

**UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS**

**FACULTAD DE INGENIERÍA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



Presentado por

**ESTEFANIE HERMINIA NERALY APAZA ROMAN**

**IMPLEMENTAR UN PROTOTIPO PARA MONITOREAR  
EL NIVEL DE GAS EN BALONES DOMESTICOS**

Asesor:

**Msc. JUAN JOSÉ ORE CERÓN**

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE  
INGENIERO DE SISTEMAS**

**ANDAHUAYLAS – APURÍMAC–PERU**

**2019**



## APROBACION DEL ASESOR

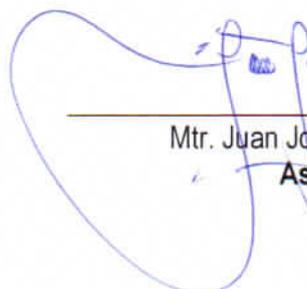
Quién suscribe:

Mtr. Juan José Oré Cerrón, docente asociado adscrito al Departamento Académico de Ingeniería y Tecnología Informática por la presente:

**CERTIFICA,**

Que, el Bachiller en Ingeniería de Sistemas Estefanie Herminia Nerally Apaza Román ha culminado satisfactoriamente el Proyecto de Tesis intitulado: **"IMPLEMENTAR UN PROTOTIPO PARA MONITOREAR EL NIVEL DE GAS EN BALONES DOMESTICOS"** para optar el Título Profesional de Ingeniero de Sistemas.

Andahuaylas, 14 de octubre del 2019



---

Mtr. Juan José Oré Cerrón  
**Asesor**



---

Bach. Estefanie Herminia Nerally Apaza Román  
**Tesista**



## FACULTAD DE INGENIERÍA

### ACTA DE SUSTENTACIÓN DE TESIS

En la Av. José María Arguedas del Local Académico SL01 (Ccoyahuacho) en el auditorio de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional José María Arguedas ubicado en el distrito de San Jerónimo de la Provincia de Andahuaylas, siendo las 02:30 horas del día 24 de julio del año 2019, se reunieron los docentes: Dra. Cecilia Edith García Rivas Plata, Dr. Yalmar Temistocles Ponce Atencio, Dr. Julio César Huanca Marín, en condición de integrantes del Jurado Evaluador del Informe Final de Tesis intitulado: "IMPLEMENTAR UN PROTOTIPO PARA MONITOREAR EL NIVEL DE GAS EN BALONES DOMÉSTICOS", cuyo autor es el Bachiller en Ingeniería de Sistemas **ESTEFANIE HERMINIA NERALY APAZA ROMÁN**, el asesor Mtr. Juan José Oré Cerrón, con el propósito de proceder a la sustentación y defensa de dicha tesis.

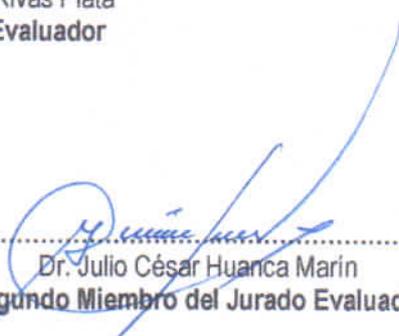
Luego de la sustentación y defensa de la tesis, el Jurado Evaluador **ACORDÓ: aprobar** por **unanimidad** al Bachiller en Ingeniería de Sistemas **ESTEFANIE HERMINIA NERALY APAZA ROMÁN**, obteniendo la siguiente calificación y mención:

Nota escala vigesimal		Mención
Números	Letras	
18	dieciocho	Excelente

En señal de conformidad, se procedió a la firma de la presente acta en 03 ejemplares.

