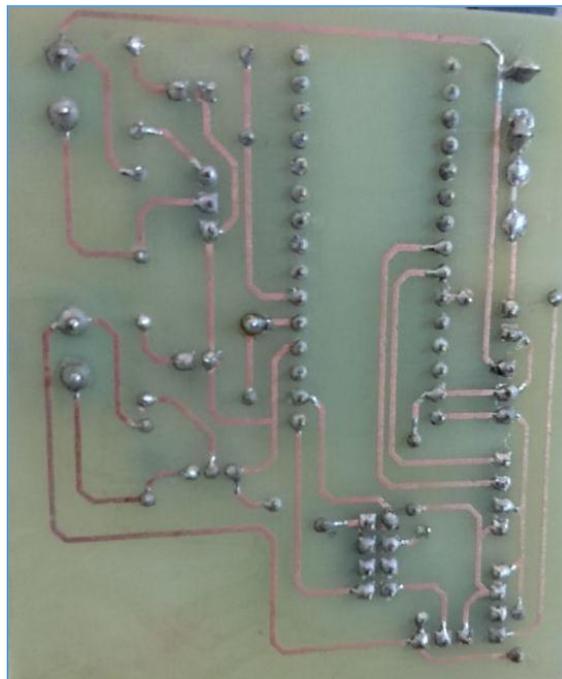


Figura 3.4 Fotografía de la tarjeta electrónica elaborada
Elaborado por: Autor

Los componentes que se usaron son los siguientes:

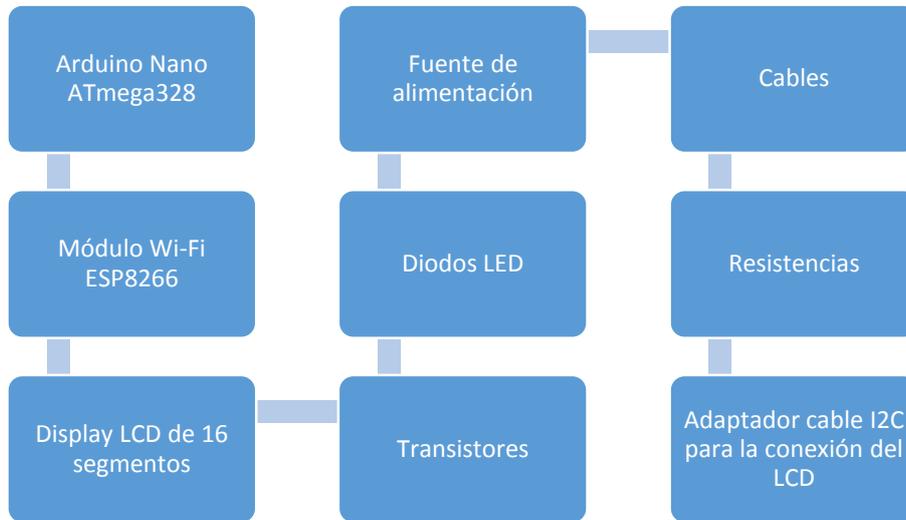


Figura 3.5 Componentes usados.
Elaborado por: Autor

En la Figura 3.5 se observa que los componentes que se utilizaron para el proyecto fueron los siguientes. Arduino Nano, Módulo Wi-Fi ESP8266, Display LCD de 16 segmentos, Transistores, Diodos LED, Fuente de Alimentación, Cables de conexión, Resistencias, Adaptador de cables para la conexión del LCD.

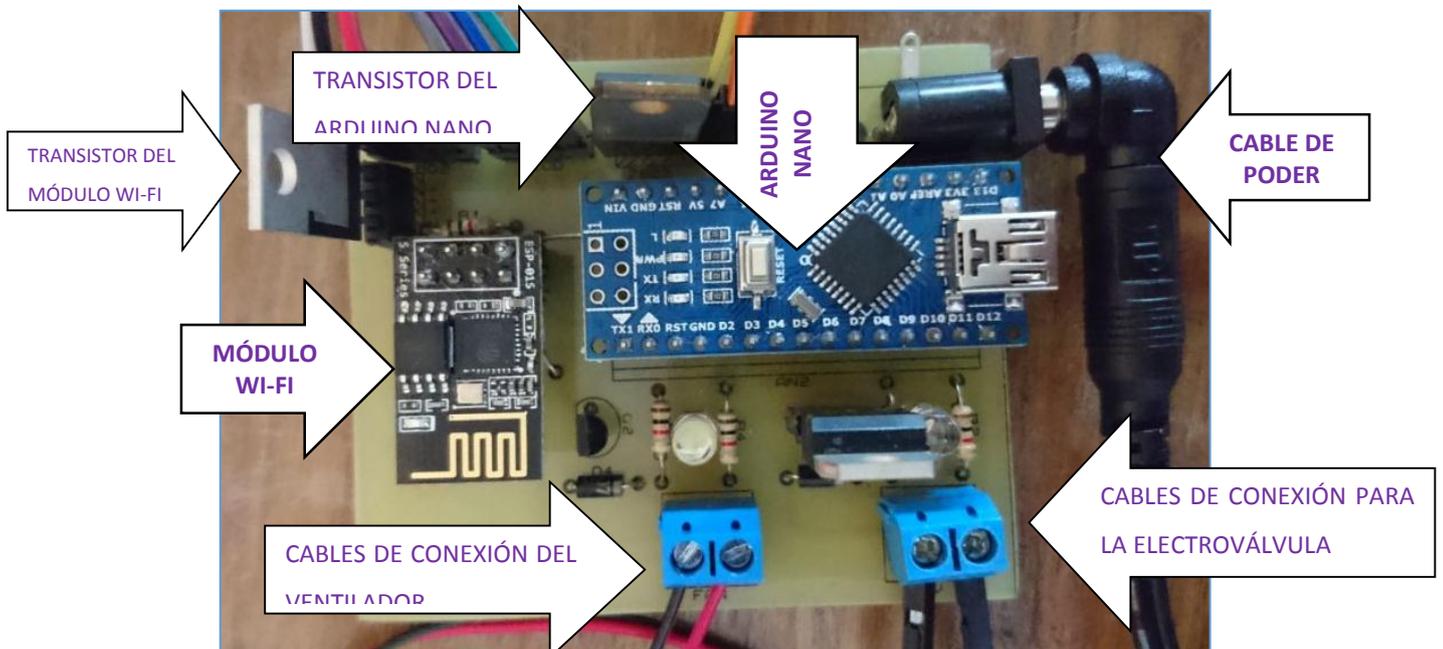


Figura 3.6 PCB del Prototipo completo
Elaborado por: Autor

En la Figura 3.6 se muestra la tarjeta de circuito impreso terminada, con los elementos y dispositivos colocados de manera correcta, se ha empleado cables de conexión y sócalos para que sea más sencillo el reemplazar los módulos de ser necesario.

En la Figura 3.7, se observan los componentes principales y aquellos dispositivos necesarios para su conexión o interconexión, como son los cables de conexión, sócalos, borneras, entre otros.

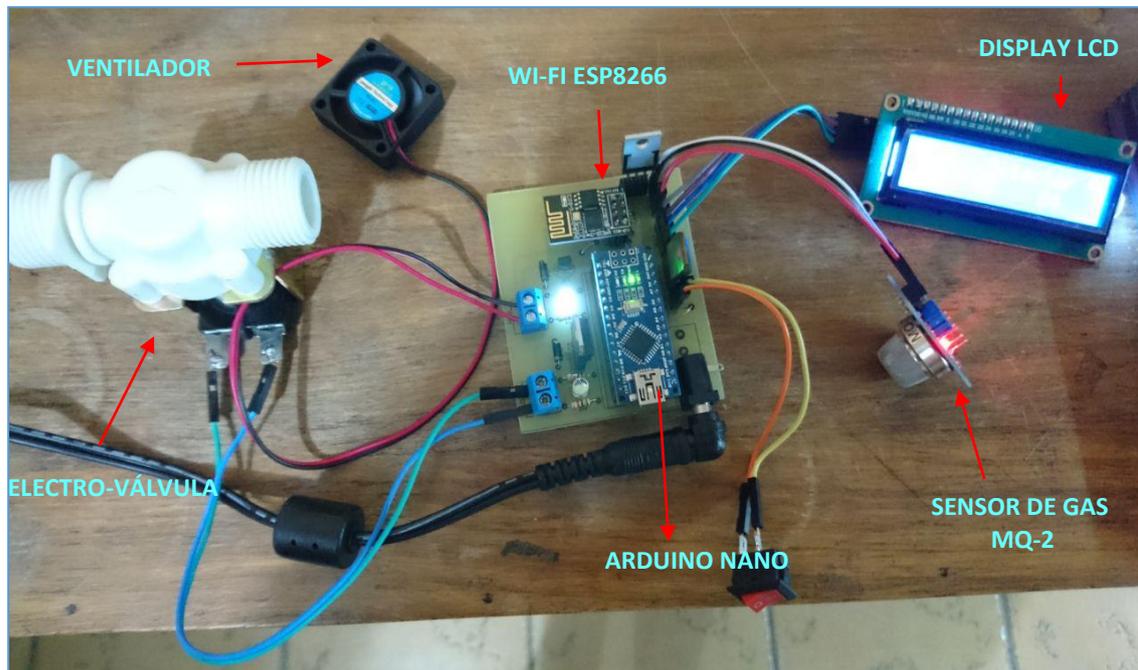


Figura 3.7 Tarjeta del Prototipo Completa
Elaborado por: Autor

3.1.3 Implementación del Hardware

En el circuito armado de la Figura 3.7, se identifican los componentes utilizados para comprobar la conectividad entre el Arduino Nano y los diferentes módulos como: el módulo WI-FI, el sensor de gas, así como el funcionamiento de la electroválvula y el ventilador.

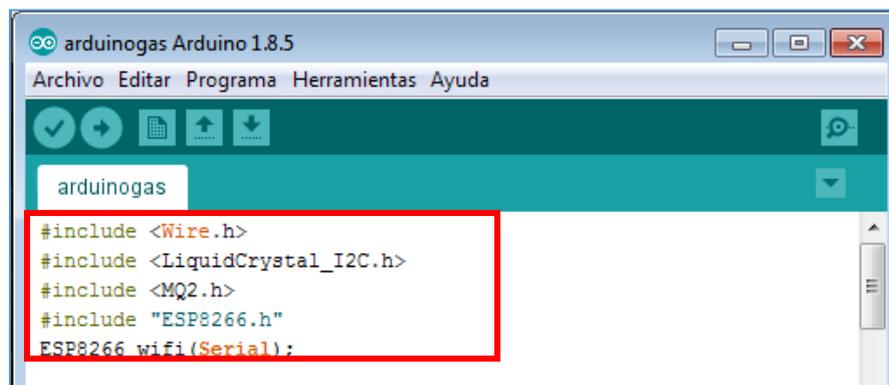
El LCD (pantalla de cristal líquido) muestra el aviso cuando exista la fuga de gas, se visualizan los valores de sensibilidad que el usuario podrá configurar mediante la aplicación Android, pues va a ser más o menos sensible a la detección del gas, según su configuración.

3.1.4 Implementación del Software

Las instrucciones creadas para el prototipo se desarrollaron en el programa Arduino y fueron cargadas directamente en el módulo Arduino Nano ATmega328, por medio de un cable USB.

3.1.5 Programación en Arduino

A continuación, se detalla el conjunto de instrucciones de la programación para el Arduino Nano. Se inicia colocando las librerías necesarias para el manejo del Display LCD, el conector I2C (para el Display LCD), el sensor MQ-2, el módulo WI-FI, como se muestra en la figura 3.8.

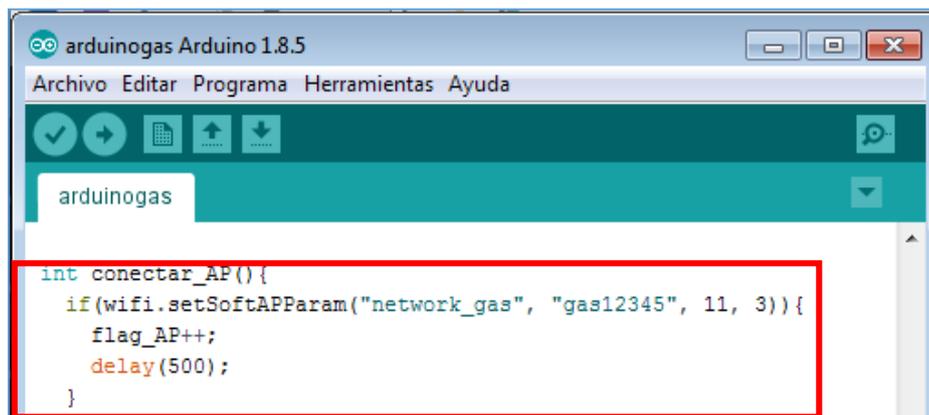


The screenshot shows the Arduino IDE interface for a project named 'arduinogas'. The code editor displays the following code, which is highlighted with a red box:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <MQ2.h>
#include "ESP8266.h"
ESP8266_wifi(Serial);
```

Figura 3.8 Códigos de Programación en Arduino
Elaborado por: Autor

En la Figura 3.9 es la programación en Arduino comprende varios aspectos, la configuración del módulo Wi-Fi, la configuración de los niveles de sensibilidad del sensor de gas, y la comunicación entre los diferentes módulos.



The screenshot shows the Arduino IDE interface for the same project. The code editor displays the following code, which is highlighted with a red box:

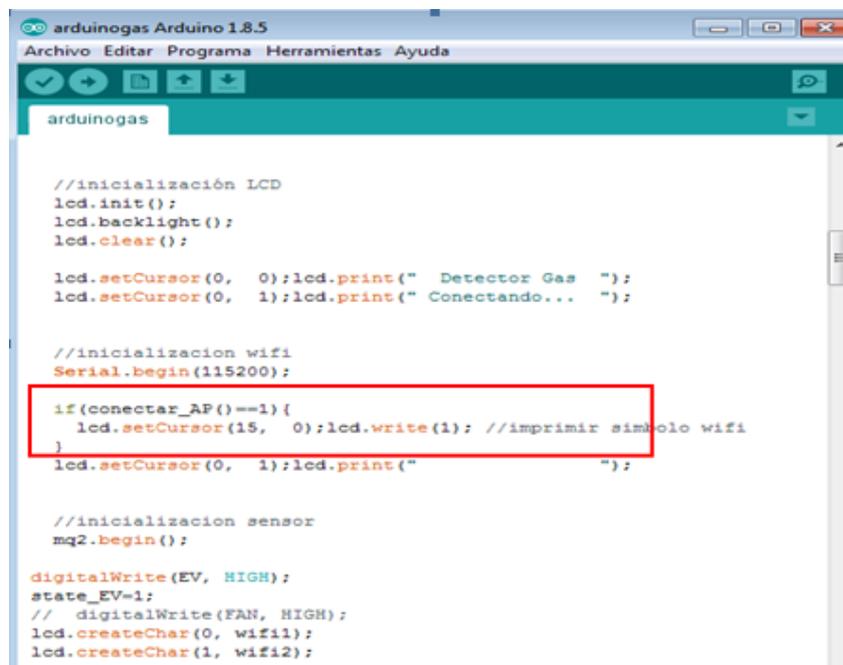
```
int conectar_AP(){
  if(wifi.setSoftAPParam("network_gas", "gas12345", 11, 3)){
    flag_AP++;
    delay(500);
  }
}
```

Figura 3.9 Sección de Programación del Arduino Nano

Elaborado por: Autor.

En el *software* de Arduino, se define la configuración del módulo Wi-Fi para la conexión a la red, es decir el SSID⁷ y la contraseña de acceso a la red inalámbrica. El SSID es el nombre con el que se identifica a la red inalámbrica en este caso “*network_gas*” y la contraseña “*gas12345*”, como se muestra en la Figura 3.9.

Siguiendo con la programación del Arduino, en la Figura 3.10 se visualiza la inicialización del módulo LCD, y se encera para que no aparezcan símbolos que no se identifiquen o no sean parte de la información que se quiere ver.



