



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELECTRÓNICO EN REDES Y COMUNICACIÓN DE DATOS

**TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA
DETECCIÓN Y MONITORÉO DE LA CONCENTRACIÓN DE GAS
GLP MEDIANTE ARREGLO DE SENsoRES ORIENTADO AL
SECTOR RESIDENCIAL**

AUTOR: RAMÍREZ PATAJALO GUSTAVO ANDRÉS

DIRECTOR: ING GORDILLO RODOLFO

SANGOLQUÍ

2016



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

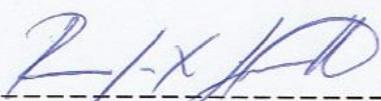
DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA DETECCIÓN Y MONITOREO DE LA CONCENTRACIÓN DE GAS GLP MEDIANTE ARREGLO DE SENsoRES ORIENTADO AL SECTOR RESIDENCIAL**” realizado por el señor **GUSTAVO ANDRÉS RAMÍREZ PATAJALO**, ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **GUSTAVO ANDRÉS RAMÍREZ PATAJALO** para que lo sustente públicamente.

Sangolquí, 13 de Septiembre del 2016



ING. RODOLFO GORDILLO
DIRECTOR



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

AUTORÍA DE RESPONSABILIDAD

Yo, **GUSTAVO ANDRÉS RAMÍREZ PATAJALO**, con cédula de identidad N° 172133812-5, declaro que este trabajo de titulación "**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA DETECCIÓN Y MONITOREO DE LA CONCENTRACIÓN DE GAS GLP MEDIANTE ARREGLO DE SENsoRES ORIENTADO AL SECTOR RESIDENCIAL**" ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Sangolquí, 13 de Septiembre del 2016


GUSTAVO ANDRÉS RAMÍREZ PATAJALO

C.C.: 172133812-5



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA EN REDES Y COMUNICACIÓN DE DATOS

AUTORIZACIÓN

Yo, **GUSTAVO ANDRÉS RAMÍREZ PATAJALO**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación "**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA DETECCIÓN Y MONITORÉO DE LA CONCENTRACIÓN DE GAS GLP MEDIANTE ARREGLO DE SENsoRES ORIENTADO AL SECTOR RESIDENCIAL**" cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Sangolquí, 13 de Septiembre del 2016



GUSTAVO ANDRÉS RAMÍREZ PATAJALO

C.C.: 1721338125

DEDICATORIA

Dedico este trabajo a mis padres y hermanos, quienes me brindaron todo su apoyo y han sido mi fuente de inspiración para seguir a lo largo de esta carrera hasta culminarla.

A mis abuelos que me bendicen cada día desde el cielo y la familia que me llena de orgullo y fuerza para superarme.

RAMÍREZ PATAJALO GUSTAVO ANDRÉS

AGRADECIMIENTO

Agradezco de manera especial a mis padres por su ejemplo y la inspiración que han sembrado en mí para buscar siempre la superación personal.

Agradezco de manera especial a mi director de proyecto, el Ingeniero Rodolfo Gordillo por su colaboración y las facilidades que me ha brindado para el desarrollo de este proyecto.

Agradezco de manera especial al Dr. Nikolai Espinosa Ortiz, por su enorme colaboración y paciencia en las aulas como en la dirección de carrera.

RAMÍREZ PATAJALO GUSTAVO ANDRÉS

ÍNDICE DE CONTENIDOS

CARÁTULA.....	I
DEDICATORIA	V
AGRADECIMIENTO	VI
ÍNDICE DE CONTENIDOS	VII
ÍNDICE DE TABLAS	IX
ÍNDICE DE FIGURAS.....	X
RESUMEN.....	XIV
ABSTRACT.....	XV
CAPÍTULO I.....	1
GENERALIDADES	1
1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES	2
1.3 JUSTIFICACIÓN.....	3
1.4 ALCANCE.....	3
1.5 OBJETIVOS DEL PROYECTO	4
1.5.1 OBJETIVO GENERAL.....	4
1.5.2 OBJETIVOS ESPECÍFICOS	4
CAPITULO II	6
FUNDAMENTOS	6
2.1 GASES Y SENSORES.....	6
2.1.1 CONCEPTOS BÁSICOS DE LOS GASES	6
2.1.2 SENSORES DE GASES Y SUS CARACTERÍSTICAS.....	17
2.1.3 PRINCIPIOS DE FUNCIONAMIENTO DEL SENSOR DE GAS SEMICONDUCTOR	19
2.1.4 PARÁMETROS ESTÁTICOS	21
2.1.5 UNIDADES DE MEDIDA.....	24
2.2 MARCO REGLAMENTARIO.....	25
2.2.1 NTE INEN 2 260:2010 SEGUNDA REVISIÓN	25
2.2.2 NCh 2245 Of. 2009	25
2.2.3 MN-DT-600_A.....	26
2.3 COMPONENTES DEL HARDWARE	26
2.3.1 ARDUINO.....	26
2.3.2 ARDUINO GENUINO MEGA 2560 REV3	28
2.3.3 SHIELD ETHERNET PARA ARDUINO	30
2.3.4 SENSOR DE GAS GLP MQ-6	32
2.4 SOFTWARE PARA DESARROLLO.....	35
2.4.1 ARDUINO Y IOT	35
2.4.2 PRINCIPIOS BÁSICOS DE REDES	39
2.4.3 ENTORNO DE PROGRAMACIÓN DE ARDUINO (IDE).....	49
2.4.4 PLATAFORMA MÓVIL ANDROID	51

CAPÍTULO III	55
ESTRUCTURA Y MODELO DEL PROTOTIPO.....	55
3.1 CRITERIOS DE DISEÑO DEL SISTEMA	55
3.1.1 HARDWARE	55
3.1.2 SOFTWARE	56
3.2 DESARROLLO DE HARDWARE	57
3.2.1 INSTALACIÓN DE LOS SENsoRES	57
3.2.2 DISEÑO DE LA MAQUETA	64
3.2.3 CALIBRACIÓN DEL SISTEMA	66
3.2.4 ALIMENTACIÓN ELÉCTRICA	73
3.2.5 MANTENIMIENTO	75
3.3 DESARROLLO DE SOFTWARE	76
3.3.1 PROGRAMACIÓN PARA LA TARJETA ARDUINO	76
3.3.2 MUESTREO Y LÍMITES PERMITIDOS	79
3.3.3 PROGRAMACIÓN DEL CÓDIGO HTML.....	80
3.3.4 SERVICIO AL CLIENTE DE RED LOCAL	83
3.3.5 SERVICIO AL CLIENTE DE RED REMOTA.....	85
3.3.6 PROGRAMACIÓN PARA LA APP ANDROID	89
 CAPÍTULO IV	98
ANÁLISIS DE RESULTADOS	98
4.1 RESULTADOS DEL SISTEMA	98
4.1.1 PUESTA EN MARCHA DEL SISTEMA	98
4.1.2 COMPORTAMIENTO DEL SISTEMA A EXPOSICIÓN DE GLP	104
4.2 DISCUSIÓN DE LOS RESULTADOS	116
 CAPÍTULO V	118
CONCLUSIONES Y LINEAS FUTURAS DE INVESTIGACIÓN.....	118
5.1 CONCLUSIONES	118
5.2 RECOMENDACIONES.....	119
5.3 TRABAJOS FUTUROS	119
REFERENCIAS	120
ANEXOS	115

ÍNDICE DE TABLAS

Tabla 1	Características físico-químicas del GLP	11
Tabla 2	LEL de gases combustibles.....	13
Tabla 3	Potencia en los aparatos de gas	15
Tabla 4	Potencia Aproximada de los artefactos domésticos a Gas	15
Tabla 5	Unidades de medida de gases	24
Tabla 6	Especificaciones técnicas Arduino Mega	30
Tabla 7	Tecnologías Ethernet	41
Tabla 8	Diferencias entre UDP y TCP	44
Tabla 9	Requerimientos de sistema para Android Studio.....	54
Tabla 10	Área de cobertura de sensores.....	59
Tabla 11	Distancias mínimas entre tuberías	64
Tabla 12	Datos aproximados de la curva del MQ-6	68
Tabla 13	Potencia de consumo del sistema.....	74
Tabla 14	Mantenimiento según el tipo de sensor.....	76

ÍNDICE DE FIGURAS

Figura 1 Rombo de clasificación de riesgos GLP	11
Figura 2 Límites de inflamabilidad del GLP para aire con 21% O ₂	14
Figura 3 Estructura básica de funcionamiento de un sensor	17
Figura 4 Estructura interna de semiconductor	18
Figura 5 Modelo de barrera de potencial de entre-grano (ausencia de gases)	19
Figura 6 Modelo de barrera de potencial de entre-grano (presencia de gases)	20
Figura 7 Desviación media cuadrática (MDQ)	21
Figura 8 Histéresis	22
Figura 9 Repetibilidad	22
Figura 10 Comparación de resolución	23
Figura 11 El universo Arduino	28
Figura 12 Configuración de Arduino Mega	29
Figura 13 Shield Ethernet	31
Figura 14 Sensor de gas GLP Mq-6	33
Figura 15 Características de sensibilidad de MQ-6	34
Figura 16 Aplicación de Internet de las Cosas	36
Figura 17 Plataformas del Internet de las cosas	37
Figura 18 Internet de las Cosas con Ubidots	37
Figura 19 Formato de los datos HTML	38
Figura 20 Estructura general de una línea de código con etiquetas HTML	39
Figura 21 Formato etiqueta HTML	40
Figura 22 Interacción cliente - servidor WEB	45
Figura 23 Petición HTTP	46
Figura 24 Respuesta HTTP	46
Figura 25 Topología de red cliente - servidor WEB	47
Figura 26 Tecnologías agrupadas bajo el concepto de AJAX	48
Figura 27 Diferencia entre modelos de aplicación WEB	49
Figura 28 Secuencia de programación en el microcontrolador	51
Figura 29 Arquitectura Android	52
Figura 30 Ciclo de vida de un Activity	53
Figura 31 Esquema general de funcionamiento	55

Figura 32 Diagrama de bloques del circuito	56
Figura 33 Tareas del programa principal	57
Figura 34 Ubicación normal de detector de gas pesado.....	60
Figura 35 Ubicación pendiente de detector de gas pesado	60
Figura 36 Ubicación de sensores	61
Figura 37 Cableado del sensor	63
Figura 38 Cableado de ventilador	63
Figura 39 Ducto de ventilación del sistema.....	65
Figura 40 Estructura de la maqueta.....	66
Figura 41 Ecuación potencial del sensor.....	69
Figura 42 Circuito de conexión del sensor MQ-6.....	70
Figura 43 Captura de datos GLP.....	71
Figura 44 Código de captura datos GLP	72
Figura 45 Datos de gas, escenario inicial.....	72
Figura 46 Datos de gas, escenario controlado.....	73
Figura 47 Datos de gas, escenario de peligro.....	73
Figura 48 Sistema electrónico del prototipo	74
Figura 49 Diagrama de flujo del programa principal.....	77
Figura 50 Diagrama de flujo de la función Setup	78
Figura 51 Diagrama de flujo de la función Loop.....	79
Figura 52 Funciones del programa HTML	81
Figura 53 Aplicación Web usuario local.....	84
Figura 54 Creación de variables en Ubidots	85
Figura 55 Envió de un dato por medio del sitio Ubidots	86
Figura 56 Codificación del mensaje al API de Ubidots.....	87
Figura 57 Visualización de variables en Ubidots.....	88
Figura 58 Notificaciones en Ubidots	88
Figura 59 Creación de nuevo proyecto Android.....	89
Figura 60 Configuración del nuevo proyecto	90
Figura 61 Configuración de mínima versión Android.....	90
Figura 62 Creación de Activity	91
Figura 63 Configuración de la Activity	91

Figura 64 LinearLayout(vertical)	92
Figura 65 Widge LargeText.....	92
Figura 66 Configuración de ImageView()	93
Figura 67 Métodos del MainActivity	93
Figura 68 Configuración de permisos	94
Figura 69 Interfaz de la app	95
Figura 70 Iconos personalizados de la app	95
Figura 71 Servicio notificación Android	96
Figura 72 Escenarios para app notificación	96
Figura 73 Comunicación API Pushingbox	97
Figura 74 Notificación en Android	97
Figura 75 Reconocimiento del servidor en la red local	98
Figura 76 Proceso de inicialización del servidor	99
Figura 77 Requerimiento de datos hacia el servidor.....	99
Figura 78 Tramas de datos del servidor	99
Figura 79 Confirmación de recepción de datos por usuario remoto	100
Figura 80 Señal estable del sensor 1	100
Figura 81 Señal estable del sensor 2	101
Figura 82 Señal estable del sensor 3	101
Figura 83 Base de datos de Planta 1	102
Figura 84 Base de datos de Planta 2 exportada a Microsoft Excel	102
Figura 85 Transmisión de un paquete inicial	103
Figura 86 Transmisión de un paquete final.....	103
Figura 87 Paquetes inicial-final en transmisión.....	104
Figura 88 Valores iniciales desde el servidor	104
Figura 89 Valores iniciales desde el cliente local	105
Figura 90 Valores iniciales desde el cliente remoto	105
Figura 91 Incremento de gas en Planta 1	106
Figura 92 Concentración de gas en el tiempo Planta 1	106
Figura 93 Señal de concentración de gas de Planta 1	107
Figura 94 Notificaciones recibidas por el teléfono celular del usuario.....	107
Figura 95 Notificación para Android	107

Figura 96 Notificación a e-mail	108
Figura 97 Concentración de gas en Planta 1	108
Figura 98 Concentración de gas en Planta 1	109
Figura 99 Notificaciones recibidas por el teléfono celular del usuario.....	109
Figura 100 Notificación para Android	110
Figura 101 Notificación a e-mail	110
Figura 102 Concentración de gas en Planta 2	110
Figura 103 Concentración de gas en Planta 1	111
Figura 104 Concentración de gas en Planta 3	111
Figura 105 Incremento de gas en Planta 2	112
Figura 106 Concentración de gas en el tiempo Planta 2	112
Figura 107 Contaminación de todas las plantas debido a la planta 2	113
Figura 108 Notificación para Android	113
Figura 109 Notificación a e-mail	113
Figura 110 Concentración de gas en el tiempo Planta 3	114
Figura 111 Notificación para Android	114
Figura 112 Notificación a e-mail	115
Figura 113 Concentración de gas en el tiempo Planta 3	115
Figura 114 Notificación para Android	115
Figura 115 Notificación a e-mail	116

RESUMEN

En el presente trabajo se analiza el funcionamiento y las herramientas compatibles con el Hardware libre Arduino que es utilizado como infraestructura para implementar un algoritmo basado en lenguaje C++ bajo distribución de software libre, con el fin de implantar un prototipo para detectar la concentración excesiva de gas combustible casero, quien tiene las funciones de mantener al usuario al tanto del estado del sistema que se analiza y tomar acciones preventivas sobre el mismo ante comportamientos que atenten su equilibrio. El proyecto de investigación se enfoca en la recolección de datos tomados del medio ambiente por medio de sensores analógicos de gas los cuales son calibrados de acuerdo a las especificaciones del fabricante para evitar medidas erróneas, la tarjeta Arduino como unidad de procesamiento de datos es configurada para trabajar conjuntamente con un Shield Ethernet para comunicar al dispositivo con el mundo exterior utilizando lenguajes que se emplea para el desarrollo de páginas de internet como también API's para conectarse a servicios contenidos en la nube. Parte del proyecto es desarrollar una aplicación bajo el sistema operativo Android quien facilitara la interfaz para comunicarse con el sistema dentro y fuera de la red local. El estudio finaliza con el análisis de resultados obtenidos de las pruebas realizadas en una maqueta que simula una edificación domiciliaria típica.

PALABRAS CLAVE:

- ARDUINO
- API
- HTML
- NETWORKING

ABSTRACT

In this work the operation and compatible tools discussed with free Arduino Hardware which is used as infrastructure to implement a system based on C ++ language under the free software distribution algorithm, in order to implement a prototype to detect excessive concentration of combustible gas home, who has the functions to keep the user aware of the state of the system is analyzed and take preventive action on it to behaviors that threaten their balance. The research project focuses on collecting data from the environment through analog gas sensors which are calibrated according to the manufacturer's specifications to avoid erroneous measurements, Arduino card as the data processing unit is configured to work in conjunction with an Ethernet Shield device to communicate with the outside world using languages used for developing websites as API's to connect to cloud content services. Part of the project is to develop an application under the Android operating system who facilitate the interface to communicate with the system inside and outside the local network. The study concludes with an analysis of results obtained from tests on a model that simulates a typical home building.

KEYWORDS:

- ARDUINO
- API
- HTML
- NETWORKING

CAPÍTULO I

GENERALIDADES

1.1 Introducción

La presente investigación tuvo como finalidad desarrollar un sistema para preservar la seguridad de personas y bienes que en algún momento podrían estar en riesgo, debido a los incidentes que conlleva la concentración desmedida de gas doméstico dentro de edificaciones residenciales, es necesario resaltar que la prevención es el aspecto más importante de la seguridad.

El principal interés de esta investigación fue desarrollar un sistema de prevención de accidentes como incendios o inhalación por la presencia de gas, que además de ser económico aproveche la tecnología que está al alcance de la población común.

Con el desarrollo de la investigación se determinaron las propiedades y comportamiento de los gases, el gas licuado de petróleo denominado GLP es una mezcla de gas inflamable que no presenta color pero si un ligero olor en altas concentraciones. Con respecto al peligro para la salud asociado con escapes del gas no es su toxicidad, sino asfixia por desplazamiento de aire en un espacio confinado, al alcanzar largas distancias en presencia de una fuente de ignición puede regresar en llamas o formar mezclas explosivas con el aire.

Se llegó a definir los componentes apropiados para el diseño del prototipo. La plataforma de desarrollo Arduino se considera como el corazón del proyecto, pues es el medio por el cual los datos son leídos y traducidos para visualizarlos en la WEB, teniendo una comunicación directa con el medio físico que se monitorea y el usuario final, quien visualiza el comportamiento de las variables medidas y toma acciones.

1.2 Antecedentes

Resulta obvio decir que las nuevas tecnologías forman parte cada día más de todos los ámbitos y aspectos que envuelven la vida cotidiana de los ciudadanos. Tanto es así, que estas se utilizan sin que en muchos casos, uno sea consciente de que las está usando.

La tecnología en general, en la que están incluidas tecnologías de la informática, las comunicaciones, electrónica y la industria en general, no han escatimado esfuerzo para poder desarrollarse rápidamente a favor de nuestra seguridad en el entorno laboral como en el hogar.

Existen empresas dedicadas a elaboración de equipos de medición de gases y vapores en el ambiente con instrumentos colorimétricos o monitores de lectura directa. Drägerwerk Ingeniería es una empresa Alemana que desarrolla, produce y distribuye soluciones fiables para medición de gases en diferentes situaciones y para diverso campos de aplicación, desde equipos de detección móviles y estacionarios para grandes plantas industriales hasta productos de seguridad para pequeñas y medianas empresas. (Drägerwerk, 2016)

A nivel residencial se han desarrollado pocos avances fuera de lo normal, como dispositivos estáticos que necesitan la presencia continua del usuario para avistar algún evento o alarma activa. En el campo de investigación se puede citar la monografía desarrollada en la Universidad de San Buenaventura, en el cual se implementa un “detector de fuga de gas integrando un chip Motorola y electrónica digital” (Colmenares, 2016), obteniendo como producto un sistema que se limita a detectar una fuga de gas en el ambiente. En el trabajo de grado de la Universidad Nueva Esparta, en este se desarrolla un “prototipo para la detección de fugas de gas en tuberías” (Gianfranco, 2013), en la cual detecta una fuga y desactiva una electroválvula localmente situada, en ambos casos si el usuario se mantiene al margen del sistema en ningún momento puede enterarse de los acontecimientos.

1.3 Justificación

Según la Secretaría General de Planificación, en su portal de datos abiertos, “Combustibles utilizados por los hogares para cocinar”, la población de la ciudad de Quito utiliza gas en sus hogares un promedio del 78,62% al 98,84% (Planificación, 2016).

A medida que aumenta el consumo de gas, también aumenta los accidentes ocasionados con el empleo de esta fuente de energía. Muchos de los accidentes relacionados con gas se dan por la mala manipulación, deficiencia en la instalación, falta de mantenimiento preventivo y correctivo.

El monitoreo constante y personalizado del ambiente es de gran ayuda para reducir accidentes y resguardar vidas humanas como bienes materiales. Para ello la tecnología al alcance de la mayoría de la población como el internet se ha convertido en un excelente medio de comunicación que se puede emplear para implementar un sistema que brinde seguridad a un bajo costo.

1.4 Alcance

El desarrollo de este trabajo de investigación se limita a elaborar los siguientes ítems:

Implementar un prototipo para detección de la concentración de gas GLP a nivel residencial, la infraestructura del prototipo tiene como base la utilización de la tecnología Arduino, sus Shield's y el sensor de gas GLP Mq-6 para recolectar datos, procesarlos y tomar decisiones de acuerdo a los umbrales limitados por las normas de regulación.

Implementar una maqueta de un edificio de tres plantas. Cada planta dispondrá de un sensor de gas GLP que enviarán los datos medidos al circuito fuera de la maqueta.

El gas GLP será aislado en las paredes de la maqueta con plástico acetato y el exceso de gas expulsado por un extractor de aire.

Configurar la tarjeta Arduino como un servidor web para interactuar con los clientes que conectados a la red local o una red remota con conexión a internet, podrán monitorear y visualizar estadísticas históricas de datos tomados.

Diseñar una página web para recolectar los datos tomados por el prototipo y procesarlos para ser presentados a los usuarios.

Desarrollar una aplicación móvil en el sistema operativo Android, la cual notificará a los usuarios al presentarse una concentración de gas GLP que supere el nivel umbral permitido en el ambiente.

1.5 Objetivos del proyecto

1.5.1 Objetivo general

Diseñar e implementar un prototipo para detectar la concentración de gas GLP a en el ámbito residencial.

1.5.2 Objetivos específicos

- Implementar el prototipo basado en la plataforma Arduino, siguiendo las normas para desarrollo de sistemas de gestión ambiental del INEN.
- Configurar un servidor web para mostrar los datos de sensores mediante la tarjeta Arduino que se conecta a una red Ethernet.
- Desarrollar una aplicación móvil para notificar al usuario cuando el nivel de contaminación por gas supere el umbral permitido.
- Diseñar una página web que permita comunicarse con el prototipo para monitorearlo y visualizar estadísticas históricas de datos tomados.

- Extender el control del prototipo de la red local a una red remota en cualquier lugar con conexión a internet.

CAPITULO II

FUNDAMENTOS

2.1 Gases y sensores

2.1.1 Conceptos básicos de los gases

2.1.1.1 Gas

Se denomina gas el estado de agregación de la materia que bajo ciertas condiciones de temperatura y presión permanece en estado gaseoso. Para su estudio se clasifica a los gases como gas ideal y gas real.

Un gas ideal es un gas teórico compuesto de un conjunto de partículas puntuales con desplazamiento aleatorio que no interactúan entre sí. El concepto de gas ideal es útil porque el mismo se comporta según la ley de los gases ideales, por medio de una ecuación de estado simplificada se puede analizar un elemento mediante la mecánica estadística. En cierta ocasión se puede considerar que los gases de masa molecular no muy alta a presiones no muy bajas y a temperaturas no excesivamente bajas se comportan como gases ideales, tales como el oxígeno, el nitrógeno, el hidrógeno o el dióxido de carbono.

El gas real, en cambio, es aquel que posee un comportamiento termodinámico y que no sigue la misma ecuación de estado de los gases ideales. Los gases se consideran como reales a presión elevada y poca temperatura.

2.1.1.2 Propiedades de los gases

El estado gaseoso es un estado disperso de la materia, es decir, que las moléculas del gas están separadas unas de otras por distancias mucho mayores del tamaño del diámetro real de las moléculas. Resuelta entonces, que el volumen ocupado por el

gas (V) depende de la presión (P), la temperatura (T) y de la cantidad o número de moles (n).

Volumen de un gas: son las dimensiones del espacio que ocupa un gas. En un sistema cerrado, el gas ocupa todo el volumen del sistema. Así por ejemplo, cuando un gas es metido a un recipiente, se expande uniformemente para ocupar todo el recipiente. Cuando un gas es sacado del recipiente al ambiente tenderá a expandirse por la atmósfera.

Presión de un gas: fuerza dividida entre el área que ejercen las moléculas de gas al chocar unas con otras y con las paredes de un sistema. A la vez, por la 3^a ley de Newton de Acción y Reacción, también es la fuerza por unidad de área que ejercen las paredes del sistema sobre el gas.

Temperatura de un gas: percepción macroscópica de la energía interna que contiene un gas. La energía interna es aquella energía que poseen las moléculas para moverse: rotar, vibrar y desplazarse. A mayor temperatura, mayor energía interna contiene el gas. (Dorame, 2016)

Las propiedades de la materia en estado gaseoso son:

- Se adaptan a la forma y el volumen del recipiente que los contiene. Un gas, al cambiar de recipiente, se expande o se comprime, de manera que ocupa todo el volumen y toma la forma de su nuevo recipiente. Se dejan comprimir fácilmente. Al existir espacios intermoleculares, las moléculas se pueden acercar unas a otras reduciendo su volumen, cuando aplicamos una presión.
- Se difunden fácilmente. Al no existir fuerza de atracción intermolecular entre sus partículas, los gases se esparcen en forma espontánea.
- Se dilatan, la energía cinética promedio de sus moléculas es directamente proporcional a la temperatura aplicada.

2.1.1.3 Tipos de gases combustibles

Según su utilidad el gas combustible se utiliza para producir energía térmica siguiendo un proceso de combustión.

Gas natural: es una importante fuente de energía no renovable que está formado por una mezcla de gases ligeros depositados en yacimientos de petróleo.

Gas licuado del petróleo (GLP): Es la mezcla de gases condensables que se encuentran en el gas natural o disuelto en el petróleo. Los componentes del GLP a temperatura y presión ambientales son gases fáciles de condensar, de ahí su nombre. (Tecnolowikia, 2016)

2.1.1.4 Gases licuados del petróleo (GLP)

Reciben el nombre de Gases Licuados del Petróleo (GLP), las mezclas comerciales de hidrocarburos en los que el butano o el propano son dominantes.

En su estado natural son gaseosos, pero en recipientes cerrados y temperatura ambiente, una gran parte de los mismos están en fase líquida, ocupando un volumen 250 veces inferior al que ocuparían en estado vapor.

Los GLP, una vez extraídos como gases de refinería o instalaciones de gas natural asociados, son comprimidos hasta licuarlos y mantenidos en ese estado en grandes almacenamientos, normalmente anexos a las instalaciones de refinerías o portuarias, llamados terminales.

Los depósitos de almacenamiento son de varios tipos:

- Depósitos a presión
- Depósitos refrigerados, o de presión atmosférica
- Depósitos semirrefrigerados

- Depósitos subterráneos: en cavidad salina o galerías mineras convencionales.

El alto poder calorífico de los GLPs en relación con el resto de los gases comerciales, presenta la ventaja de que para obtener una misma cantidad de energía total, se requieren menores diámetros de las conducciones utilizadas para la distribución del gas en fase gaseosa, si bien las presiones no pueden sobrepasar un determinado valor porque se producirían condensaciones en las canalizaciones.

Sus características de alta pureza, y la homogeneidad de sus componentes, hacen que sea muy fácil ajustar el aire necesario para que la combustión de estos productos, sea la estequiométrica.

Además, y esta es una de sus principales características, están prácticamente exentos de azufre y otras sustancias como metales. Su campo de aplicación, es en general el calentamiento directo, y aquellas instalaciones en las que la accesibilidad del gas propano en depósito, la ausencia de azufre y todas las ventajas que conlleva el ser un combustible gaseoso, le proporcionan un valor añadido frente a otros combustibles. (Repsol, 2016)

Su comercialización está claramente diferenciada en función de la forma de almacenamiento y distribución, en:

- GLP envasado, que se utiliza casi exclusivamente como combustible doméstico para la cocina, el agua caliente y la calefacción.
- GLP almacenado en depósito fijo, que se utiliza principalmente en los sectores domésticos, residencial, comercial, en la agricultura, y en determinadas industrias
- GLP automoción, para su uso como carburante.

2.1.1.5 Características físico-químicas del GLP

Las propiedades de un GLP se alteran según su composición. Estas características son necesarias definirlas con el objeto de utilizarlas para el cálculo que realizan las refinerías, establecer su calidad, las normas de manejo y seguridad del producto.

El GLP es inodoro para su comercialización es mezclado con una substancia odorizante.

El GLP es incoloro tanto en estado líquido como gaseoso. Cuando existe fuga se lo detecta visualmente porque forma hielo en el orificio por donde sale, debido al cambio de estado de líquido a gaseoso.

El GLP es más pesado que el aire en estado gaseoso, por lo que una nube de gas tenderá a permanecer a nivel del suelo siendo un peligro eminente ya que puede encontrar con facilidad una fuente de ignición, sea chispa o llama lo que le hará explotar. También puede causar asfixia en la gente ya que reemplaza al aire.

La temperatura de ebullición es la temperatura a la cual un líquido se toma el estado de gas, a una determinada presión. La mayoría de compuestos cambian su estado en ambientes mayores a los cero grados.

Su densidad es la relación entre la masa de un cuerpo y el volumen que ocupa ese cuerpo, se expresa en kg/m³. La densidad tomada como referencia para los gases es la del aire a 1 atmósfera de presión y 20 °C y su valor es 1,20 kg/m³. (Tecnolowikia, 2016)

Temperatura de inflamación es la temperatura a la cual, los gases se encienden en presencia de una chispa o una llama. GLP es inflamable, las concentraciones necesarias de gas en aire, la temperatura y estado para que una mezcla sea inflamable, oscilan entre los datos de la tabla 1.