  
.....  
Dra. Cecilia Edith García Rivas Plata  
Presidente del Jurado Evaluador

  
.....  
Dr. Yalmar Temistocles Ponce Atencio  
Primer Miembro del Jurado Evaluador

  
.....  
Dr. Julio César Huanca Marín  
Segundo Miembro del Jurado Evaluador



## APROBACIÓN DEL JURADO DICTAMINADOR

LA TESIS: "IMPLEMENTAR UN PROTOTIPO PARA MONITOREAR EL NIVEL DE GAS EN BALONES DOMESTICOS" para optar el Título Profesional de INGENIERO DE SISTEMAS, ha sido evaluada por el Jurado Dictaminador conformado por:

**PRESIDENTE:** DRA. CECILIA EDITH GARCÍA RIVAS PLATA  
**PRIMER MIEMBRO:** DR. YALMAR TEMISTOCLES PONCE ATENCIO  
**SEGUNDO MIEMBRO:** DR. JULIO CÉSAR HUANCA MARÍN

Habiendo sido aprobado por UNANIMIDAD, en la ciudad de Andahuaylas el día 24 del mes de Julio de 2019.

Andahuaylas, 14 de octubre de 2019.

---

Dra. CECILIA EDITH GARCÍA RIVAS PLATA  
PRESIDENTE DEL JURADO DICTAMINADOR

---

Dr. YALMAR TEMISTOCLES PONCE ATENCIO  
PRIMER MIEMBRO DEL JURADO DICTAMINADOR

---

Dr. JULIO CÉSAR HUANCA MARÍN  
SEGUNDO MIEMBRO DEL JURADO DICTAMINADOR



## DECLARACIÓN JURADA DE AUTENTICIDAD

Yo, **Estefanie Herminia Neraly Apaza Román**, identificado (a) con DNI N° 70685622 de la **Escuela Profesional de ingeniería de Sistemas**. Declaro bajo juramento que el **Proyecto Titulado**: "IMPLEMENTAR UN PROTOTIPO PARA MONITOREAR EL NIVEL DE GAS EN BALONES DOMESTICOS". Es auténtico y no vulnera los derechos de autor. Además, su contenido es de entera responsabilidad del autor (es) del proyecto, quedando la UNAJMA exenta de toda responsabilidad en caso de atentar contra la Ley de propiedad intelectual y derechos de autor.

**Andahuaylas, 14 de Octubre de 2019.**

.....  
Firma

N° DNI: 70685622

E-mail: neralyar@gmail.com

N° Celular:993951436

## **DEDICATORIA**

A mi madre por el esfuerzo, dedicación y su amor incondicional, por enseñarme a no rendirme ante las adversidades.

A mi abuela y mi hermana quienes fueron un soporte fundamental durante estos años, por enseñarme a no desmayar en el camino, por apoyarme en todas las caídas que tuve, por enseñarme que la vida esta echa solo para aquel que saben luchar por sus sueños y que lo mejor está por venir. A ellas estaré eternamente agradecida

*Estefanie Herminia Neraly Apaza Roman*

## **AGRADECIMIENTOS**

Ala mujer que me dio la vida, Marina Román Rosales por todo el esfuerzo que hizo por sus dos hijas, por apoyarme incondicionalmente durante estos años de quien siento una extraordinaria admiración.

A mi familia cercana quienes de una u otra manera me brindaron su apoyo, así mismo a mis amigos que siempre estuvieron en los malos buenos momentos.

Al MSc Juan José Ore Cerón por su apoyo como asesor en el desarrollo de mi proyecto de tesis.

Agradezco a las personas que quedan en el anonimato que influyeron en mi formación con su apoyo y ejemplo de superación.