```
arduinoogas Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

arduinoogas

//inicialización LCD
lcd.init();
lcd.backlight();
lcd.clear();

lcd.setCursor(0, 0);lcd.print(" Detector Gas ");
lcd.setCursor(0, 1);lcd.print(" Conectando... ");

//inicializacion wifi
Serial.begin(115200);

if(conectar_AP()==1){
  lcd.setCursor(15, 0);lcd.write(1); //imprimir simbolo wifi
}
lcd.setCursor(0, 1);lcd.print(" ");

//inicializacion sensor
mq2.begin();

digitalWrite(EV, HIGH);
state_EV=1;
// digitalWrite(FAN, HIGH);
lcd.createChar(0, wifil);
lcd.createChar(1, wifl2);
```

Figura 3.10 Inicialización del LCD

Elaborado por: Autor.

Programa en Android Studio

Continuando con la implementación del *software* del proyecto, y como ya se mencionó anteriormente, se recalca que Android Studio es un programa que permite realizar aplicaciones para telefonos inteligentes, sobre la plataforma Android.

Android Studio maneja un panel que representa en un árbol de jerarquías, los directorios que se crean y se los puede visualizar desde Windows

⁷ SSID: (Service Set Identifier) o identificador de paquetes de servicio, el nombre con el que se identifica una red Wi-Fi.

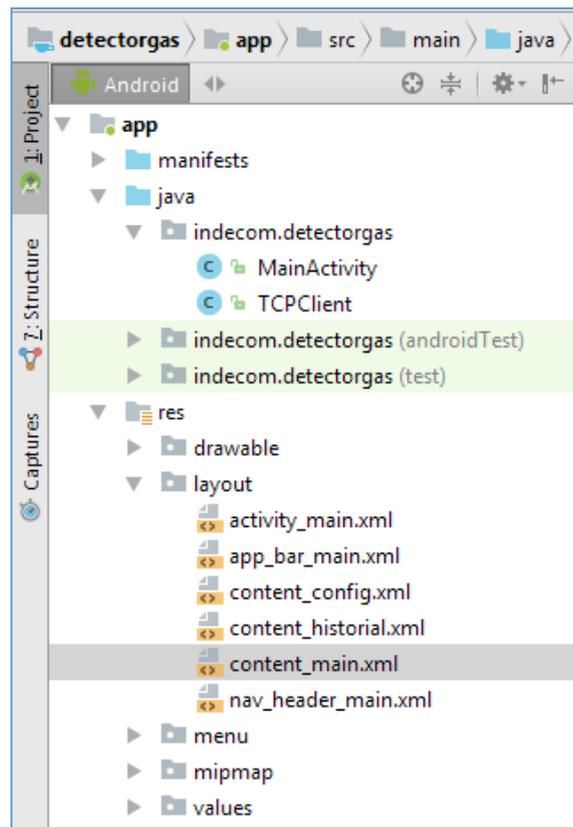


Figura 3.11 Árbol de jerarquías de Android Studio
Elaborado por: Autor

Por defecto se abre en el área de trabajo dos archivos: MainActivity.java y content_main.xml, la primera contiene la lógica de la aplicación, es decir el funcionamiento, este archivo es el que determina con funciona la aplicación, por ejemplo al pulsar un botón que es lo que va a ocurrir, o al desplegar un menú que es lo que tiene que aparecer, el segundo hace referencia al aspecto visual o a la interfaz de la aplicación, es decir, como se visualiza la aplicación, colores, formas, botones, audio, video, entre otros.

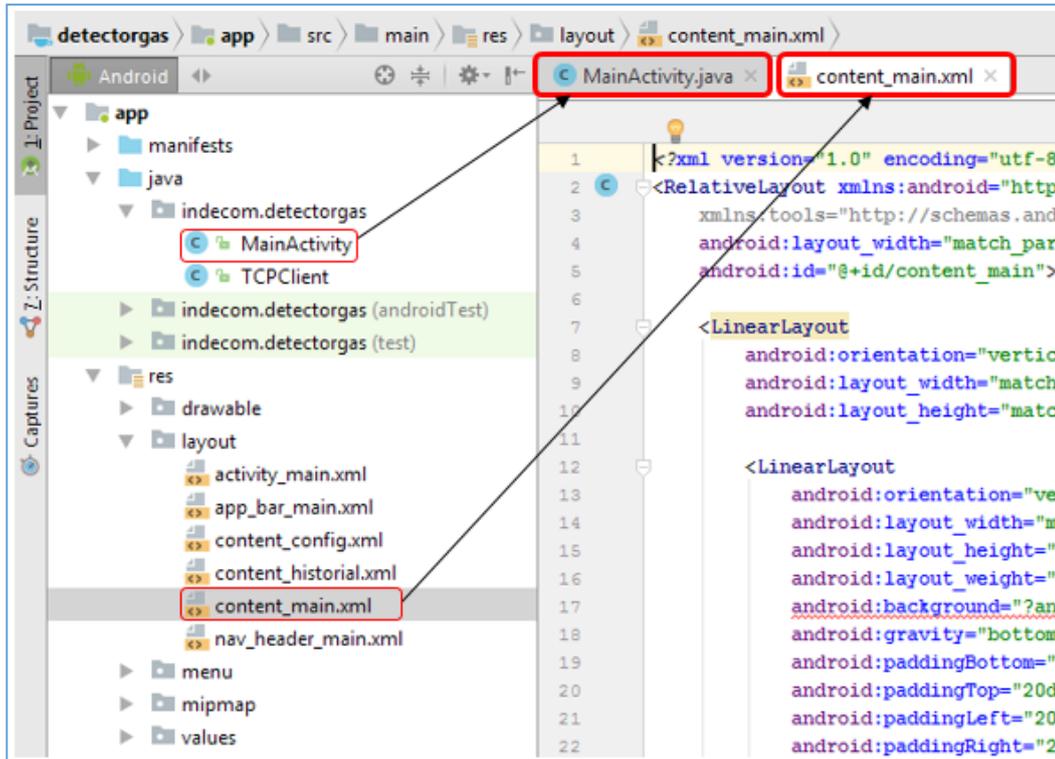


Figura 3.12 Archivos por defecto del área de trabajo de Android Studio

Elaborado por: Autor.

Existen dos maneras de trabajar en la parte visual de Android Studio, que sería en forma visual o mediante códigos, en el archivo `content_main.xml` se observa dos pestañas inferiores, *Design* y *Text*. *Design* habilita todos los componentes que se van a agregar a la aplicación, en este caso los botones, en cambio si se escoge la opción *Text*, permitirá realizar la programación en el archivo xml con su modificación.

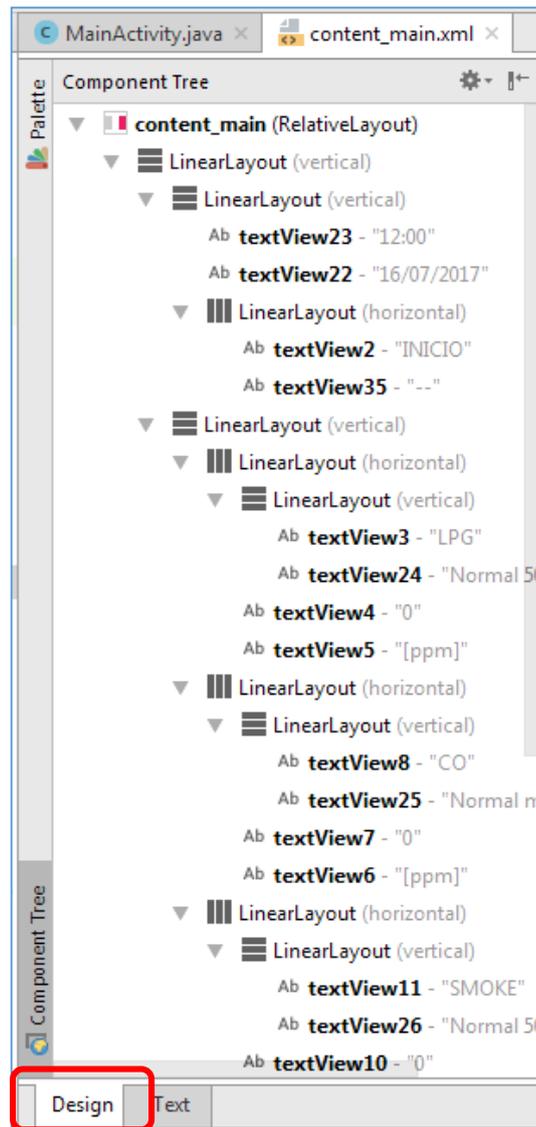


Figura 3.13 Componentes del archivo content_main.xml

Elaborado por: Autor.

Al escoger la opción *Design* se elabora la aplicación con el uso del menú de la parte izquierda, con los componentes necesarios y arrastrarlos hacia la derecha según el lugar donde va a ir ubicado. Cada botón tiene propiedades que puede ser configuradas como: color, tamaño y función que va a realizar, entre otras. Una vez realizada la configuración y programación de la aplicación, en la opción *Text* se modificará el código de la programación.

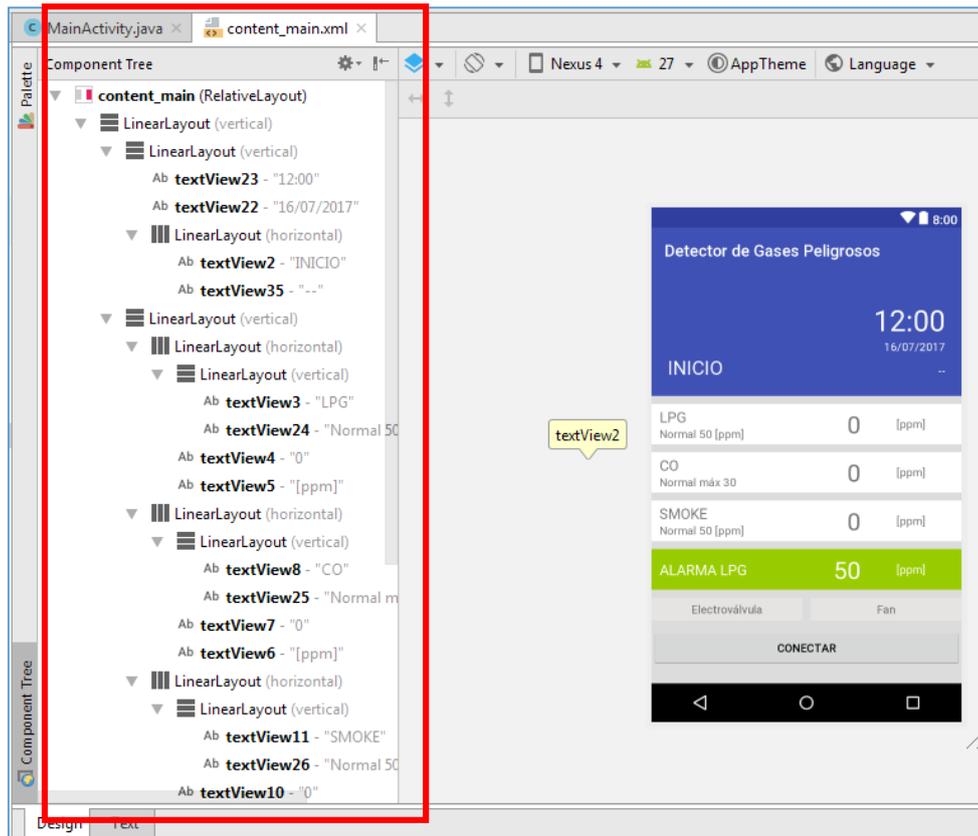


Figura 3.14 Menú Design y opciones de configuración de la aplicación
Elaborado por: Autor.

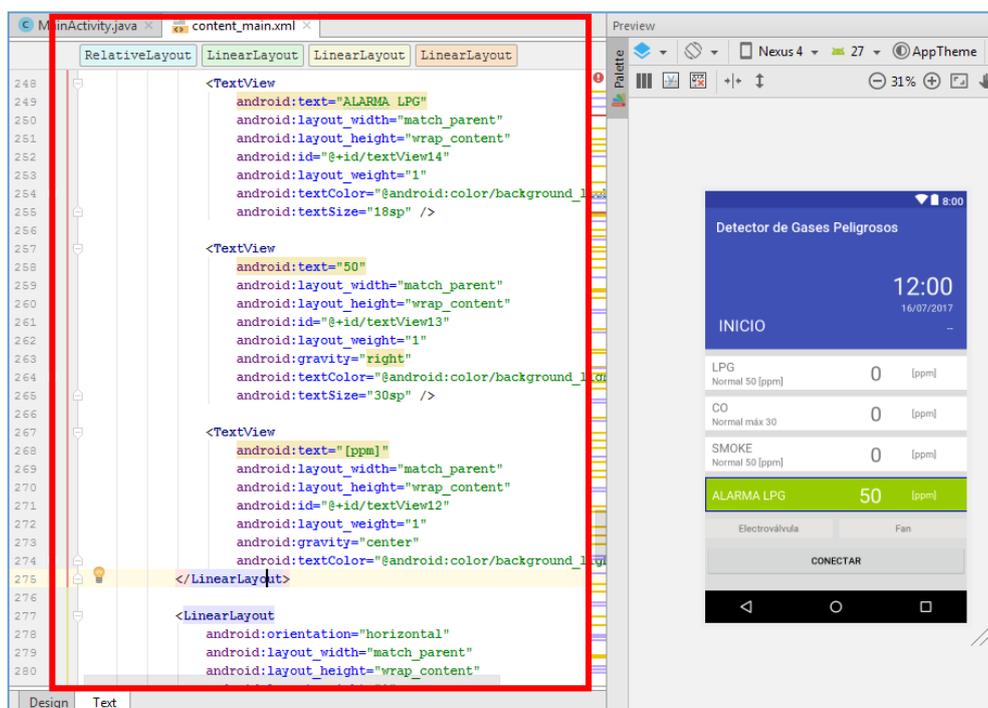


Figura 3.15 Aspecto de la aplicación
Elaborado por: Autor.

En la opción recursos, existe un submenú *mipmap* que permitirá elaborar el ícono que va a aparecer en teléfono inteligente cuando se instale la aplicación (Figura 26).

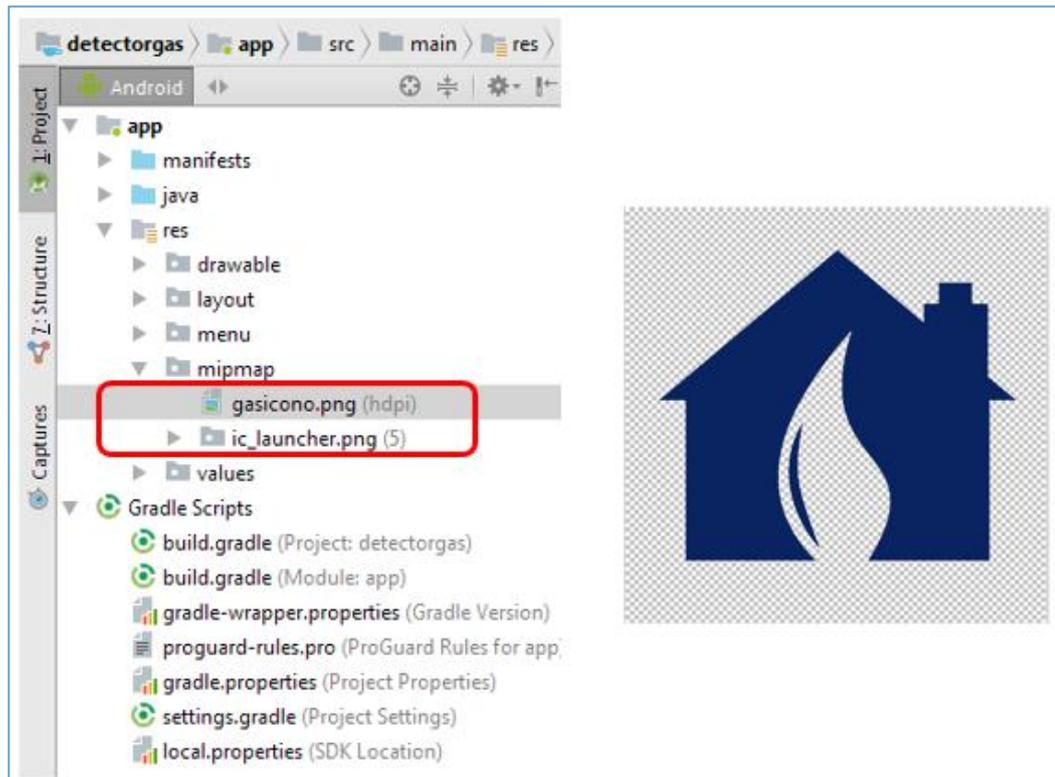


Figura 3.16 Icono de la aplicación en Android Studio

Elaborado por: Autor.

Código Android Studio

```

package indecom.detectorgas;

import android.media.MediaPlayer;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.GridView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;

```