Tabla 1
Características físico-químicas del GLP

Parámetro	Valor	Unidad
Estado físico	Gas	
Concentración	95% aprox. Dependiendo del tipo GLP	
Formula	C3H8 + C4H10 (Propano 60%-Butano 40%)	
Estabilidad	Extremadamente inflamable y combustible.	
Temperatura de auto ignición	480 – 549	°C
Punto de ebullición	-42	°C
Punto de inflamación	-104	°C
Temperatura de la llama	1980	°C
Velocidad de propagación de la llama	448	m/s
Presión de vapor a 20 °C	126.406	psi
Densidad de vapor	1.55	G/m ³
Poder calorífico	27.9	kWh/m ³
LEL	1.8	%
UEL	9.3	%

Fuente: (Pemex, 2016)

Clasificación de riesgos de GLP

Cada compuesto químico posee un código estándar para conocer su grado de sensibilidad al tratarlo, para el caso del GLP se muestra en la figura 1.

- █ Salud
- █ Inflamable
- █ Reactividad
- █ Riesgos Especiales

- Riesgo
- 0 Insignificante
 - 1 Ligero-Suave
 - 2 Moderado-Medio
 - 3 Alto-Severo
 - 4 Muy Alto-Extremo

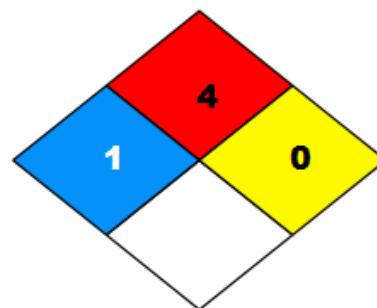


Figura 1 Rombo de clasificación de riesgos GLP

Fuente: (Pemex, 2016)

2.1.1.6 Niveles de detección

Los niveles de concentración que se pueden detectar varían según el fin previsto: seguridad ocupacional o detección de una fuga de gas.

Seguridad ocupacional: Generalmente, las leyes y las normas concernientes a la seguridad ocupacional solicitan que los gases se supervisen a niveles expresados en promedios ponderados en el tiempo (**TWA**) de la exposición de larga (8 horas) y corta duración (10 minutos).

Cada país establece sus propios límites de exposición ocupacional (**OEL**, Occupational Exposure Limit).

Detección de gases: el sector típico que se emplean todos los niveles de detección de gases son refrigeración y aire acondicionado. El valor umbral límite (**TLV**) de larga duración es el umbral máximo de alarma adecuado para los detectores de gas, la mayoría de las normas tienen su origen en la norma ISO 5149 y, por lo tanto, son muy similares.

En Europa, la norma EN 378 2000 especifica ‘límites prácticos’ el umbral máximo de alarma. Dichos límites prácticos se fundamentan en varios criterios. Para halocarbonos en la mitad de la concentración que puede producir asfixia debido al desplazamiento del oxígeno o que tiene efectos narcóticos o de sensibilización del corazón. (Murco, 2016)

LEL (LIE): se aplica a las concentraciones bajas de gases combustibles y significa Lower Explosive Limit (límite inferior de explosividad). Expresa la composición de volumen de un gas inflamable con respecto el aire. Por debajo de este límite no se formará ninguna atmósfera explosiva. Los niveles de concentración con interés de detección de gases se expresan en % de LEL. Es decir, es la relación del volumen del gas con el LEL.

Los valores de LEL pueden ser muy diferentes según los gases combustibles y varían de una autoridad a otra. Normalmente son los siguientes indicados en la tabla 2:

Tabla 2
LEL de gases combustibles

GAS	LEL	GAS	LEL
Amoniaco	15%	Monóxido de carbono	12,5%
Butano	1,9%	Isobutano	1,8%
GLP	1,9%	Hidrógeno	4 %
Metano	5,0%	Gasolina	1,4%
Tolueno	1,1%	Propano	2,1%

Fuente: (Murco, 2016)

Así como el calor debe sobrepasar lo suficiente para alcanzar la temperatura de ignición la relación combustible/comburente (aire) debe estar dentro de los límites de inflamabilidad.

Reactividad - Mezclas de Aire + GLP

En caso de los materiales inflamables estas proporciones se determinan como porcentajes en volumen de gas o vapor en aire. Por lo tanto los porcentajes mínimos y máximos de gas o vapor combustible necesarios para formar mezclas inflamables, constituyen los límites inferior (LIE) y superior (LSE) de inflamabilidad respectivamente. La diferencia entre ambos límites define lo que se conoce como rango inflamable.

En la figura 2 se visualiza las zonas A y B. Un contenedor en condiciones ideales de homogeneidad, las mezclas de aire con menos de 1.8% y más de 9.3% de gas licuado no explotarán, aun cuando esté presente de una fuente de ignición. Sin embargo, a nivel práctico deberá desconfiar de las mezclas cuya composición se acerque a la zona explosiva, porque con una fuente de ignición se desencadena una explosión. (Pemex, 2016)

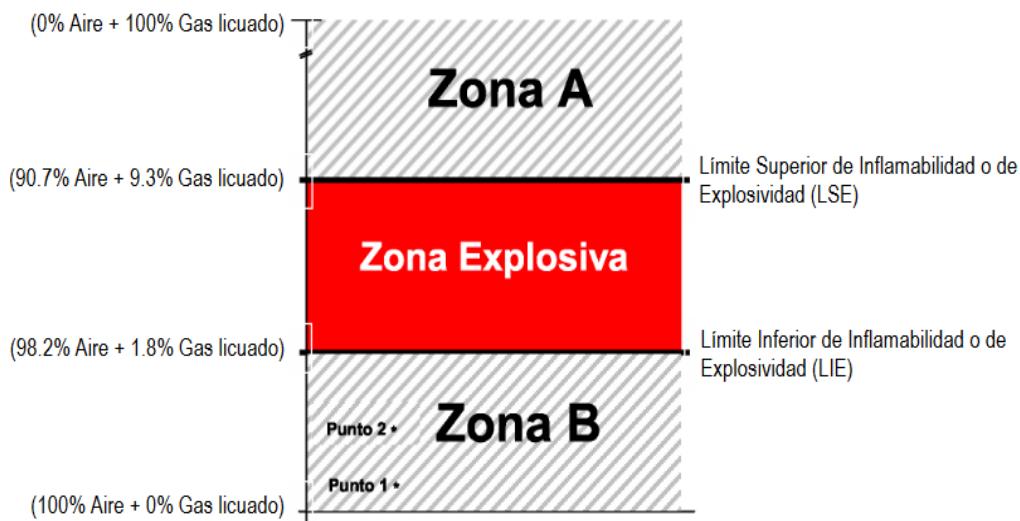


Figura 2 Límites de inflamabilidad del GLP para aire con 21% O₂

Fuente: (Pemex, 2016)

- Punto 1 = 20% del LIE: Valor de ajuste de las alarmas en los detectores de mezclas explosivas.
- Punto 2 = 60% del LIE: Se ejecutan acciones de paro de bombas, bloqueo de válvulas, etc., antes de llegar a la Zona Explosiva

2.1.1.7 Potencia

Para estimar la cantidad de gas requerido en una instalación, se debe determinar directamente a partir del consumo térmico nominal indicado por los fabricantes de los artefactos y equipos de gas a utilizar. En los aparatos se especifica su consumo calorífico o potencia inferior (P_i), en los cálculos se trabaja con la potencia (nominal) superior (P_s). La diferencia radica que con la potencia inferior se considera el poder calorífico inferior (H_i), mientras que con la potencia superior es el (H_s) que se utiliza. La relación entre ellos es: H_s=1.1*H_i, por un lado y P_s=1.1*P_i por otro. En la tabla 3 se resume las potencias presentes en los aparatos.

Tabla 3**Potencia en los aparatos de gas**

POTENCIA INFERIOR o Consumo calorífico (Pi)	POTENCIA SUPERIOR (Ps)	POTENCIA (Pu)	ÚTIL
Potencia consumida referida al Hi $P_i = \text{caudal de gas} \times H_i \text{ (en kW)}$	Potencia producida referida al Hs $Ps = \text{caudal de gas} \times H_s \text{ (en kW)}$ Potencia nominal de utilización simultánea. Hace referencia a ΣPs Considerando simultaneidad.	Potencia transmitida al agua $= \text{caudal de agua} \times Ce \times \Delta t$ $Pu = Ps - \text{pérdidas (en kW)}$	

Ce = calor mísico (específico). Para el agua vale 1/860.

Fuente: (SECNORMATIVA, 2016)

Una aproximación de la potencia consumida por los distintos aparatos cuyo funcionamiento está basado en gas combustible se observa en la tabla 4:

Tabla 4**Potencia Aproximada de los artefactos domésticos a Gas**

Artefacto	Consumo térmico nominal	
	(Mcal/h)	(kW)
Cocina	8	9.3
Encimera	4	4.7
Horno	5	5.8
Calefón:		
5 (l/min)	11	13
5 (l/min)	20	23
10 (l/min)	26	30
13 (l/min)	30	35
16 (l/min)		
Estufa:		
Mural chica	3	3.5
Mural grande	5	5.8
Ambiental	12	14
Secadora de ropa	17	20

Fuente: (Planificación, 2016)

2.1.1.8 Caudal

Con GLP se trabaja normalmente con caudales expresados en kg/h. Tan solo, cuando se han de calcular las conducciones, se han de expresar los caudales en m³/h. La transformación de la expresión másica a volúmica se realiza dividiendo por la masa en volumen del gas en fase gaseosa.

$$Q_v = \frac{Q_m}{\rho} \quad \text{Ecuación 1}$$

El caudal de gas necesario para obtener la potencia superior (Ps) o aprovechar la potencia útil (Pu), se deduce conociendo el Poder Calorífico superior (Hs) del gas:

$$Q = \frac{P_s}{H_s} \quad \text{Ecuación 2}$$

Caudal de simultaneidad

Cuando existen más de dos aparatos en una instalación individual, el caudal resultante no se puede considerar como la suma de todos ellos, puesto que no es previsible que se utilicen necesariamente a la vez.

Siendo las potencias nominales de los aparatos PA, PB, PC, PD..., los caudales nominales respectivos serán:

$$Q_A = \frac{P_A}{H_s}, Q_B = \frac{P_B}{H_s}, Q_C = \frac{P_C}{H_s}, Q_D = \frac{P_D}{H_s}$$

El caudal resultante se calcula utilizando la siguiente fórmula:

$$Q_{SI} = Q_A + Q_B + \frac{Q_C + Q_D + \dots + Q_N}{2} \quad \text{Ecuación 3}$$

Donde, QSI es el caudal máximo probable o de simultaneidad (en m³/h o en kg/h)

QA y QB son los caudales nominales de los aparatos de mayor consumo (en m³/h o en kg/h).

QC, QD y QN son los caudales nominales del resto de los aparatos (en m³/h o en kg/h).

2.1.2 Sensores de gases y sus características

2.1.2.1 Sensor

Un sensor puede definirse como un dispositivo que convierte magnitudes físicas (luz, presión, temperatura, gasto, etc.) en valores medibles de dicha magnitud, mostrando los cambios en las propiedades medidas en respuesta a una variación de la misma. Estos dispositivos pueden ser mecánicos, eléctricos o en forma de fibra óptica.

Un sensor químico es un dispositivo analítico capaz de detectar determinadas substancias de manera continua y reversible, el cual consta de un transductor acoplado a una fase de reconocimiento o químicamente selectiva. En la zona de reconocimiento del sensor, la información química se transforma en una forma de energía medible por el transductor. Este dispositivo es capaz de transformar la energía que lleva la información química de la muestra en una señal analítica utilizable. La figura 3 indica la estructura básica de funcionamiento de un sensor.

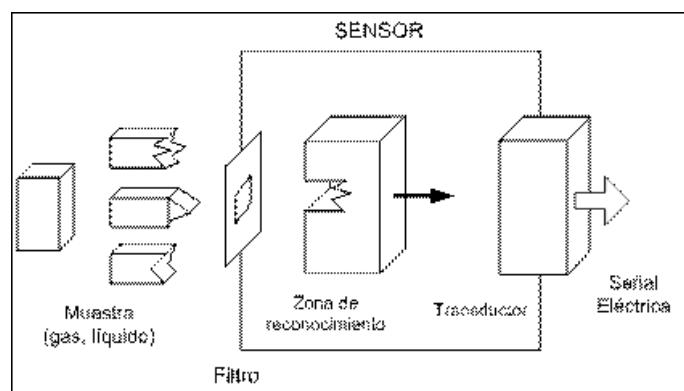


Figura 3 Estructura básica de funcionamiento de un sensor

Fuente: (Casella, 2016)

Considerando el origen de la información química que se genera en el sensor y las características del transductor que lo conforma, es posible establecer una clasificación de los sensores en grupos que se han sido desarrollados en los últimos años.

2.1.2.2 Sensores detectores de gases semiconductores

La medición está basada en variaciones inducidas de conductividad eléctrica, por absorción de gas, en la superficie de un óxido metálico (SnO_2 por ejemplo). Este semiconductor se coloca en la superficie de un substrato (tubo o plato), en la figura 4 se indica la estructura de este semiconductor.

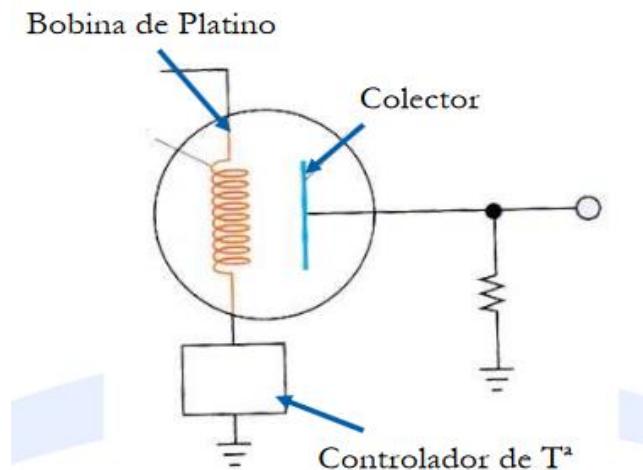


Figura 4 Estructura interna de semiconductor

Fuente: (Casella, 2016)

Principio de Detección

El filamento interno se calienta mediante una corriente eléctrica que conduce, por lo que el sustrato aumenta la temperatura hasta que alcanza entre 300 y 500 °C.

La sensibilidad del SnO_2 a los diferentes gases varía con la temperatura. Por lo tanto, es necesario determinar la temperatura de trabajo para que el sensor funcione con la máxima sensibilidad.

Gases Detectables

Gases tóxicos e inflamables: COV, hidrocarburos (tolueno, xileno...), vapores de hidrocarburos (keroseno...), cetonas (MEK...), esteres, (acetato de metilo, etil éter...), alcoholes (metanol...) y los principales FREONES.

Ventajas y Limitaciones

- Alta sensibilidad.
- Tiempo de vida amplio (~ 5 años)
- Muy sensibles a interferentes y venenos.
- Cambios en la concentración de oxígeno, en la temperatura o en la velocidad del aire pueden tener un efecto significativo en la sensibilidad.

2.1.3 Principios de funcionamiento del sensor de gas Semiconductor

Normalmente óxido metálico es utilizado como material sensor sea el caso del dióxido de estaño SnO_2 . Cuando un cristal de óxido metálico como SnO_2 es calentado a una temperatura elevada, la superficie del cristal absorbe el oxígeno y una carga negativa. Posteriormente los electrones en la superficie del cristal son transferidos al oxígeno absorbido, dando como resultado una carga positiva. De este modo, el potencial en la superficie actúa como una barrera contra el flujo de electrones como se muestra en la figura 5.

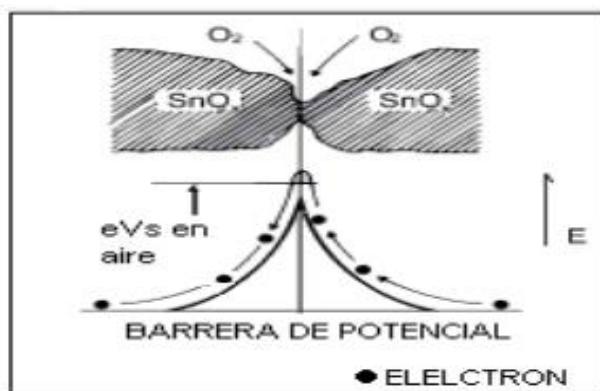


Figura 5 Modelo de barrera de potencial de entre-grano (ausencia de gases)

Fuente: (Villa, 2010)

En la parte interior del sensor la corriente atraviesa la unión de los micro-cristales de óxido de estaño (SnO_2). En la unión del material detector del oxígeno absorbido forma una barrera de potencial la cual proviene que las cargas se muevan con gran libertad. La resistencia eléctrica del sensor es proporcional a esta barrera de potencial. En la presencia de gas reductor, la densidad del oxígeno cargado negativamente disminuye, el cual produce que la altura de la barrera en la unión se disminuya, apreciado en la figura 6.

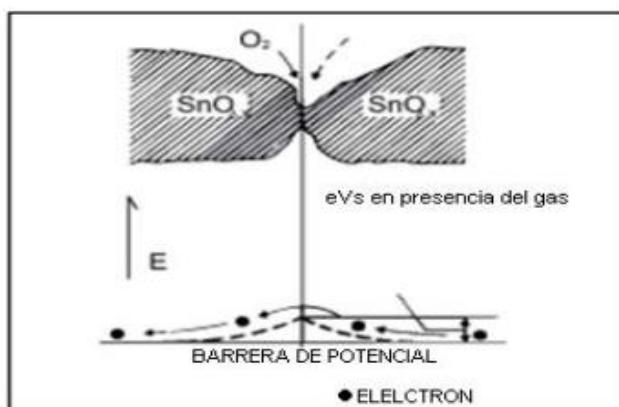


Figura 6 Modelo de barrera de potencial de entre-grano (presencia de gases)

Fuente: (Villa, 2010)

La relación entre la resistencia del sensor y la concentración de gas reductor puede ser expresada por encima de un cierto rango de concentración de gas, con la siguiente ecuación:

$$R_s = A[C]^{-\alpha} \quad (\text{Resistencia con respecto al sensor de gas})$$

Donde:

R_s = resistencia eléctrica del sensor

A = constante

C = concentración de gas

α = pendiente de la curva R_s

En general los sensores de gas pueden detectar gas a bajas y altas concentraciones, siendo capaces de percibir moléculas de todo tipo de gases según sea su función. (Villa, 2010)

2.1.4 Parámetros estáticos

2.1.4.1 Rango

Es el valor máximo que el sensor puede registrar. Por encima de este valor la medida es defectuosa.

2.1.4.2 Precisión (Accuracy)

Tiene que ver con cuán cercana es la medida registrada por el sensor a la medida real. Por lo general en la etapa de calibración se ajusta la precisión del sensor al comparar la medida esperada o real con un equipo confiable antes calibrado.

El error absoluto máximo es considerado como la suma algebraica de todos los errores que influyen en la medición, como lo son; la desviación media cuadrática (MDQ) es una medida para la calidad del ajuste matemático de la respuesta del sensor a una entrada constante. Este es una función de la linealidad del transductor (la proximidad de una curva de calibración a una línea recta especificada) y del procedimiento de calibración (el coeficiente de la escala y la función polinómica utilizada), como se muestra en la figura 7.

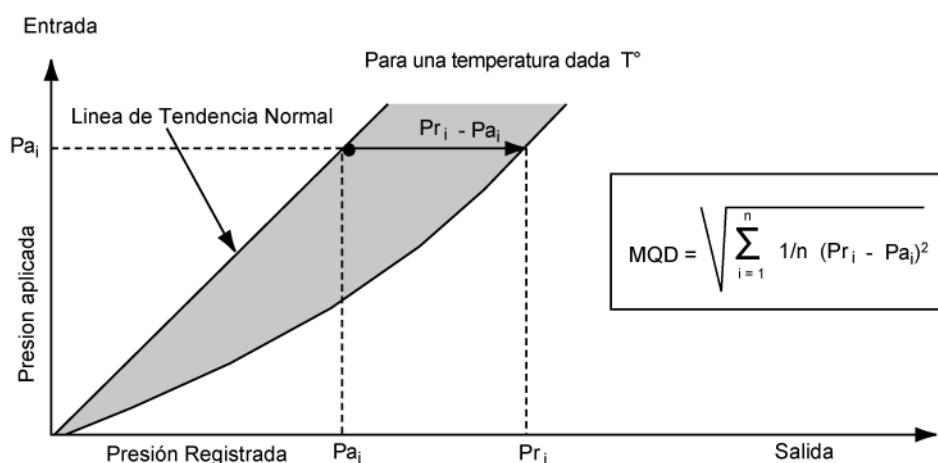


Figura 7 Desviación media cuadrática (MDQ)

Fuente: (Drake, 2015)

2.1.4.3 Histéresis

Es la más alta discrepancia de la señal enviada entre un incremento y decremento de un valor de medida para un mismo nivel de medida, como se muestra en la figura 8.

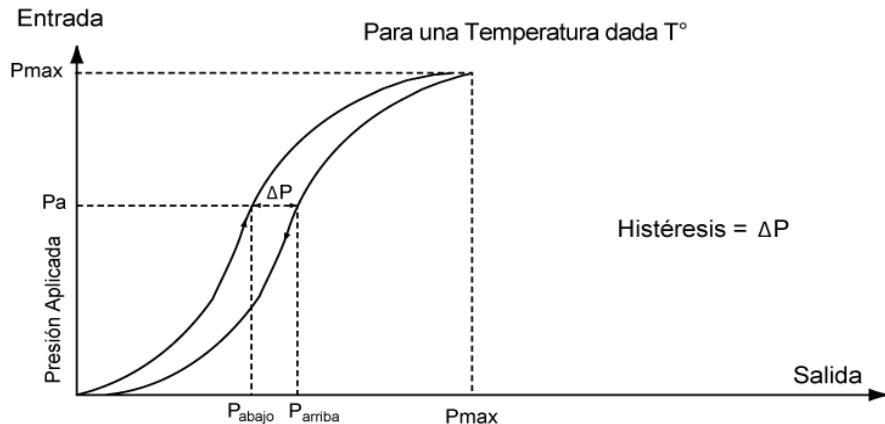


Figura 8 Histéresis

Fuente: (Drake, 2015)

2.1.4.4 Repetibilidad

Se define como la discrepancia entre dos mediciones consecutivas de una determinada variable. La repetición no se ve afectada por la histéresis, ya que las mediciones se realizan utilizando el mismo procedimiento, se muestra en la figura 9.

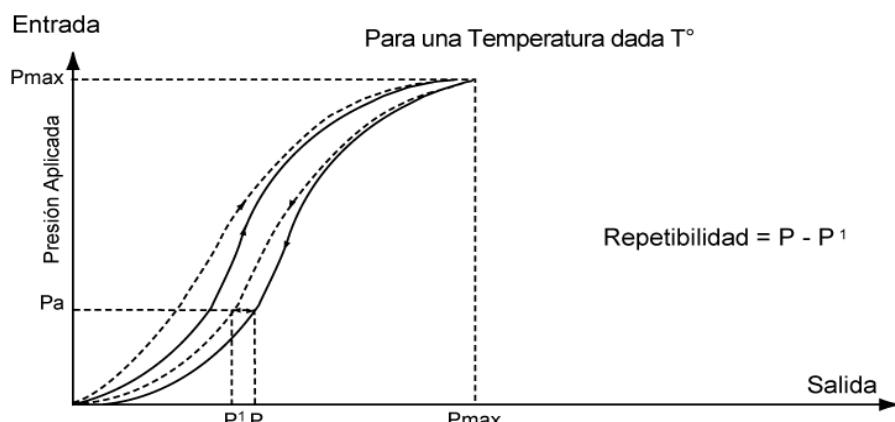


Figura 9 Repetibilidad

Fuente: (Drake, 2015)

2.1.4.5 Resolución

Se refiere al valor mínimo del cambio de variable medida que puede registrar el sensor. Este parámetro depende del sensor y del módulo ADC, para traducir la señal analógica a digital (según el número de bits asignados). En la figura 10 se indica una diferencia entre un sensor de presión con diferente resolución.

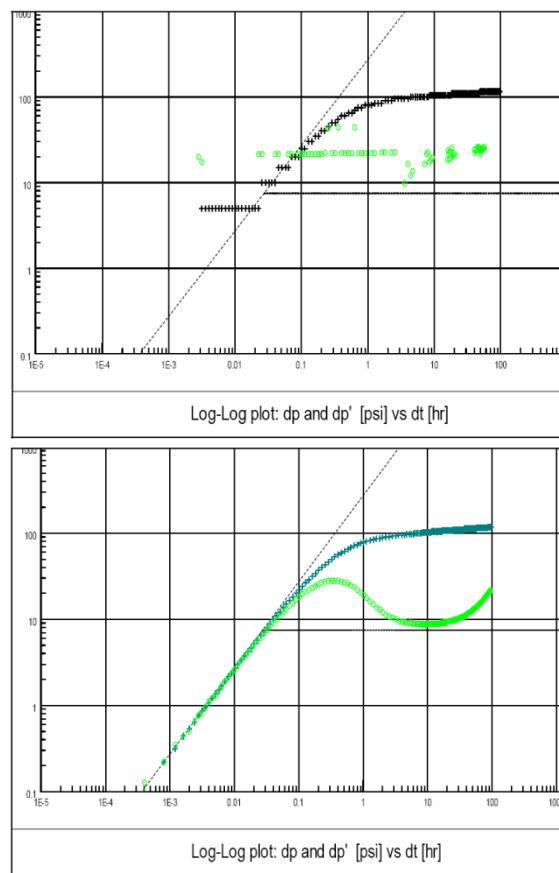


Figura 10 Comparación de resolución

Fuente: (Drake, 2015)

2.1.4.6 Frecuencia de muestreo (Sampling Rate)

Se refiere a la frecuencia de adquisición de los datos medidos. Por ejemplo, un valor tomado cada segundo, es muy común en sistemas de adquisición de tiempo real.

2.1.5 Unidades de medida

2.1.5.1 Cantidad de gas

Se puede medir en unidades de masa, usualmente en gramos. De acuerdo con el sistema de unidades SI, la cantidad también se expresa mediante el número de moles de sustancia, esta puede calcularse dividiendo el peso del gas por su peso molecular. El volumen es el espacio ocupado por un cuerpo, las unidades de volumen:

- $m^3 = 1000$ litros
- Litro = 1000 centímetros cúbicos (c.c)
- 1 c.c = 1 mililitro

Las unidades de medida que se utilizan al estudiar las propiedades de los gases son las siguientes de la tabla 5:

Tabla 5

Unidades de medida de gases

Variable	Unidad
Presión	Atmosfera (atm.)
Temperatura	Kelvin ($^{\circ}$ K)
Cantidad de gas	Moles (n)
Volumen	Litros (L)

Fuente: (Villa, 2010)

2.1.5.2 ppm (partes por millón)

Al igual que el "porcentaje (%)", ppm es una unidad que representa una relación. Mientras que porcentaje significa un número determinado de partes por cada cien partes; ppm se refiere a un número determinado de partes de un millón de partes. La unidad ppm se utiliza para concentraciones más bajas y no depende de la presión ni

de la temperatura. Si están presentes concentraciones mayores, éstas se especifican en porcentaje (%).

Los porcentajes se convierten como sigue:

$$10.000 \text{ ppm} = 1\%$$

$$1.000 \text{ ppm} = 0,1\%$$

$$100 \text{ ppm} = 0,01\%$$

$$10 \text{ ppm} = 0,001\%$$

$$1 \text{ ppm} = 0,0001\%$$

Una concentración de oxígeno de 21% vol. correspondería a una concentración de 210.000 ppm de O₂. (Testo, 2016)

2.2 Marco reglamentario

2.2.1 NTE INEN 2 260:2010 Segunda revisión

INSTALACIONES DE GASES COMBUSTIBLES PARA USO RESIDENCIAL, COMERCIAL E INDUSTRIAL. REQUISITOS.

Esta norma establece los requisitos técnicos y las medidas de seguridad mínimas que se deben practicar al diseñar, construir, ampliar, reformar, revisar y operar las instalaciones receptoras de gases combustibles para uso residencial, comercial e industrial; así como los requerimientos mínimos de los sitios donde se ubiquen los equipos y artefactos que consumen gases combustibles, las condiciones técnicas de su conexión, ensayos de comprobación y su puesta en marcha. (INEN, 2010)

2.2.2 NCh 2245 Of. 2009

TARJETA DE DATOS DE SEGURIDAD SOBRE GAS LICUADO DE PETRÓLEO.

La tarjeta de emergencia contiene información de los componentes del gas y grado de toxicidad del compuesto, los efectos que puede producir en el cuerpo humano ante su exposición y las medidas que se deben tomar ante un accidente o fuga. (INDURA, 2009)

2.2.3 MN-DT-600_A

NOTAS GENERALES PARA LA CALIBRACIÓN, MANTENIMIENTO E INSTALACIÓN DE LOS DETECTORES DE GAS

El manual refiere algunos de los factores que deben tenerse en cuenta para la selección de la posición idónea de los detectores de gas en diferentes escenarios, expresa diferentes variables del ambiente que pueden regir en el comportamiento de las lecturas de gas. (Honeywell, 2016)

2.3 Componentes del hardware

2.3.1 Arduino

Es una plataforma de hardware libre, asentada en una placa con un microcontrolador y un entorno de desarrollo, creada para facilitar el uso de la electrónica en gran variedad de proyectos. Sus creadores son el zaragozano David Cuartielles, ingeniero electrónico y docente de la Universidad de Malmö, Suecia y Massimo Banzi, italiano, diseñador y desarrollador Web. El proyecto fue concebido en Italia en el año 2005.

Hasta principios de 2015 la web oficial de Arduino era www.arduino.cc mantenida por los creadores de Arduino y todo su equipo. En febrero de 2015 se hizo público la ruptura entre los fundadores de Arduino y el fabricante de las placas de Arduino, la división de Arduino en dos partes ha provocado que haya dos tipos de placas con marcas diferentes. En la página arduino.org se encuentra disponibles productos que en el arduino.cc no están y viceversa.

Arduino es una plataforma abierta que facilita la programación de un microcontrolador. Los microcontroladores nos rodean en nuestra vida diaria, usan los sensores para escuchar el mundo físico y los actuadores para interactuar con el mundo físico. Los microcontroladores leen sobre los sensores y escriben sobre los actuadores.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring (lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java) y el cargador de arranque (boot loader) que corre en la placa.