*Estefanie Herminia Neraly Apaza Roman*

## INDICE GENERAL

INTRODUCCIÓN .....	14
1. CAPITULO I: PROBLEMA DE INVESTIGACION .....	15
1.1. REALIDAD PROBLEMÁTICA.....	15
1.2. FORMULACIÓN DEL PROBLEMA .....	17
1.3. OBJETIVOS .....	17
1.3.1. OBJETIVO GENERAL .....	17
1.3.2. OBJETIVOS ESPECÍFICOS.....	17
1.4. JUSTIFICACIÓN .....	18
1.5. VIABILIDAD DE LA INVESTIGACIÓN .....	19
1.5.1. VIABILIDAD TÉCNICA .....	19
1.5.2. VIABILIDAD ECONÓMICA .....	19
1.5.3. VIABILIDAD OPERATIVA.....	19
1.5.4. VIABILIDAD LEGAL .....	19
1.5.5. VIABILIDAD SOCIAL.....	20
1.6. LIMITACIÓN DEL ESTUDIO.....	20
2. CAPITULO II: ANTECEDENTES .....	22
2.1. ANTECEDENTES INTERNACIONALES .....	22
2.2. ANTECEDENTES NACIONALES.....	24
3. CAPITULO III: MARCO TEÓRICO .....	26
3.1. INTERNET DE LAS COSAS .....	26
3.2. DISPOSITIVO MÓVIL.....	27
3.2.1. TIPOS DE DISPOSITIVOS MOVILES.....	27
3.3. SISTEMAS OPERATIVOS.....	28
3.4. SISTEMAS OPERATIVOS MOVILES .....	28
3.5. APLICATIVO MÓVIL .....	31
3.5.1. APLICACIONES NATIVAS .....	31

3.6.	ANDROID .....	32
3.7.	ANDROID STUDIO .....	33
3.8.	ARDUINO .....	35
3.8.1.	CARACTERÍSTICAS .....	36
3.8.2.	VENTAJAS .....	37
3.8.3.	TIPOS DE ARDUINO .....	38
3.9.	ARDUINO IDE .....	43
3.9.1.	ESCRITURA DE BOCETOS .....	43
3.10.	MODULO BLUETOOTH .....	46
3.11.	CELDA DE CARGA CON TRANSMISOR HX711 .....	47
3.11.1.	TRANSMISOR DE CELDA DE CARGA HX711.....	48
3.12.	GAS PROPANO .....	49
3.13.	METODOLOGÍA .....	49
3.13.1.	METODOLOGÍA V.....	50
4.	CAPITULO IV: METODOLOGÍA DE LA INVESTIGACIÓN.....	53
4.1.	METODOLOGÍA DE DESARROLLO DEL PROYECTO.....	53
4.1.1.	NIVEL1 DESARROLLO: FASE DE ESPECIFICACIONES.....	54
A.	REQUERIMIENTOS FUNCIONALESONAL.....	54
B.	REQUERIMIENTOS NO FUNCIONALES.....	54
C.	ARDUINO NANO.....	55
4.1.2.	NIVEL 2 DESARROLLO: DISEÑO DE ALTO NIVEL .....	56
A.	ESTABLECIMIENTO DE ACTORES.....	56
B.	ESTABLECIENDO USUARIOS.....	57
C.	DIAGRAMA DE CASO DE USO GENERAL.....	57
D.	SUB CASO DE USO 1 .....	58
E.	SUB CASO DE USO 2 .....	59
F.	DIAGRAMA DE DESPLIEGUE .....	59
4.1.3.	NIVEL3 DESARROLLO: DISEÑO EN DETALLE .....	60
A.	DIAGRAMA DE ACTIVIDADES .....	60
4.1.4.	NIVEL 4 DESARROLLO: IMPLEMENTACIÓN .....	63
A.	IMPLEMENTACIÓN DEL HARDWARE .....	63
B.	IMPLEMENTACIÓN DEL SOTFWARE.....	74

<b>5. CAPITULO V: RESULTADOS .....</b>	<b>84</b>
5.1. TEST UNITARIO .....	84
5.1.1. PRUEBA DE HARDWARE .....	84
5.1.2. PRUEBAS DE SOFTWARE .....	84
5.2. TEST DE INTEGRACION .....	86
<b>6. CAPITULO VI: DISCUSIÓN .....</b>	<b>92</b>
CONCLUSIONES .....	93
RECOMENDACIONES .....	94
REFERENCIAS BIBLIOGRAFICAS .....	95
ANEXO1: DATOS GENERALES .....	99
1. TITULO DEL PROYECTO .....	99
2. AUTOR DEL PROYECTO .....	99
3. ASSESSOR DEL PROYECTO .....	99
4. LINEA DE INVESTIGACION .....	99
5. AREA PRIORIZADA DEL PROYECTO .....	99
6. INSTITUCION Y LUGAR DE EJECUCION DEL PROYECTO .....	99
7. DURACION DEL PROYECTO .....	99
ANEXO2: CRONOGRAMA DE EJECUCION DEL PROYECTO .....	100
ANEXO3: PRESUPUESTO .....	101