```

import android.widget.TextView;
import android.widget.Toast;

import java.text.SimpleDateFormat;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity
implements NavigationView.OnNavigationItemSelectedListener {
private TCPClient mTcpClient;
private RelativeLayout main,historial,test,config;
private TextView textView35,textView36,textView37,textAlarma,textAlarma2;
private EditText editAlarma;
private GridView grid;
private Button
buttonEVON,buttonEVOFF,buttonFANON,buttonFANOFF,buttonCONECT,buttonEDIT,b
uttonRST;
private String[] myStringArray={" "};
private char [] arrayChar={},intChar={};
private String st_send="",st_reciv="",st_aux="";
private int
lpg,co2,smoke,estado_ev,estado_fan,conexion=0,flag_conect=0,alarmaLPG=10,
flag_alarma=0;
private MediaPlayer alarmadisp;
static final String[] reporte_string = new String[] {
"Evento1", " ", " ",
"Evento2", " ", " ",
"Evento3", " ", " ",
"Evento4", " ", " ",
"Evento5", " ", " ",
"Evento6", " ", " ",
"Evento7", " ", " ",
"Evento8", " ", " ",
"Evento9", " ", " ",
"Evento10", " ", " ",
"Evento11", " ", " ",
"Evento12", " ", " ",
"Evento13", " ", " ",
"Evento14", " ", " ",
"Evento15", " ", " ",
"Evento16", " ", " ",
"Evento17", " ", " ",
"Evento18", " ", " ",
"Evento19", " ", " ",
"Evento20", " ", " "};

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
Toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

main = (RelativeLayout) findViewById(R.id.content_main);
historial = (RelativeLayout)
findViewById(R.id.content_historial);
config = (RelativeLayout) findViewById(R.id.content_config);
grid = (GridView) findViewById(R.id.grid);

buttonEVON = (Button) findViewById(R.id.button6);

```

```

buttonEVOFF = (Button) findViewById(R.id.button5);
buttonFANON = (Button) findViewById(R.id.button3);
buttonFANOFF = (Button) findViewById(R.id.button2);
buttonCONECT = (Button) findViewById(R.id.button8);
buttonRST = (Button) findViewById(R.id.button4);
buttonEDIT = (Button) findViewById(R.id.buttonEdit);

textView35 = (TextView) findViewById(R.id.textView35);
textView36 = (TextView) findViewById(R.id.textView36);
textView37 = (TextView) findViewById(R.id.textView37);
editAlarma = (EditText) findViewById(R.id.editText);
textAlarma = (TextView) findViewById(R.id.textView30);
textAlarma2 = (TextView) findViewById(R.id.textView13);

alarmadisp = MediaPlayer.create(this, R.raw.alarmadisp );

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_1, reporte_string);
grid.setAdapter(adapter);

alarmadisp.start();

new connectTask().execute("");

/*
    for(int x=0;x<1;x++){
        try {
            Thread.sleep(500);
        } catch (Exception e) {
            e.printStackTrace();
        }
        if (mTcpClient != null) {
            mTcpClient.sendMessage("E");
        }
    }
*/

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    this, drawer, toolbar, R.string.navigation_drawer_open,
    R.string.navigation_drawer_close);
drawer.setDrawerListener(toggle);
toggle.syncState();

NavigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);

tiempo();
textAlarma.setText(""+alarmaLPG);

buttonEVON.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (mTcpClient != null) {
            mTcpClient.sendMessage("ON1");
        }
    }
});

```

```

        buttonEVOFF.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (mTcpClient != null) {
                    mTcpClient.sendMessage("OF1");
                }
            }
        });
        buttonFANON.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (mTcpClient != null) {
                    mTcpClient.sendMessage("ON2");
                }
            }
        });
        buttonFANOFF.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (mTcpClient != null) {
                    mTcpClient.sendMessage("OF2");
                }
            }
        });
        buttonCONNECT.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //int i = TCPClient.getState();
                if (conexion==0) {
                    new connectTask().execute("");

                    for(int x=0;x<1;x++){
                        try {
                            Thread.sleep(500);
                        } catch (Exception e) {
                            e.printStackTrace();
                        }
                    }
                    if (mTcpClient != null) {
                        mTcpClient.sendMessage("E");
                    }
                }
            }
        });
        buttonRST.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (mTcpClient != null) {
                    mTcpClient.sendMessage("RST");
                }
            }
        });
        buttonEDIT.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

```

```

if (!editAlarma.getText().toString().isEmpty()) {
if
((Integer.parseInt(editAlarma.getText().toString())>0)&&(Integer.parseInt
(editAlarma.getText().toString())<=100)) {
//alarmaLPG = Integer.parseInt(editAlarma.getText().toString());
//Toast.makeText(MainActivity.this, "ALARMA LPG:
"+alarmaLPG+" [ppm]", Toast.LENGTH_SHORT).show();
//textAlarma.setText(""+alarmaLPG);
if (mTcpClient != null) {
mTcpClient.sendMessage("A
"+String.format("%03d",Integer.parseInt(editAlarma.getText().toString())
);
}
editAlarma.setText("");
} else {
Toast.makeText(MainActivity.this, "ERROR!!!
INGRESE VALOR VÁLIDO", Toast.LENGTH_SHORT).show();
editAlarma.setText("");
}
}
else{
Toast.makeText(MainActivity.this, "ERROR!!! INGRESE
VALOR VÁLIDO", Toast.LENGTH_SHORT).show();
}
});
}

@Override
public void onBackPressed() {
DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
if (drawer.isDrawerOpen(GravityCompat.START)) {
drawer.closeDrawer(GravityCompat.START);
} else {
super.onBackPressed();
}
}

@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {
// Handle navigation view item clicks here.
int id = item.getItemId();

if (id == R.id.nav_home) {
main.setVisibility(View.VISIBLE);
historial.setVisibility(View.GONE);
config.setVisibility(View.GONE);
} else if (id == R.id.nav_historial) {
main.setVisibility(View.GONE);
historial.setVisibility(View.VISIBLE);
config.setVisibility(View.GONE);
} else if (id == R.id.nav_config) {
main.setVisibility(View.GONE);
historial.setVisibility(View.GONE);
config.setVisibility(View.VISIBLE);
}
}

```

```

        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
return true;
    }

//se ejecuta cada 1 segundo
public void tiempo (){
new CountdownTimer(1000,500){
    @Override
public void onTick(long millisUntilFinished) {
    }
    @Override
public void onFinish() {

        TextView textView23 = (TextView)
findViewById(R.id.textView23);
        TextView textView22 = (TextView)
findViewById(R.id.textView22);

        Calendar c = Calendar.getInstance();
SimpleDateFormat dh = new SimpleDateFormat("HH:mm");
SimpleDateFormat df = new SimpleDateFormat("yyyy/MM/dd");
textView23.setText(""+dh.format(c.getTime()));// muestra
hora del sistema
textView22.setText(""+df.format(c.getTime()));// muestra fecha del
sistema

        // generación de reportes
if ((conexion==0)&&(flag_conect==0)){
    Toast.makeText(MainActivity.this, "DESCONECTADO" ,
Toast.LENGTH_SHORT).show();
    botonCONECT.setVisibility(View.VISIBLE);
    flag_conect=1;
    actualizar_hist("DESCONEC");
}

if ((conexion==1)&&(flag_conect==1)){
    Toast.makeText(MainActivity.this, "CONECTADO" ,
Toast.LENGTH_SHORT).show();
    botonCONECT.setVisibility(View.INVISIBLE);
    flag_conect=0;
    actualizar_hist("CONECT");
}

if((lpg>=alarmaLPG)&&(flag_alarma==0)){
    Toast.makeText(MainActivity.this, "ALARMA LPG" ,
Toast.LENGTH_SHORT).show();
    actualizar_hist("ALARMA LPG");
    alarmadisp.start();
    flag_alarma=1;
}

if(lpg<(alarmaLPG-1)){
    flag_alarma=0;
}

int i = TCPClient.getState();

if((conexion==0)|| (i==0)) {

```

```

        textView35.setText("DESCONECTADO");
        textView36.setText("DESCONECTADO");
        textView37.setText("DESCONECTADO");
        conexion=0;
new connectTask().execute("");
for(int x=0;x<1;x++){
try {
                Thread.sleep(500);
            } catch (Exception e) {
                e.printStackTrace();
            }
if (mTcpClient != null) {
                mTcpClient.sendMessage("E");
            }
        }
        tiempo();
    }.start();
}

public class connectTask extends AsyncTask<String,String,TCPCClient>
{//tarea en segundo plano para conexion
@Override
protected TCPCClient doInBackground(String... message) {

        mTcpClient = new TCPCClient(new TCPCClient.OnMessageReceived()
{
                @Override
//here the messageReceived method is implemented

public void messageReceived(String message) {
                publishProgress(message);
            }
        });
        mTcpClient.run();//ejecuta tcp cliente wifi

return null;
    }
    @Override
protected void onProgressUpdate(String... values) {
super.onProgressUpdate(values);

        myStringArray = null;
        arrayChar = values[0].toCharArray();
        st_reciv=values[0];

        textView35.setText("CONECTADO");
        textView36.setText("CONECTADO");
        textView37.setText("CONECTADO");
        conexion=1;

if(arrayChar[0]=='A'){
                myStringArray = values[0].split(" ");
//Toast.makeText(MainActivity.this, "ENTER "+ myStringArray[1],
Toast.LENGTH_SHORT).show();

alarmaLPG = Integer.parseInt(myStringArray[1]);
                Toast.makeText(MainActivity.this, "ALARMA EDITADA LPG:

```

```

"+alarmaLPG+" [ppm]", Toast.LENGTH_SHORT).show();
        textAlarma.setText(""+alarmaLPG);
        textAlarma2.setText(""+alarmaLPG);
    }

    if(arrayChar[0]=='R') {
        Toast.makeText(MainActivity.this, "RESET DISPOSITIVO ",
        Toast.LENGTH_SHORT).show();
        actualizar_hist("RESET DISP");
    }

    if(arrayChar[0]=='E') {
        myStringArray = values[0].split(" ");
        //Toast.makeText(MainActivity.this, "--"+ myStringArray[4]+"--
        "+myStringArray[5]+"--" , Toast.LENGTH_SHORT).show();
        lpg = Integer.parseInt(myStringArray[1]);
        co2 = Integer.parseInt(myStringArray[2]);
        smoke = Integer.parseInt(myStringArray[3]);
        estado_ev = Integer.parseInt(myStringArray[4]);
        estado_fan = Integer.parseInt(myStringArray[5]);

        TextView textView4 = (TextView)
        findViewById(R.id.textView4);
        TextView textView7 = (TextView)
        findViewById(R.id.textView7);
        TextView textView10 = (TextView)
        findViewById(R.id.textView10);
        LinearLayout led_ev1 = (LinearLayout)
        findViewById(R.id.led_ev1);
        LinearLayout led_fan1 = (LinearLayout)
        findViewById(R.id.led_fan1);
        FrameLayout led_ev2 = (FrameLayout)
        findViewById(R.id.led_ev2);
        FrameLayout led_fan2 = (FrameLayout)
        findViewById(R.id.led_fan2);

        textView4.setText(""+ lpg);
        textView7.setText(""+ co2);
        textView10.setText(""+ smoke);

        if (estado_ev == 1) {

            led_ev1.setBackgroundColor(getResources().getColor(R.color.green));
            led_ev2.setBackgroundColor(getResources().getColor(R.color.green));
        } else {

            led_ev1.setBackgroundColor(getResources().getColor(R.color.silver));
            led_ev2.setBackgroundColor(getResources().getColor(R.color.silver));
        }

        if (estado_fan == 1) {

            led_fan1.setBackgroundColor(getResources().getColor(R.color.green));
            led_fan2.setBackgroundColor(getResources().getColor(R.color.green));
        } else {

            led_fan1.setBackgroundColor(getResources().getColor(R.color.silver));

```

```

led_fan2.setBackgroundColor(getResources().getColor(R.color.silver));
    }
}
}
}

```

```

public void actualizar_hist(String evento){

    Calendar c = Calendar.getInstance();
    SimpleDateFormat dh = new SimpleDateFormat("HH:mm:ss");
    SimpleDateFormat df = new SimpleDateFormat("yyyy/MM/dd");

    for(int i=59;i>=5;i--){
        reporte_string[i]=reporte_string[i-3];
        reporte_string[i-1]=reporte_string[i-4];
        reporte_string[i-2]=reporte_string[i-5];
        i--;
        i--;
    }
    reporte_string[0]=evento;
    reporte_string[1]="" +df.format(c.getTime());
    reporte_string[2]="" +dh.format(c.getTime());

}

}

```

Código Arduino

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <MQ2.h>
#include "ESP8266.h"
ESP8266 wifi(Serial);

LiquidCrystal_I2C lcd(0x3f,16,2); // 27/3F/set the LCD address to 0x20
for a 16 chars and 2 line display

////////////////////////////////////
//          VARIABLES SALIDAS
////////////////////////////////////

const int EV = 2;
const int FAN = 3;
const int pin = A6;

int flag_AP=0,flag_JOIN=0;
unsigned int lpg, co, smoke,lpg_aux=-1,co_aux=-1,smoke_aux=-1,lpg_alarma=10,t=0;
unsigned long timer_1=0;
int state_EV=0,state_FAN=0;

byte wifi1[] = {
    0x01, 0x03, 0x07, 0x0F, 0x1F, 0x00, 0x00, 0x00
};
byte wifi2[] = {
    0x1C, 0x02, 0x19, 0x05, 0x15, 0x00, 0x00, 0x00

```

```

};

MQ2 mq2 (pin);

void setup() {

  pinMode (EV, OUTPUT);
  pinMode (FAN, OUTPUT);

  //inicialización LCD
  lcd.init();
  lcd.backlight();
  lcd.clear();

  lcd.setCursor(0, 0);lcd.print("A 10 Detector ");
  lcd.setCursor(0, 1);lcd.print(" Conectando... ");

  //inicializacion wifi
  Serial.begin(115200);

  if(conectar_AP()==1){
  lcd.setCursor(15, 0);lcd.write(1); //imprimir simbolo wifi
  }

  lcd.setCursor(0, 1);lcd.print(" Calentando... ");
  //inicializacion sensor
  mq2.begin();

  if((mq2.readLPG()<2) || (t < 500 )){
    delay(1000);
    t++;
  }

  lcd.setCursor(0, 1);lcd.print("

");

digitalWrite (EV, HIGH);
state_EV=1;
// digitalWrite (FAN, HIGH);
lcd.createChar(0, wifi1);
lcd.createChar(1, wifi2);

}

void loop() {
  //float* values= mq2.read(true); //set it false if you don't want to
  print the values in the Serial

  lpg = mq2.readLPG();
  co = mq2.readCO();
  smoke = mq2.readSmoke();

  if(lpg_aux!=lpg){
    lcd.setCursor(0, 1);lcd.print("

");
    lcd.setCursor(0, 1);lcd.print("L");lcd.setCursor(1, 1);lcd.print(lpg);
  }
  lpg_aux=lpg;

  if(co_aux!=co){
    lcd.setCursor(5, 1);lcd.print("

");
    lcd.setCursor(5, 1);lcd.print("C");lcd.setCursor(6, 1);lcd.print(co);
  }
}

```

```

}
co_aux=co;

if(smoke_aux!=smoke){
    lcd.setCursor(11, 1);lcd.print("      ");
    lcd.setCursor(11, 1);lcd.print("S");lcd.setCursor(12,
1);lcd.print(smoke);
}
smoke_aux=smoke;

if(lpg>=lpg_alarma){
    digitalWrite(EV, LOW);
    state_EV=0;
    digitalWrite(FAN, HIGH);
    state_FAN=1;
}

//lectura desde cliente via wifi
uint8_t buffer[128] = {0};
uint8_t mux_id;
uint32_t len = wifi.recv(&mux_id, buffer, sizeof(buffer), 100);
if (len > 0) {
    if (buffer[0]=='A'){
        lpg_alarma=(buffer[2]-48)*100+(buffer[3]-48)*10+(buffer[4]-48)*1;
        lcd.setCursor(2, 0);lcd.print(" ");
        lcd.setCursor(2, 0);lcd.print(lpg_alarma);
        wifi.send(mux_id, buffer, len);
    }

    if ((buffer[0]=='R') && (buffer[1]=='S') && (buffer[2]=='T')) {
        wifi.send(mux_id, buffer, len);
        digitalWrite(EV, HIGH);
        state_EV=1;
        digitalWrite(FAN, LOW);
        state_FAN=0;
    }

    if ((buffer[0]=='O') && (buffer[1]=='N') && (buffer[2]=='1')) {
        //buffer[0]='O';buffer[1]='N';buffer[2]=10;
        //wifi.send(mux_id, buffer, 3);
        digitalWrite(EV, HIGH);
        state_EV=1;
    }

    if ((buffer[0]=='O') && (buffer[1]=='F') && (buffer[2]=='1')) {
        //buffer[0]='O';buffer[1]='N';buffer[2]=10;
        //wifi.send(mux_id, buffer, 3);
        digitalWrite(EV, LOW);
        state_EV=0;
    }

    if ((buffer[0]=='O') && (buffer[1]=='N') && (buffer[2]=='2')) {
        //buffer[0]='O';buffer[1]='N';buffer[2]=10;
        //wifi.send(mux_id, buffer, 3);
        digitalWrite(FAN, HIGH);
        state_FAN=1;
    }

    if ((buffer[0]=='O') && (buffer[1]=='F') && (buffer[2]=='2')) {
        //buffer[0]='O';buffer[1]='N';buffer[2]=10;
        //wifi.send(mux_id, buffer, 3);
        digitalWrite(FAN, LOW);
        state_FAN=0;
    }
}
}