Al ser open-hardware, tanto su diseño como su distribución son libres. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

Con la posibilidad de utilizarlo para cualquier tipo de proyecto, al ser una plataforma que facilita una primera toma de contacto con la electrónica y la programación de microcontroladores, se puede encontrar con limitaciones en cuanto a hardware. No obstante, se pueden utilizar una variedad de módulos para la comunicación y sensores tales como termómetro, higrómetro, barómetros, sensores de proximidad, infrarrojos, etc.



Figura 11 El universo Arduino

Fuente: (Crespo, 2016)

Arduino es una tecnología que tiene una rápida curva de entrada con básicos conocimientos de programación y electrónica, que permite automatizar cualquier cosa, hacer agentes autónomos, desarrollar proyectos en el ámbito de las Smart Cities, el Internet de las cosas, dispositivos wearables, salud, ocio, educación, robótica, etc...

2.3.2 Arduino Genuino Mega 2560 rev3

Arduino Mega a más de mayores puertos I/O, tiene una memoria de mayor capacidad lo que permite utilizarlo para códigos muy extensos o que requieran de una gran cantidad de variables.

El Arduino Mega está basado en el microcontrolador ATMega2560. Tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16 Mhz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conectando a la PC por medio de un cable USB o con una fuente de alimentación externa. El Arduino Mega es compatible con la mayoría de los Shields diseñados para Arduino Duemilanove, diecimila o UNO.

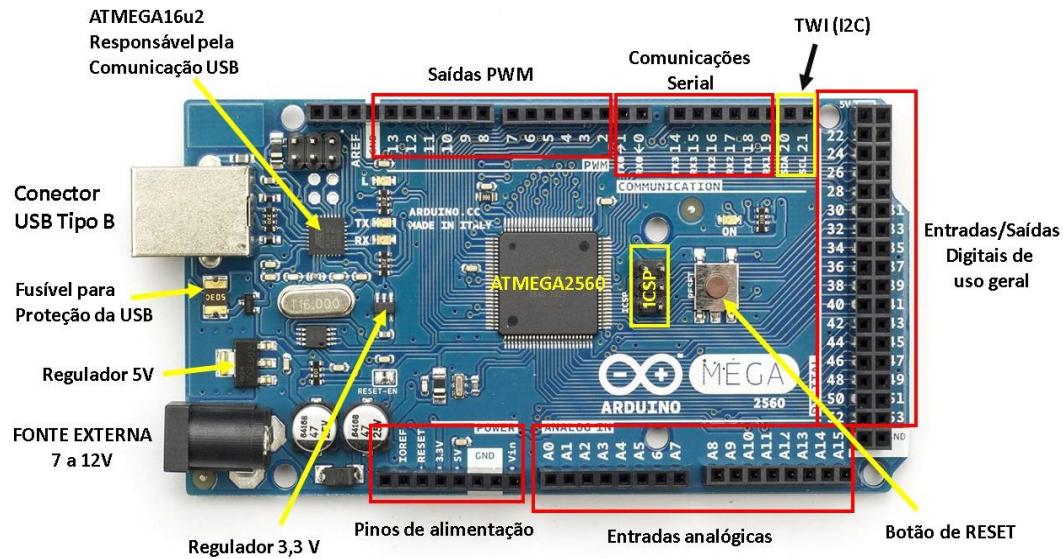


Figura 12 Configuración de Arduino Mega

Fuente: (Crespo, 2016)

Esta nueva versión de Arduino Mega 2560 adicionalmente a todas las características de su sucesor, el Arduino Mega ahora utiliza un microcontrolador ATMega8U2 en vez del chip FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo inf es necesario para Windows) además ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc. (ARDUINO, 2016)

Especificaciones detalladas de Arduino Mega:

Tabla 6
Especificaciones técnicas Arduino Mega

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54
PWM Digital I/O Pins	14
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 kB
Flash Memory for Bootloader	8 kB
SRAM	8 kB
EEPROM	4 kB
Clock Speed	16 MHz

Fuente: (ARDUINO, 2016)

2.3.3 Shield ethernet para Arduino

Las Shields son placas de circuitos modulares que se montan unas encima de otras para dar funcionalidad extra a Arduino. Los Shields son apilables, pero debemos tener en cuenta que cuando instalamos un Shield, habrá que leer su documentación que nos dirá si inhabilita algunas de los pines I/O o si usa un bus y que requisitos tiene para su utilización, así como el software o librería que necesita para su uso. Generalmente las Shields tienen un ejemplo o una librería para su uso.

El Arduino Ethernet Shield permite conectar el Arduino a Internet. Está basado en el chip Wiznet W5100 con un buffer interno de 16 kb que es el encargado de dar una dirección IP al Arduino con soporte de los protocolos TCP y UDP. Soporta hasta cuatro conexiones de socket simultáneas. Tiene un conector Ethernet RJ-45 con una velocidad de 10/100Mb.

Se usa la librería <Ethernet.h> para escribir el programa. También tiene un slot de tarjetas micro-SD para almacenar datos para servir a la red. Para esto, se usa la librería <SD.h>. Para comunicarse con la placa Arduino.

Utiliza el puerto SPI (pins 50, 51 ,52 y 53) con la librería <SPI.h>, y el pin 4 si trabaja con una tarjeta micro-SD. Estos pins no pueden ser usados como entradas o salidas digitales.

También tiene un botón de Reset que resetea tanto la placa Arduino como el W5100. Gracias a la Ethernet Shield se puede usar el Arduino como servidor web o como cliente web.

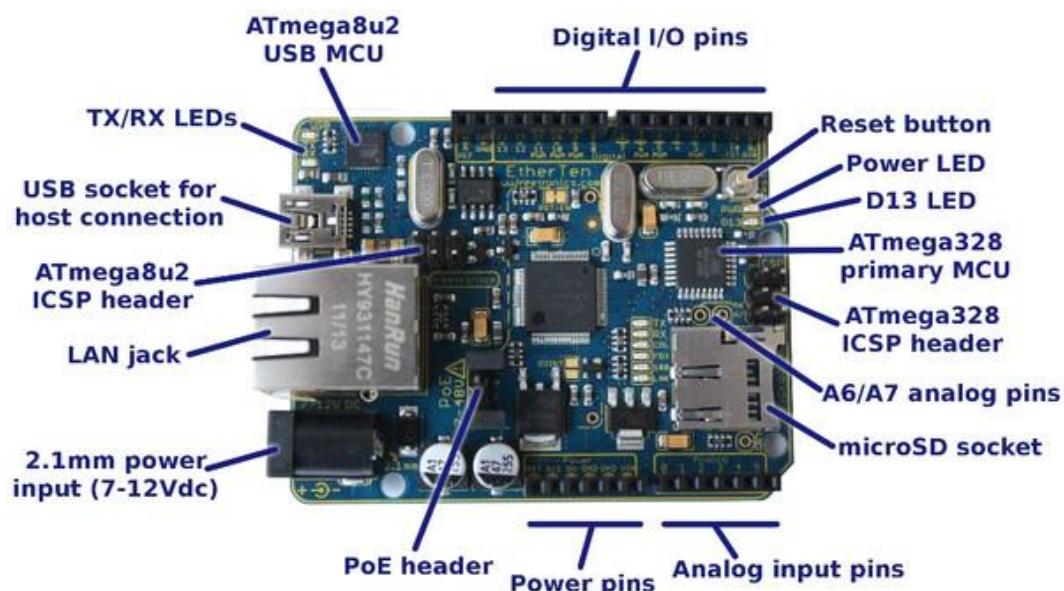


Figura 13 Shield Ethernet

Fuente: (FjRamire, 2016)

El Shield actual tiene una alimentación a través de Ethernet (PoE), módulo diseñado para extraer energía de una convencional de par trenzado Categoría 5 cable Ethernet:IEEE802.3af compliant. El Shield dispone de un conector RJ45 para conectarse a una red Ethernet. (FjRamire, 2016)

El Shield contiene un número de Leds informativos:

- PWR: indica que Arduino y el Shield están alimentados
- LINK: indica la conexión a una red y parpadea cuando el Shield transmite o recibe datos
- FULLD: indica que la conexión de red es full duplex
- 100M: indica la presencia de una conexión de red 100 Mb/s (en lugar de 10 Mb/s)
- RX: parpadea cuando el Shield recibe datos
- TX: parpadea cuando el Shield envía datos
- COLL: parpadea cuando se detectan colisiones de red

2.3.4 Sensor de gas GLP MQ-6

Es un sensor que varía su resistencia en función de la absorción de gas en la superficie de un óxido calentado sobre un semiconductor (Si).

Se utilizan en equipos de detección de fugas de gas en la familia y la industria, son adecuados para la detección de GLP, mayormente compuesto por propano y butano. Evita el ruido de los gases de alcohol, de cocina y el humo del cigarrillo. (Datasheet, 2016)

Características:

- El MQ-6 detecta concentraciones de gas dentro de las 200 a 10000 ppm.
- Presenta una resolución de 0.196 ohm/ppm.
- Mantiene un rango de resistencia de 10k – 60 k ohm
- Resistencia de carga 2k – 47 k ohm, Ideal 20 k ohm
- Alimentación 5.0 +/- 0.2 V CA o CD
- Concentración de oxígeno al 21% afectaría la sensibilidad, valor mínimo 2 %.
- Temperatura de funcionamiento (- 10 C) – 50 C, ideal 20 C +/- 2 C
- Humedad 65% +/- 5%

La salida del sensor se da por la ecuación:

$$V_0 = (R_s * V_s) / (R_s + RL) \quad Ecuación\ 4$$

$V_{o_min} = 0.83\text{ V}$

$V_{o_max} = 4.83\text{ V}$

R_s = Resistencia del sensor

V_s = Alimentación

RL = Resistencia de carga

V_o = Voltaje de salida

Circuito estándar de uso:

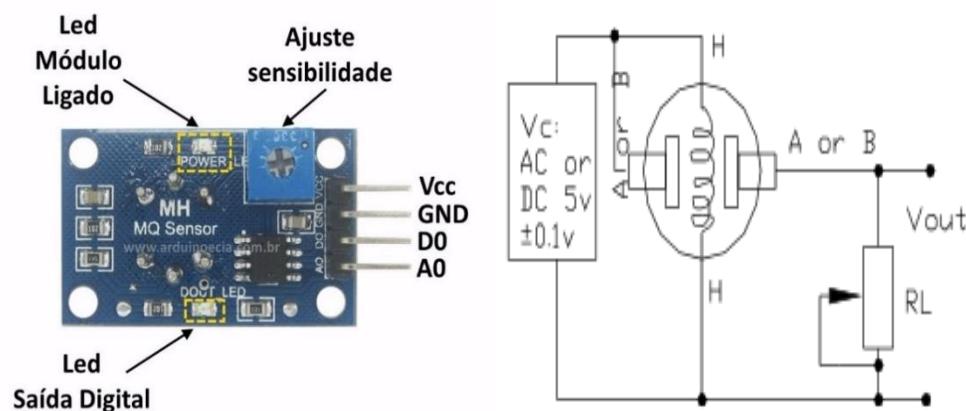


Figura 14 Sensor de gas GLP Mq-6

Fuente: (Datasheet, 2016)

Características de sensibilidad, trabajando bajo los parametros ideales.

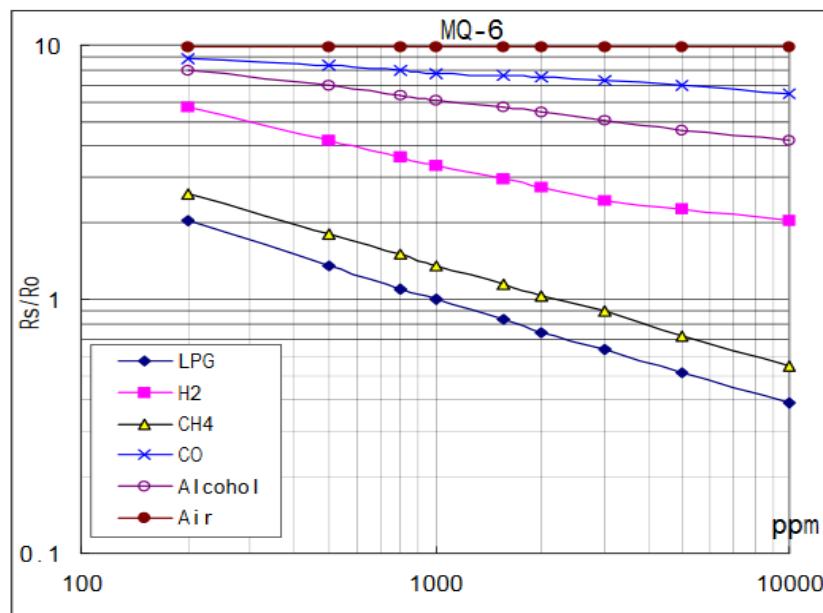


Figura 15 Características de sensibilidad de MQ-6

Fuente: (Datasheet, 2016)

Ajuste de sensibilidad

El valor de la resistencia de MQ-6 es la diferencia de diversos tipos y diversos gases de concentración. Por lo tanto, cuando se usa este ajuste de los componentes, la sensibilidad es muy necesaria, se recomienda calibrar el detector de 1000 ppm de concentración de GLP en el valor del aire y el uso de la resistencia de carga (RL) sobre 20kΩ (1 a 47 kΩ).

Al medir con precisión, el punto de alarma adecuado para el detector de gas debe ser determinado después de tener en cuenta la influencia de la temperatura y la humedad.

2.4 Software para desarrollo

2.4.1 Arduino y IOT

2.4.1.1 Que es IOT

La internet de las cosas (en inglés Internet of things, abreviado IoT) es esencialmente un sistema de máquinas u objetos equipados con tecnologías de recopilación de datos, de manera que esos objetos pueden comunicarse entre sí.

IoT es un nuevo paradigma que está ganando terreno en el escenario de las telecomunicaciones inalámbricas modernas. De hecho, ha sido clasificado por el Consejo de Inteligencia de los Estados Unidos como una de las 6 tecnologías civiles disruptivas con impacto sobre la soberanía mundial estadounidense.

Básicamente, IoT consiste en una “evolución” de lo que hasta ahora se conocía como comunicación “Machine-to-machine” (M2M), expandiéndolo hacia sistemas “Machine-to-Infrastructure” e incorporando otros conceptos como; La comunicación entre redes de diferente naturaleza mediante uso de protocolos de comunicación estándar y acceso de la información a través de la “nube”, donde podrá ser tratada a nivel de aplicación.

IoT persigue que, una gran variedad de cosas u objetos (sistemas de identificación por RFID, sensores, actuadores, smartphones, etc.) sean capaces de interactuar y cooperar con los demás para alcanzar objetivos comunes. De esta forma, todas estas soluciones dejarían de ser aplicaciones “stand-alone” pasando a convertirse en aplicaciones “en red”.

Una aplicación completamente desarrollada bajo IoT, se desenvuelve en tres diferentes niveles: la electrónica, las comunicaciones y el software expresado en la figura 16.



Figura 16 Aplicación de Internet de las Cosas

Fuente: (Ubidots, 2016)

Arduino es un elemento que nos permite de forma sencilla y económica conectar cualquier cosa a Internet. Con un Arduino y un sencillo módulo Ethernet o Wifi podemos conectar a Internet sensores para informar, controlar motores o bombillas desde cualquier parte del mundo o mandar un SMS o email cada vez que se abra la puerta de casa.

En muy poco tiempo se podrá conectar todo a la red. El principal inconveniente que está retrasando este avance es la limitación de direccionamiento que ofrece el actual protocolo ipv4, donde no sería posible interconectar tal cantidad de dispositivos y objetos. Por este motivo es ya inminente la transición a ipv6, el nuevo protocolo de direccionamiento que permitirá la interconexión casi ilimitada de cualquier objeto y la evolución e implementación del Internet of Things.

Una vez esquivado dicho obstáculo, y con el progresivo abaratamiento de esta tecnología con el tiempo, no solo va a ser posible, si no necesario debido a su utilidad y la gran cantidad de aplicaciones que podremos realizar con el Internet de las cosas.

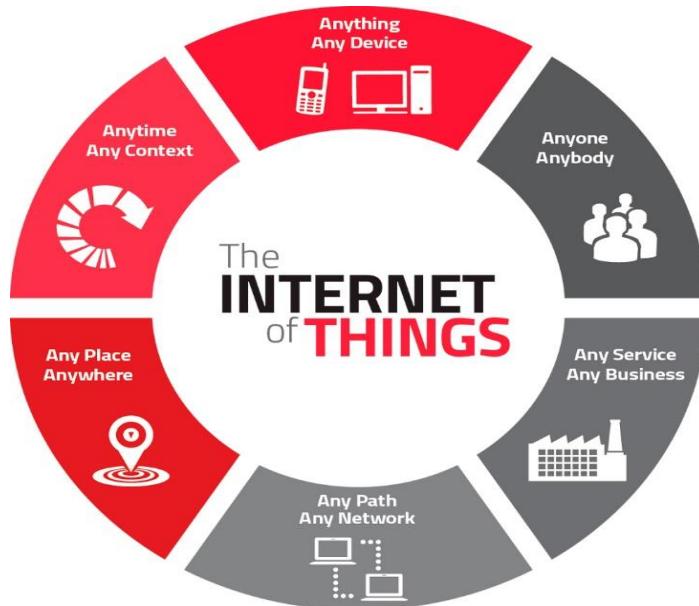


Figura 17 Plataformas del Internet de las cosas

Fuente: (Crespo, 2016)

2.4.1.2 Internet de las Cosas con Ubidots

Ubidots es un servicio en la nube que permite almacenar e interpretar información de hardware en tiempo real, permitiendo la creación de aplicaciones para el Internet de las Cosas de una manera fácil, rápida y divertida.

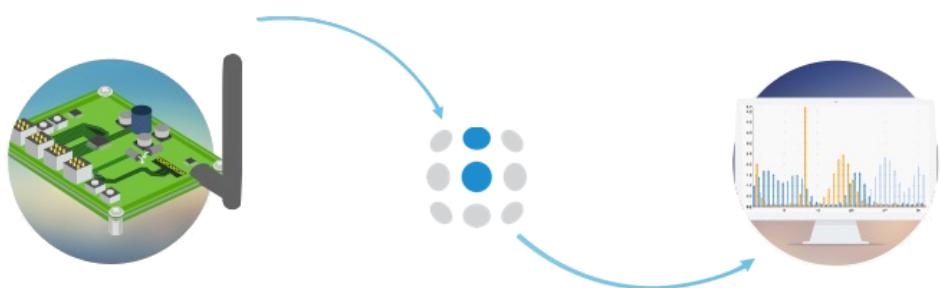


Figura 18 Internet de las Cosas con Ubidots

Fuente: (Ubidots, 2016)

Gracias a ésta herramienta, se ahorra tiempo y dinero al momento de desarrollar aplicaciones como sistemas de telemetría GPS, sistemas para monitoreo de temperatura, aplicaciones para contar vehículos en una calle, etc.

Del mismo modo que las personas utilizan un navegador para visitar páginas web a través de URLs, botones y campos de texto, los dispositivos electrónicos tienen su propia forma de interactuar con los sistemas web. Puesto que los dispositivos no son tan inteligentes como los humanos, ellos necesitan una “página web especial”, con una estructura estandarizada y unos comandos preestablecidos. Éstas “páginas web para dispositivos” son lo que se conoce como un API REST.

“API” significa “Interfaz de Programación de Aplicaciones” (Application Programming Interface) y define cómo diferentes componentes de un sistema de software deben interactuar entre ellos. API en esta aplicación específica la interacción entre los dispositivos y Ubidots.

El servicio utiliza el protocolo HTTP, el mismo protocolo que usado por los navegadores para comunicarse con las páginas web. El API de Ubidots implementa los principales cuatro métodos de HTTP:

- GET Para la lectura de información
- POST Para la creación de información
- PUT Para la edición de información
- DELETE Para el eliminado de información

Como último y no menos importante, un API también debe describir el formato de los datos. Los más comunes son XML y JSON, siendo éste último el más popular, en gran parte por su simplicidad. En comparativa se indica una misma estructura de datos representada en ambos formatos como lo indica la figura 19. (Ubidots, 2016)

XML	JSON
<pre><item> <value>1000</value> <context> <estado>encendido</estado> </context> </item></pre>	<pre>{ "value": 1000, "context": { "estado": "encendido" } }</pre>

Figura 19 Formato de los datos HTML

Fuente: (Ubidots, 2016)

2.4.2 Principios básicos de redes

2.4.2.1 HTML

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), se refiere al lenguaje de marcado para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código propio (denominado código HTML) para la creación de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

El HTML se escribe en forma de «etiquetas», su contenido se rodea por corchetes angulares (<,>,/); siguiendo la estructura general de la figura 20. El HTML también puede describir de forma limitada la apariencia de un documento, y puede incluir o hacer referencia a un tipo de programa llamado script, quien influye en el comportamiento de los navegadores web y otros procesadores de HTML.



Figura 20 Estructura general de una línea de código con etiquetas HTML

Fuente: (Wikipedia, 2016)

HTML utiliza etiquetas o marcas, que expresan el comienzo y final, en las cuales se determinan el orden que debe aparecer en el navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

Toda etiqueta se identifica porque está encerrada entre los signos menor que y mayor que (<>), y algunas adoptan atributos que pueden tomar algún valor. En general las etiquetas se aplicarán de dos formas especiales:

- Se abren y se cierran, como por ejemplo: negrita, que se vería en su navegador web como negrita.
- No pueden abrirse y cerrarse, como <hr />, que se vería en su navegador web como una línea horizontal.
- Otras que pueden abrirse y cerrarse, como por ejemplo <p>.
- Las etiquetas básicas o mínimas son:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Ejemplo1</title>
  </head>
  <body>
    <p>Párrafo de ejemplo</p>
  </body>
</html>
```

Figura 21 Formato etiqueta HTML

Fuente: (Wikipedia, 2016)

2.4.2.2 Ethernet

Ethernet, es un estándar referente a redes de área local para hardware como computadores con acceso al medio por detección de la onda portadora y con detección de colisiones (CSMA/CD).

Hace ya mucho tiempo que Ethernet consiguió situarse como el principal protocolo del nivel de enlace. Ethernet 10Base2 consiguió, ya en la década de los 90s, una gran aceptación en el sector. Hoy por hoy, 10Base2 se considera como una "tecnología de legado" respecto a 100BaseT. Hoy los fabricantes ya han desarrollado adaptadores capaces de trabajar tanto con la tecnología 10baseT como la 100BaseT y esto ayuda a una mejor adaptación y transición.

Las tecnologías Ethernet que existen se diferencian en estos conceptos entre ellos:

- Velocidad de transmisión_ Velocidad a la que transmite la tecnología.

- Tipo de cable_ Tecnología del nivel físico que usa la tecnología.
- Longitud máxima_ Distancia máxima que puede haber entre dos nodos adyacentes (sin estaciones repetidoras).
- Topología_ Determina la forma física de la red.

A continuación se especifican los anteriores conceptos en las tecnologías más importantes:

Tabla 7

Tecnologías Ethernet

Tecnología	Velocidad de transmisión	Tipo de cable	Distancia máxima	Topología
10Base2	10 Mbit/s	Coaxial	185 m	Bus (Conector T)
10BaseT	10 Mbit/s	Par Trenzado	100 m	Estrella (Hub o Switch)
10BaseF	10 Mbit/s	Fibra óptica	2000 m	Estrella (Hub o Switch)
100BaseT4	100 Mbit/s	Par Trenzado (categoría 3UTP)	100 m	Estrella. Half Duplex (hub) y Full Duplex (switch)
100BaseTX	100 Mbit/s	Par Trenzado (categoría 5UTP)	100 m	Estrella. Half Duplex (hub) y Full Duplex (switch)
100BaseFX	100 Mbit/s	Fibra óptica	2000 m	No permite el uso de hubs
1000BaseT	1000 Mbit/s	4 pares trenzado (categoría 5e ó 6UTP)	100 m	Estrella. Full Duplex (switch)
1000BaseSX	1000 Mbit/s	Fibra óptica (multimodo)	550 m	Estrella. Full Duplex (switch)
1000BaseLX	1000 Mbit/s	Fibra óptica (monomodo)	5000 m	Estrella. Full Duplex (switch)

Fuente: (Parziale, 2011)

2.4.2.3 Dirección IP

Una dirección IP es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red del protocolo TCP/IP.

A través de internet, los ordenadores se conectan entre sí mediante sus respectivas direcciones IP. Sin embargo, a los seres humanos nos es más cómodo utilizar otra notación más fácil de recordar y utilizar, como los nombres de dominio; la traducción entre unos y otros se resuelve mediante los servidores de nombres de dominio DNS.

En su versión 6.55, una dirección IP se implementa con un número de 32 bits que suele ser mostrado en cuatro grupos de números decimales de 8 bits (IPv4). Cada uno de esos números se mueve en un rango de 0 a 255 (expresado en decimal), o de 0 a FF (en hexadecimal), o de 0 a 11111111 (en binario). Las direcciones IP se pueden expresar como números de notación decimal: se dividen los 32 bits de la dirección en cuatro octetos.

El valor decimal de cada octeto puede ser entre 0 y 255 (el número binario de 8 bits más alto es 11111111 y esos bits, de derecha a izquierda, tienen valores decimales de 1, 2, 4, 8, 16, 32, 64 y 128, lo que suma 255 en total).

En la expresión de direcciones IPv4 en decimal se separa cada octeto por un carácter “.”. Cada uno de estos octetos puede estar comprendido entre 0 y 255, salvo algunas excepciones. Los ceros iniciales, si los hubiera, se pueden obviar.

Una dirección IP fija es una IP asignada por el usuario de manera manual, que es lo que se realizará en este proyecto.

2.4.2.4 Dirección MAC

En redes de computadoras la dirección MAC (Media Access Control Address o dirección de control de acceso al medio) es un identificador de 48 bits (6 bytes) que corresponde de forma única a una tarjeta o interfaz de red. Es individual, cada dispositivo tiene su propia dirección MAC determinada y configurada por el IEEE (los últimos 24 bits) y el fabricante (los primeros 24 bits) utilizando el OUI. La mayoría de los protocolos que trabajan en la capa 2 del modelo OSI usan una de las tres numeraciones manejadas por el IEEE: MAC-48, EUI-48 y EUI-64 las cuales han sido diseñadas para ser identificadas globalmente únicos. No todos los protocolos de comunicación usan direcciones MAC, y no todos los protocolos requieren identificadores globalmente únicos.

Las direcciones MAC son únicas a nivel mundial, puesto que son escritas directamente, en forma binaria, en el hardware en su momento de fabricación. Debido a esto, las direcciones MAC son a veces llamadas “Direcciones Quemadas Dentro” (BIA, por las siglas Burned-in Address).

La dirección Mac es un número único de 48 bits asignado a cada tarjeta de red. Se conoce también como la dirección física en cuanto identifica dispositivos de red.

La dirección MAC es utilizada en varias tecnologías entre las que se incluye:

- a) Ethernet.
- b) 802.5 o redes en anillo a 4 Mbps o 16 Mbps Token Ring.
- c) 802.11 redes inalámbricas (WIFI).
- d) ATM

En este proyecto se asignará la dirección MAC de una manera directa.

2.4.2.5 Protocolo TCP y UDP

Los dos protocolos más comunes de la capa de Transporte del conjunto de protocolos TCP/IP son el Protocolo de control de transmisión (TCP) y el Protocolo de datagramas de usuario (UDP). Ambos protocolos gestionan la comunicación de múltiples aplicaciones

En general, las diferencias en cómo entregan los datos UDP y TCP son sencillas, TCP transmite datos asegurándose que el canal está disponible y asegura la entrega, mientras UDP transmite datos de manera simultánea en pequeñas cantidades sin asegurar su recepción. Debido a su menor control es utilizado en la mayoría de aplicaciones de tiempo real.

En la tabla 8 se comparan las diferencias en el modo de administrar la comunicación TCP/IP según se utilice UDP o TCP para el transporte de datos.

Tabla 8

Diferencias entre UDP y TCP

UDP	TCP
Servicio sin conexión; no establece una sesión entre los hosts.	Servicio orientado a la conexión; se establece una sesión entre los hosts.
UDP no garantiza ni confirma la entrega, y no secuencia los datos.	TCP garantiza la entrega mediante el uso de confirmaciones y la entrega secuenciada de datos.
Los programas que utilizan UDP son responsables de proporcionar la confiabilidad necesaria para el transporte de datos.	Los programas que utilizan TCP proporcionan la seguridad del transporte de datos confiable.
UDP es rápido, tiene requisitos de carga pequeños y puede admitir la comunicación punto a punto y de un punto a varios puntos.	TCP es más lento, tiene requisitos de carga mayores y sólo admite la comunicación punto a punto.

Fuente: (Parziale, 2011)

UDP y TCP utilizan puertos para identificar las comunicaciones para cada programa TCP/IP.

El campo de puerto tiene una longitud de 16 bits, lo que permite un rango que va desde 0 a 65535, pero no todos los puertos son de libre uso por lo que se detalla algunas normas sobre ellos:

- El puerto 0 es un puerto reservado, pero es un puerto permitido si el emisor no espera recibir mensajes del receptor.
- Los puertos que van del 1 al 1023 reciben el nombre de puertos “bien conocidos”, y en sistemas operativos tipo Unix es necesario acceder como súper usuario para enlazarse con uno de estos puertos.
- Los puertos que van desde 1024 al 49151 son los llamados puertos “registrados” y son de libre utilización.
- Los puertos del 49152 al 65535 son puertos “efímeros”, de tipo temporal y se utilizan sobre todo por los clientes al conectarse con el servidor.

2.4.2.6 Web server

Un servidor web es una computadora que guarda y transmite datos vía Internet. La información que almacenan es en forma de página Web.

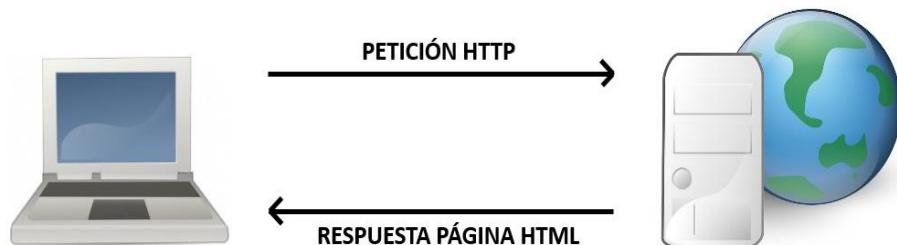


Figura 22 Interacción cliente - servidor WEB

Fuente: (Crespo, 2016)

Cuando un cliente (navegador Web) accede a un servidor Web, envía una petición HTTP que recibe el servidor Web. Luego este envía la información a través de protocolo HTTP al cliente en formato HTML.

La petición HTTP que envía el cliente al servidor sería algo parecido al contenido de la figura 23.