## INDICE DE FIGURAS

FIGURA 1: ÁRBOL DE PROBLEMAS .....	17
FIGURA 2: FUNCIONALIDADES DEL INTERNET DE LAS COSAS.....	26
FIGURA 3: SISTEMA DE CAPAS DE ANDROID .....	33
<b>FIGURA 4: REQUISITOS DEL SISTEMA .....</b>	<b>34</b>
FIGURA 5: PLACA ARDUINO UNO.....	36
FIGURA 6: CARACTERÍSTICAS DEL ARDUINO NANO .....	43
FIGURA 7: VERSIÓN ARDUINO IDE.....	45
<b>FIGURA 8: ENTORNO DE PROGRAMACIÓN ARDUINO IDE 1.8.5 .....</b>	<b>45</b>
FIGURA 9: CELDA DE CARGA. ....	48
FIGURA 10: CELDA DE CARGA HX711.....	48
FIGURA 11: MODELO EN V.....	50
FIGURA 12: MODELO EN V DEL CICLO DE VIDA.....	52
FIGURA 13: DIAGRAMA DE CASO DE USO GENERAL. ....	58
FIGURA 14: DIAGRAMA DE SUB CASO DE USO1.....	58
FIGURA 15: DIAGRAMA DE SUB CASO DE USO2.....	59
FIGURA 16: DIAGRAMA DE DESPLIEGUE .....	59
FIGURA 17: DIAGRAMA DE ACTIVIDAD ELEGIR CAPACIDAD .....	61
FIGURA 18: DIAGRAMA DE ACTIVIDAD MONITOREO.....	61
FIGURA 19: DIAGRAMA ALERTA DE AGOTAMIENTO. ....	62
FIGURA 20: ARDUINO .....	64
FIGURA 21: PRIMERA PANTALLA DEL ARDUINO.....	65
FIGURA 22: DRIVER DE INSTALACIÓN ARDUINO .....	65
FIGURA 23: ARDUINO NANO.....	66
FIGURA 24: MODULO BLUETOOTH.....	66
FIGURA 25: TRANSMISOR HX711.....	67
FIGURA 26: LEDS DE SALIDA .....	67
FIGURA 27: SIMULACIÓN DE LA CELDA DE CARGA.....	68
FIGURA 28: INTERIOR DE LA PLACA .....	68
FIGURA 29: CIRCUITO INTEGRADO .....	69
FIGURA 30: ESTRUCTURA DEL PROTOTIPO FINAL.....	70
FIGURA 31: CAPTURA DE LOS DATOS DEL ARDUINO NANO.....	71
FIGURA 32: CONFIGURACIÓN DEL ARDUINO.....	72
FIGURA 33: CÓDIGO DE CONDICIONES.....	73
FIGURA 34: SEGUNDA PARTE DEL CÓDIGO .....	73
FIGURA 35: ANDROID STUDIO. ....	74
FIGURA 36: IMÁGENES DE INSTALACIÓN DE LIBRERÍAS.....	74
FIGURA 37: CREACIÓN DE UN PROYECTO.....	75
FIGURA 38: APP LOGIN .....	77
FIGURA 39: LOGIN.AP.....	79
FIGURA 40: VENTANA DE SELECCIÓN DE BLUETOOTH.....	79
FIGURA 41: CÓDIGO FUENTE FORMULARIO DE SELECCIONAR CAPACIDAD.....	81
FIGURA 42: FORMULARIO ALERTA .....	82
FIGURA 43: CARGAR LA INFORMACIÓN A ARDUINO NANO.....	85