```

```

    }

    if(millis()>=timer_1+1000){
        buffer[0]='E';
        buffer[1]=' ';
        if(lpg>=0){
            buffer[6]=48+(lpg/1)%10;
            buffer[5]=48+(lpg/10)%10;
            buffer[4]=48+(lpg/100)%10;
            buffer[3]=48+(lpg/1000)%10;
            buffer[2]=48+(lpg/10000)%10;
        }
        buffer[7]=' ';
        if(co>=0){
            buffer[12]=48+(co/1)%10;
            buffer[11]=48+(co/10)%10;
            buffer[10]=48+(co/100)%10;
            buffer[9]=48+(co/1000)%10;
            buffer[8]=48+(co/10000)%10;
        }
        buffer[13]=' ';
        if(smoke>=0){
            buffer[18]=48+(smoke/1)%10;
            buffer[17]=48+(smoke/10)%10;
            buffer[16]=48+(smoke/100)%10;
            buffer[15]=48+(smoke/1000)%10;
            buffer[14]=48+(smoke/10000)%10;
        }
        buffer[19]=' ';
        if(state_EV==1)
            buffer[20]=48+1;
        else
            buffer[20]=48+0;
        buffer[21]=' ';
        if(state_FAN==1)
            buffer[22]=48+1;
        else
            buffer[22]=48+0;

        buffer[23]=' ';
        buffer[24]=13;
        wifi.send(mux_id, buffer, 25);
        timer_1=millis();
    }

}

int conectar_AP(){
    if(wifi.setSoftAPParam("network_gas", "gas12345", 11, 3)){
        flag_AP++;
        delay(500);
    }

    if (wifi.setOprToSoftAP()) { // establece red wifi
        flag_AP++;
        delay(500);
    }

    if (wifi.enableMUX()) { // conexion con multiples clientes max 4

```

```
    flag_AP++;
    delay(500);
}

if (wifi.startTCPServer(8080)) { //configuracion como servidor
flag_AP++;
delay(500);
}

if(flag_AP==4){
return 1;
}
else
return 0;
}
```

3.1.6 Implementación total del sistema

Por último, se realizaron las adaptaciones necesarias para que el circuito se pueda instalar correctamente, como se visualiza en la Figura 3.17.

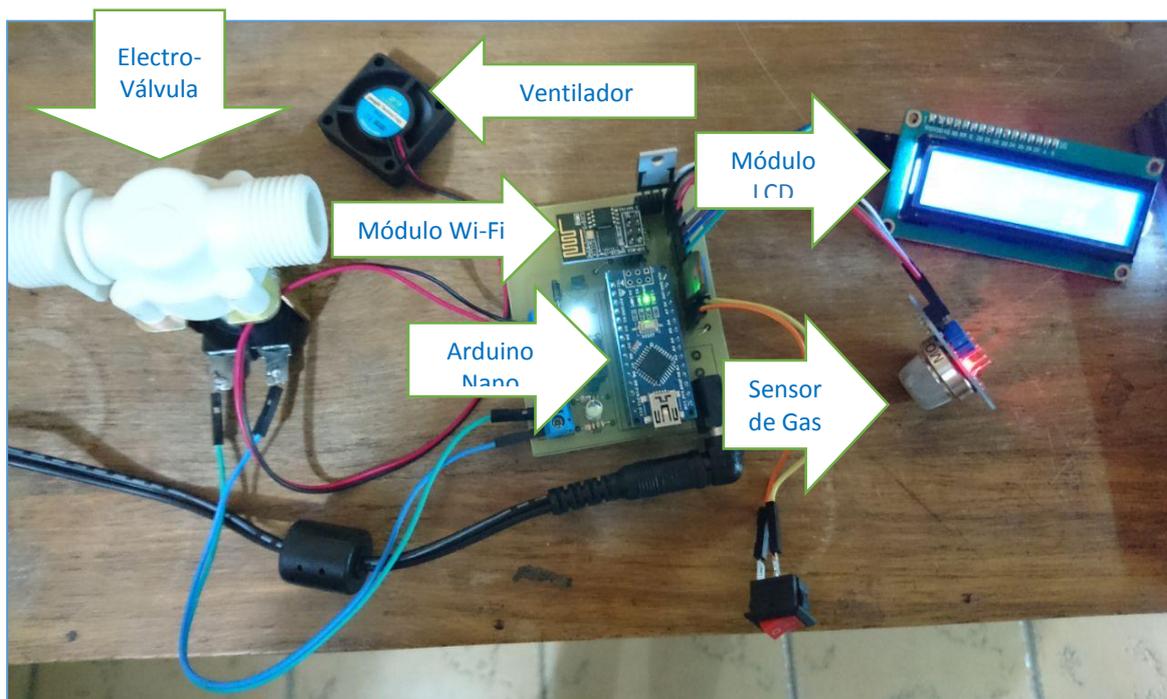


Figura 3.17 Circuito Total del prototipo

Elaborado por: Autor.

En la Figura 3.18 se puede apreciar el funcionamiento del módulo LCD y la aplicación configurada en el teléfono inteligente. En el LCD se muestra la información del sensor, en cambio en aplicación móvil la configuración de sensibilidad del sensor y las alarmas que se han activado.



Figura 3.18 Visualización de Información
Elaborado por: Autor.

El diseño del circuito está elaborado para implementar en una maqueta, diseñada para evaluar de manera óptima el funcionamiento del circuito, donde se aprecia una casa modelo, que consta de dormitorios, sala, comedor y cocina, como se observa en la figura a continuación.

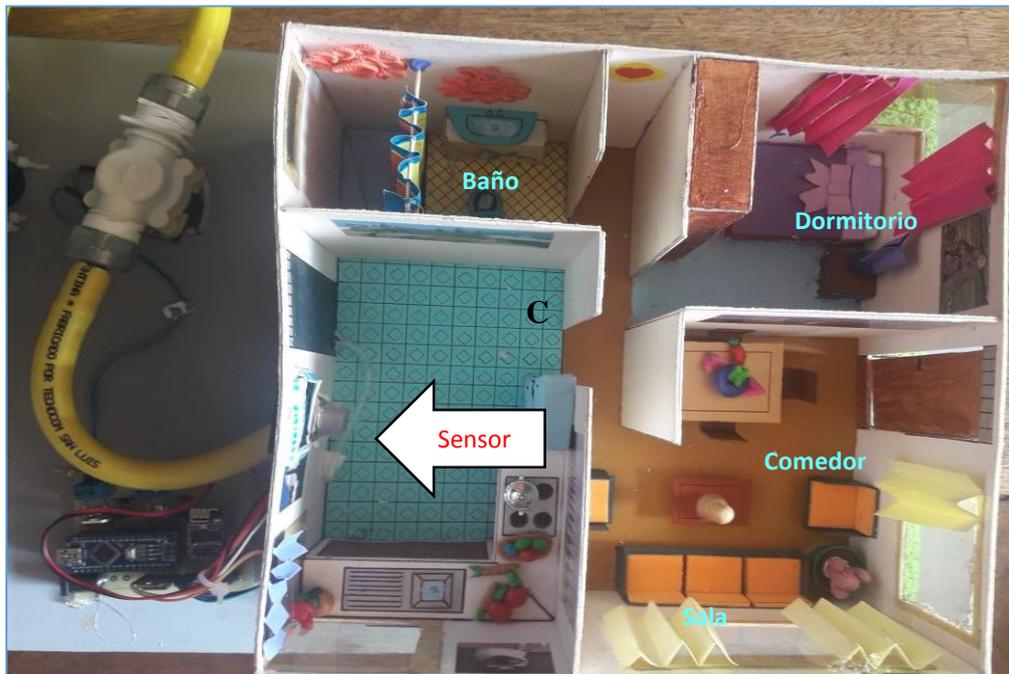


Figura 3.19 Instalación de Sensor en maqueta diseñada
Elaborado por: Autor.

El sensor de gas MQ-2 se instala en la cocina, en un lugar determinado para identificar de manera correcta la fuga, y actuar de manera inmediata para evitar daños físicos como materiales.



Figura 3.20 Instalación de sensor MQ-2

Elaborado por: Autor.

El esquema del prototipo se encuentra constituido por un módulo Arduino Nano, una electroválvula, el sensor de gas MQ-2, un módulo Wi-Fi, acceso a Internet, como se ilustra en la Figura 3.21.

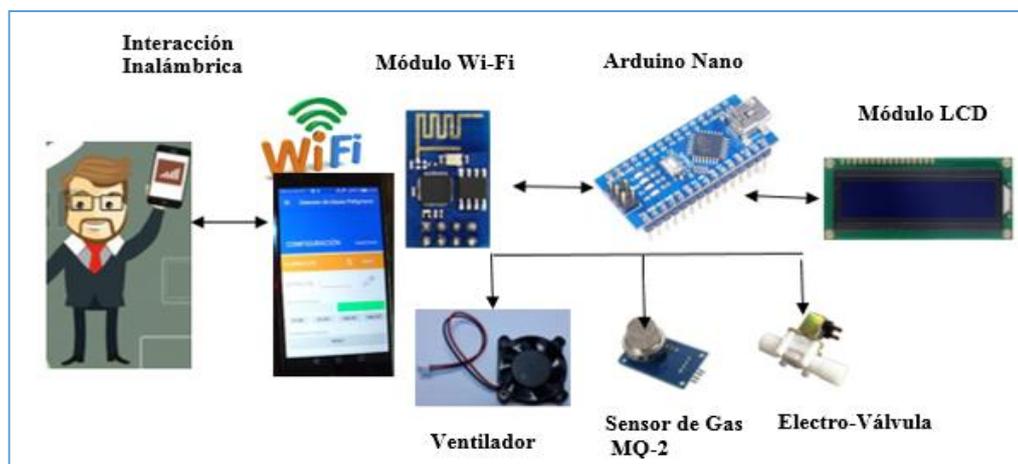


Figura 3.21 Esquema del prototipo

Elaborado por: Autor.

El usuario que tiene instalada la aplicación para Android va a configurar los parámetros de sensibilidad para el sensor de gas. Mediante la señal WI-FI va a obtener información cuando la alarma se active, en este caso cuando el sensor MQ-2 detecte el gas.

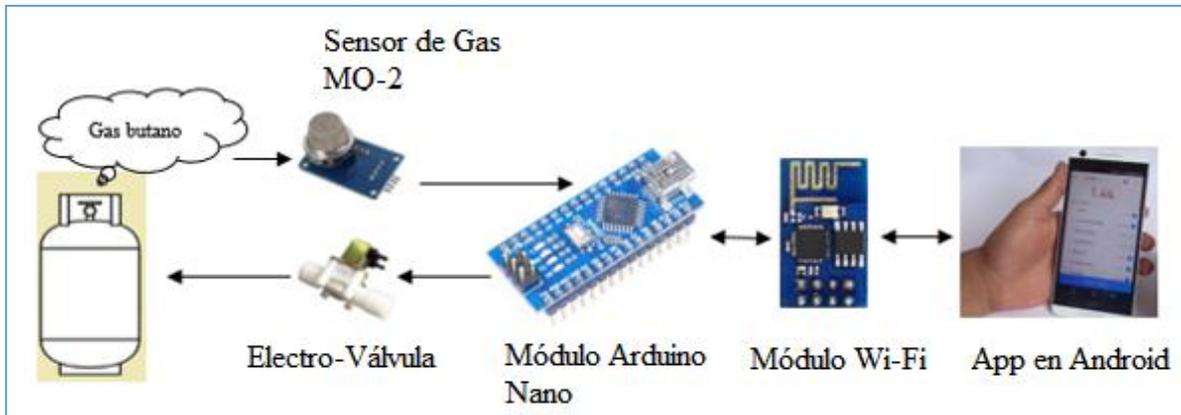


Figura 3.22 Esquema de la alarma

Elaborado por: Autor.

La figura anterior, muestra el funcionamiento del detector de gas, utiliza el módulo Arduino nano para la parte del control así, el sensor, la electroválvula, el módulo WI-FI, el ventilador, se conectan a este y realizan la detección de la fuga de gas y también el aviso al usuario mediante la aplicación en Android.

Finalmente se muestra la vista externa del prototipo en la Figura 3.23.



Figura 3.23 Vista externa del prototipo

Elaborado por: Autor

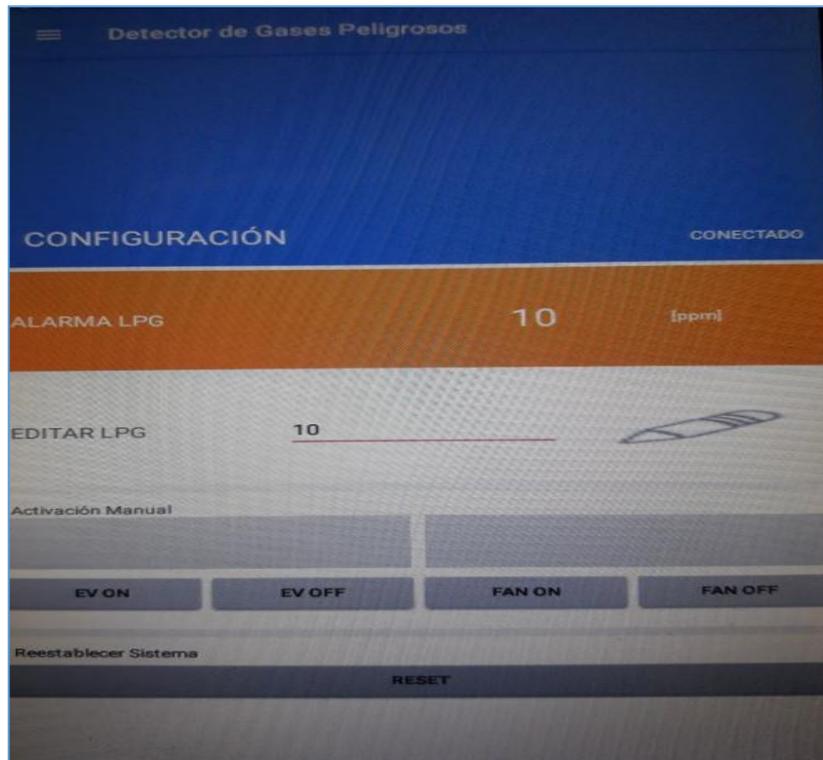


Figura 3.24 App Sistema detector
Elaborado por: Autor

3.2 Pruebas de Funcionamiento

La construcción y diseño del prototipo siguió una estructurada y sistemática secuencia, lo que ayudó a obtener un producto eficaz, que junto con las respectivas pruebas de funcionamiento permitió el perfeccionamiento del mismo.

Concretamente, las pruebas realizadas fueron las siguientes:

3.2.1 Chequeo inicial

Tabla 3.1 Pruebas chequeo inicial

Nº	FUENTE12 VDC	SWITCH ENCENDIDO	LCD	OBSERVACIONES
1	✓	✓	✓	Registra 5 VDC (Flujo en pines)
2	x	✓	x	Comprobar que switch enciende Revisar adaptador Revisar cables del adaptador
3	✓	x	x	Revisar interruptor Revisar cables LCD
4	✓	✓	✓	

Elaborado por: Autor

Las pruebas de la Tabla 3.1 se realizaron de la siguiente forma:

1era.- Si se tiene conectado una fuente a 12 VDC, el switch encendido, el display LCD se va a encender con un voltaje de 5 VDC.

2da.- Si no está conectada la fuente de 12 VDC, el switch encendido, el display LCD no se va a encender, en ese caso hay revisar que el switch encienda, revisar adaptador, y por último revisar cables del adaptador.

3era.- Está conectada la fuente de 12 VDC, el switch no está encendido, el display LCD no se va a encender ahí, en este caso se tiene que revisar interruptor, y revisar cables LCD.