```
GET / HTTP/1.1
Host: 192.168.1.100 (dirección IP del servidor)
Accept: text/html
User-Agent: Mozilla/5.0 (Windows) (Navegador y sistema operativo)
[Línea en blanco]
```

Figura 23 Petición HTTP

Fuente: (Crespo, 2016)

Y la respuesta del servidor al cliente sería similar al contenido de la figura 24.

```
HTTP/1.1 200 OK
Date: Sun, 25 Des 2014 17:42:10 GMT
Content-Type: text/html
<html>
  <!-- código HTML -->
</html>
```

Figura 24 Respuesta HTTP

Fuente: (Crespo, 2016)

En resumen, Arduino está conectado a la red con una IP. El código HTML de la web que enviará lo escribiremos en el sketch. Cuando se escribe esta dirección IP en un buscador Web de un ordenador conectado a la misma red (o desde fuera a través de Internet si al abrir un puerto que direccione a la IP del Arduino), está enviando una petición HTTP. Cuando el Arduino lea esta petición, enviará el código HTML para poder visualizar la página Web en el navegador, se indica la idea en la figura 25.

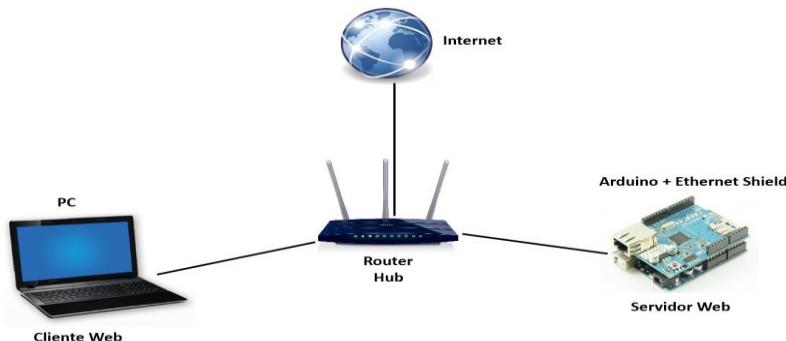


Figura 25 Topología de red cliente - servidor WEB

2.4.2.7 Web server embebido en Arduino

Arduino es un elemento que nos permite de forma sencilla y económica conectar cualquier cosa a Internet. Con un Arduino y un sencillo módulo Ethernet o Wifi podemos conectar a Internet sensores para informar, controlar motores o bombillas desde cualquier parte del mundo o mandar un SMS o email cada vez que se abra la puerta de casa.

2.4.2.8 Webserver con Ajax

Mediante Ajax podemos actualizar los datos de la web embebida en Arduino sin necesidad de cargar toda la web, sino solo mandando los datos actualizados, economizando los datos mandados a través de la red.

En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML".

Ajax es la composición de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.

- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

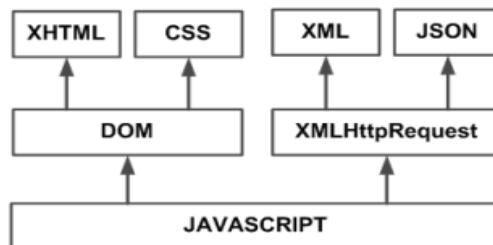


Figura 26 Tecnologías agrupadas bajo el concepto de AJAX

Fuente: (Crespo, 2016)

El desarrollar de aplicaciones AJAX requiere tener un conocimiento avanzado de todas y cada una de las tecnologías mencionadas.

En las aplicaciones web tradicionales, cada acción del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. El servidor devuelve una nueva página HTML una vez procesada la petición del usuario.

En la figura 27, el modelo tradicional de las aplicaciones web se ilustra en la imagen de la izquierda. La imagen de la derecha indica el nuevo modelo propuesto por AJAX.

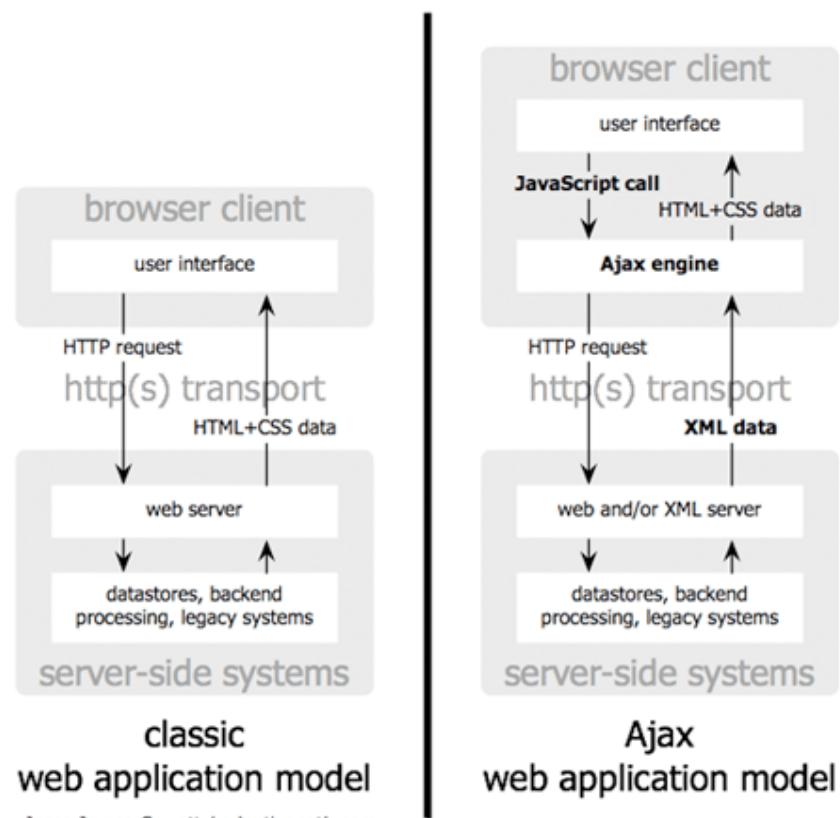


Figura 27 Diferencia entre modelos de aplicación WEB

Fuente: (Crespo, 2016)

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

2.4.3 Entorno de programación de Arduino (IDE)

IDE_ entorno de desarrollo integrado, llamado IDE (sigla en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación, que ha sido empaquetado como un programa de aplicación, es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además en el caso de Arduino

incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware a través del puerto serie.

Arduino facilita un IDE de desarrollo bastante fácil de utilizar además de amigable. Este IDE se lo puede encontrar para MAC OS, Linux y Windows. La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el popular lenguaje de programación de alto nivel Processing.

Los programas de Arduino están compuestos por un solo fichero con extensión “ino”, aunque es posible organizarlo en varios ficheros. El fichero principal siempre debe estar en una carpeta con el mismo nombre que el fichero.

El lenguaje de programación de Arduino está basado en C++ y aunque la referencia para el lenguaje de programación de Arduino se la puede consultar en la página oficial de la plataforma (www.arduino.cc/), también es posible usar comandos estándar de C++ en la programación de Arduino referente a librerías, funciones, estructuras de control, operadores, tipos de datos, casting, arrays, strings, etc.

GCC es un conjunto de compiladores que se considera el estándar para los Sistemas Operativos derivados de UNIX y requiere de un conjunto de aplicaciones conocidas como binutils que son unas herramientas de programación para la manipulación de código de objeto.

GCC está construido para ejecutarse en un sistema como Linux, Windows o Mac OS y generar código para un microcontrolador AVR, entonces se denomina avr-gcc, compilador que usa el IDE de Arduino para convertir el sketch en C++ a un fichero binario (.hex) que es el que se carga en la flash del MCU y que ejecuta.

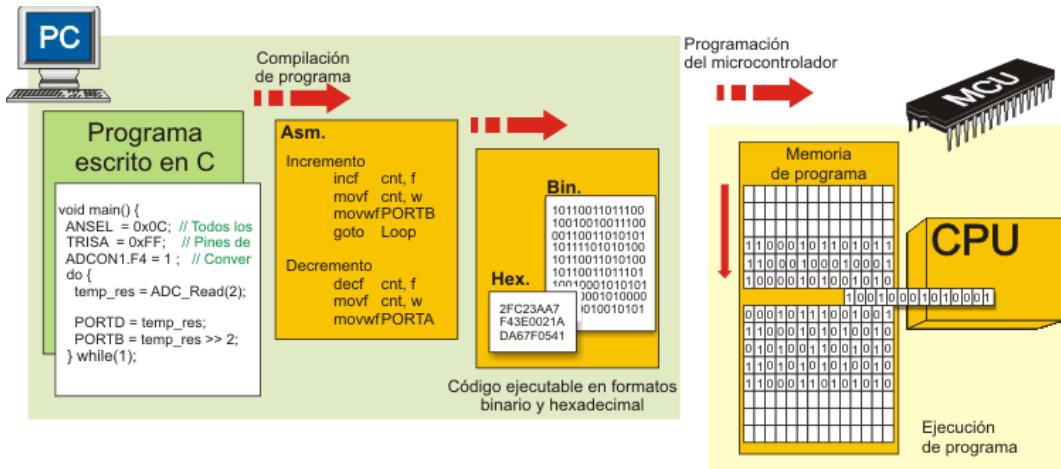


Figura 28 Secuencia de programación en el microcontrolador

Fuente: (Crespo, 2016)

2.4.4 Plataforma móvil Android

Las plataformas móviles son la base para cualquier dispositivo móvil. Proporcionan el SDK, herramientas y el sistema operativo que permiten el desarrollo de aplicaciones para esa plataforma. Por lo general, las propias plataformas tienen su propio modelo de distribución.

Android es una plataforma móvil lanzado por la Open Handset Alliance, que consisten en una pila de software compuesto por un sistema operativo, middleware y la clave de AP-8.

El desarrollo del software está ligado a la utilización de la máquina virtual Dalvik (Android Runtime) que permite el uso de Java como lenguaje de programación. La mayoría de las bibliotecas que son compatibles con JDK pueden ser desplegadas en un dispositivo Android. Sin embargo, algunos de ellos pueden presentar problemas de compatibilidad con el compilador, y por lo tanto no son capaces de ejecutarse.

La arquitectura de Android se compone de varias capas que se basan en un núcleo Linux como se muestra en la figura 29. La capa de aplicación incluye un conjunto de aplicaciones por defecto en el sistema operativo, como el calendario, contactos, etc. El marco de aplicación constará de los servicios predeterminados para la gestión de

los recursos de hardware (sensores, pantalla, etc.), software (alarmas, servicios de fondo) y la integración con recursos externos (sistemas de información de ubicación, notificación de servicios, etc.) (Menéndez, 2016)

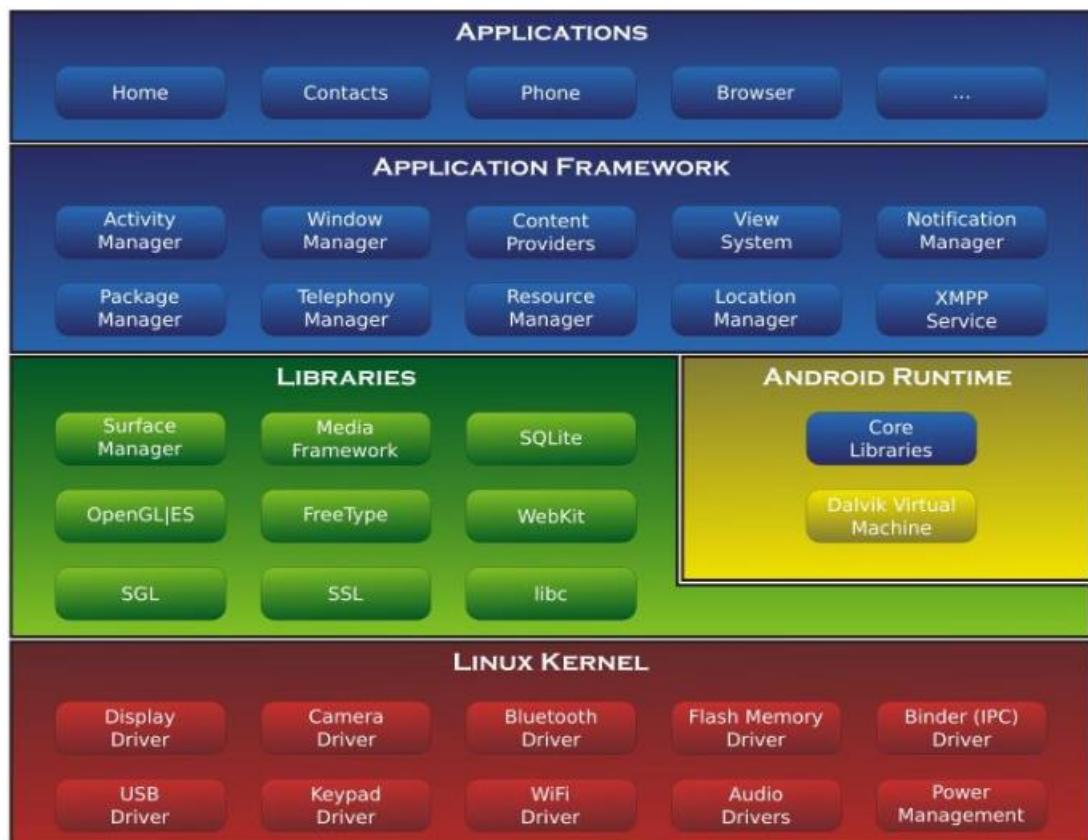


Figura 29 Arquitectura Android

Fuente: (Menéndez, 2016)

- Framework de las aplicaciones: Acceso al API para rehusar componentes o modificarlos.
- Aplicaciones: cualquier tipo de aplicación escrita en Java.
- Bibliotecas en C/C++: el desarrollador puede usarlas a través del Framework.
- Runtime de Android: bibliotecas del lenguaje Java y única instancia en la máquina virtual Dalvik.
- Núcleo Linux: Capa de abstracción del hardware y servicios de seguridad, gestión de memoria, de procesos, pila de red, modelo de los controladores, etc.

Cada uno de los componentes básicos de Android tiene un ciclo de vida bien definido, esto implica que el desarrollador puede controlar en cada momento en qué estado se encuentra dicho componente, pudiendo así programar las acciones.

El componente Activity, probablemente el más importante tiene el ciclo de vida que muestra la figura 30.

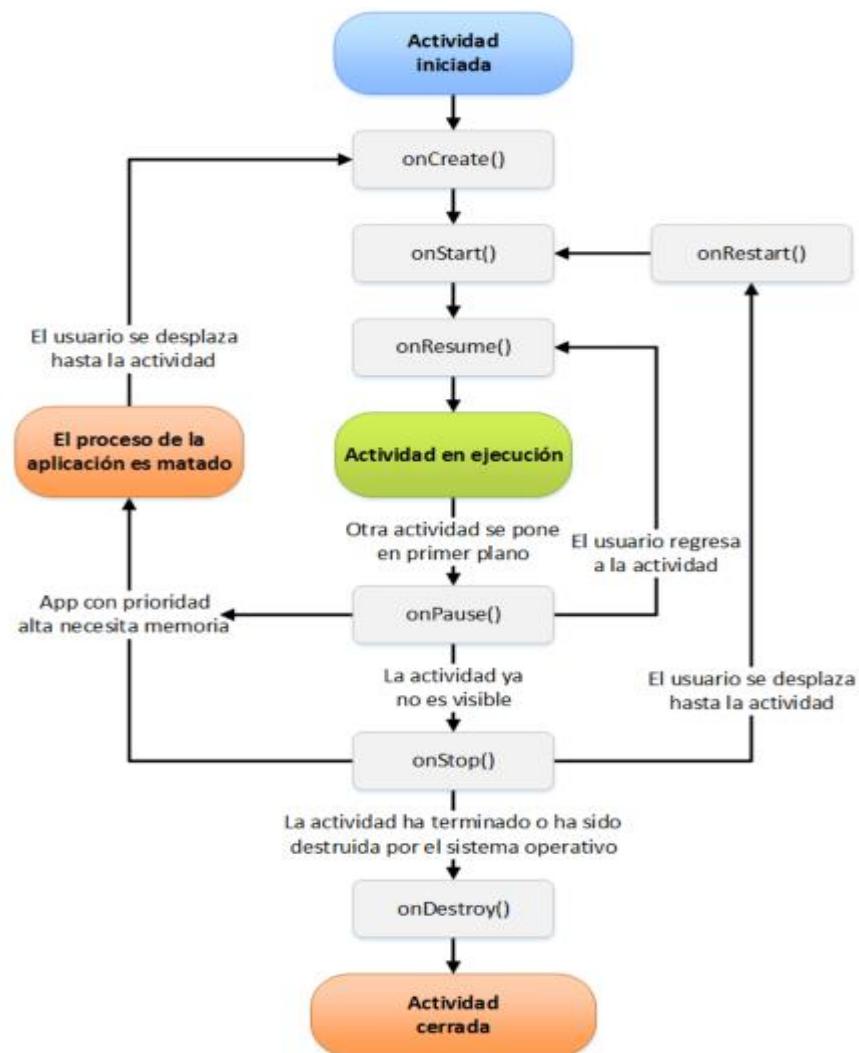


Figura 30 Ciclo de vida de un Activity

Fuente: (Menéndez, 2016)

Se puede observar que entre los principales eventos que se tienen son:

- `OnCreate()`, `onDestroy()`: Abarca todo el ciclo de vida. Cada uno de estos métodos representan el principio y el fin de la actividad.
- `onStart()` hasta `onStop()`: Representan la parte visible del ciclo de vida. Desde `onStart()` hasta `onStop()`, la actividad será visible para el usuario.
- `onResume()`, `onPause()`: Delimitan la parte útil del ciclo de vida. Desde `onResume()` hasta `onPause()`, la actividad no es visible.

2.4.4.1 Android Studio

Android Studio es un entorno de desarrollo integrado para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains, y es publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux. Los requerimientos del sistema se muestran en la tabla 9.

Tabla 9

Requerimientos de sistema para Android Studio

Windows	Mac OS	Linux
Microsoft Windows 8/7/Vista/2003 (32 o 64 bit)	Mac OS X 10.8.5 o superior, hasta la 10.9 (Mavericks)	GNOME o entorno de escritorio KDE
Mínimo de 2 GB de RAM, recomendado 4 GB de RAM		
400 MB de espacio en disco		
Necesita de al menos 1 GB para Android SDK, emulador de imágenes del sistema, y cachés		
Resolución mínima de pantalla de 1280 x 800		
Java Development Kit (JDK) 7 o superior		

Fuente: (Menéndez, 2016)

CAPÍTULO III

ESTRUCTURA Y MODELO DEL PROTOTIPO

3.1 Criterios de diseño del sistema

En este capítulo se da a conocer el conjunto de especificaciones que conforman el sistema, permitiendo su implementación.

Después de realizar un análisis de los materiales para el diseño del hardware y las plataformas de desarrollo para software del sistema, se consigue implementar un sistema autónomo el cual se encarga de la adquisición de datos del ambiente de interés, procesarlos para su envío a usuarios que están conectados en un entorno de red globalizado, presentar un entorno web que se ajusta a las necesidades visuales del usuario y la oportuna acción en el medio medido para la prevención de accidentes.

3.1.1 Hardware

3.1.1.1 Esquema general de funcionamiento

El diseño del sistema de detección y monitoreo de concentración de gas GLP en el ámbito residencial basado en la plataforma Arduino se puede expresar en un diagrama de bloques general que sintetiza su funcionamiento, tal como se muestra en la figura 31.

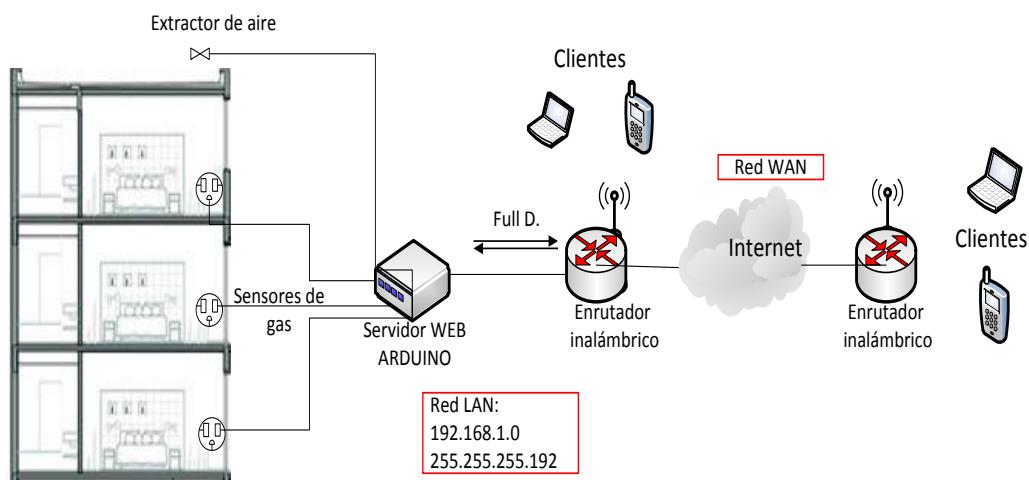


Figura 31 Esquema general de funcionamiento

3.1.1.2 Diagrama de bloques

La figura 32 muestra las conexiones que establece la placa con los diferentes módulos para su normal funcionamiento.

- Control del sistema: Realiza una acción que regula el estado del sistema para controlar la concentración de gas.
- Sensores de gas: Son los sensores de gas LP, los cuales toman la medida del ambiente y entregan un valor de voltaje proporcional a la concentración del gas.
- Placa Arduino: es el corazón del sistema encargado de procesar información y establecer comunicación con los diferentes módulos.
- Alimentación: Se encarga de proporcionar la energía necesaria a la placa para permanecer activa.
- Enrutador: Este dispositivo se encarga de la comunicación entre la placa y los clientes que verifican el estado del sistema.

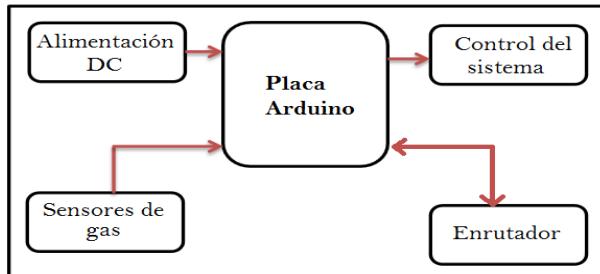


Figura 32 Diagrama de bloques del circuito

3.1.2 Software

3.1.2.1 Diagrama de bloques

La figura 33 representa las tareas que ejecuta el Sketch, se encarga de leer los módulos físicos conectados a la placa como los sensores de gas y la memoria SD.

Otro aspecto importante es mantener la comunicación con el cliente para atender sus requerimientos como enviar los datos procesados, la comunicación serial para monitorear el proceso de ejecución del programa y la comunicación con el servidor de notificaciones en el caso que amerite informar al cliente sobre algún evento crítico.

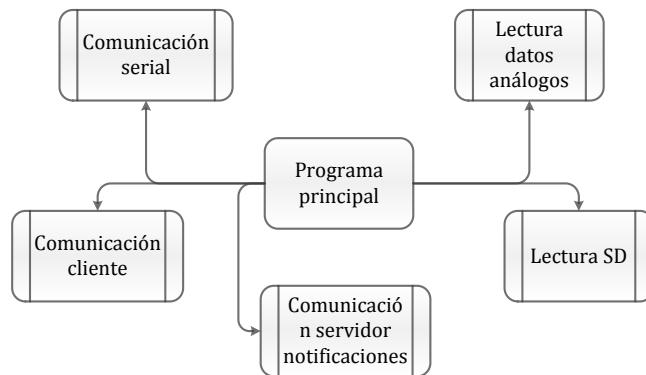


Figura 33 Tareas del programa principal

3.2 Desarrollo de hardware

3.2.1 Instalación de los sensores

3.2.1.1 Consideraciones importantes

Algunos de los factores que se deben tenerse en cuenta para la selección del detector, su calibración y posición idónea de los detectores.

- Si el área es interior o exterior;
- Las fugas en potencia: localización, naturaleza (ej.: la densidad, presión, volumen, temperatura de la fuente y distancia);
- El dato químico-físico del gas o vapor presente;
- La presencia de líquido de baja volubilidad. En este caso son necesarios detectores junto a la fuente;
- La naturaleza y la concentración típica del gas fugado (ej.: chorro presión, pérdida lenta, derrame de líquido);

- La presencia de cavidades o chorro;
- El movimiento del aire: - en interior: ventilación natural y forzada; - en exterior: velocidad y dirección del viento; Los flujos térmicos eventuales pueden influir en la distribución de la mezcla gas-aire
- Los efectos de la temperatura, las condiciones ambientales de la instalación;
- La cantidad de personas presentes en el área y la localización de las mismas;
- La localización de las posibles fuentes de incendio;
- La ubicación de los detectores instalados que no deben ser vulnerables a los golpes y el agua durante el funcionamiento normal;
- La ubicación de los detectores debe permitir una fácil calibración y mantenimiento;
- La situación de estructuras (paredes, mamparas, separadores, etc.) que favorezcan la acumulación de gas;

3.2.1.2 Instalación y cobertura

Los sensores deben instalarse en todas las áreas donde puedan formarse acumulaciones peligrosas de gas. Puede que no sean áreas en las que se produzca una fuga sino donde haya un movimiento de aire reducido.

El gas más pesado suele acumularse en pozos o cuevas y el gas más ligero suele acumularse en cavidades superiores. Los sensores deben ubicarse junto a la mayoría de las fuentes, aunque, para evitar falsas alarmas, no deben estar cerca de elementos que desprendan gas en su funcionamiento normal. En general, en lugares abiertos, las pequeñas fugas pueden dispersarse sin causar acumulaciones peligrosas. Cuando es necesario detectar la fuga de gas de un área específica, los sensores pueden colocarse a intervalos regulares a lo largo del perímetro del área. (Honeywell, 2016)

No obstante, esta sistematización para detectar algunas fugas específicas puede originar alarmas no deseadas, por lo que se aconseja utilizarla sólo cuando hay un peligro significativo para las personas o cosas en el interior de perímetro.

Para evitar daños en el sensor producidos por tareas de soldadura o de pintura, debe instalar el sensor una vez hayan acabado los trabajos de construcción.

Área de cobertura

Es muy difícil establecer la cobertura en distancia desde el detector, dando valores absolutos para una correcta instalación. Se pude seguir el siguiente valor de referencia de la tabla 10, teniendo presente que a diferencia de la detección de incendio donde el humo se expande bastante uniforme, el gas baja o asciende formando estratos o nubes que se posicionan en puntos distintos según las condiciones ambientales y del aire.

Tabla 10

Área de cobertura de sensores

Cobertura media en m ²	Área con geometría normal		Área con geometría estrecha Corredores-Conducciones- Galerías- Túneles	
	Gas ligero	Gas pesado	Gas ligero	Gas pesado
Catalítico industrial	80 – 100		50 – 80	30 – 50
Catalítico profesional	100 – 300	80 – 100	60 – 150	50 – 80
Semicond. Industrial	100		80	
Célula	100 – 300		60 – 150	
Electroquímica				

Fuente: (Honeywell, 2016)

Ubicación: gas pesado

Áreas normales.- Los detectores se instalarán junto al suelo a 30cm del mismo, montado sobre el muro lateral, columnas, etc. y colocados de modo que al menos uno esté próximo al punto de posible fuga de gas.

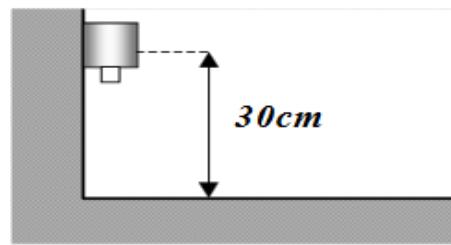


Figura 34 Ubicación normal de detector de gas pesado

Fuente: (Honeywell, 2016)

Áreas con pozos o en pendiente.- Los detectores deben instalarse siempre en el punto más bajo del suelo con pendiente y en el interior del pozo.

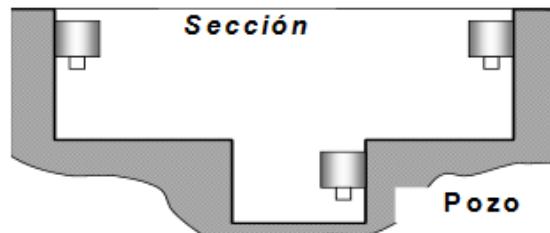


Figura 35 Ubicación pendiente de detector de gas pesado

Fuente: (Honeywell, 2016)

Es preferible no instalar detectores en zonas muertas, a menos que pueda ser una zona de posible fuga de gas.

Para la aplicación se ha deducido instalar los sensores a 15cm de altura como lo muestra la figura 36, pues el espacio de prueba tiene medidas inferiores a las que es aplicable la normativa.



Figura 36 Ubicación de sensores

3.2.1.3 Seguridad con la electrónica

Los dispositivos para la tecnología de medición y control, presentan un consumo de energía relativamente bajo ya que son diseñados de manera muy inteligente con respecto a su protección contra explosión, pues en el caso de un primer o segundo fallo, está asegurado que no se produzcan chispas con suficiente energía ni superficies con suficiente calor, así no hay riesgo de ignición.

Las descargas eléctricas del dispositivo deben tener cierta energía mínima, las temperaturas de superficie de componentes eléctricos o electrónicos no deben sobrepasar la temperatura de ignición, 480 – 549 °C para GLP.

Si los circuitos electrónicos de un producto están diseñados como corresponde y la energía eléctrica almacenada (esto es capacidades e inductancias) así como la fuerza eléctrica (eso es corriente y voltaje eléctrico) están limitados a ciertos valores

máximos por fusibles, resistores para limitación de corriente y diodos Zener para limitación de voltaje. Un circuito electrónico que no puede actuar como una fuente de ignición se conoce como un producto intrínsecamente seguro.

Los productos intrínsecamente seguros pueden ser revisados mientras que están conectados. Su diseño sofisticado, ligero y sencillo permite que los cables se puedan desconectar y los sensores sustituir sin desclasificación de la zona porque con toda seguridad no puede haber chispas ni superficies calientes.

Una carcasa antideflagrante es un método pesado y puramente mecánico para evitar la ignición de una atmósfera explosiva. El tipo de protección intrínsecamente seguro simplemente deja que la explosión se efectúe en el interior de la carcasa y de manera fiable evita un flashback. Así las carcasa del tipo de protección antideflagrante deben diseñarse de tal manera que puedan resistir la presión de la explosión interna. (Drager, 2016).

En el prototipo diseñado únicamente el receptor de cada sensor tiene contacto con el gas dentro de la maqueta, dejando aislado el cableado y las patillas de conexión fuera de la misma figura 37. Para el caso de los ventiladores figura 38, tienen la misma configuración con el cableado alineado por el exterior de la maqueta hacia la fuente de poder.



Figura 37 Cableado del sensor

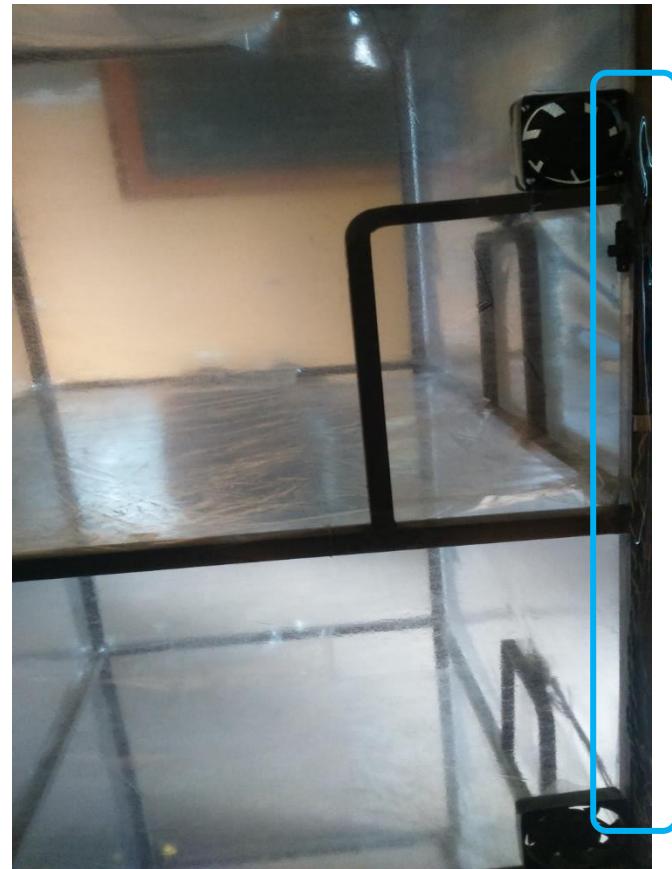


Figura 38 Cableado de ventilador

3.2.2 Diseño de la maqueta

Los sensores deben conectarse a la unidad de control externa al sistema (considerando la mínima dimensión del cable, el aislamiento, etc.) y utilizando mangueras, hilos y sistemas de conexión u otros sistemas adaptados y aprobados para la clasificación del área. Lubrificar las conexiones verificando que el lubricante no contenga siliconas que puedan deteriorar el sensor.

En muchas ocasiones, la orientación del detector puede venir especificada por el fabricante. Se debe llevar a cabo un drenaje adecuado para minimizar las condensaciones y depósitos en el instrumento, en la manguera y los conductores. En la tabla 11 se aprecia la distancia que se debe guardar entre las diferentes instalaciones de servicio por seguridad.

Tabla 11

Distancias mínimas entre tuberías

TUBERIA PARA OTROS SERVICIOS	CURSO PARALELO (cm)	CRUCE (cm)
Conducción agua caliente	3	1
Conducción eléctrica	3	1
Conducción de Vapor	5	5
Chimeneas y ductos de extracción de gases de combustión	40	40
Suelo por donde discurren	3	Ninguno
Las tuberías vistas que discurran por locales comerciales e industriales deben colocarse a una separación mínima de 1 cm de las paredes, pisos y techos.		

Fuente: (INEN, 2010)

Debe existir una ventilación segura para cualquier gas potencialmente explosivo introducido en el sistema. En la figura 39 se indica la el ducto de ventilación de la maqueta que servirá para la circulación de corriente de aire y para drenar el gas que permanece en el interior de la maqueta.



Figura 39 Ducto de ventilación del sistema

Efectos de absorción en materiales y paredes

Es recomendable que los tubos sean preferiblemente PTFE (Teflón), Viton, o acero inoxidable. Algunos gases se absorben con mayor facilidad en las superficies de materiales pudiendo influir en la señal de medida especialmente para medida de ppm. La figura 40 presenta la estructura metálica de la maqueta y el revestimiento con plástico acetato. El diseño de la maqueta se la observa en el anexo A.



Figura 40 Estructura de la maqueta

3.2.3 Calibración del sistema

Los sensores deben estar instalados de forma que se pueda acceder a ellos con facilidad para permitir que se pueda llevar a cabo la calibración, mantenimiento e inspección. Siempre que sea posible, se debe acceder al sensor periódicamente vía remota para adoptar alguna forma de calibración.

Al efectuar la calibración de los sensores de gases, se deberá hacer en condiciones de laboratorio y siempre con una muestra del gas a detectar que no deberá exceder a una concentración mayor del 40% del LEI pues no se puede garantizar que la respuesta del sistema sea lineal.

En el caso de detectores con rango LIE, el nivel de alarma debe ser el más bajo posible teniendo en cuenta el evitar las falsas alarmas.

La regulación debe realizarse de acuerdo con las instrucciones del fabricante.

Reglas generales para la calibración

- a) Calibrar siguiendo las instrucciones del fabricante y utilizando el equipamiento recomendado
- b) Calibrar antes de la instalación y después establecer un plan de calibraciones y mantenimiento.
- c) Inspeccionar con regularidad para comprobar los posibles defectos, daños o deterioros
- d) Probar y recalibrar antes de cada uso si se trata de detectores portátiles
- e) Calibrar sobre el gas para el que resulta menos sensible si se utilizan para la detección de más de un gas.

Las instrucciones de uso, de verificación y operativa de sistemas fijos deben estar siempre disponibles.

A fin de efectuar el mantenimiento, deben realizarse planos de la instalación que muestren todos los componentes del sistema (unidad de control, sensores, cajas de distribución, etc.) así como los esquemas de conexionado y distribución de las mangueras conectadas o relativas al sistema. Se deben conservar los informes de instalación para posibles intervenciones futuras.

3.2.3.1 Escalando los sensores MQ-6

Para que el sensor funcione correctamente y con mayor precisión necesita un pre-calentamiento. Este dependerá de cuánto tiempo ha permanecido inactivo. Inicialmente se necesita un pre-calentamiento de mínimo 12hrs para tener una fidelidad decente. A partir de este mínimo el sensor podrá apagarse y solo requerirá aproximadamente 10-15 min para volver a estar listo.

Necesitamos escalar el valor leído por los sensores a las unidades correspondientes de la medición del gas, el problema de esto es que la relación entre

la lectura analógica y el valor real no es lineal. Por lo que necesitamos estimar la curva que nos da el datasheet.

La curva de característica de la figura 23 sirve para estimar una ecuación por medio de regresión de los datos de la tabla 12. Se procede a ingresar datos de la curva de GLP a una hoja de cálculo, con la mayor cantidad de datos para graficar y obtener la ecuación general.

Tabla 12
Datos aproximados de la curva del MQ-6

Ppm	Rs/R ₀
200	2
300	1,8
400	1,62
500	1,49
600	1,36
700	1,3
800	1,24
900	1,16
1000	1
2000	0,73
3000	0,64
4000	0,57
5000	0,51
6000	0,48
7000	0,45
8000	0,43
9000	0,405
10000	0,385

La figura 41 representa la ecuación extraída después de realizar la regresión.

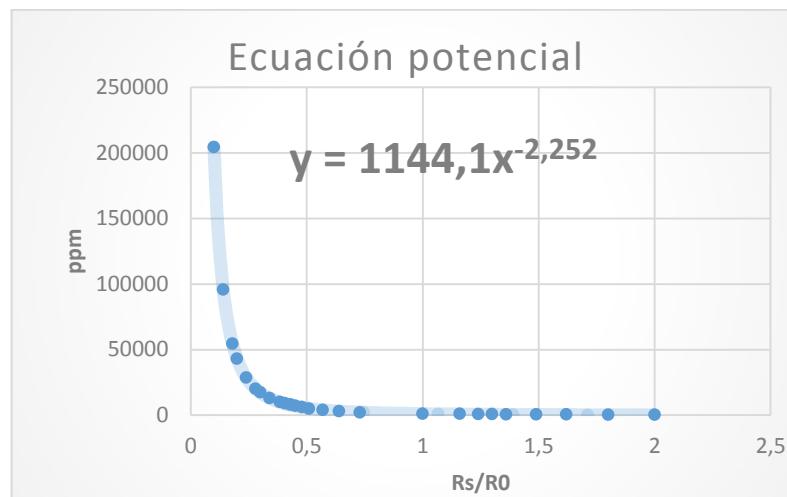


Figura 41 Ecuación potencial del sensor

La ecuación que obtenemos es:

$$\text{ppm} = 1144,1(\text{Rs}/\text{R}_0)^{-2,252} \quad \text{Ecuación 5}$$

Los valores ppm se refieren el valor de concentración de GLP, Ro es una constante que equivale al valor de la resistencia del sensor cuando se lo expone a una concentración de 1144 ppm y Rs es la resistencia del sensor, el cual leemos desde Arduino.

Las características de fidelidad del sensor también dependerán de una resistencia de carga RL, la cual básicamente se conecta en serie con la resistencia variable del sensor. Así esta tendrá un divisor de voltaje que puede variar dependiendo de la concentración de sustancia/aire (A mayor resistencia de cargar se tiene un rango más amplio de medición, a costa de un mayor tiempo de precalentamiento y de espera.).

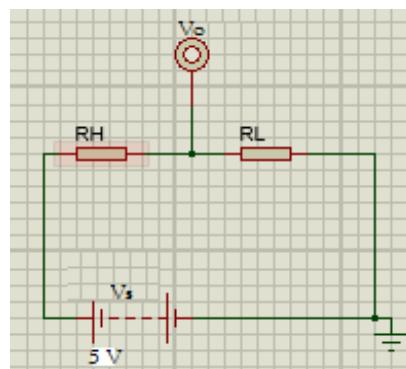


Figura 42 Circuito de conexión del sensor MQ-6

De la ecuación 4 para calcular el valor de R_s la despejamos, sabiendo que la resistencia de carga R_L en la mayoría de módulos es de 1K.

$$R_s = R_L \left(\frac{V_s}{V_0} - 1 \right) \quad \text{Ecuación 6}$$

El valor de R_o se calcula en laboratorio, R_o es el valor de R_s cuando se usa una muestra de aire con 1144 ppm. Otra forma de calibrar es usando concentraciones conocidas medidas con un equipo apropiado.

Sin embargo un método para conseguir de forma aproximada el valor de R_o y así tener la ecuación resuelta. En el gráfico de la ecuación potencial se observa que el valor de R_s/R_o se acerca 0,1 para valores superiores a la máxima concentración de GLP que el sensor puede censar, en los puntos cercanos a este punto, la variación de R_s/R_o es mínima por lo que los errores que pueden darse tener acá para calcular R_o son pequeños.

Entonces asumiendo que el sensor se satura con 150 K ppm al cual según la gráfica le corresponde un R_s/R_o de 0.12, los puntos cercanos por mayor variación de GLP que exista el R_s/R_o tendrán variaciones mínimas. Para estar en este punto generamos un ambiente con bastante GLP y si medimos con nuestro sensor obtendremos un voltaje de 4.18 V.