4ta.- Está conectada la fuente de 12 VDC, el switch está encendido, el LCD Display se va a encender.

3.2.2 Pruebas de Hardware

Medición de gases

Tabla 3.2 Pruebas medición de gases

OPCIONES		OBSERVACIONES
Al aire libre	0 ppm	No registra lectura
Humo CO ₂	1° vez	No registra lectura
	2° vez	No estaba encendida la electroválvula
	3° vez	Lectura de Humo normal de acuerdo a los ppm configuradas en la app

Elaborado por: Autor.

En la Tabla 3.2 se realizaron los siguientes tests:

1era.- Al aire libre se configura en la aplicación en el dispositivo móvil con el mínimo valor que es 1 no registra lectura en el display.

2da.- Se realizó con CO₂ se configura en la aplicación en el dispositivo móvil con el mínimo valor que es 1 no registra lectura en el display. Y no se enciende la electroválvula.

3era.- Se realizó la tercera prueba se configura en la aplicación del dispositivo móvil con valores mínimos y máximos y no se registra lectura en el display y no se enciende.

Activar/Desactivar**Tabla 3.3 Pruebas activar/desactivar**

COMPONENTES		OBSERVACIONES
Electroválvula	Error	Cambiar de botones en la app
Ventilador	Error	

Elaborado por: Autor

En la Tabla 3.3 se realizó las siguientes pruebas:

1era.- En la electroválvula, no se activa en la aplicación móvil es porque está mal digitada en la aplicación de la Tableta móvil, esto se configura nuevamente digitando el botón de electroválvula.

2da.- En el ventilador, no se activa en la aplicación móvil es porque está mal digitada en la aplicación de la Tableta móvil, esto se configura nuevamente digitando el botón de ventilador.

Diario/Registro**Tabla 3.4 Pruebas diario/registro**

	OBSERVACIONES
No hay registro de humo en el histórico de eventos	Agregar (Time lapse) Fecha y hora

Elaborado por: Autor

La app trabaja en segundo plano.

En la Tabla 3.4 se realizó la siguiente prueba:

1era.- Hay un historial de los diferentes monitoreos donde dan hora y fecha y esto se actualiza deslizando la aplicación en segundo plano de arriba hacia abajo y ahí se actualiza el historial. Tiene que estar configurado el dispositivo móvil en horas y fechas actuales.

Android 7.0 (No WI-FI) / APP Android Studio**Tabla 3.5 Pruebas android**

Observaciones		
No se enlaza con dispositivo	1	Tiene clave de acceso WI-FI gas12345
	2	No tiene seguridad

Elaborado por: Autor.

En la Tabla 3.5 se realizaron las siguientes pruebas:

1era.- No se enlaza con el dispositivo móvil, revisar el acceso WI-FI con la siguiente clave gas12345 todo unido y minúsculas.

2da.- No se enlaza el módulo WI-FI, con el dispositivo móvil, revisar que este encendido en Switch, y revisar la fuente de alimentación para el correcto funcionamiento.

Electroválvula**Tabla 3.6 Pruebas electroválvula**

Observaciones	
No se activa	Alimentación mal conectada Tip 122 averiado (sustituirlo) Válvula mal conectada en el sentido del flujo del GLP o hay fugas en las uniones de la electroválvula con la manguera de gas

Elaborado por: Autor.

En la Tabla 3.6 se realizaron las siguientes pruebas

1era.- Si la electroválvula no se activa, revisar alimentación, revisar configuración en la aplicación móvil.

2da.- Si la electroválvula no se activa, se revisará el transistor TIP 122 y si está quemado hay que sustituir.

3era.- Si la electroválvula no se activa, se revisara si no está mal conectada en el sentido del flujo GLP, o también revisar posibles fugas en las uniones de la electroválvula con las mangueras de gas.

3.2.3 Pruebas de Software

Android 6.0, Marshmallow, API 23, por razones de seguridad de Android

En esta versión se presenta un nuevo modelo de permisos en el cual los usuarios ahora pueden administrar directamente los permisos de la app durante el tiempo de ejecución. En esta versión se presentan nuevas optimizaciones de ahorro de energía para apps y dispositivos inactivos.

Además de nuevas funciones y capacidades, en Android 6.0 (nivel de API 23) se incluyen diversos cambios en el sistema y modificaciones en los comportamientos de la API.

3.2.4 Pruebas de GLP

Tabla 3.7 Pruebas GLP

		1° vez Configurado	Lectura	Display
Ventilador	Electroválvula	2 ppm	120 ppm	✓
✓	✓	10 ppm	230 ppm	✓
		30 ppm	260 ppm	✓
		1,32 ppm	No decimales solo enteros	✓
Observaciones	1-5 ppm ideal para condiciones de seguridad			

Elaborado por: Autor.

En la Tabla 3.7 se realizaron las siguientes pruebas:

1era.- Se encenderá el ventilador y la electroválvula una vez configurado en la aplicación móvil del dispositivo configurando a 2ppm le detectarán, y da una lectura en el display de 120ppm.

2da.- Se encenderá el ventilador y la electroválvula una vez configurado en la aplicación móvil del dispositivo configurando a 10ppm le detectarán, y da una lectura en el display 230ppm.

3era.- Se encenderá el ventilador y la electroválvula una vez configurado en la aplicación móvil del dispositivo configurando a 30ppm le detectarán, y da una lectura en el display de 260ppm.

4ta.- No se encenderá el ventilador y la electroválvula una vez configurado en la aplicación móvil del dispositivo configurando a 1.32ppm no le detectarán, y no dará una lectura en el display porque son solo valores enteros y no decimales.

5ta.- Para seguridad de usuario la configuración en la aplicación del dispositivo móvil debe ser de 1-5ppm, ya que son condiciones ideales para la seguridad.

3.3 Análisis de Resultados

Como se puede observar, en cada una de las pruebas realizadas, no solo se encuentra la falla identificada, sino que además se presenta la solución a implementar, la cual una vez realizada permitió arreglar o corregir los inconvenientes obtenidos en un inicio con el prototipo construido.

Respecto a la pruebas como tal, es importante indicar que en su mayoría fueron desajustes de *hardware* o *software*, que al ejecutar los respectivos ajustes, se logró obtener un sistema estable y eficiente de prevención de desastres por fuga de gas en un hogar. Su unidad de medida del paso del gas GLP son partículas por minuto (ppm), el número de partículas por minuto eso le asigna el usuario ya que solo puede ingresar valores enteros que son 1 hasta el 99.

En las pruebas de velocidad y respuesta es de 2 mili segundos (ms) y es al instante el tiempo de respuesta, al momento de que ingresa a la electroválvula el GLP detecta y se cierra y la respuesta es inmediata.

La velocidad de 9600 es la velocidad de transmisión del módulo Wi-Fi y se mide en baudios. Con lo referente al historial del monitoreo se almacena dentro de la aplicación de la Tableta y se actualiza automáticamente y para poder ver las actualizaciones se desliza la ventana hacia abajo que está función actúa en segundo plano.

Las pruebas se realizó con un tanque de gas conectado a través de la válvula de gas con una manguera conectada hacia la electroválvula cuando se deja fugas el GLP se cierra la electroválvula y no permite el paso del GLP. Además en estas pruebas se utilizó un tanque de gas de uso doméstico de 15 kilos.

CONCLUSIONES

- Al realizar la simulación en proteus, se conoció que existen librerías que no se encuentran en el programa y es necesario descargarlas para desarrollar la simulación de manera apropiada.
- Con el software de programación Android Studio se tiene una visión más enfocada a la creación de aplicaciones para dispositivos móviles, por su entorno de desarrollo, cuanto más complicada sea la aplicación, mayor será el esfuerzo de programación.
- La aplicación creada en la plataforma de Android Studio permite recibir alertas de fugas de gas, para tomar acción inmediata.
- El prototipo se desarrolla en la maqueta creada de acuerdo al ambiente necesario para realizar las oportunas pruebas.
- Al tener conocimiento previo a cerca de Arduino Nano por realizar distintos proyectos o prácticas, se pudo elegir este Arduino Nano debido a su tamaño pequeño y bajo costo, pues se verificó que tiene las mismas funcionalidades que el Arduino Uno.
- El proyecto se probó en el diseño de la maqueta, realizando pruebas de fugas de gas y verificando que el sistema de ventilación funcione correctamente y el gas no se quede en el ambiente.
- Los componentes usados determinan la viabilidad del proyecto, ya que son fáciles de conseguir y no son costosos.
- A pesar de que el prototipo fue realizado específicamente para una cocina, puede adaptarse para detectar fugas en otro tipo de espacios, como lavanderías.

- El prototipo construido cumple con el objetivo inicialmente propuesto, ya que se ha desarrollado un sistema automático para prevenir explosiones de gas licuado de petróleo en un domicilio.
- La aplicación APP INVENTOR es un servicio basado en la nube además todas las tareas se realiza en un navegador conectado a internet, ya que Android Studio no solo se descarga la aplicación y se puede realizar la programación.

RECOMENDACIONES

- Cuando se realice la programación de la aplicación en Android Studio se debe verificar la versión de esta, porque de no ser compatible el programa no se va a ejecutar en un teléfono que no sea compatible y presentará errores.
- Verificar los niveles de voltaje y corriente a los que trabaja cada dispositivo electrónico, para evitar daño por sobre voltaje o disminuir su vida útil.
- Al conectar el módulo Wi-Fi con el punto de acceso a Internet, se presentó problemas de intermitencia cuando se conectaba por debajo del punto de acceso, eso va a mejorar adaptando un módulo que tenga mayor alcance o mayor potencia.
- La electroválvula que se usó puede ser adaptada directamente a la válvula del tanque de gas, para obtener una mayor durabilidad se puede construir de mercurio, según los estándares de tubería tradicional que se usan en residencias.
- La aplicación elaborada en Android Studio soporta la versión de Android 6.0, no mayores, se podrá mejorar cambiando la configuración de la aplicación, para que soporte versiones mayores, pues van apareciendo cada vez más en el mercado de la telefonía celular.
- Es importante incentivar a la investigación, construcción y desarrollo de nuevos proyectos que permitan detectar o crear un sistema de alarmas ante cualquier exposición tóxica, además de motivar y despertar un mayor interés en la creación de aplicaciones que sea compatible con teléfonos celulares, porque es una tecnología que se encuentra en apogeo, beneficiando a la sociedad ecuatoriana.
- Como se menciona en la recomendación anterior este documento puede servir de base para la creación de dispositivos detectores para ser implementados en otro tipo de lugares tipo de gases como bodegas o para uso industrial, y podría ser adaptado además para otro tipo de gases.

REFERENCIAS BIBLIOGRÁFICAS

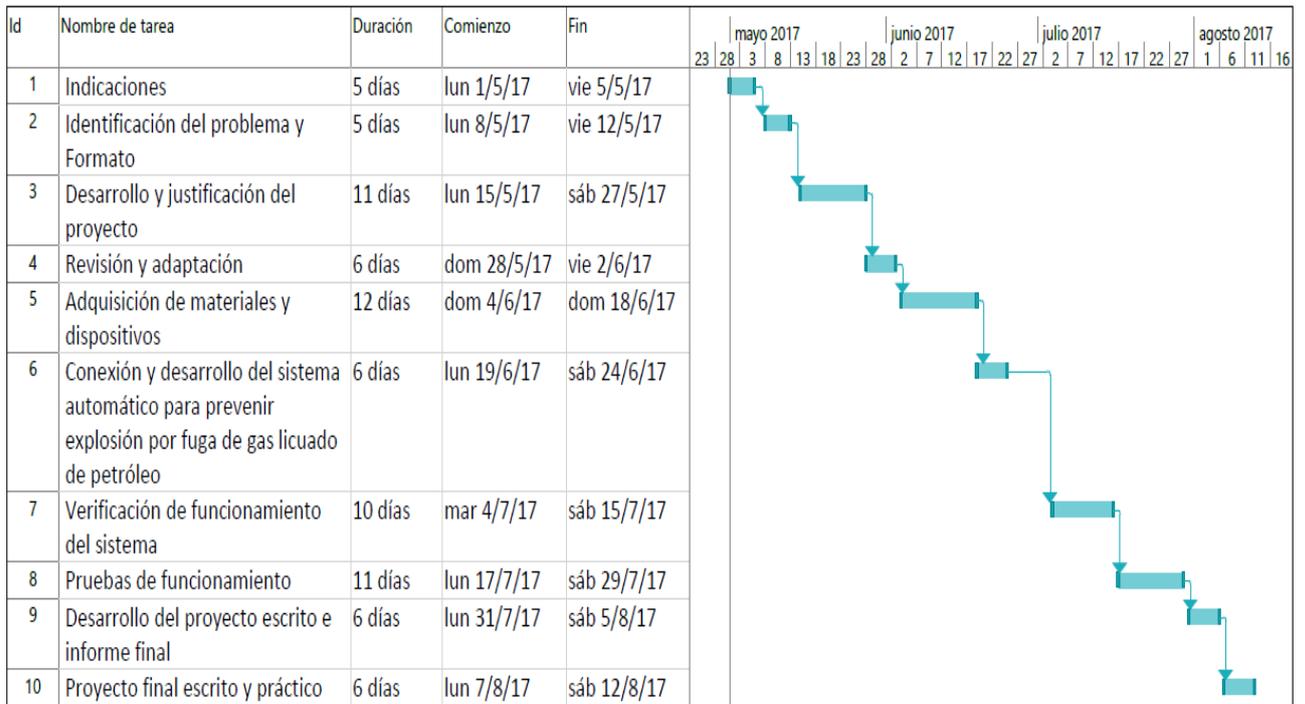
- 330ohms. (2016). *Cómo funcionan los sensores de gas?* Recuperado el 21 de Marzo de 2018, de <https://www.330ohms.com/blogs/blog/sensores-de-gas-como-funcionan>
- AG Electrónica. (2016). *Módulo inalámbrico ESP8266*. Recuperado el 19 de Marzo de 2018, de <http://www.agspecinfo.com/pdfs/I/IM151228.PDF>
- AG Electrónica. (2018). *Módulo WIFI ESP8266*. Recuperado el 19 de Marzo de 2018, de <http://www.agspecinfo.com/pdfs/E/ESP-12F.PDF>
- Altec. (2018). *¿Qué son las Electroválvulas?* Recuperado el 21 de Marzo de 2018, de <http://www.altecdust.com/soporte-tecnico/que-son-las-electrovalvulas>
- Andes. (2015). *Explosiones de tanques de gas en enero dejaron cuatro muertos en Ecuador*. Recuperado el 23 de Marzo de 2018, de <https://www.andes.info.ec/es/noticias/actualidad/1/35456/explosiones-tanques-gas-enero-dejaron-cuatro-muertos-ecuador>
- Android Studio FAQs. (2016). *Android Studio: ventajas, desventajas y principales características*. Recuperado el 23 de Marzo de 2018, de <https://androidstudiofaqs.com/conceptos/ventajas-desventajas-android-studio>
- Android Studio. (s.f.). *Android*. Recuperado el 23 de Marzo de 2018, de <https://developer.android.com/studio/index.html>
- Aprendiendo Arduino. (2018). *Wifi en Arduino*. Recuperado el 19 de Marzo de 2018, de <https://aprendiendoarduino.wordpress.com/tag/genuino-mkr1000/>
- Ardudino. (2018). *Arduino nano*. Recuperado el 19 de Marzo de 2018, de <http://arduino.cl/arduino-nano/>
- Asconumatics. (s.f.). *Tecnología electroválvulas y válvulas*. Recuperado el 21 de Marzo de 2018, de https://www.asconumatics.eu/images/site/upload/_es/pdf1/00005es.pdf
- BCE. (2018). *Cifras del Sector Petrolero - Serie Cifras Petroleras 2007-2018*. Recuperado el 19 de Marzo de 2018, de <https://www.bce.fin.ec/index.php/hidrocarburos>
- Castellanos, G. (17 de Marzo de 2015). *Cinco datos básicos para evitar la explosión del cilindro de gas. El Comercio*. Obtenido de <http://www.elcomercio.com/actualidad/gas-fugas-explosiones-recomendaciones-bomberos.html>
- Dayre. (2013). *¿Qué es el GLP?* Recuperado el 19 de Marzo de 2018, de <http://www.talleredayre.es/que-es-glp-ventajas-propiedades-funcionamiento.html>

- Distritec. (2013). *¿Qué son las electroválvulas?* Recuperado el 21 de Marzo de 2018, de <https://www.distritec.com.ar/novedad-detalle.php?titulo=%BFQU%C9%20SON%20LAS%20ELECTROV%C1LVULAS?>
- Dubois, R. (2006). *Introducción a la refinación del petróleo*. Argentina: EUDEBA.
- El Nuevo Diario. (2015). *La manipulación correcta del cilindro de gas*. Recuperado el 19 de Marzo de 2018, de <https://www.elnuevodiario.com.ni/infografia/3208/>
- Electronilab. (2018). *ESP8266 – ESP-01 – Módulo Transceptor WiFi Serial*. Recuperado el 19 de Marzo de 2018, de <https://electronilab.co/tienda/esp8266-modulo-wifi-serial-transceptor/>
- ENI Ecuador. (2018). *El Regulador para Cilindros de uso doméstico*. Recuperado el 23 de Marzo de 2018, de <http://www.eniecuador.ec/index.php/tecnologia-e-innovacion/reguladores>
- INEN. (2010). *Instalaciones de gases combustibles para uso residencial, comercial e industrial NTE INEN 2 260:2010*. Ecuador: La Institución.
- Líderes. (s.f.). *El subsidio opaca el negocio del gas*. Recuperado el 19 de Marzo de 2018, de <http://www.revistalideres.ec/lideres/subsidio-opaca-negocio-gas.html>
- LPG. (2015). *Origen del GLP*. Obtenido de <https://www.wlpga.org/wp-content/uploads/2015/10/WLPGA-EE-PDF-ES.V1.pdf>
- Osinergmin. (2018). *Diferencias Físico - Químicas del Gas Natural el GLP*. Recuperado el 19 de Marzo de 2018, de <http://srvgart07.osinerg.gob.pe/webdgn/contenido/diferencias-fisico-quimicas-gn-glp.html>
- Parra, E. (2003). *Petróleo y gas natural*. España: Akal.
- Pascual, A. (2007). *Estándares de Tecnologías Inalámbricas*. Recuperado el 19 de Marzo de 2018, de http://www.itrainonline.org/itrainonline/mmtk/wireless_es/files/02_es_estandares-inalambricos_guia_v02.pdf
- Pastor, D. (2017). *¿Qué es y para que sirve el SSID de la conexión Wi-Fi?* Recuperado el 1 de Junio de 2018, de <https://easysmartech.com/ssid-wifi/>
- Propanogas. (s.f.). *¿Qué son los gases licuados del petróleo (propano y butano)?* Recuperado el 23 de Marzo de 2018, de <https://propanogas.com/faq/gas-licuado>

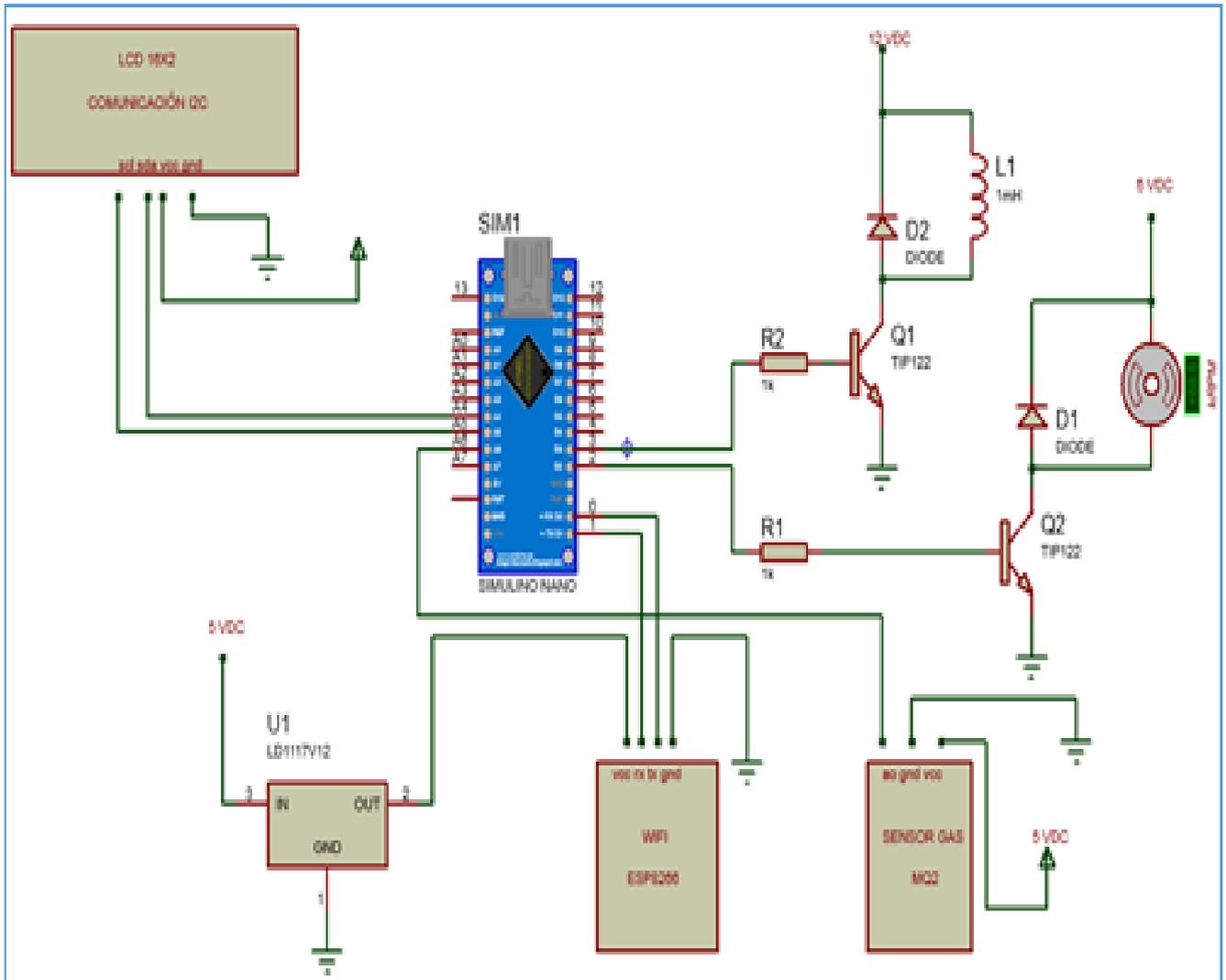
- Realpe, A. (2017). *Desarrollo de un prototipo de estacionamiento automatizado para bicicletas*. Recuperado el 19 de Marzo de 2018, de <http://repositorio.uisrael.edu.ec/bitstream/47000/1416/1/UISRAEL-EC-ELDT-378.242-2017-036.pdf>
- Recope. (2017). *Gas Licuado de Petróleo (GLP)*. Recuperado el 19 de Marzo de 2018, de <https://www.recope.go.cr/gas-licuado-de-petroleo-glp/>
- Shell. (s.f.). *Guía del LPG*. Recuperado el 19 de Marzo de 2018, de <https://www.shell.com.ar/empresas/gas-lpg-en-los-negocios/guia-del-lpg.html>
- Torres, J. (2008). *Sensores de gas*. Recuperado el 21 de Marzo de 2018, de <https://nanotecnologia.fundaciontelefonica.com/2008/03/19/sensores-de-gas/>
- ValorTop. (2017). *¿Qué es WiFi? ¿Qué significa y para qué sirve?* Recuperado el 1 de Junio de 2018, de <http://www.valortop.com/blog/que-es-wifi-que-significa-y-para-que-sirve>
- Venegas, D., Ayabaca, C., Celi, S., & Rocha, J. (2018). El riesgo en el almacenamiento de GLP en el Ecuador. *INNOVA*, 3(1), págs. 19-29. Recuperado el 21 de Marzo de 2018, de <http://www.journaluidegye.com/magazine/index.php/innova/article/view/331/548>.
- Villegas, L. (2012). *Espectro ensanchado por secuencia directa (DSSS)*. Recuperado el 1 de Mayo de 2018, de <http://investigaciondsss.blogspot.com/>
- Vincze, J. (2014). *Diseño de una red de telecomunicaciones para soportar los servicios de voz, datos y videos entre el hospital Chiquinquirá y el hospital Central*. Recuperado el 19 de Marzo de 2018, de <http://200.35.84.131/portal/bases/marc/texto/2701-14-07089.pdf>

ANEXOS

Anexo 1 – Cronograma



Anexo 2 – Diagrama electrónico completo

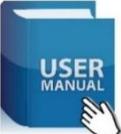


Anexo 3 – Manual de Usuario



MANUAL DE USUARIO

DETECTOR DE FUGAS DE GLP

Precauciones de seguridad	
	El detector es un prototipo electrónico por lo cual es muy importante que no entre en contacto con ningún tipo de sustancia líquida.
 PROHIBIDO GOLPEAR	No golpee el prototipo.
	No exponer el prototipo al fuego.
NO ABRIR	No abra el localizador y tampoco manipule los elementos que se encuentran en su interior.
	Use el prototipo como indica el manual.

NOTA: Verificar que el prototipo se encuentre correctamente asegurado cerca del cilindro de GLP sin que éste interfiera en su correcto uso.

Estimado usuario:

- *Gracias por preferir GAS PROTECT basado en Tecnología Arduino, un sistema seguro, confiable y sencillo de utilizar.*
- *Este manual muestra cómo operar el dispositivo sin problemas. Asegúrese de leer este manual antes de usar el equipo.*
- *Por favor, lea estas instrucciones cuidadosamente antes de utilizar este dispositivo.*
- *Por favor, mantenga este manual para futuras consultas.*
- *El fabricante no asume ninguna responsabilidad por cualquier error u omisión en este documento.*

Descripción de los componentes externos del dispositivo

La figura siguiente muestra la parte exterior del prototipo, se puede observar una caja compacta, el interruptor de encendido del prototipo, con sus respectivas salidas para los componentes principales como el sensor y el ventilador.



Figura 1. Vista externa del prototipo

Requerimientos mínimos

- El teléfono inteligente ó Tableta debe contar con conectividad bluetooth y tener la versión Android 6.0.1 como mínimo para su correcto funcionamiento.
- El terminal celular también debe contar con conectividad WI-FI.

Instalación de aplicación en el dispositivo móvil

El CD adjunto contiene la aplicación a utilizar.

Se conecta el teléfono inteligente encendido a la computadora por medio de un cable USB, éste mostrará un mensaje "Permitir acceso a los datos del dispositivo" en el cual se debe escoger la opción "PERMITIR".

El CD se debe insertar en la unidad de DVD-RW ó CD-RW en la computadora, luego se debe ir a "INICIO" --> "EQUIPO" --> "ABRIR" --> "UNIDAD D:" --> "ABRIR", aquí se encuentra el archivo "Gas Protect", se da clic derecho y se escoge la opción "Copiar", luego se debe ir Al explorador de Windows y buscamos "ESTE EQUIPO" --> "ABRIR" -->

“PHONE” --> “ABRIR” --> “DOWN LOAD” y se da clic derecho dentro de esta carpeta y se escoge la opción pegar.

La aplicación ha sido copiada en el teléfono inteligente en la carpeta “DOWNLOAD”.

Una vez descargada la aplicación en el dispositivo móvil, se procede a la instalación de

esta, para ello se selecciona el archivo  y mostrará la siguiente pantalla indicando el avance de instalación.

Una vez finalizada la instalación se muestra la siguiente imagen, donde se genera el ícono, como se ve en la figura, luego de esto se presiona en abrir para inicializar la aplicación.



Figura 2. Aplicación Instalada

La figura 3 muestra la aplicación ya instalada en el teléfono inteligente, luce con un aspecto sencillo y amigable con los botones necesarios para la su interacción entre el módulo y el teléfono inteligente, utilizando tecnología Wi-Fi presente en ambos dispositivos.

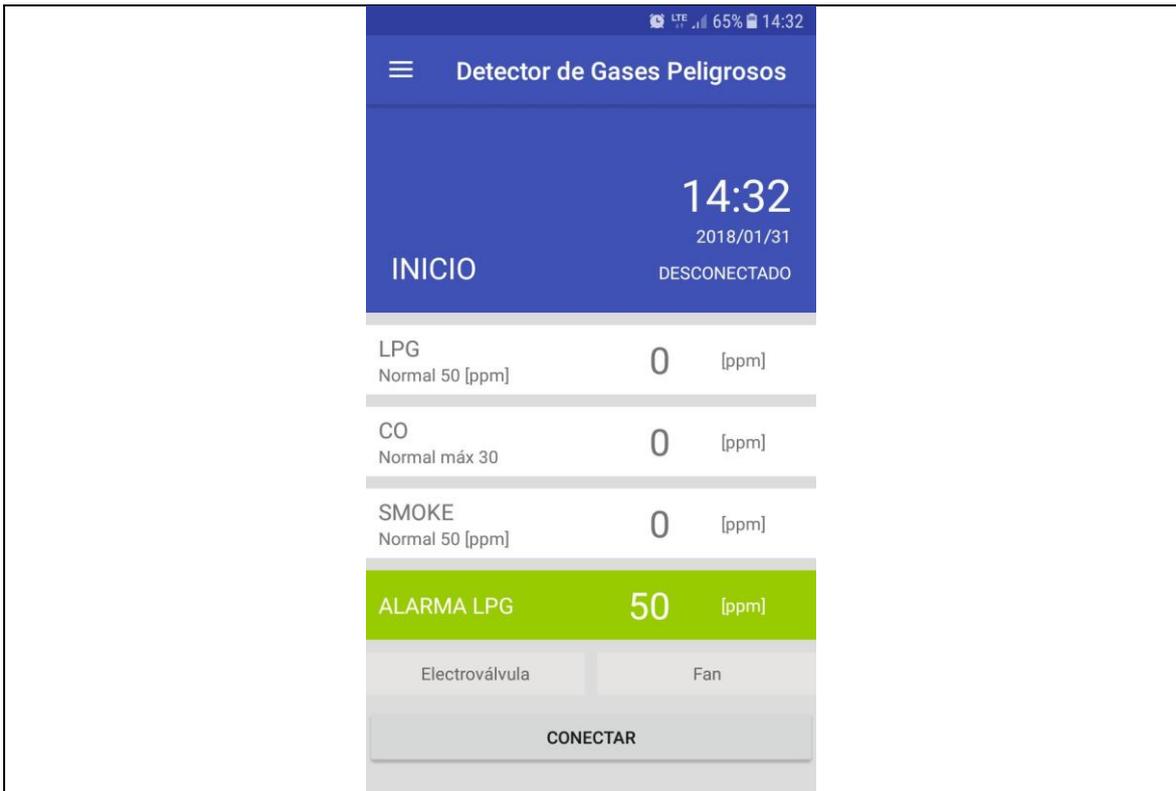


Figura 4. Aplicación instalada en el teléfono inteligente

Montaje del detector

Nuestro personal calificado se encargará de instalar este dispositivo, de acuerdo con la distribución del espacio y localización del cilindro de gas en su domicilio.

Guía de uso

- El prototipo debe ser instalado por el personal calificado para mantener la cadena de calidad en nuestro producto.
- El prototipo debe ser sujeto correctamente cerca del cilindro de GLP, evitando que tenga contacto con los víveres.
- Una vez ya sujeto el prototipo se procede a colocar el ventilador y el sensor elementos esenciales para el control del entorno, donde se encuentra instalado.
- El teléfono inteligente donde está instalada la aplicación de detección del prototipo debe estar cargado con un mínimo del 70% como se indica en la figura.



Figura 5. Nivel de batería del dispositivo móvil

- Encender la opción “Wi-Fi” en el teléfono inteligente.

- Instalar la aplicación creada para la conectividad con el prototipo de acuerdo con las indicaciones mostradas en la sección de este manual denominada: **Instalación de aplicación en el dispositivo móvil**, donde finalmente obtendremos un ícono  en el Menú de aplicaciones del teléfono.
- Verificar la conectividad entre el prototipo y el teléfono inteligente presionando el botón de conectar en la aplicación como se muestra en la figura 8, inmediatamente el sistema mostrará lo niveles de gas presentes en el ambiente donde se encuentre instalado el sensor.



Figura 6. Pulsador desde el prototipo localizador y notificación en el teléfono móvil.

Solución de problemas más comunes

- **El prototipo no enciende**
 - Se debe verificar que el prototipo esté conectado con la fuente de alimentación para ello utilice la fuente apropiada que viene dentro de los componentes del producto.
- **El Wi-Fi del teléfono inteligente no se empareja con el Wi-Fi del prototipo**
 - El teléfono está descargado, recargarlo utilizando el cargador.
 - Verificar que la opción **Wi-Fi** del teléfono inteligente esté encendida.
 - Verificar que la distancia entre el prototipo y el teléfono inteligente no sea mayor de 20 metros.
- **La aplicación no muestra los niveles de gas del prototipo**

- Verificar que el sensor se encuentre activado, para ello verifique el LED de encendido de este.
- Verificar que el dispositivo se encuentre conectado con la fuente de alimentación.