```

MQ-6-2

1 void setup() {
2   Serial.begin(9600);
3 }
4
5 void loop() {
6
7   int adc_MQ = analogRead(A0); //Lemos la señal
8   float voltaje = adc_MQ * (5.0 / 1023.0);
9
10  Serial.print("adc:");
11  Serial.print(adc_MQ);
12  Serial.print("    voltaje:");
13  Serial.println(voltaje);
14  delay(300);
15 }
16

```

adc:854	voltaje:4.17
adc:853	voltaje:4.17
adc:855	voltaje:4.18
adc:855	voltaje:4.18
adc:855	voltaje:4.18
adc:853	voltaje:4.17
adc:852	voltaje:4.16
adc:851	voltaje:4.16
adc:849	voltaje:4.15
adc:848	voltaje:4.14
adc:847	voltaje:4.14
adc:846	voltaje:4.13
adc:845	voltaje:4.13
adc:843	voltaje:4.12
adc:842	voltaje:4.12
adc:842	voltaje:4.12

Autoscroll

Figura 43 Captura de datos GLP

, que equivale a un $R_s=196$, entonces:

$$(R_s/R_o)=0.12 \text{ si } R_s=196 \text{ entonces } R_o=1633$$

Con R_o calculado, ya tenemos una solución de la ecuación y con esto obtener los valores aproximados de concentración de GLP.

$$\text{ppm}=1144,1(R_s/1633)^{(-2,252)}$$

Ecuación 7

El Sketch en Arduino representando las ecuaciones obtenidas, en la figura 44:

```

MQ-6-2\$

1 void setup() {
2   Serial.begin(9600);
3 }
4 void loop() []

5
6 int adc_MQ = analogRead(A0); //Lemos la salida analógica del MQ
7 float voltaje = adc_MQ * (5.0 / 1023.0); //Convertimos la lectura en un valor de voltaje
8 float Rs=1000*((5-voltaje)/voltaje); //Calculamos Rs con un RL de 1k
9 double glp=1144.1*pow(Rs/1633, -2.252); // calculamos la concentración de glp con la ecuación obtenida.
10 //-----Enviamos los valores por el puerto serial-----
11 Serial.print("adc:");
12 Serial.print(adc_MQ);
13 Serial.print("    voltaje:");
14 Serial.print(voltaje);
15 Serial.print("    Rs:");
16 Serial.print(Rs);
17 Serial.print("    glp:");
18 Serial.print(glp);
19 Serial.println("ppm");
20 delay(333);
21 }

```

Figura 44 Código de captura datos GLP

Los resultados leídos para un estado pasivo se muestran en la figura 45:

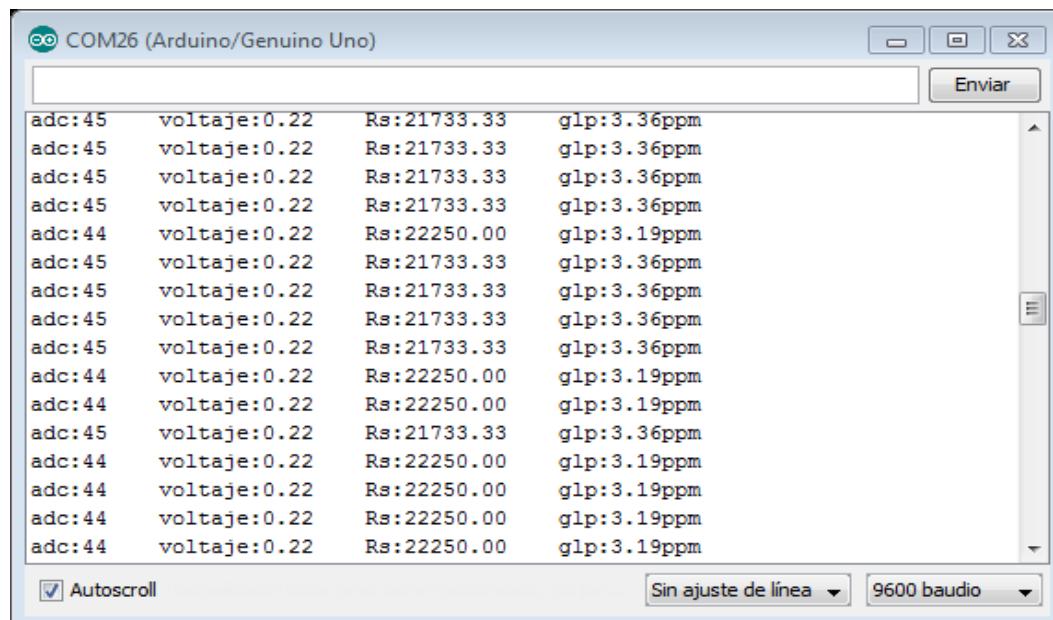


Figura 45 Datos de gas, escenario inicial

Los resultados leídos para un estado medio estable se muestran en la figura 46:

The screenshot shows the Arduino Serial Monitor window titled "COM26 (Arduino/Genuino Uno)". The main text area displays a series of data entries, each consisting of four fields: ADC value, voltage, resistance, and gas concentration (GLP). The data is as follows:

```

adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:212  voltaje:1.04  Rs:3825.47  glp:168.23ppm
adc:212  voltaje:1.04  Rs:3825.47  glp:168.23ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:212  voltaje:1.04  Rs:3825.47  glp:168.23ppm
adc:212  voltaje:1.04  Rs:3825.47  glp:168.23ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm
adc:212  voltaje:1.04  Rs:3825.47  glp:168.23ppm
adc:212  voltaje:1.04  Rs:3825.47  glp:168.23ppm
adc:213  voltaje:1.04  Rs:3802.82  glp:170.49ppm

```

At the bottom of the window, there are several controls: a checked "Autoscroll" checkbox, a "Sin ajuste de línea" dropdown, a "9600 baudio" dropdown, and an "Enviar" button.

Figura 46 Datos de gas, escenario controlado

Los resultados leídos para un estado alto de cambio brusco se muestran en la figura 47:

The screenshot shows the Arduino Serial Monitor window titled "COM26 (Arduino/Genuino Uno)". The main text area displays a series of data entries, each consisting of four fields: ADC value, voltage, resistance, and gas concentration (GLP). The data is as follows:

```

adc:259  voltaje:1.27  Rs:2949.81  glp:302.09ppm
adc:268  voltaje:1.31  Rs:2817.16  glp:335.07ppm
adc:435  voltaje:2.13  Rs:1351.72  glp:1751.25ppm
adc:835  voltaje:4.08  Rs:225.15   glp:99161.52ppm
adc:851  voltaje:4.16  Rs:202.12   glp:126444.51ppm
adc:854  voltaje:4.17  Rs:197.89   glp:132602.18ppm
adc:855  voltaje:4.18  Rs:196.49   glp:134740.98ppm
adc:858  voltaje:4.19  Rs:192.31   glp:141432.15ppm
adc:857  voltaje:4.19  Rs:193.70   glp:139154.68ppm
adc:856  voltaje:4.18  Rs:195.09   glp:136924.73ppm
adc:855  voltaje:4.18  Rs:196.49   glp:134740.98ppm
adc:855  voltaje:4.18  Rs:196.49   glp:134740.98ppm
adc:854  voltaje:4.17  Rs:197.89   glp:132602.18ppm
adc:854  voltaje:4.17  Rs:197.89   glp:132602.18ppm
adc:852  voltaje:4.16  Rs:200.70   glp:128455.11ppm

```

At the bottom of the window, there are several controls: a checked "Autoscroll" checkbox, a "Sin ajuste de línea" dropdown, a "9600 baudio" dropdown, and an "Enviar" button.

Figura 47 Datos de gas, escenario de peligro

3.2.4 Alimentación eléctrica

Para determinar una fuente de alimentación necesaria se procede a sumar la potencia que consume cada elemento activo del sistema mostrado en la figura 48, como lo muestra la tabla 13.

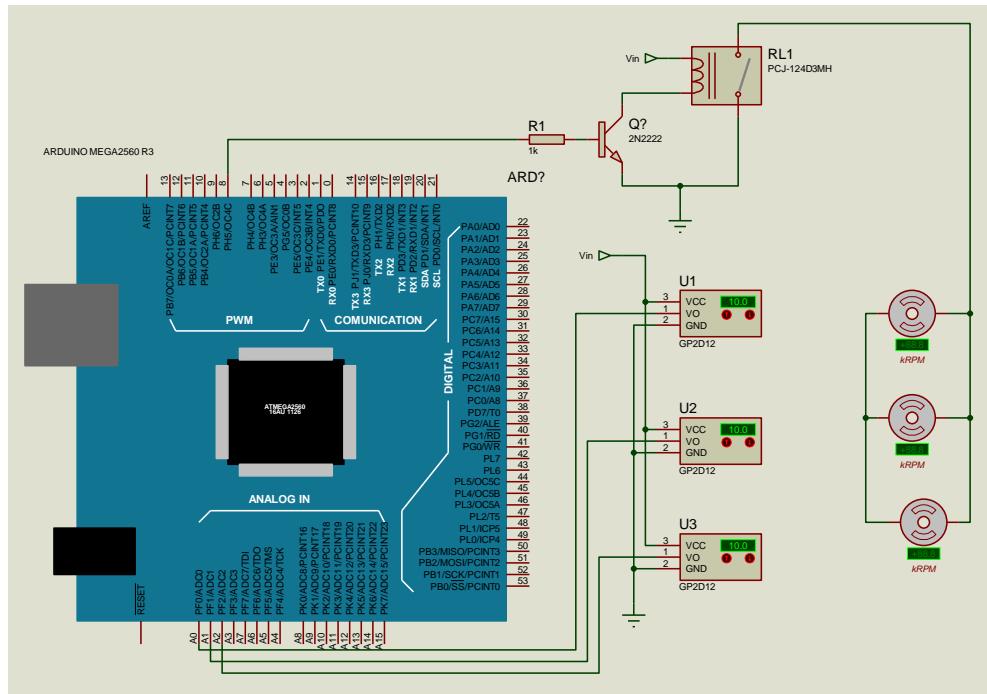


Figura 48 Sistema electrónico del prototipo

Tabla 13

Potencia de consumo del sistema

Material	Cantidad	Potencia nominal (W)	Potencia total (W)
Ventilador	3	1,68	5,04
Sensor	3	0,9	2,7
Tarjeta Arduino	1	0,765	0,765
			8,505 W

Para alimentar el prototipo se utiliza una fuente de voltaje regulada de PC, la cual satisface con mucho las exigencias de consumo total de energía.

3.2.5 Mantenimiento

El mantenimiento de los sistemas de detección de gases inflamables es un factor muy importante que puede influir en la fiabilidad del sistema.

Los sistemas para gases inflamables varían, en cuanto a aplicación, los portátiles (normalmente utilizados cuando se debe comprobar la presencia de gas) y los fijos que raramente están en contacto con el gas durante su vida.

Sistemas fijos

Inspección visual periódica.- Se debe realizar una inspección visual periódica del panel de control. Las verificaciones deben registrarse y archivarse con fecha y firma y con una breve descripción de lo realizado.

Los problemas encontrados deben resolverse rápidamente.

Calibración periódica y operativa del sistema.- Esta comprobación es de vital importancia para la conservación y fiabilidad del sistema.

La verificación se efectúa sometiendo cada elemento sensor al gas de calibración y realizando un ciclo completo de prueba del sensor y de la electrónica. Se debe anotar de forma detallada toda la información sobre las características del sistema, su particularidad y la vida de sus componentes etc. Siguiendo esta verificación, se pueden prevenir posibles problemas antes de que se produzcan.

El plan de seguridad debe prever la rotación de los equipos para efectuar calibraciones o su sustitución.

A continuación se presenta en la tabla 14 el mantenimiento recomendado según el tipo de sensor.

Tabla 14
Mantenimiento según el tipo de sensor

Modelo	Frecuencia	Acción de mantenimiento	Accesorios necesarios
Todos	Cada 6 meses	Verificación del cero y calibración de las sondas.	Regulador de caudal. Célula de flujo.
	En caso de alarma	Verificación del cero y calibración de las sondas. Cambio del sensor si fuera necesario.	Mezcla patrón. Regulador de caudal. Célula de flujo.
Gases combustibles	Cada 6 meses	Chequeo de los filtros para su limpieza	
	Cada 4 años	Cambio del sensor si fuera necesario.	
Tóxicos: H₂s, NH₃, Cl₂ y CO	Cada 2 años	Cambio de la célula sensora si fuera necesario.	
Tóxico; O₂	Una vez al año	Cambio de la célula sensora si fuera necesario.	

Fuente: (Drager, 2016)

3.3 Desarrollo de software

3.3.1 Programación para la tarjeta Arduino

3.3.1.1 Programa general

La placa utilizada en el proyecto tiene como base un microcontrolador, este tiene la característica de siempre estar ejecutando una línea de código de manera secuencial. El programa cargado en el Arduino, al ser este encendido empieza por habilitar las librerías que se utilizará, registrar las variables globales y setear los puertos de la placa a utilizar.

Después de esta configuración procede a ejecutar cíclicamente el código del programa principal, se puede resumir el sketch de Arduino en el siguiente diagrama de flujo.

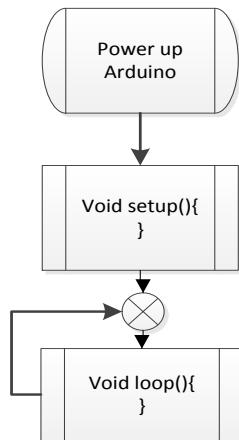


Figura 49 Diagrama de flujo del programa principal

3.3.1.2 Programa de configuración Setup Arduino

En el sketch se inicializa por única vez las variables de programa, el estado y método de trabajo de los puertos de la placa, como lo indica la figura 50.

Es primordial realizar las configuraciones iniciales para el debido funcionamiento de las herramientas utilizadas como lo son la comunicación Serial, Ethernet y la tarjeta SD que contiene el programa HTML.

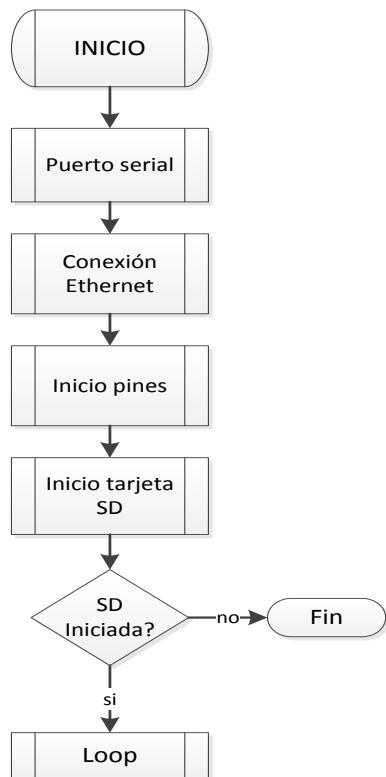


Figura 50 Diagrama de flujo de la función Setup

3.3.1.3 Programa principal loop Arduino

Después de inicializar Setup, Arduino mantiene siempre ejecutando la función loop (programa principal) figura 51, en esta sección del sketch se comprueba si se ha establecido una conexión con un cliente y atender el requerimiento del mismo.

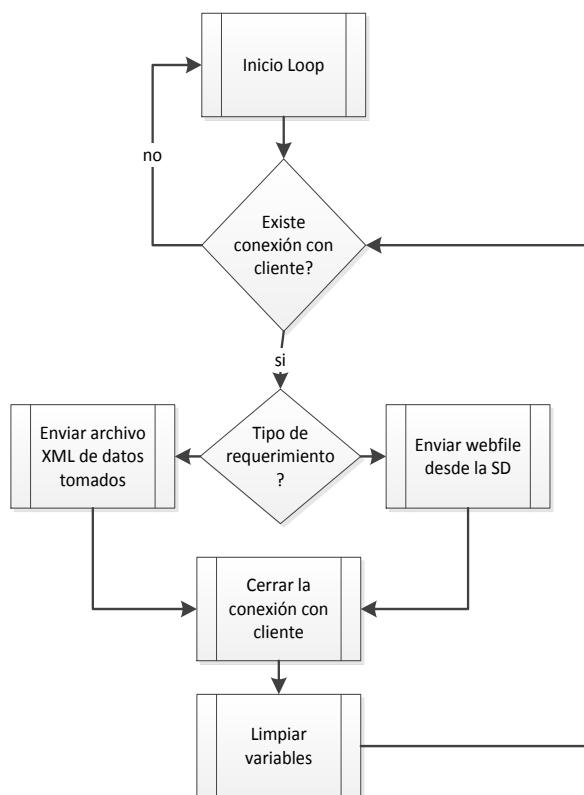


Figura 51 Diagrama de flujo de la función loop

3.3.2 Muestreo y límites permitidos

Un aspecto importante a más de la lectura eficiente de datos es saber discriminar cuales de estos representan una anomalía en el sistema y evitar falsas alarmas debido a los picos que se pueden presentar en algún momento por la naturaleza del mismo sistema.

El valor del LIE de 1,8 % que corresponde al límite de gas combustible para producir una explosión, sirve de referencia para calcular el límite o valor umbral que debe tolerar el sistema de control el cual es incluido en la programación de Sketch de Arduino.

Por seguridad se ha optado adoptar un valor correspondiente al 50% del valor del LEI el cual expresado en valores ppm, se lo obtiene:

- El 50% del valor de LEI corresponde a 0,9%
- Para expresar % a ppm se realiza una regla de tres; 100% equivale a 1'000.000 ppm entonces

$$\text{Umbral} = (1'000.000 \text{ ppm} * 0,9\%) / 100\% = 9000 \text{ [ppm]}$$

Definido en el código del programa principal:

```
int umbral = 9000;
```

Para evitar una falsa lectura el código se comporta de manera redundante ante la medición de las variables antes de lanzar las notificaciones y activar el hardware para contrarrestar el posible peligro.