Correo electrónico y teléfonos de soporte técnico

Para contactarnos puede ubicarnos en la siguiente página web: www.gasprotect.com.ec

- Teléfonos de contactos:
Teléfono de oficina: 022555741 ext. 117
Teléfono celular: 0995062389
- Correo electrónico:
servicioalcliente@gasprotect.com.ec

Anexo 4 – Manual Técnico



MANUAL TÉCNICO

DETECTOR DE FUGAS DE GLP

1. OBJETIVOS DEL MANUAL Y ALCANCES DEL SISTEMA

El objetivo de este manual es dar a conocer los elementos que conforman este prototipo y su funcionamiento. Guiar para resolver las fallas de hardware y software más comunes y simples que se pueden presentar durante el uso del prototipo. Y dar indicaciones de cómo realizar un mantenimiento preventivo y correctivo del prototipo.

El prototipo está diseñado para detectar fugas de GLP en el ambiente y enviar la información, mediante la utilización de tecnología de comunicación Wi-Fi y una aplicación en Android para teléfonos inteligentes, la notificación indicará al usuario que existe una fuga de gas y el sistema automáticamente cerrará el flujo de GLP y evacuará dichos gases mediante un sistema de ventilación.

2. Especificaciones técnicas

2.1. Tecnología Arduino

Arduino es una plataforma electrónica de código abierto cuya política es contar y proporcionar software y hardware fáciles de usar.

El módulo Arduino tiene una interfaz de entrada que puede estar directamente conectada a los periféricos por medio de los puertos de entrada y salida, el objetivo es traspasar la información al microcontrolador, el cual se encarga de procesar los datos.

También tiene una interfaz de salida que lleva la información procesada a los periféricos, los cuales se encargan de hacer uso final de la información, que en algunos casos puede tratarse de otra placa, una pantalla o un altavoz encargada de mostrar los datos.

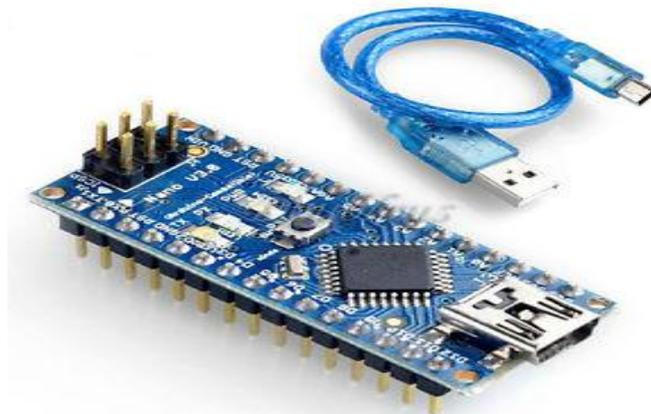


Figura 1. Módulo Arduino Nano ATmega328

2.1.1. Características básicas del Arduino Nano ATmega328

- Voltaje operativo: 5 V
- Voltaje de entrada al regulador: 7 V a 12 V
- Voltaje de entrada límite: 6 V a 20 V

- Pines I/O: 14 (6 pines para salida PWM)
- Entradas análogas: 8
- Corriente I/O: 40 mA
- Memoria flash: 32 KB
- SRAM: 2 KB
- Dimensiones: 0.73' x 1.70'
- Compatibilidad con Windows: 32 y 64 bits
- Peso: 7 gr.

2.1.2. Tecnología Android

Es un sistema operativo **creado para teléfonos con pantalla táctil** (teléfonos inteligentes o de última generación).

La versión de Android que se va a utilizar en este proyecto es: Marshmallow 6.0.1.

2.1.2.1. Android Marshmallow

Tiene como característica el no ceder todo el permiso de uso en las aplicaciones, utilizando un **opt-in** como **sistema regulador, donde el usuario al bajar una aplicación toma la decisión de que permitir y que no permitir.**

La versión de Android Marshmallow está cargada en el terminal de prueba, **que para el proyecto es: Tablet Alcatel**

2.2. Solución Tecnológica del proyecto

La Figura 2, representa el boceto del presente proyecto, es decir el proceso que se sigue para demostrar el funcionamiento de este; desde la conexión del sensor de gas en el cilindro, el proceso de control por parte del microcontrolador Arduino, y por último la interacción con el teléfono celular mediante la aplicación elaborada en Android Studio.

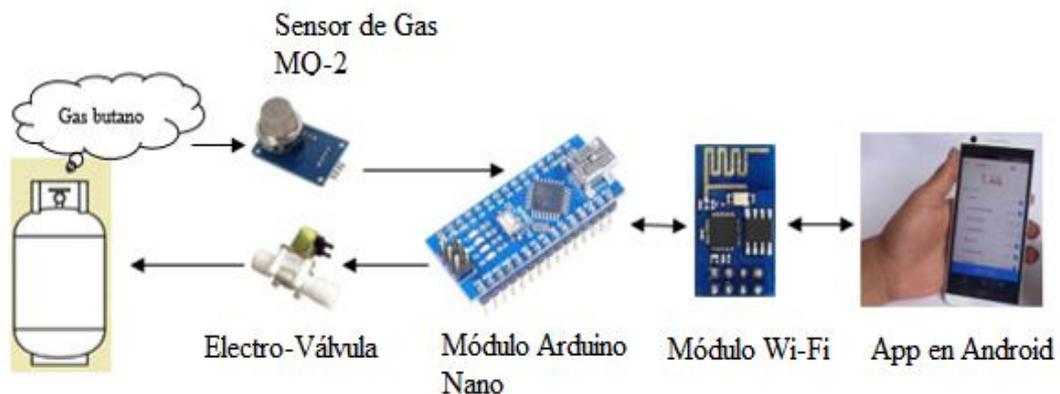


Figura 2. Esquema del Proyecto

2.3. Diagrama de Flujo del Sistema

En primer lugar se va a encender el prototipo, con el *Switch* que se encuentra instalado en él, esto permitirá la conexión del teléfono celular con el módulo Wi-Fi, mediante el teléfono celular se va a ingresar a la aplicación creada en Android Studio, para configurar el umbral de sensibilidad del sensor, si fuese necesario, esto va a permitir que el sensor sea más o menos sensible al momento de detectar el gas, depende de la configuración que se realice en el microcontrolador Arduino, el sensor realiza detecciones periódicas y constantes del ambiente, cuando este se encuentre con un nivel considerado de gas, el sensor va a alarmarse y enviar una señal, dicha alarma va a llegar mediante la aplicación al teléfono celular. Después de detectar la alarma la electroválvula cerrara el paso de gas y se activara el ventilador para disipar el gas que se encuentre en el lugar, como el sensor va a detectar continuamente el nivel de gas, una vez que el aire esté libre de esta sustancia toxica la electroválvula se abre y el ventilador se apagará, como se aprecia en la Figura 3.

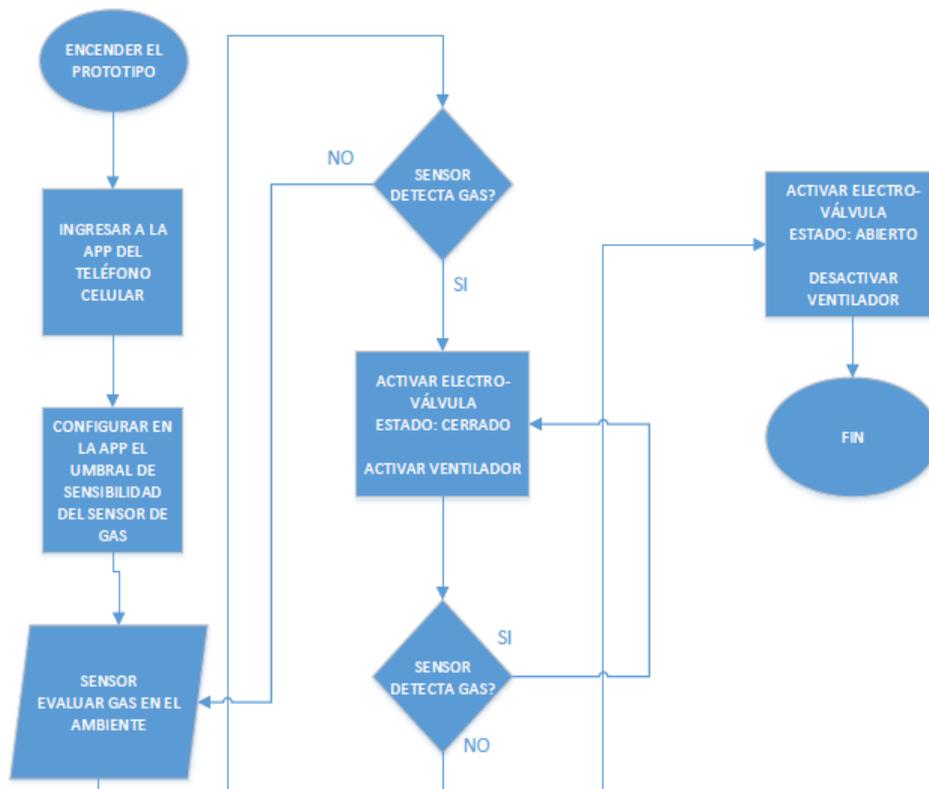


Figura 3. Diagrama de flujo del funcionamiento del Sistema

2.4. Diagrama de Bloques

En esta parte del proyecto se muestra los componentes electrónicos y en base a que fueron seleccionados para la elaboración y construcción del prototipo.

2.4.1. Diseño del Hardware

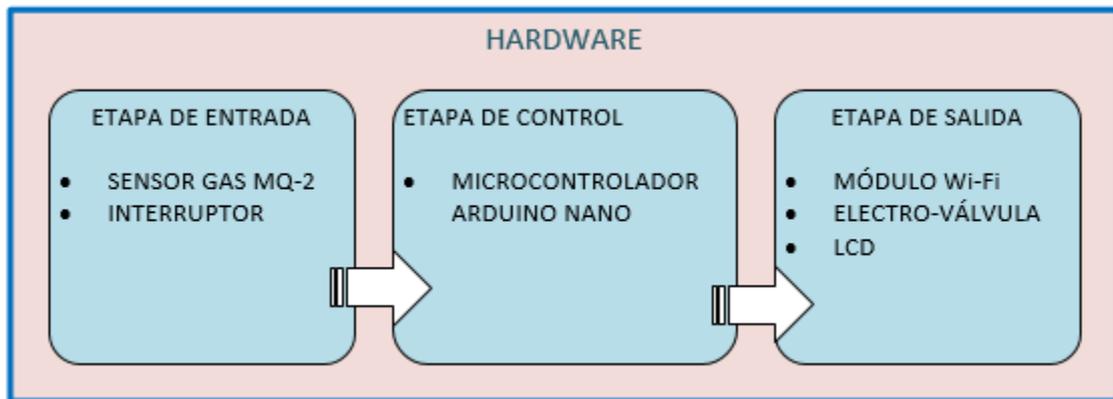


Figura 4. Diagrama de bloques del *hardware* del prototipo

2.4.1.1. Etapa de Entrada

En la Figura 4, se muestran los componentes del *hardware*, en la etapa de entrada se utiliza el sensor de gas MQ-2, el cual está conectado de manera analógica, con esta conexión se medirá la cantidad de gas en el aire en ppm (partes por millón).

Adicional al sensor también se tiene un interruptor, que permitirá el encendido y apagado del prototipo.

2.4.1.2. Etapa de Control

En esta etapa el módulo Arduino Nano (seleccionado porque su tamaño es reducido, posee una memoria de 16 KB, funciona con un voltaje que puede oscilar entre 5 voltios a 12 voltios) se interconecta con el módulo Wi-Fi ESP-12F (seleccionado por su bajo consumo de energía, por su confiabilidad para el envío de datos hacia Internet, permite configuración de parámetros mediante comandos AT), ambos poseen la programación necesaria para su intercomunicación.

2.4.1.3. Etapa de Salida

El módulo Arduino Nano, interactúa con el módulo Wi-Fi, electroválvula, que permite verificar notificaciones del estado del sensor, y la entrega de esa información al usuario mediante el teléfono celular, el LCD (display de cristal líquido, *liquid crystal display*), el cual exhibe mensajes, por ejemplo: La detección del gas, símbolo de la conexión Wi-Fi, entre otros.

2.4.2. Diseño de Software

En la Figura 5 se muestran las etapas de funcionamiento que debe proporcionar el software del proyecto, las cuales se detallan a continuación.



Figura 5. Diagrama de bloques del *software* del prototipo

2.4.2.1. Etapa de Entrada

En esta etapa se realizará la activación del hardware del proyecto, mediante la señal inalámbrica del Wi-Fi, que va a transmitir la información hacia la siguiente etapa.

2.4.2.2. Etapa de Control

La etapa de control consta de la aplicación creada en Android Studio, que permitirá la configuración y modificación de los niveles de detección del sensor de gas, y además recibirá las alarmas de notificaciones en caso de haber una fuga de gas.

2.4.2.3. Etapa de Salida

La información y configuraciones que se realicen se enviarán mediante la señal inalámbrica, que enviará la información para ser transmitida hacia el módulo Wi-Fi.

En la Figura 6, se muestra el diagrama esquemático del proyecto, donde se visualizan los componentes que son parte del diseño como: el microcontrolador Arduino Nano, el módulo LCD, el sensor de gas, el módulo Wi-Fi, entre otros, que se encuentran en el programa Proteus de simulación.

Proteus tiene dos tipos de *software* de simulación: ISIS y ARES, el primero permite elaborar el diseño del circuito con los dispositivos electrónicos necesarios para realizar la simulación de manera correcta, además este ambiente permite cargar programas elaborados en lenguajes de programación como C++, pues su sistema es compatible, y el segundo permite llevar el diseño electrónico como un circuito impreso.

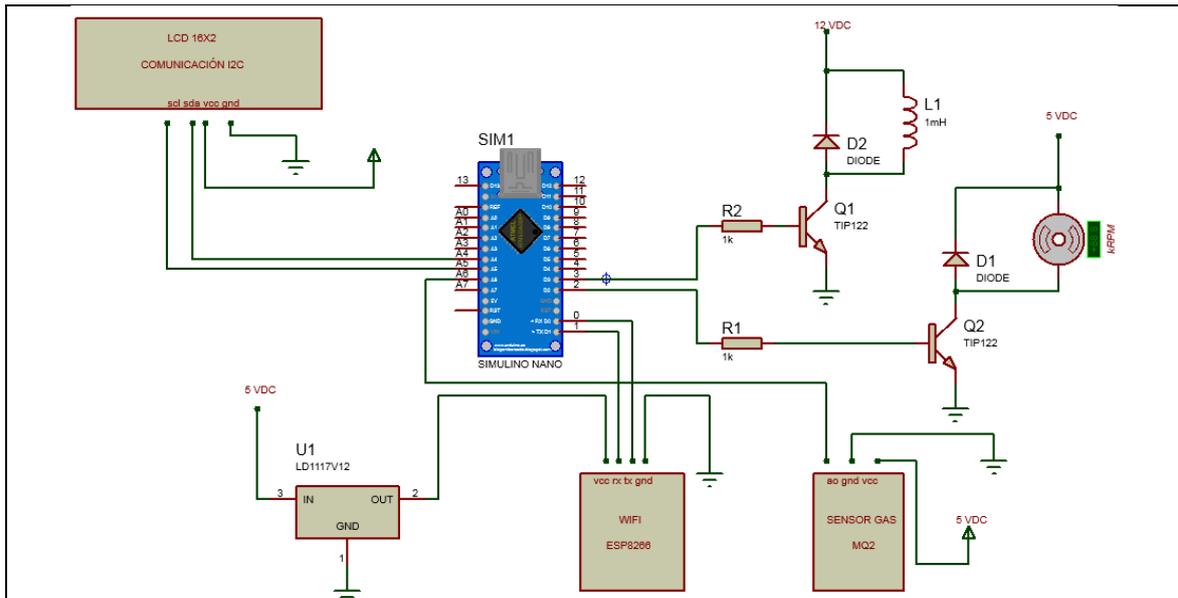


Figura 6. Diagrama Esquemático

Al microcontrolador Arduino Nano se encuentran conectados los módulos, Wi-Fi, el sensor de gas, el ventilador, la electroválvula, y demás componentes electrónicos, por este motivo es el encargado de la parte de control del Prototipo, además en el microcontrolador se encuentra la programación necesaria que se va a ejecutar.