```
if (analog_val0 >= umbral){  
    delay(5);  
    analog_val0 = analogRead(0);  
    if (analog_val0 >= umbral){  
        actvent(DEVID1);}}
```

3.3.3 Programación del código HTML

El desarrollo del código se lo puede probar en un editor de texto simple como el blog de notas propio de Windows, el documento final tiene que ser guardado en la memoria SD con el nombre index.htm.

3.3.3.1 Programa principal HTML

El programa HTML tiene la función de representar en un entorno gráfico las variables medidas por la tarjeta Arduino, desde el lado del cliente realiza peticiones de dichos datos y los grafica por medio de un Gauge. Las funciones principales de las aprecia en la figura 52.

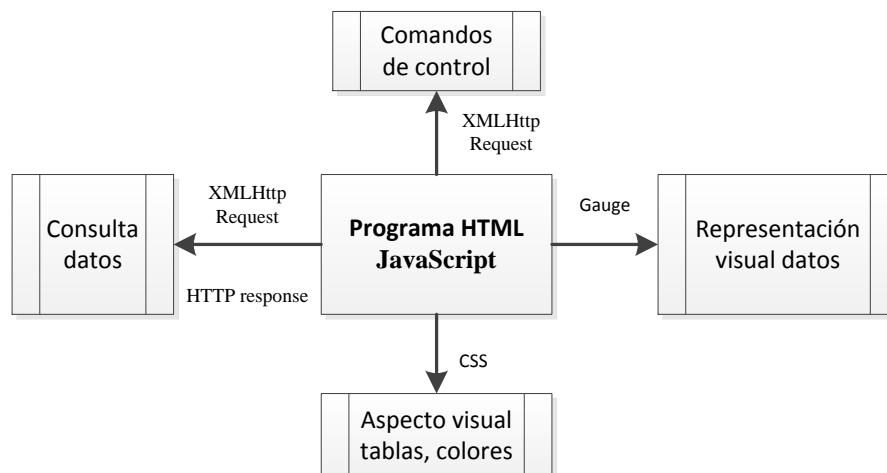


Figura 52 Funciones del programa HTML

JavaScript es un lenguaje, cuyos comandos se ejecutan al lado del cliente, es decir en el navegador web cuando el usuario visita una página HTML. AJAX es básicamente el uso de un conjunto de funciones de JavaScript, optimiza recursos de la red al sólo consultar la información que ha cambiado.

El proceso de intercambio de datos se puede resumir de la siguiente manera:

1. AJAX Request desde el browser

La página HTML realiza consultas al servidor cada cierto tiempo, en este caso cada segundo por medio del comando

- `setTimeout('GetArduinoInputs()', 1000);`

Llama a ejecutar a la función JavaScript

- `GetArduinoInputs()`

Esta función realiza lo siguiente

- Genera un número aleatorio para enviar la solicitud GET: `nocache = "&nocache=" + Math.random() * 1000000;`
- Crea un objeto `XMLHttpRequest()` llamado `request: new XMLHttpRequest();`
- Asigna una función para manejar la respuesta del servidor web: `request.onreadystatechange = function() {};`

- Se instala una solicitud GET HTTP para enviar al servidor web:
`request.open("GET", "ajax_switch" + nocache, true);`
- Envía la solicitud HTTP: `request.send (null);`

2. Respuesta del Web Server Arduino

- Cuando el servidor web Arduino recibe la solicitud GET HTTP, envía de vuelta una respuesta HTTP estándar seguido de texto que representa el estado de los canales analógicos. El estado del interruptor y el texto enviado se obtiene de la función `GetArduinoInputs()` propia de la Arduino.

3. Respuesta del navegador JavaScript Handles

- La respuesta HTTP del servidor web de Arduino es manejada por el código JavaScript. La función de controlador de eventos JavaScript se ejecuta cuando se recibe la respuesta de la Arduino (la función que controla los eventos es la función sin nombre asignado a `request.onreadystatechange`).

Si la respuesta recibida es correcta (`'analog1'[0]` y no está vacío, entonces esta línea de código JavaScript se ejecuta:

```
document.getElementById("input1").innerHTML = this.responseText.  
getElementsByName('analog1')[0].childNodes[0].nodeValue;
```

- JavaScript encuentra el párrafo en el código HTML que está marcado con la etiqueta de identificación `input1` y sustituye el texto actual con el texto recibido del Arduino. El HTML para este párrafo es el siguiente:

```
<td> <span id="input1">...</span> ppm</td>
```

3.3.3.2 Gauge representación datos

A través de una API completa, se puede añadir, eliminar y modificar series y puntos de ejes o modificar en cualquier momento después de creación de gráficos. En combinación con jQuery, MooTools o API Ajax de Prototype, abre soluciones como gráficos en vivo en constante actualización con los valores del servidor, los datos suministrados por el usuario y más.

La representación de datos se lo realiza insertando los datos captados del servidor como lo muestran las líneas de código:

```
// Seteo para actualización cada segundo
var series1 = this.series[0];
setInterval(function () {
    var x = (new Date()).getTime(), // current time
    y = input1;
    series1.addPoint([x, data_val1/1023], true, true);
```

Realizando una ajuste de los datos a mostrar en la página.

3.3.4 Servicio al cliente de red local

El dispositivo Router en una red LAN, asigna direcciones IP a sus clientes por medio del protocolo DHCP de acuerdo al rango de direcciones que el administrador ha configurado; el servidor Arduino es a su vez un cliente que utiliza el Router como vía de transmisión de tráfico. Es así que el servidor debe contar con una dirección IP que lo identifica dentro de dicho rango, pero de preferencia esta debe ser fija; en el código de programa del servidor la asignación de la dirección IP, MAC y el puerto por el cual se presta el servicio se las realiza con los siguientes comandos.

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192,168,1,8);
```

```
EthernetServer server(80);
```

Para inicializar el Shield Ethernet con los identificadores asignados y presentar la dirección IP por medio de terminal serial se ejecutan los comandos.

```
Ethernet.begin(mac,ip);
Serial.println(Ethernet.localIP());
```

Para que el cliente local pueda acceder a la interfaz figura 53, que alberga el servidor web basta con digitar la dirección IP del servidor seguido del puerto que permite comunicarse con la aplicación. Debido a que las direcciones suelen cambiar con el tiempo y resultan tediosas de memorizar para los clientes, se realiza una pequeña configuración normalmente en el Router para configurar el servicio que brinda DNS para cambiar la dirección IP con un dominio “conjunto de caracteres”, que brinda mayor facilidad al cliente; a continuación se indica las dos vías de como ingresar al servidor.

192.168.1.8:8080 // Vía dirección IP

http://firesglp.ddns.net:8080 // Vía DNS



Figura 53 Aplicación Web usuario local

3.3.5 Servicio al cliente de red remota

Para comenzar se tiene que conectar Arduino al API de Ubidots, para ello se necesita un Token que actúa como llave de acceso de la aplicación y un ID que identifica a las variables medidas del sistema, los cuales se obtienen creando una cuenta en Ubidots e insertando el dispositivo y las variables que se desea trabajar mostrado en la figura 54.

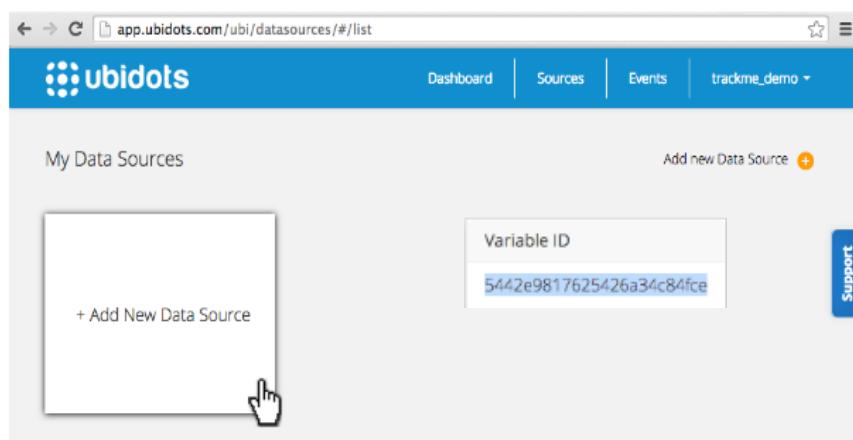


Figura 54 Creación de variables en Ubidots

Con el ID de la variable y el Token, ya se puede enviar datos a Ubidots. Se puede simular un dispositivo enviando datos desde el mismo sitio. El método HTTP a utilizar será el POST, ya que según el estándar éste se refiere a la creación de un nuevo elemento dentro del API, en éste caso a la creación de un “value”, un dato enviado se observa en la figura 55.

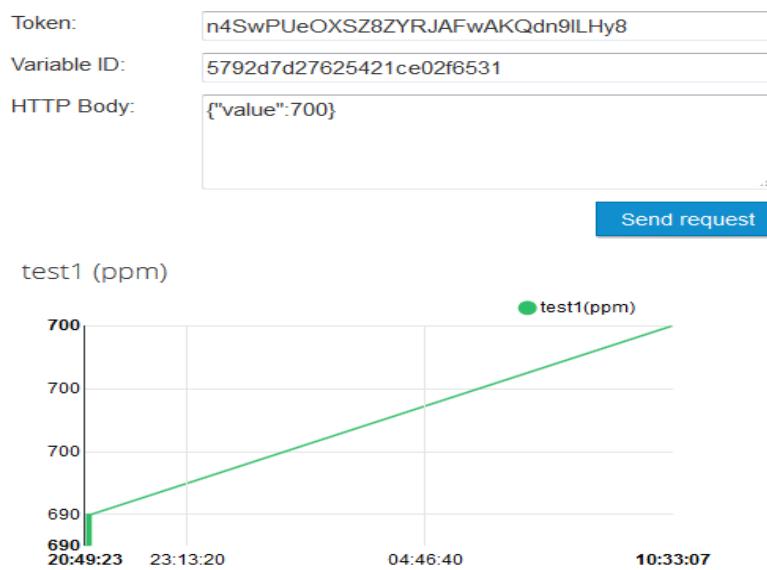


Figura 55 Envió de un dato por medio del sitio Ubidots

Una petición HTTP contiene dos partes: el encabezado o headers y el cuerpo o body, donde el encabezado Content-Length especifica la longitud de del cuerpo o body, en éste caso se refiere a la cantidad de caracteres de: {"value":700}. Enviado el dato el API deberá responder un mensaje tipo "201", indicando que el valor fue exitosamente creado, como se observa en la figura 56.

The screenshot shows a mobile application interface. At the top, there is a header bar with the text "Http Request". Below this, the request details are displayed:

```
POST /api/v1.6/variables/5792d7d27625421ce02f6531/values
HTTP/1.1
X-Auth-Token: n4SwPUEOXSZ8ZYRJAFwAKQdn9ILHy8
Host: things.ubidots.com
Connection: close
Content-Type: application/json
Content-Length: 13

{"value":700}
```

At the bottom, there is a header bar with the text "API response". Below this, the response details are displayed:

```
{"url": "http://things.ubidots.com/api/v1.6/values/57963133762542731536dd64", "value": 700, "timestamp": 1469460787325, "created_at": "2016-07-25T15:33:07.325"}
```

Figura 56 Codificación del mensaje al API de Ubidots

Una vez confirmada la recepción del dato, éste debe aparecer en nuestra cuenta de Ubidots, para visualizar la variable es necesario asignar cualquiera de los Widgets disponibles con la ID de la variable creada, como el ejemplo dela figura 57.

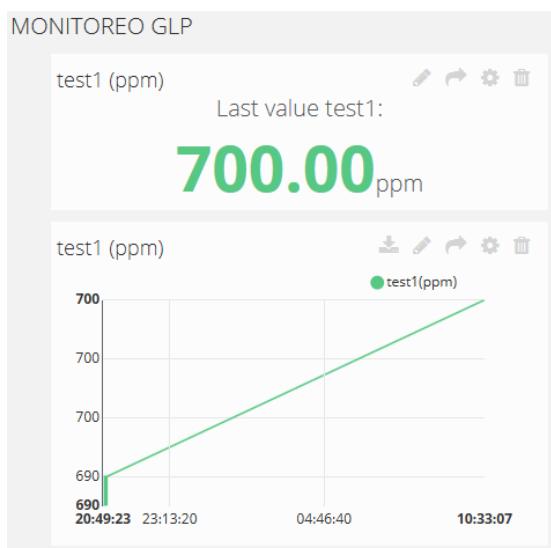


Figura 57 Visualización de variables en Ubidots

De la misma forma se puede crear eventos que responden a las el cambio de las variables que se encararan de enviar notificaciones a nuestros dispositivos electrónicos, mostrado en la figura 58.

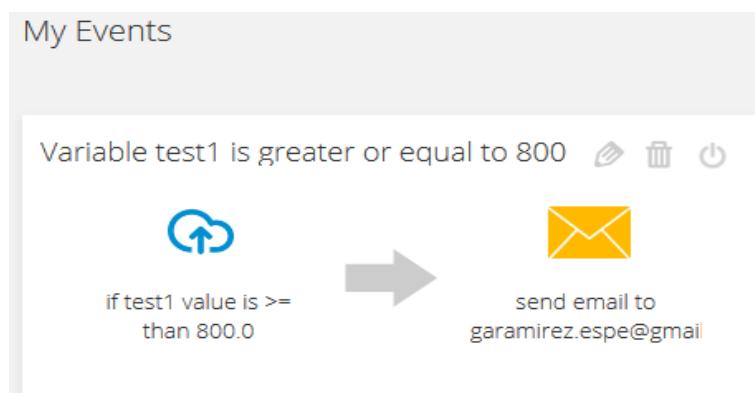


Figura 58 Notificaciones en Ubidots

La implementación en Arduino se realiza utilizando la librería que ofrece Ubidots, instalando la librería se tiene que configurar el ID y Token de la aplicación para empezar a transmitir datos.

Parámetros de identificación del cliente:

```
#include <UbidotsEthernet.h>
#define TOKEN "n4SwPUeOXSZ8ZYRJAFwAKQdn9ILHy8"
```

Ubidots client(TOKEN);

Parámetros de identificación de los datos:

```
client.add("5792d7d27625421ce02f6531", A0);
client.sendAll();
```

3.3.6 Programación para la App Android

Desde la interfaz, proceder a crear un nuevo proyecto seguido del nombre y ubicación de la aplicación.

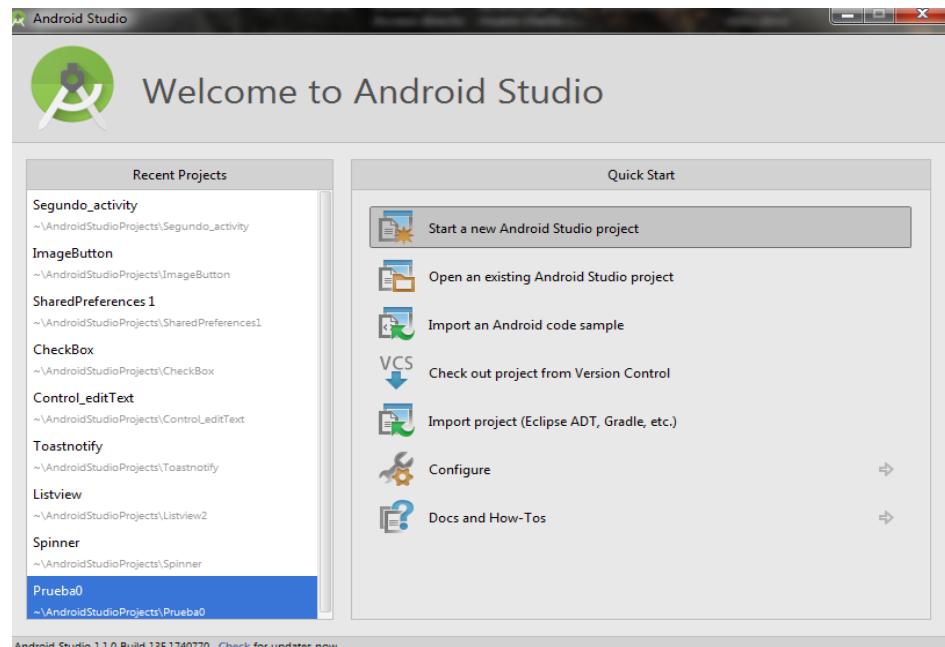


Figura 59 Creación de nuevo proyecto Android

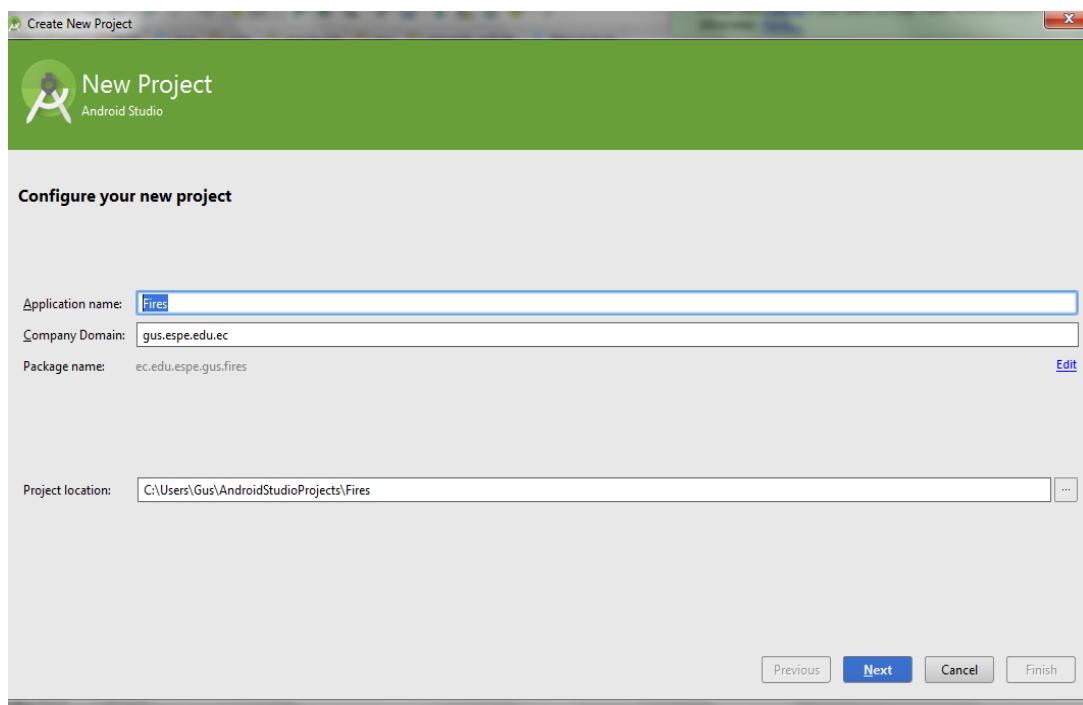


Figura 60 Configuración del nuevo proyecto

A continuación especificar la versión de Android minima donde ese ejecutara la aplicación, para este caso se elige la versión 4.1 (Jelly Bean).

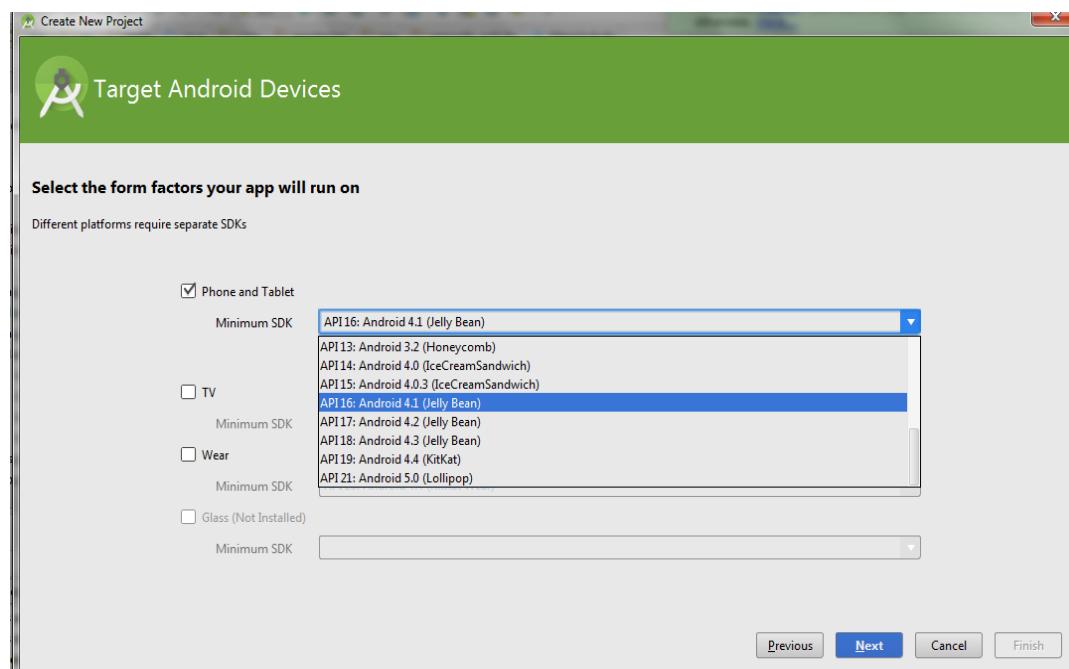


Figura 61 Configuración de mínima versión Android

Como siguiente paso, especificar la estructura de la aplicación para generar el código básico que presente una ventana seleccionar “Blank Activity”.

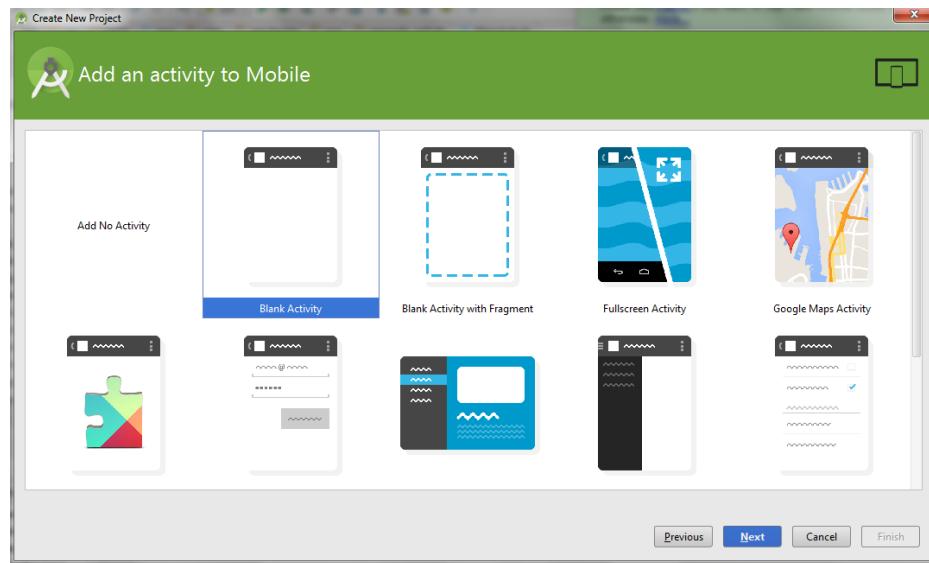


Figura 62 Creación de Activity

Para terminar la configuración se tiene que indicar el nombre de la ventana principal de la aplicación (Activity Name), dejandolos por defecto como propone Android Studio.

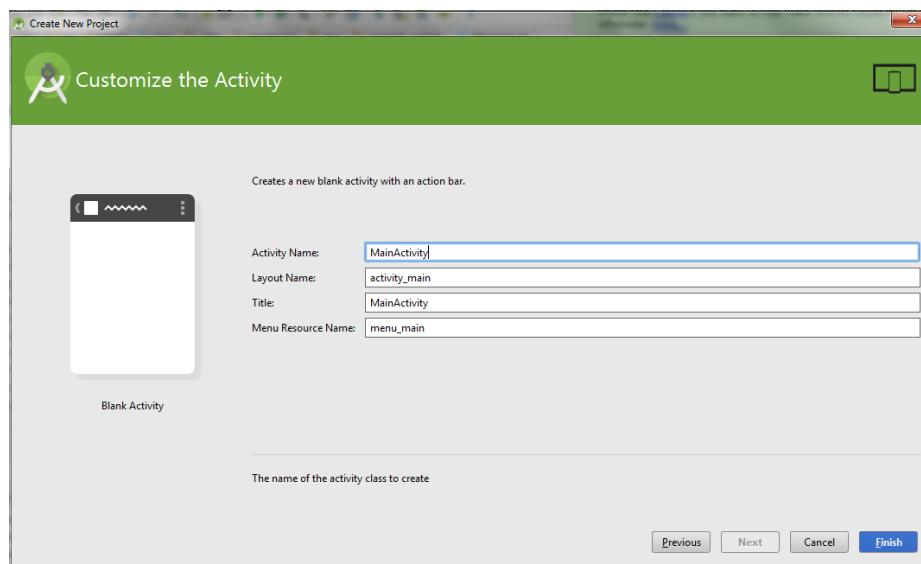


Figura 63 Configuración de la Activity

Situados en la paleta de herramientas, arrastrar un *LinearLayout(vertical)* a la pantalla de diseño con el fin de alinear los componentes que se van a ubicar.

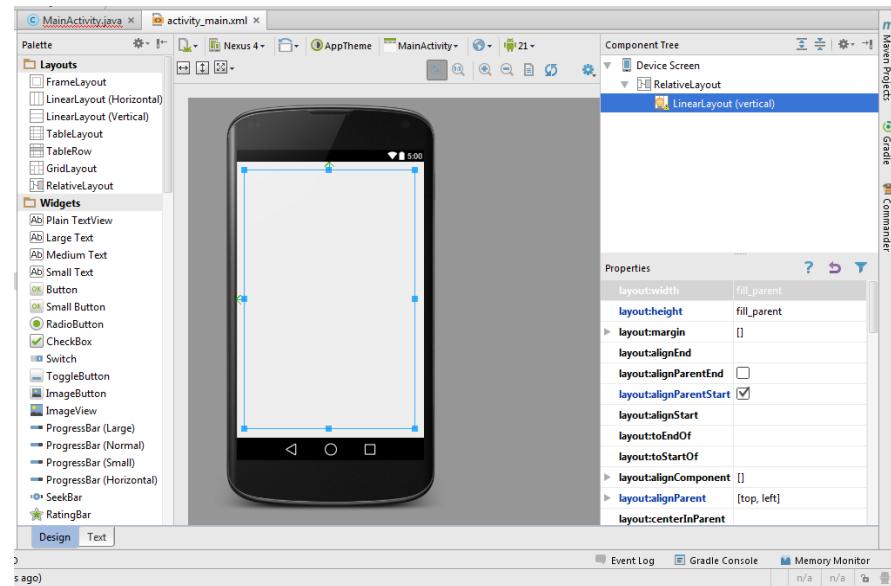


Figura 64 LinearLayout (vertical)

En la primera posición arrastrar un widget, LargeText para colocar una leyenda

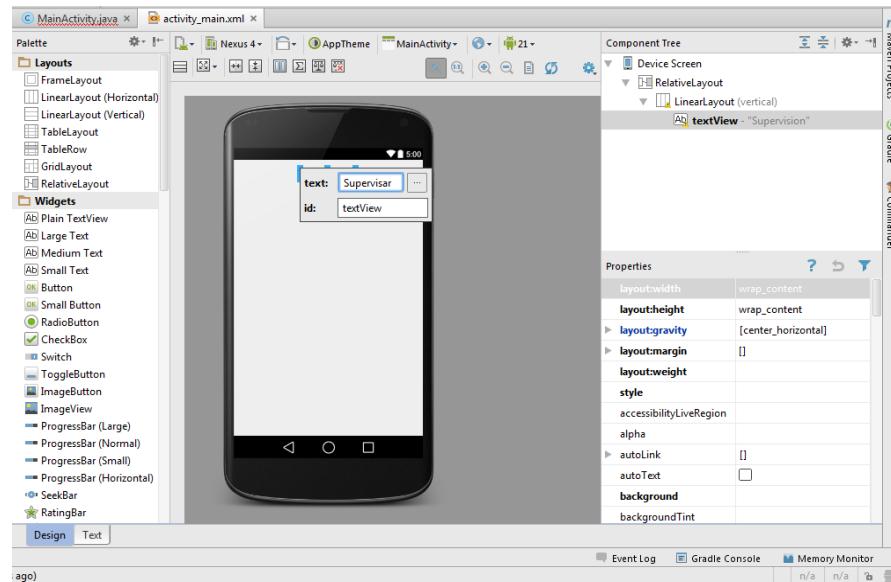


Figura 65 Widge LargeText

Elegimos luego un *ImageView()*, en las propiedades de la misma elegir desde un directorio la imagen a colocar y referenciar el nombre del método al que responderá.

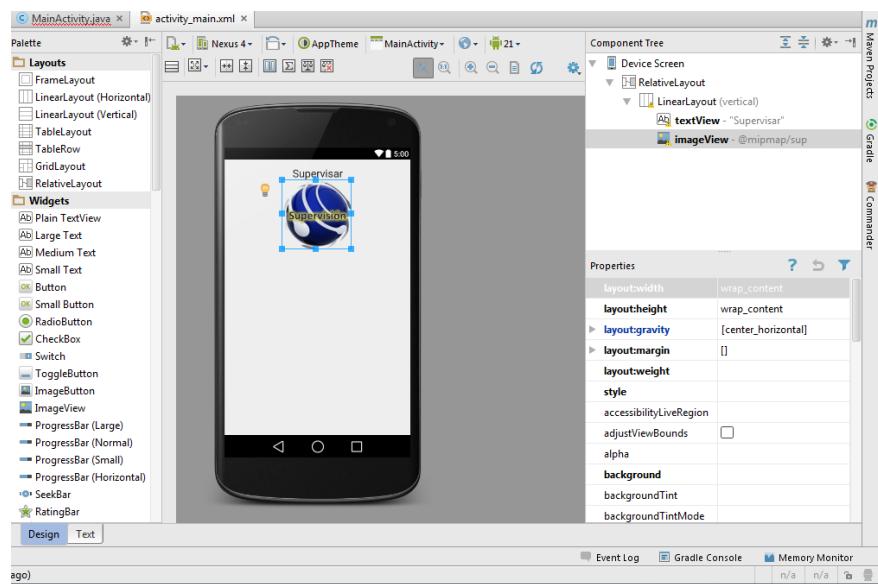


Figura 66 Configuración de ImageView()

Pasando al fichero MainActivity, se va a designar una actividad a cada acción que se proceda en la pantalla del usuario.

```
public void revisar1(View view){
    //El brindamos el dato necesario a Uri
    Uri uriUrl = Uri.parse("http://www.miIP.com");
    //Especificamos la acción a realizar con el ACTION_VIEW
    //para que elija lo mas razonable
    Intent intent = new Intent(Intent.ACTION_VIEW, uriUrl);
    startActivity(intent);
}

public void revisar2(View view){
    Uri uriUrl = Uri.parse("https://www.pushingbox.com/dashboard.php");
    //Especificamos la acción a realizar con el ACTION_VIEW
    //para que elija lo mas razonable
    Intent intent = new Intent(Intent.ACTION_VIEW, uriUrl);
    startActivity(intent);
}

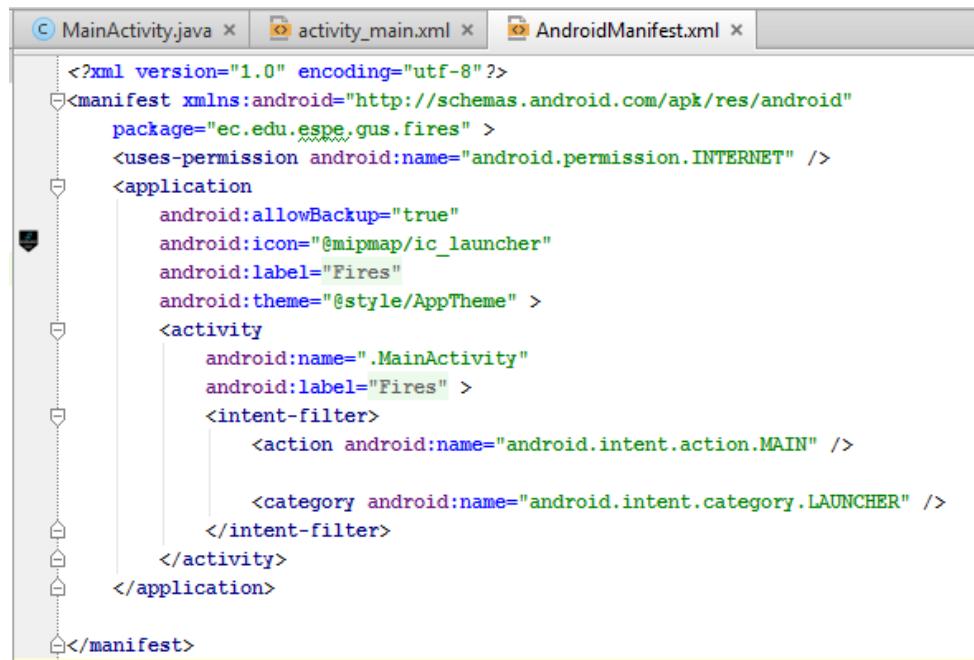
public void finalizar(View view) { finish(); }
```

Figura 67 Métodos del MainActivity

Los controles de tipo Button comandan los métodos mostrados en la Figura 67, en cada uno de ellos creamos un objeto de la clase Intent y le pasamos como parámetros la referencia de la clase Uri. Posteriormente se realiza la llamada al

método starActivity pasando el objeto de la clase Intent para ser visualizado en una nueva pantalla.

Como ultima configuración importante, ya que la aplicación accede a internet es necesario realizar una configuración en archivo “AndroidManifest.xml” figura 68, podemos ubicar este archivo.



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ec.edu.espe.gus.fires" 
    uses-permission android:name="android.permission.INTERNET" />
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Fires"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".MainActivity"
        android:label="Fires" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
</manifest>
```

Figura 68 Configuración de permisos

La interfaz de la aplicación finalizada que da como se muestra en la Figura 69



Figura 69 Interfaz de la App

Es posible obtener un pack de iconos para la aplicación desde la web de Github como lo indica la Figura 70.

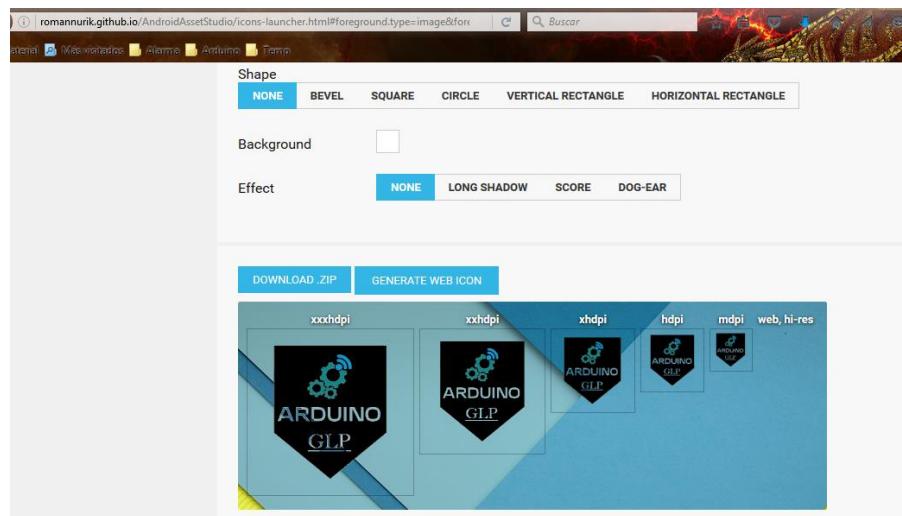


Figura 70 Íconos personalizados de la App

La notificación para el sistema operativo Android se la consigue instalando la App Newtify, disponible en la tienda Play Store. La web de Pushingbox ofrece el servicio de notificaciones a los diferentes sistemas celulares, e-mail y redes sociales

como lo indica la figura 71, en este se puede crear servicios y escenarios para interactuar con los componentes del servidor. El servicio genera un Token que se debe registrar en la aplicación Android.

The screenshot shows a web-based interface for managing services. At the top, there's a header 'Add a service' with a sub-instruction: 'Here you can select one of the various services that are available on PushingBox.' Below this is a blue button labeled 'Add a service'. The main section is titled 'Your services' with the sub-instruction: 'Here you can find the services you have subscribed to.' It lists two services: 'Email Gustavo Ramirez' (with an icon of an eagle) and 'Arduino-serv' (with an icon of a computer monitor and a flame). Each service entry includes its name, email or ID, and 'Edit' and 'Delete' buttons.

Figura 71 Servicio notificación Android

Cada escenario genera un ID que lo identifica dentro del servicio.

Your virtual scenarios

Here you can find all your scenarios previously created. Click on Manage to add actions.

Scenario name	DeviceID	
Demo scenario	v07AA37FE683E1CA	Test Manage Delete
Planta1	v2EB89B23CCD801D	Test Manage Delete
Planta2	v8812C652E181E16	Test Manage Delete
Planta3	vC3D8879144BE41A	Test Manage Delete

Figura 72 Escenarios para App notificación

La comunicación entre el servidor Arduino y el servicio de Pushingbox se lo realiza por medio de su API, la programación dentro del Sketch de Arduino se muestra en la figura 73, cada vez que se genera una medida que supera al umbral permitido, se hace un llamado a la función lanza la notificación de acuerdo al ID del escenario que recibe.

```

void sendToPushingBox(char devid[]){
    cliapp.stop();
    Serial.println("Conectando server notificacion...");
    // Conectando al servidor de notificaciones
    if (cliapp.connect(serverName, 80)) {
        if(DEBUG){Serial.println("Conectado");}
        Serial.println("Enviando request...");
        // Enviando datos al API Pushingbox
        cliapp.print("GET /pushingbox?devid=");
        cliapp.print(devid);
        cliapp.println(" HTTP/1.1");
        cliapp.print("Host: ");
        cliapp.println(serverName);
        cliapp.println("User-Agent: Arduino");
        cliapp.println();
    }
    else {
        Serial.println("Conexion fallo...");
    }
}

```

Figura 73 Comunicación API Pushingbox

La notificación en un terminal Android se visualiza como lo muestra la figura 74, a más de la notificación push se reproduce en audio el contenido del mensaje.

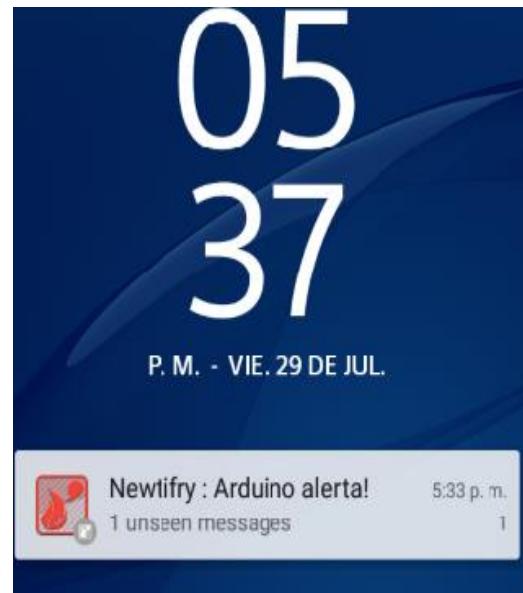


Figura 74 Notificación en Android

CAPÍTULO IV

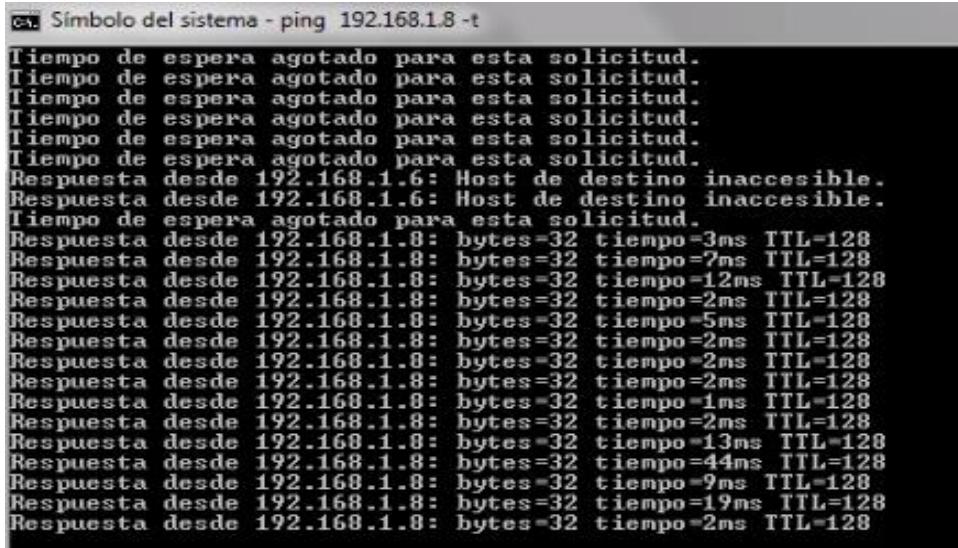
ANÁLISIS DE RESULTADOS

4.1 Resultados del sistema

En este capítulo se presentan los resultados de las pruebas que se realizaron al sistema implementado, con la captura de imágenes que exponen la conducta del sistema ante el cambio de la variable medida, desde el punto de vista del cliente y el propio servidor configurado en Arduino.

4.1.1 Puesta en marcha del sistema

Al encender el prototipo, el servidor empieza autenticándose en la red, desde un equipo de la misma red se puede probar la conexión con el servidor, en la figura 75 se observa el instante en que el servidor es reconocido desde la consola del sistema de Windows y un Sniffer que indica el intercambio de paquetes entre los nodos.



Time	Source	Destination	Protocol	Length	Info
3.029555	192.168.1.6	192.168.1.8	ICMP	74	Echo (ping) request id=0x0001, seq=997/58627, ttl=128 (reply in 36)
3.039076	192.168.1.8	192.168.1.6	ICMP	74	Echo (ping) reply id=0x0001, seq=997/58627, ttl=128 (request in 35)

Figura 75 Reconocimiento del servidor en la red local

Desde el punto de vista del servidor se verifica en la figura 76 el proceso que se sigue para establecer la comunicación.

```
Iniciando la SD card...
SUCCESS - SD card iniciada
SUCCESS - index.htm file hallada
SUCCESS - dispositivo IP
My direccion IP: 192.168.1.8
```

Figura 76 Proceso de inicialización del servidor

Una vez que el servidor este en marcha pasa al estado de escuchar clientes para atender sus requerimientos. En el caso de los clientes locales, es el cliente quien dispone cuando recibir la información o hacer lecturas; en la figura 77 se indica los mensajes de requerimiento de datos hacia el servidor.

GET /ajax_inputs&nocache=688898.8665909885 HTTP/1					
Time	Source	Destination	Protocol	Length	Info
27.531592	192.168.1.6	192.168.1.8	HTTP	434	GET /ajax_inputs&nocache=210416.648688466 HTTP/1.1

Figura 77 Requerimiento de datos hacia el servidor

El conjunto de datos que emite el servidor se agrupan en tramas de 60 Bytes según la capa de acceso de red para luego ser re ensamblados por el cliente y presentados al usuario, el número de paquetes depende de la cantidad de datos transmitidos como lo muestra la figura 78 capturada de un Sniffer.

Time	Source	Destination	Protocol	Length	Info
5.551147	192.168.1.8	192.168.1.6	TCP	60	[TCP segment of a reassembled PDU]
5.551150	192.168.1.8	192.168.1.6	TCP	60	[TCP segment of a reassembled PDU]
5.551152	192.168.1.8	192.168.1.6	TCP	60	[TCP segment of a reassembled PDU]
5.551152	192.168.1.8	192.168.1.6	TCP	60	[TCP segment of a reassembled PDU]
5.551154	192.168.1.8	192.168.1.6	TCP	60	[TCP segment of a reassembled PDU]
5.551154	192.168.1.8	192.168.1.6	TCP	60	[TCP segment of a reassembled PDU]
5.551156	192.168.1.8	192.168.1.6	TCP	60	[TCP segment of a reassembled PDU]
5.551668	192.168.1.6	192.168.1.8	TCP	54	6757 → 80 [ACK] Seq=1 Ack=6200 Win=17171 Len=0

Figura 78 Tramas de datos del servidor

Para el caso del cliente remoto, es el servidor Arduino quien dispone cuando enviar datos, al enviarlos y si han sido aceptados por el cliente la figura 79 indica el mensaje que recibe con su contenido.

```

Posting your variables          // variables publicadas
HTTP/1.1 200 OK                // requerimiento exitoso
Server: nginx
Date: Fri, 29 Jul 2016 21:07:33 GMT
Content-Type: application/json
Transfer-Encoding: chunked
Connection: close
Vary: Accept-Encoding
Vary: Accept
Allow: POST, OPTIONS