En la Figura 7, se encuentra la conexión del microcontrolador Arduino Nano y el módulo Wi-Fi ESP8266, la comunicación con el módulo se realiza mediante puerto serie, el módulo Wi-Fi puede configurarse mediante comandos AT la velocidad de transmisión que puede ser: 9600, 19200, 38400, 74880, 115200, 230400, 460800 y 921600, pero por defecto se usa a 9600, los comandos AT son comandos de texto enviados por Serial.

La conexión se realiza mediante los pines de transmisión y recepción de ambos módulos, el pin de transmisión del módulo Wi-Fi irá conectado al pin de la recepción de módulo Arduino y el pin de la recepción del módulo Wi-Fi irá conectado al pin de transmisión del microcontrolador.

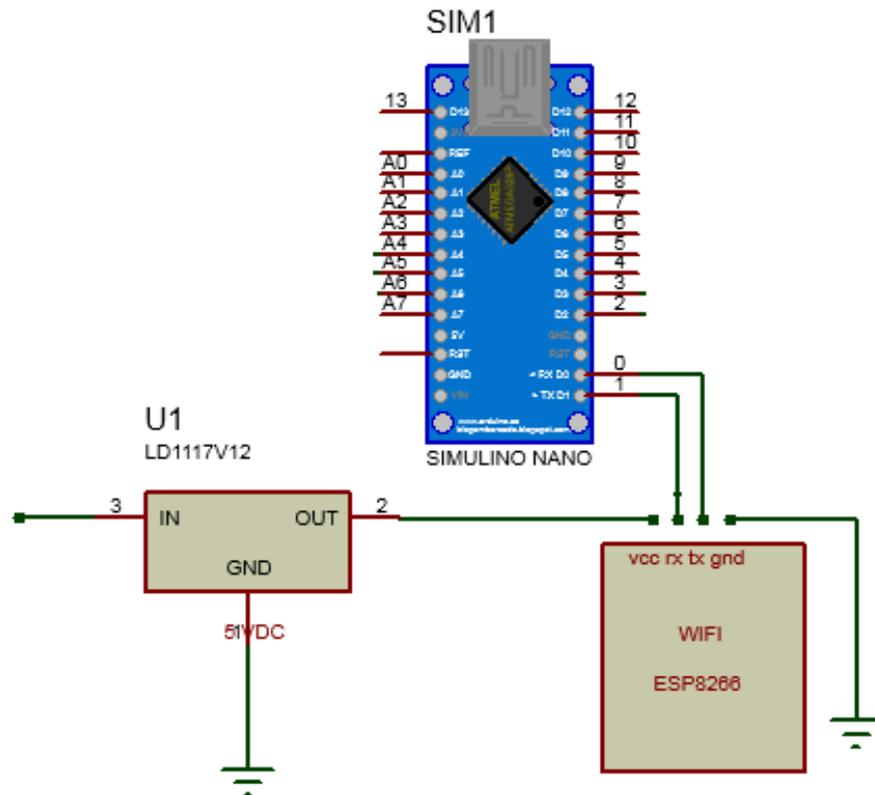


Figura 7. Conexión entre el módulo Wi-Fi y el módulo Arduino Nano

El sensor MQ-2 consta de 4 pines de conexión: V_{CC}, GND, DO y AO. DO y AO son salidas de datos, la primera digital y la segunda analógica, para el proyecto se usa la salida de datos analógica AO, esta se encuentra conectada al pin A6 del microcontrolador Arduino Nano, este pin recibe señales analógicas, como se muestra en la Figura 8. La salida analógica aumenta el valor del voltaje en relación al nivel de gas detectado. V_{CC} y GND son pines de conexión para el voltaje de alimentación del sensor.

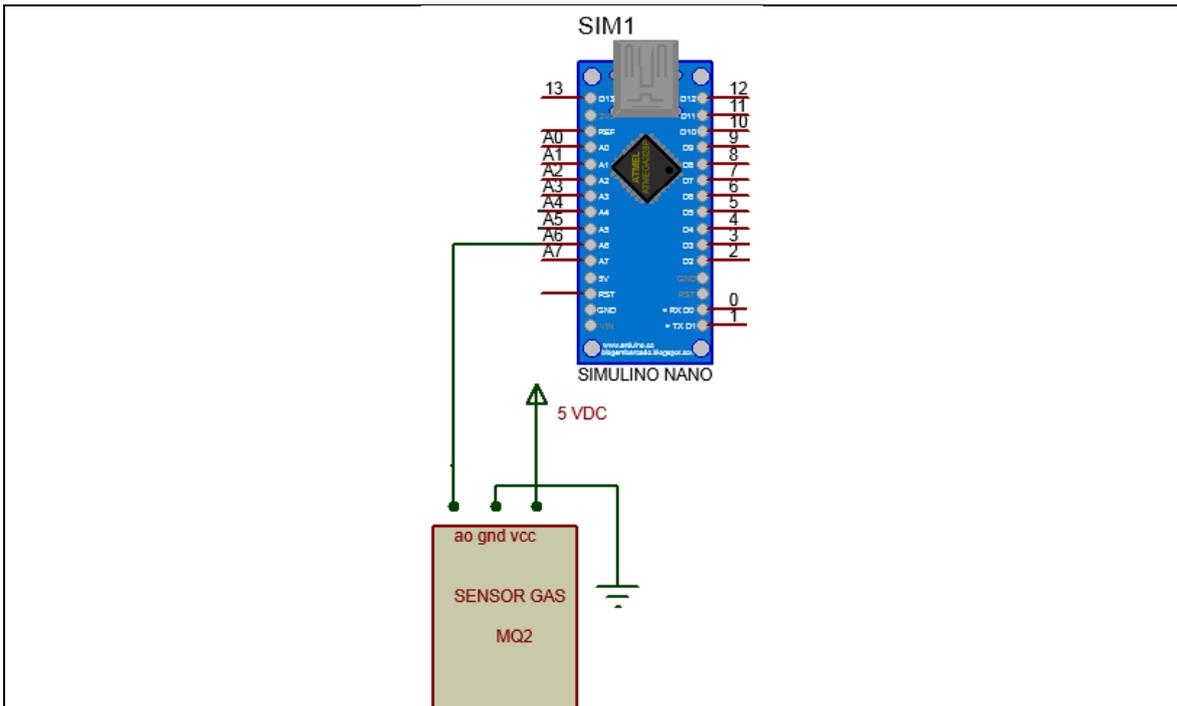


Figura 8. Conexión entre el sensor MQ-2 y módulo Arduino Nano

3.1.1. Construcción del Prototipo

Para iniciar la construcción del prototipo primero se realizaron los diagramas de las PCB (Placas de Circuito Impreso), a partir de los diagramas esquemáticos obtenidos, los cuales fueron realizados mediante el *software* de diseño Proteus con la herramienta para la creación de PCB llamada Ares. Con los diagramas se fabricaron las placas de circuito impreso correspondientes, como se muestra en la Figura 9.

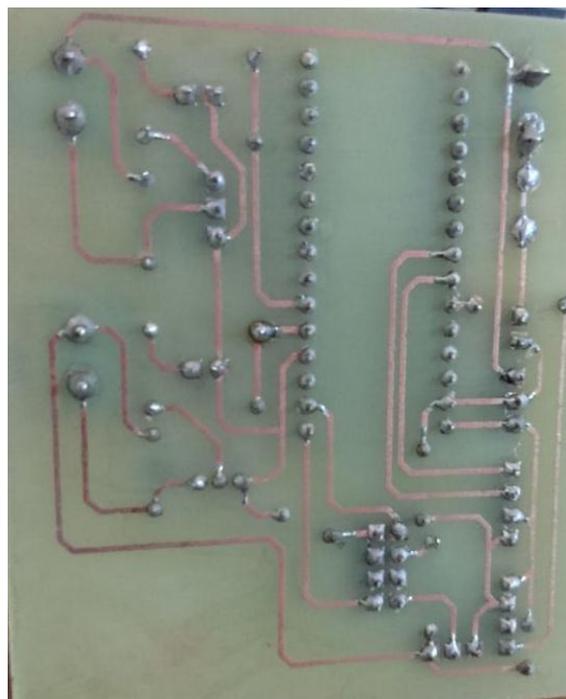


Figura 9. Fotografía de la tarjeta electrónica elaborada

Los componentes que se usaron son los siguientes:

- Arduino Nano ATmega328.
- Módulo Wi-Fi ESP8266
- Display LCD de 16 segmentos.
- Transistores.
- Diodos LED.
- Fuente de alimentación.
- Cables.
- Resistencias.
- Adaptador cable I2C para la conexión del LCD

En la Figura 10, se muestra la tarjeta de circuito impreso terminada, con los elementos y dispositivos colocados de manera correcta, se ha empleado cables de conexión y sócalos para que sea más sencillo el reemplazar los módulos de ser necesario.

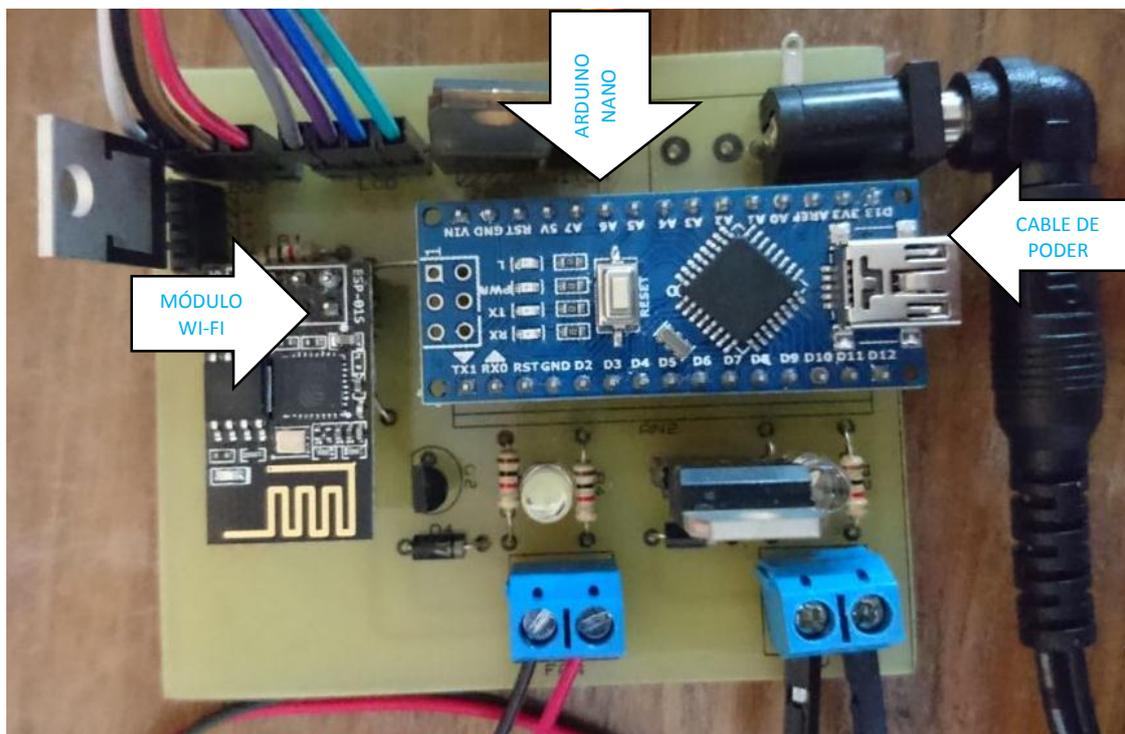


Figura 10. PCB del Prototipo completo

En la Figura 11, se observan los componentes principales y aquellos dispositivos necesarios para su conexión o interconexión, como son los cables de conexión, sócalos, borneras, entre otros.

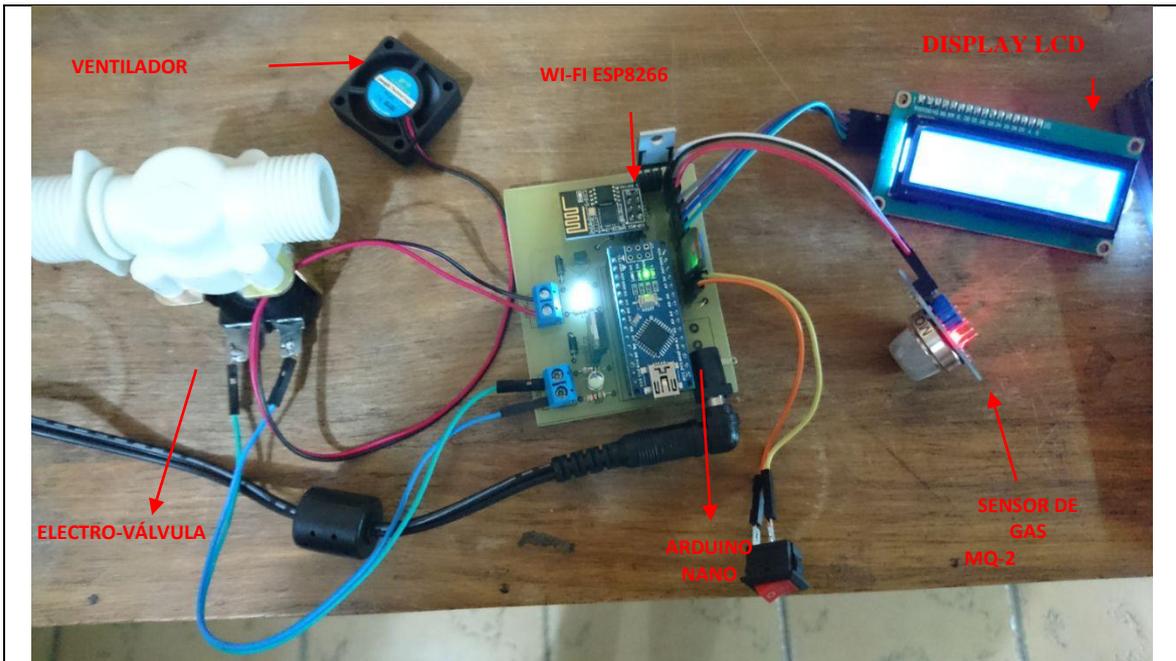


Figura 11. Tarjeta del Prototipo Completa

3. GUÍA DE MANTENIMIENTO

- Verificar que los cables del botón de encendido se encuentren bien soldados a la placa.
- Verificar que los terminales de la fuente de poder se encuentren en buen estado.
- Comprobar que el Wi-Fi del teléfono se empareje con el módulo Wi-Fi del prototipo.
- Verificar que las abrazaderas del prototipo estén muy bien sujetas al sitio donde está instalado para que éste no sufra daños por desprenderse.
- Comprobar que el sistema de apertura y cierre de válvula funcione adecuadamente.
- Verificar en el caso de una fuga si el sistema responde a la ventilación inmediata.

4. GUÍA PARA LA SOLUCIÓN DE PROBLEMAS

4.1. El dispositivo no cierra el flujo de GLP en caso de fuga

- Verificar que el dispositivo está encendido y conectado a la aplicación del teléfono inteligente.
- Verificar si el sensor está con daños físicos o roto, de estarlo se debe cambiarlo por un nuevo.
- Verificar que los pines de la electroválvula y los cables estén en buen estado, si lo están proceder con el cambio de la electroválvula por una nueva, ya que pudo haber sufrido algún tipo de daño.

4.2. El dispositivo no entrega los niveles de detección de GLP

- Verificar en primera instancia que el módulo Wi-Fi del teléfono inteligente y el del prototipo estén encendidos y emparejados.
- Verificar que el prototipo se encuentre en un espacio abierto y con una distancia entre éste y el teléfono inteligente de no más de 20 metros aproximadamente.
- Verificar en el momento de encendido del prototipo que el led indicador del módulo Wi-Fi encienda, de no hacerlo cambiar este módulo, por uno igual.
- Verificar que el conector del sensor esté correctamente conectado y que el cable no presente daños, se debe realizar una verificación de continuidad del cable, si éste presenta daños se debe reemplazarlo.

4.3. Si el prototipo no enciende

- Verificar si los terminales de fuente de poder, de estar rotos o desoldados proceder a reemplazarlos y/o realizar una nueva suelda de los mismos.
- Verificar que el interruptor principal de encendido esté funcionando correctamente, esto mediante la comprobación de sus terminales con un multímetro.

6. E-MAIL Y TELÉFONOS DEL DEPARTAMENTO TÉCNICO

- Teléfonos de contacto:
Teléfono de oficina: 022555741 ext. 117
Teléfono celular: 0995062389
- Correo electrónico:
servicioalcliente@gasprotect.com.ec