16
[{"status_code": 201}]           // fue creado con éxito la petición
0                                // + un elemento - creado.

```

Figura 79 Confirmación de recepción de datos por usuario remoto

Los sensores necesitan un intervalo de tiempo entre 5 a 10 minutos como etapa de calentamiento para trabajar con normalidad, después de transcurrido este tiempo se observa un valor promedio de ppm en cada sensor mostrado en las figuras 80 a 82.



Figura 80 Señal estable del sensor 1



Figura 81 Señal estable del sensor 2



Figura 82 Señal estable del sensor 3

La base de datos creada por cada grafica como se observa en la figura 83 tomada de la planta 1, puede ser exportada en un archivo .csv para respaldarlo, realizar análisis posteriores o graficarlos en otros sistemas de cómputo como Microsoft Excel.

Raw data	
Date	Value
2016-08-01 12:46:06 -0500	1.4077
2016-08-01 12:46:02 -0500	1.51547
2016-08-01 12:45:59 -0500	1.51547
2016-08-01 12:45:56 -0500	1.51547
2016-08-01 12:45:52 -0500	1.51547
2016-08-01 12:45:49 -0500	1.51547
2016-08-01 12:45:45 -0500	1.62791
2016-08-01 12:45:42 -0500	1.62791
2016-08-01 12:45:39 -0500	1.62791
2016-08-01 12:45:35 -0500	1.62791
2016-08-01 12:45:32 -0500	1.51547
2016-08-01 12:45:29 -0500	1.51547
2016-08-01 12:45:25 -0500	1.51547
2016-08-01 12:45:22 -0500	1.51547

Figura 83 Base de datos de Planta 1

C2	A	B	C	D	E
1	created_at,timestamp,value				
2	2016-08-01 17:46:06.178000,147007356178,1.4077				
3	2016-08-01 17:46:02.951000,1470073562951,1.51547				
4	2016-08-01 17:45:59.272000,1470073559272,1.51547				
5	2016-08-01 17:45:56.123000,1470073556123,1.51547				
6	2016-08-01 17:45:52.980000,1470073552980,1.51547				
7	2016-08-01 17:45:49.110000,1470073549110,1.51547				
8	2016-08-01 17:45:45.952000,1470073545952,1.62791				
9	2016-08-01 17:45:42.365000,1470073542365,1.62791				
10	2016-08-01 17:45:39.204000,1470073539204,1.62791				
11	2016-08-01 17:45:35.812000,1470073535812,1.62791				
12	2016-08-01 17:45:32.444000,1470073532444,1.51547				
13	2016-08-01 17:45:29.152000,1470073529151,1.51547				
14	2016-08-01 17:45:25.998000,1470073525998,1.51547				
15	2016-08-01 17:45:22.608000,1470073522608,1.51547				
values 5792d7d27625421ce02f6531					

Figura 84 Base de datos de Planta 2 exportada a Microsoft Excel

Para determinar la capacidad del enlace mínimo necesario que soporte la transferencia de información entre un cliente y el servidor, se analiza las tramas capturadas por el Sniffer asociados a la unidad de tiempo, en la figura 85 se observa la transmisión de una trama inicial en un tiempo determinado y en la figura 86 se observa la transmisión de una trama final en un tiempo determinado, cuando se ha detectado una concentración que supera al umbral permitido por el sistema.

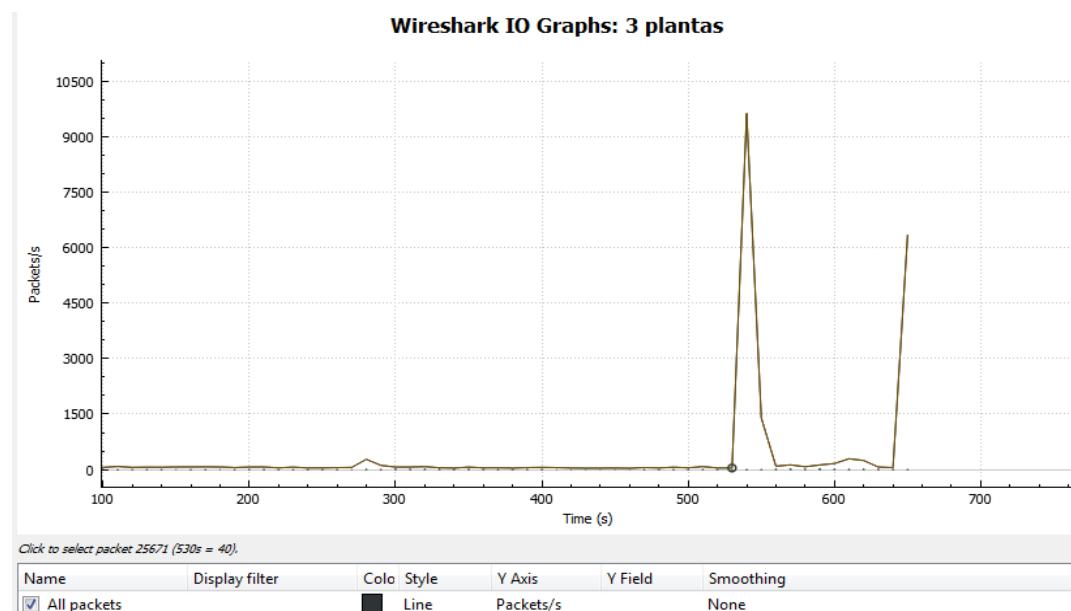


Figura 85 Transmisión de un paquete inicial

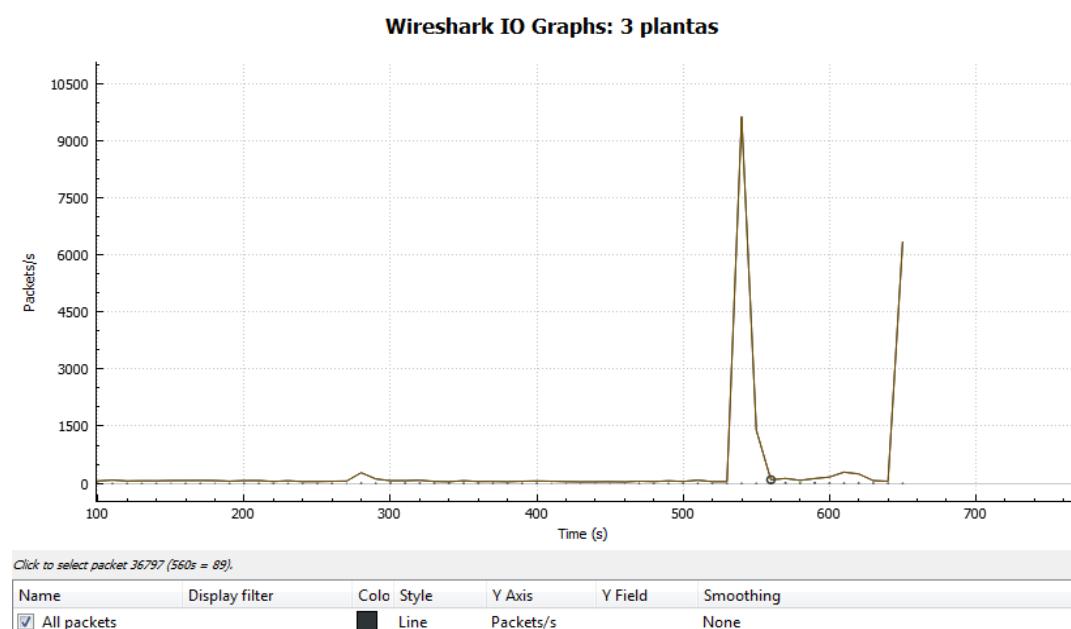


Figura 86 Transmisión de un paquete final

El ancho de banda necesario por el enlace obedece a remplazar los datos antes captados, figura 87 en la ecuación 8.

Inicial: *Click to select packet 25671 (530s = 40).*

Final: *Click to select packet 36797 (560s = 89).*

Figura 87 Paquetes inicial-final en transmisión

$$W (\text{Bps}) = \text{carga útil (Bytes)} / \text{tiempo de transmisión (s)} \quad \text{Ecuación 8}$$

Reemplazando los valores, entonces:

$$W (\text{Bps}) = (36797 - 25671) * 60 [\text{Bytes}] / (560 - 530) [\text{s}]$$

$$W = 22252 [\text{Bps}]$$

Es la capacidad mínima que requiere el enlace para abastecer la transferencia de datos que requiere el cliente cada segundo. Al encontrarse el servidor y el cliente en la misma red local con las velocidades actuales de una red Ethernet en el orden de los Gbit/s y el tiempo de procesamiento computacional siendo para este caso de 62.5 ns por instrucción, la aplicación desde el lado del cliente se puede decir que es un sistema en tiempo real y este debería ser dependiente mejor de la capacidad del transductor utilizado.

4.1.2 Comportamiento del sistema a exposición de GLP

A continuación se muestran los resultados de exponer de manera individual cada una de las plantas que contienen un sensor de gas, figura 88 a 90. El sistema inicia con valores iniciales mínimos.

Sensor:1	voltaje:0.30	Rs:15500.00	glp:7.20 ppm
Sensor:2	voltaje:0.49	Rs:9128.71	glp:23.73 ppm
Sensor:3	voltaje:0.58	Rs:7669.49	glp:35.12 ppm

Figura 88 Valores iniciales desde el servidor



Figura 89 Valores iniciales desde el cliente local



Figura 90 Valores iniciales desde el cliente remoto

4.1.2.1 Exposición de la Planta 1 a GLP

Escotilla abierta

La primera prueba se la realiza con la escotilla que conecta a la planta uno con la planta dos y tres abierta, permitiendo la circulación de corriente de aire. La figura 91 indica cómo se eleva gradualmente la concentración de gas medida en la planta uno y los ventiladores se hallan apagados.

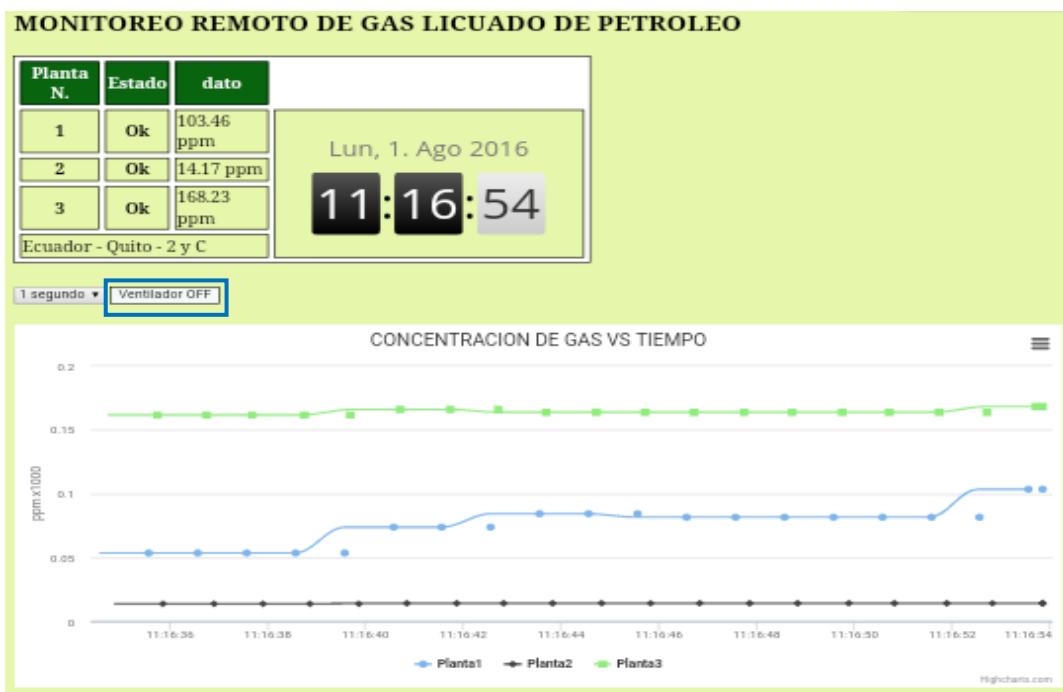


Figura 91 Incremento de gas en Planta 1

La figura 92 muestra el intervalo de tiempo que se mantiene la concentración de gas superior al umbral permitido de los 9000 ppm hasta ser controlada, aproximadamente 42 segundos tomados de la base de datos de la gráfica.

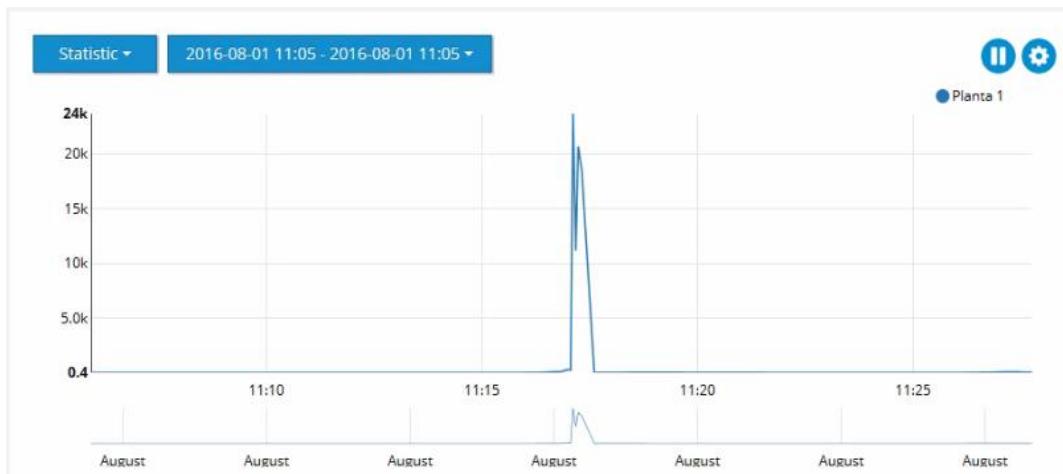


Figura 92 Concentración de gas en el tiempo Planta 1

La figura 93 indica cómo se mantiene por un instante de tiempo la concentración de gas en la planta uno, además que los ventiladores pasan a ser activados.



Figura 93 Señal de concentración de gas de Planta 1

Las figuras 94 a 96 indican las notificaciones que son recibidas por el usuario debido a la concentración excedida de gas en la planta 1.

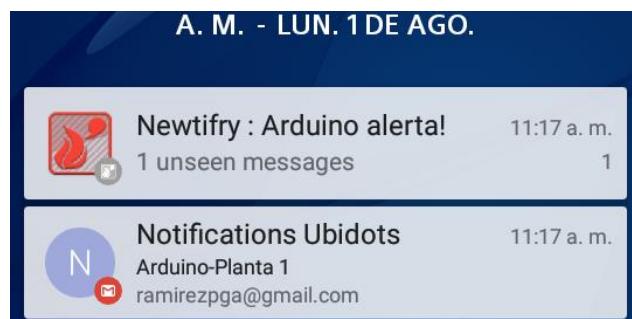


Figura 94 Notificaciones recibidas por el teléfono celular del usuario

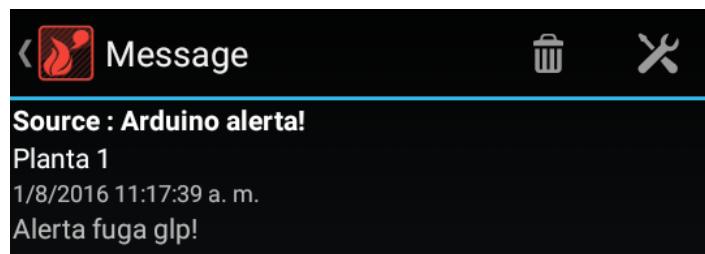


Figura 95 Notificación para Android



Figura 96 Notificación a e-mail

Escotilla cerrada

La segunda prueba se la realiza con la escotilla que conecta a la planta uno con la planta dos y tres cerrada, privando la circulación de corriente de aire. La figura 97 muestra el intervalo de tiempo que se mantiene la concentración de gas superior al umbral permitido de los 9000 ppm hasta ser controlada, aproximadamente 4 segundos tomados de la base de datos de la gráfica.



Figura 97 Concentración de gas en Planta 1

La figura 98 indica la señal del primer pico captado en la figura 97 con una duración aproximada de 4 segundos.

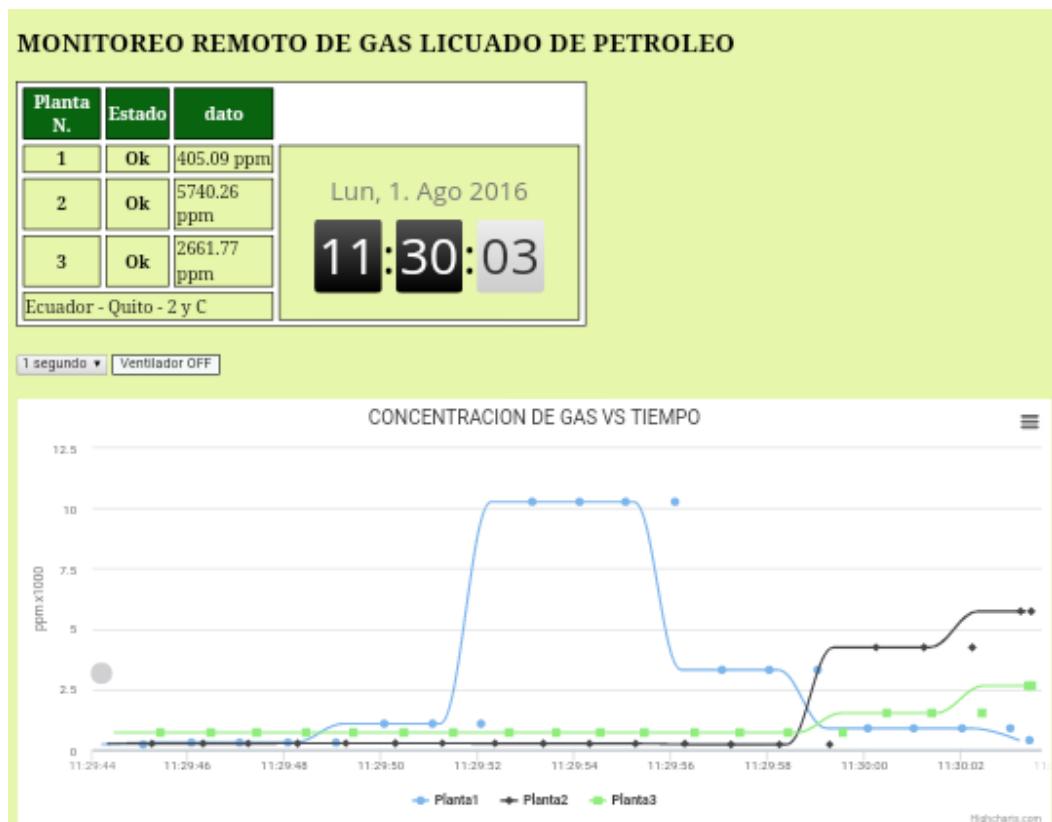


Figura 98 Concentración de gas en Planta 1

Las figuras 99 a 101 indican las notificaciones que son recibidas por el usuario debido a la concentración excedida de gas en la planta uno.

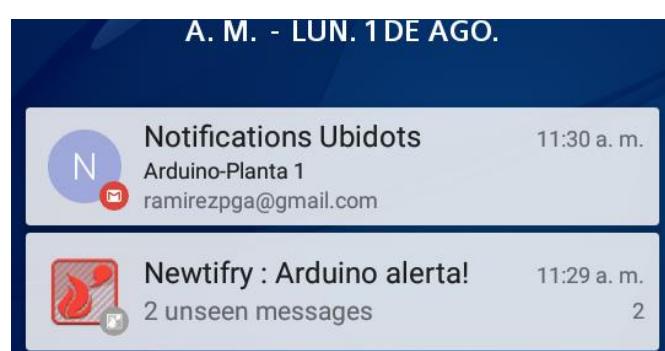


Figura 99 Notificaciones recibidas por el teléfono celular del usuario

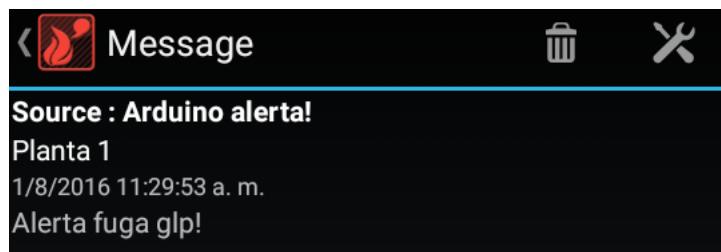


Figura 100 Notificación para Android

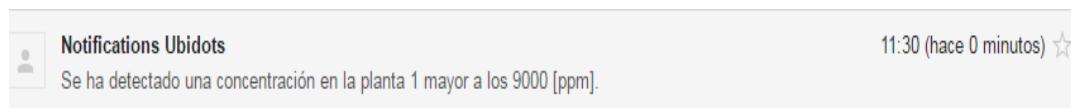


Figura 101 Notificación a e-mail

4.1.2.2 Exposición de la Planta 2 a GLP

Escotilla abierta

La primera prueba se la realiza con la escotilla que conecta a la planta dos con la planta uno y tres abierta, permitiendo la circulación de corriente de aire. La figura 102 indica cómo se eleva gradualmente la concentración de gas en esta planta en dos intentos realizados, pero a medida que transcurre el tiempo el gas se disipa hacia las plantas uno y tres como lo muestra la figura 103 y 104.



Figura 102 Concentración de gas en Planta 2

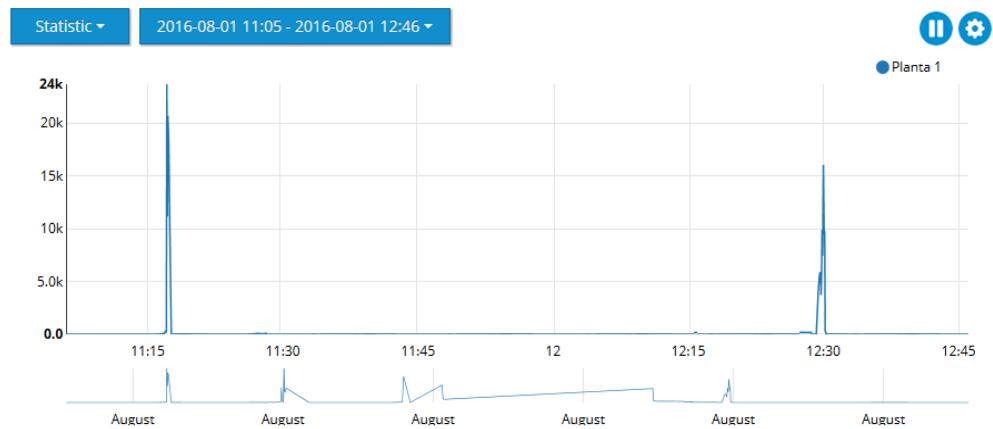


Figura 103 Concentración de gas en Planta 1



Figura 104 Concentración de gas en Planta 3

Escotilla cerrada

La segunda prueba se la realiza con la escotilla que conecta a la planta dos con la planta uno y tres cerrada, privando la circulación de corriente de aire. La figura 105 indica cómo se eleva gradualmente la concentración de gas medida en la planta dos y los ventiladores se hallan apagados.

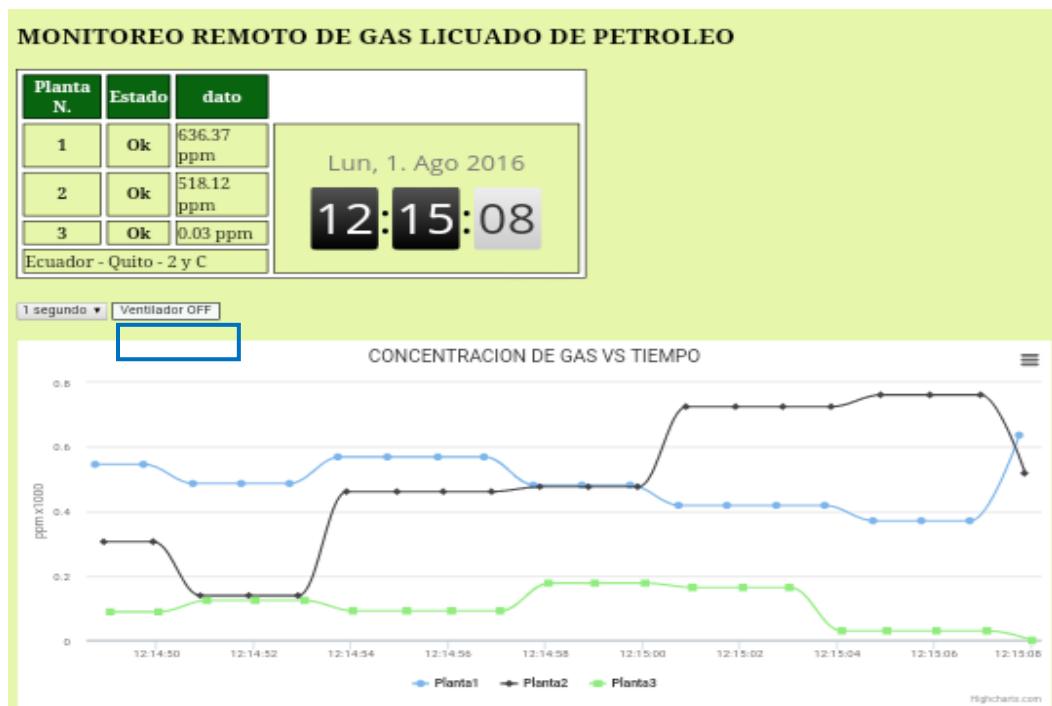


Figura 105 Incremento de gas en Planta 2

La figura 106 muestra el intervalo de tiempo que se mantiene la concentración de gas superior al umbral permitido de los 9000 ppm hasta ser controlada, aproximadamente 21 segundos tomados de la base de datos de la gráfica.

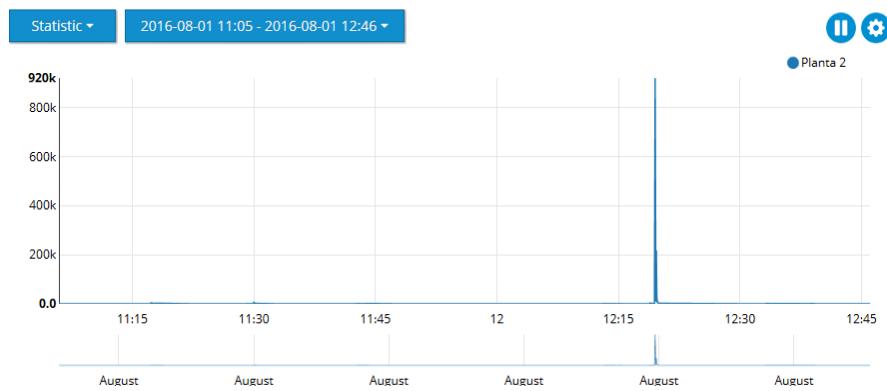


Figura 106 Concentración de gas en el tiempo Planta 2

Pese a ser superado el umbral de detección en esta planta, las otras plantas también se hallan contaminado de gas como lo muestra la figura 107.

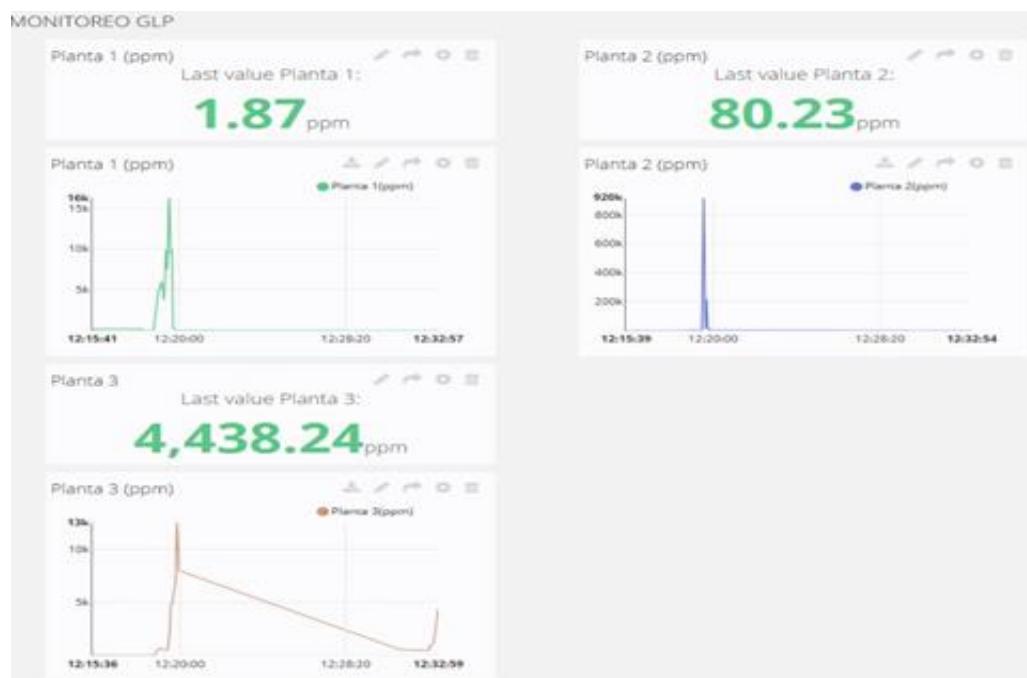


Figura 107 Contaminación de todas las plantas debido a la planta 2

De todos modos el usuario recibe la notificación de cada una de las plantas, en este caso la figura 108 y 109 indica las notificaciones correspondientes a la planta dos.

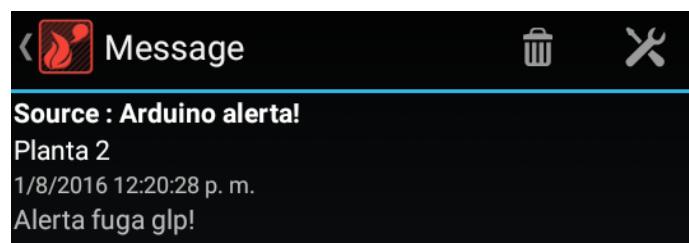


Figura 108 Notificación para Android

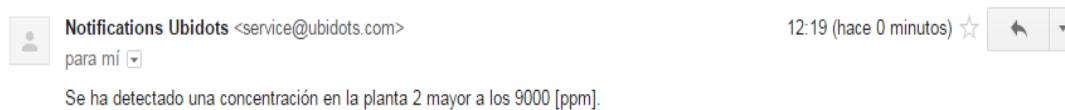


Figura 109 Notificación a e-mail

4.1.2.3 Exposición de la Planta 3 a GLP

Escotilla abierta

La primera prueba se la realiza con la escotilla que conecta a la planta tres con la planta uno y dos abierta, permitiendo la circulación de corriente de aire. La figura 110 muestra el intervalo de tiempo que se mantiene la concentración de gas superior al umbral permitido de los 9000 ppm hasta ser controlada, aproximadamente 6,5 minutos tomados de la base de datos de la gráfica.



Figura 110 Concentración de gas en el tiempo Planta 3

Las figuras 111 y 112 indican las notificaciones que son recibidas por el usuario debido a la concentración de gas excedida en la planta tres.

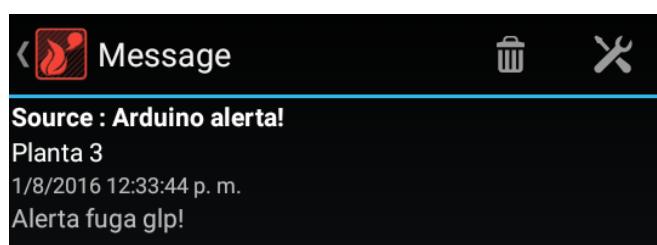


Figura 111 Notificación para Android

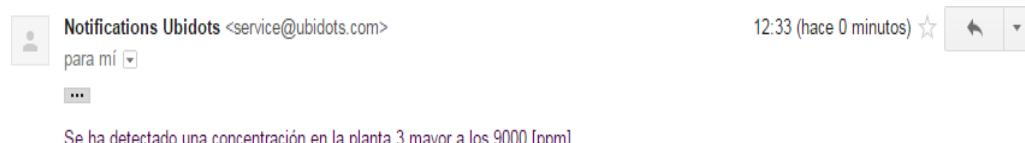


Figura 112 Notificación a e-mail

Escotilla cerrada

La segunda prueba se la realiza con la escotilla que conecta a la planta tres con la planta uno y dos cerrada, privando la circulación de corriente de aire. La figura 113 muestra el intervalo de tiempo que se mantiene la concentración de gas superior al umbral permitido de los 9000 ppm hasta ser controlada, aproximadamente 4 minutos tomados de la base de datos de la gráfica.

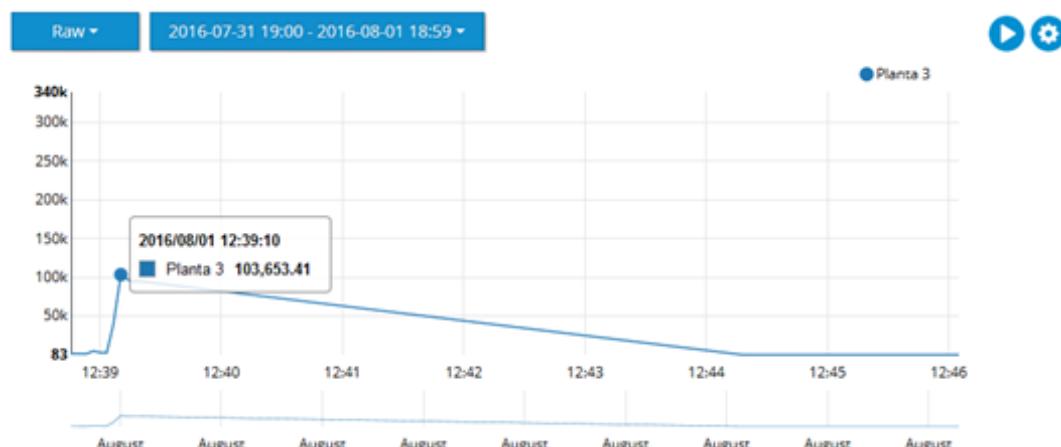


Figura 113 Concentración de gas en el tiempo Planta 3

Las figuras 114 y 115 indican las notificaciones que son recibidas por el usuario debido a la concentración de gas excedida en la planta tres.

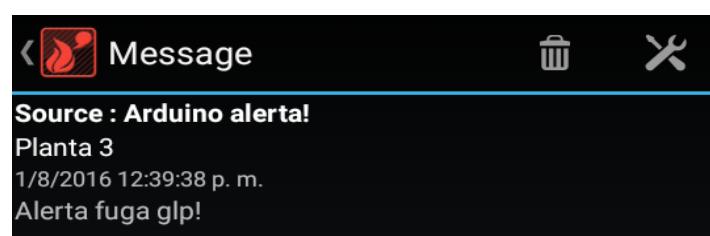


Figura 114 Notificación para Android



Figura 115 Notificación a e-mail

4.2 Discusión de los resultados

Esta investigación tuvo como propósito diseñar un prototipo cuya implementación permite detectar la concentración de gas GLP dentro de un perímetro residencial, tomando medidas de control sobre el sistema y alertando al usuario cuando las lecturas tomadas por los sensores de gas exceden los límites definidos.

Los resultados obtenidos en la investigación evidencian su cometido, las capturas de pantalla tomadas de las ventanas de monitoreo muestran como el sistema mantiene constantemente la lectura de datos en los diferentes compartimentos. Las gráficas pertenecientes al monitor o usuario local representan los datos de la variable medida con mayor resolución, es decir que se puede asimilar fácilmente el incremento o decremento de gas en el ambiente con la desventaja de tener un limitado número de datos presentes en pantalla; el análisis del ancho de banda necesario para la transmisión de datos en la red local y las características de la tarjeta de procesamiento convierten al prototipo en un sistema en tiempo real cuando está ubicado en la red LAN.

Cada compartimento de la maqueta dispone de escotillas que bloquean o permiten la circulación de corrientes de aire. Se realizaron dos pruebas en cada uno de los compartimentos con las escotillas abiertas y cerradas, para el análisis de estos resultados tiene mayor utilidad los datos tomados por el cliente remoto ya que alberga mayor cantidad de información en una sola pantalla y se puede consultar el valor de un dato en función del tiempo en su base de datos.

Los escenarios en el cual las escotillas se encuentran abiertas presentan graficas cuyos datos toman una pendiente positiva para superar el límite umbral, de hecho

esto concuerda con estudios realizados (Honeywell, 2016), pues al estar presente una corriente de aire en el escenario incide a que el gas se disperse gradualmente en el ambiente y el sensor toma medidas acorde al cambio de la variable en todo el compartimento. El intervalo de tiempo que permanece el gas en el compartimento hasta ser expulsado resulta ser mayor en comparación con las gráficas de la escotilla cerradas, debido a que se produce mayor dispersión en cuanto a la cantidad de gas en el ambiente hasta el punto de llegar a contaminar los otros compartimentos como lo evidencia la prueba en el compartimento dos; aquí también se percibe otro efecto el cual se debe a que el gas siendo más pesado que el aire al contaminar el segundo compartimento el gas desciende y es el primer compartimento quien resulta afectado.

Al realizar las pruebas con las escotillas cerradas las pendientes de las gráficas resultan más erguidas, las medidas del gas no se elevan gradualmente y nuevamente este dato concuerda con los estudios ya realizados (Gianfranco, 2013), el gas se acumula en forma de nubes y no se dispersa en el ambiente; así cuando una alta concentración es captada por el sensor, en las gráficas se observa cómo se supera el umbral de manera muy acelerada. En este caso el intervalo de tiempo que permanece el gas en el compartimento es menor por la poca cantidad de gas en el ambiente.

Las notificaciones se envían al usuario de manera simultánea que se activa el sistema para drenar el gas de los compartimentos. Comparando los dos sistemas de notificación implementados; el correo electrónico y notificación push al terminal Android, el medio de notificaciones por correo electrónico resulta más robusto pues la comparación con el umbral definido para lanzar la alarma lo hace con cada dato obtenido del sistema que se encuentra en su base de datos en la nube, lo que no ocurre con la notificación push por lo que en ciertas ocasiones no asegura conectarse con el servidor de notificaciones Pushingbox.

CAPÍTULO V

CONCLUSIONES Y LINEAS FUTURAS DE INVESTIGACIÓN

5.1 Conclusiones

Fue posible implementar un prototipo basado en la plataforma Arduino con componentes electrónicos compatibles, el cual cuenta con funciones necesarias para detectar y notificar la concentración de gas GLP.

La configuración de la tarjeta Arduino como servidor web se llevó a cabo gracias a la Shield Ethernet que permite conectarse directamente a un enrutador casero para transmitir datos sin la necesidad de hardware o sistema operativo adicional.

La programación de Sketch de Arduino requirió importar librerías ajenas a su biblioteca y API's con el fin de establecer comunicación con los servicios utilizados en la nube, obtenidos de las páginas de los desarrolladores y adaptados al código fuente.

Para el diseño del prototipo creado se acató normativas para instalaciones de gases combustibles de entidades nacionales como de importantes empresas internacionales ligadas al tema.

El planteamiento de unidades de medidas y límites umbrales permitidos de GLP, fueron establecidos de acuerdo a criterios adoptados de normativas de manipulación y prevención de desastres; exclusivos del gas licuado de petróleo debido a su composición y características químicas.

Fue necesario acondicionar la señal de los sensores de gas, ya que la relación entre la lectura analógica y el valor real no es lineal. Por lo que se necesitó estimar la curva que ofrece el datasheet para hallar una ecuación que gobierna al sensor de acuerdo al rango de lectura de la variable que interesa medir.

El almacenamiento e interpretación de información de los sensores en tiempo real gracias a los servicios en la nube, hicieron posible la creación de aplicaciones para el Internet de las Cosas de una manera fácil y rápida.

Se encontró diferencia significativa en el comportamiento de gas cuando el área contaminada presenta corrientes de aire.

5.2 Recomendaciones

Adquirir elementos electrónicos del mismo fabricante, en el caso de los sensores resulta tedioso calibrarlos para trabajar con el mismo código de programación.

Para evitar acumulación de gas en los pisos de niveles superiores al momento de drenar el sistema, utilizar ductos independientes para cada compartimento.

5.3 Trabajos futuros

Como continuación de este trabajo de titulación quedan abiertas diversas líneas de investigación para dar continuidad al esfuerzo invertido. Durante todo el desarrollo de esta tesis los objetivos han sido aspectos de diseño del prototipo, funcionalidad y su interconexión. Los principales tópicos pueden resumirse en los puntos siguientes:

Realizar una validación sobre de la implementación de un sistema de comunicación basado en tecnología inalámbrica para la ubicación de uno o más sensores en el área de medición, como puede ser el estándar IEEE 802.15.4 de redes inalámbricas de área personal (wireless personal area network, WPAN).

Debido a que este proyecto ha sido diseñado considerando un escenario ideal, se puede generalizar su utilidad atendiendo otras variables que afectan su funcionamiento como lo son la temperatura, humedad y concentración de oxígeno en el ambiente.

Referencias

- ARDUINO. (2016). *Genuino*. Obtenido de <https://www.arduino.cc>
- Casella. (2016). *Prevención de riesgos y detección de gas*. Obtenido de http://issga.xunta.es/export/sites/default/recursos/descargas/documentacion/material-formativo/relatorios/2013_05_XT_Espazos_confinados_Ivxn_Casella_Espax_a.pdf
- Colmenares. (2016). *Diseño e implementación de un prototipo para detectar fugas de gas para uso residencial*. Bogotá: Universidad de San Buenaventura .
- Crespo, E. (2016). *Aprendiendo Arduino*. Obtenido de <https://aprendiendoarduino.wordpress.com>
- Datasheet. (2016). *Sensor MQ-6*. Obtenido de <http://www.probots.co.in/Datasheets/MQ6.pdf>
- Dorame, K. (2016). *Conceptos Generales de Gases, Termoquímica y Electroquímica*. Obtenido de <http://quimicatecnologicodemexicali.blogspot.com/2012/12/unidad-v-conceptos-generales-de-gases.html>
- Drager. (2016). *Introducción a los Sistemas de Detección de Gases*. Alemania.
- Drägerwerk. (2016). *Completo sistema de detección de gas para su edificio*. Obtenido de http://www.draeger.com/sites/assets/PublishingImages/Products/gds_variogard_central_unit/ES/9094083_VarioGard_System_BR_ES.pdf
- Drake, J. (2015). *Introducción a los sistemas de instrumentación*. Santander: U. Cantabria.
- Escalona, L. (2012). Los sensores químicos y su utilidad en el control de gases contaminantes. *Revista Ingeniería UC*, 1.
- FjRamire. (2016). *Arduino Ethernet Shield*. Obtenido de <http://www.tuelectronica.es/tutoriales/arduino/arduino-ethernet-shield.html>
- Gianfranco. (2013). *Desarrollo de un prototipo para la detección de fugas de gas en tuberías*. Venezuela: Universidad Nueva Esparta.
- Honeywell, S. (2016). *Notas generales para la calibración, mantenimiento e instalación de los detectores de GAS*. Barcelona.

- INDURA. (2009). *Hoja de datos de seguridad*. Santiago.
- INEN. (2010). *Instalaciones de gases combustibles para uso residencial*. Quito: NTE INEN 2.
- Menéndez. (2016). *Android 100%*. México.
- Murco. (2016). *murcogasdetection*. Obtenido de <http://www.murcogasdetection.com/es/assets/pdfs/FAQ/LevelsToDetect.pdf>
- Parziale, L. (2011). *TCP/IP Tutorial*. USA: IBM.
- Pemex. (2016). *Hoja de datos de seguridad para sustancias químicas*. Obtenido de http://www.gas.pemex.com/NR/rdonlyres/D3D851A9-FDE6-4F68-8FD1-3CC6E50163E4/0/HojaSeguridadGasLP_v2.pdf
- Planificación, S. (2016). *Combustibles utilizados por los hogares para cocinar*. Obtenido de <http://datos.quito.gob.ec/datastreams/88/combustibles-utilizados-por-los-hogares-para-cocinar/>
- Repsol. (2016). *Los gases licuados del petróleo*. Obtenido de https://www.repsol.com/pe_es/productos_y_servicios/productos/glp_butano_y_propano/guia_de_los_glp/tipos_de_combustibles/gases_licuados/
- SECNORMATIVA, C. (2016). *Subsecretaría EFR*. Obtenido de http://www.sec.cl/pls/portal/docs/PAGE/SECNORMATIVA/COMBUSTIBLES_DECRETOS/DS%2066.PDF
- Tecnolowikia. (2016). *Gases Combustibles*. Obtenido de <https://tecnolowikia.wikispaces.com/Gases+Combustibles>
- Testo. (2016). *Unidades de medida*. Obtenido de <http://www.academiatesto.com.ar/cms/unidades-de-medida>
- Ubidots. (2016). *Internet de las Cosas con Ubidots*. Obtenido de http://ubidots.com/docs/es/get_started/introduccion.html
- Villa, G. (2010). *PROTOTIPO PARA LA DETECCIÓN DE FUGAS*. Caracas: U NUEVA ESPARTA.
- Wikipedia. (2016). *HTML*. Obtenido de [HTML](#)