FIGURA 44: CAMBIO DE PUERTO USB.....	85
FIGURA 45: PRUEBA DE CONEXIÓN CON LA BASE DE DATOS.....	86
<i>FIGURA 46: DIAGRAMA DE CONEXIÓN EN EL PROTEUZ.....</i>	<i>86</i>
FIGURA 47: TERMINAL BLUETOOTH HC-05 TRANSMISIÓN DE DATOS.....	87
FIGURA 48: INSTRUCCIONES DE CONEXIÓN BLUETOOTH ENTRE LA APLICACIÓN MÓVIL Y CIRCUITO INTEGRADO.....	88
FIGURA 49: INTERFAZ DE REGISTRO DE USUARIO.....	88
FIGURA 50: CONEXIÓN A BLUETOOTH.....	89
FIGURA 51: PRUEBA FINAL CON BALÓN DE 5KG.....	90
FIGURA 52: PRUEBA FINAL CON BALÓN DE GAS DE CAPACIDAD MEDIA.....	90
FIGURA 53: PRUEBA FINAL BALÓN DE GAS CON POCA CANTIDAD.....	91

## INDICE DE TABLAS

<b>TABLA 1: SISTEMAS OPERATIVOS EN EL PERÚ</b> .....	31
TABLA 2: HERRAMIENTAS DE ARDUINO IDE. ....	44
TABLA 3: CARACTERÍSTICAS DEL MÓDULO BLUETOOTH HC-05 .....	47
TABLA 4: FASES, TÉCNICAS Y ACCIONES DEL CICLO DE VIDA .....	53
TABLA 5: <b>CARACTERÍSTICAS ARDUINO NANO</b> .....	56
TABLA 6: CASOS DE USO .....	62
TABLA 7: CRONOGRAMA DE EJECUCIÓN DEL PROYECTO.....	100
TABLA 8: PRESUPUESTO DEL PROYECTO.....	101

## RESUMEN

En el Perú en general la gran mayoría de los usuarios utilizan el gas propano como combustible para poder preparar sus alimentos, el cual viene en varias presentaciones, en balones comerciales de 45kg, 10kg y 5kg. El balón de gas tiene una duración aproximada entre un mes y un mes y medio, sin embargo, ellos no saben con exactitud cuándo se agotará este. Esta situación provoca que el gas se agote en momentos inoportunos, por ejemplo: cuando estén preparando sus alimentos o altas horas de la noche cuando no hay un proveedor de gas propano que remplace por uno nuevo, por lo que se hace necesario contar con dispositivo de monitoreo que les ayude con estos problemas. El presente proyecto de investigación tecnológica se titula “implementar un prototipo de monitoreo de balón de gas doméstico” y se desarrolló con la finalidad de permitir a los usuarios llevar un control de peso del balón de gas en tiempo real a través de un sensor de fuerza de carga, que permite visualizar el valor exacto del contenido y mandar una alerta cuando este esté próximo a agotarse. Así mismo se ha construido un sistema formado por un dispositivo con la capacidad de monitorear el contenido de gas propano en el balón en tiempo real y comunicarlo a una aplicación móvil, siendo este como canal de comunicación entre el sistema y el usuario.

El proyecto no es limitado, ya que se puede incrementar más módulos, para realizar pedidos de balones de gas propano en línea de un proveedor, pero esto conllevaría a un mayor costo, por eso este proyecto se centra en monitorear el contenido del gas en un balón doméstico.

En el desarrollo se empleó la metodología V (verificación y validación), que permite la implementación del hardware y software. También, se emplearon una serie de tecnologías y herramientas como el uso del Arduino nano, sensores, módulos bluetooth y Android studio con el fin de lograr una aplicación funcional para teléfonos móviles de interfaz amigable.

De esta manera se pudo concluir con el desarrollo de la aplicación, obteniendo como resultado un sistema de fácil uso, interfaz amigable y un dispositivo que solo necesita ser conectado a una fuente de alimentación de 5v.

**Palabras clave:** Arduino nano, bluetooth, sensor de fuerza de carga.

## ABSTRACT

In Peru in general the vast majority of housewives use propane gas as fuel to prepare their food, which comes in several presentations in commercial tanks of 45kg, 10kg and 5kg. The gas ball lasts approximately between a month and a month and a half, but they do not know exactly when it will run out. This situation causes the gas to run out at inopportune times, for example: when they are preparing their food or late at night when there is not a propane gas supplier to replace it with a new one, so it is necessary to have a monitoring device to help them with these problems. This technological research project is titled "Implement a prototype of domestic gas tank monitoring", was developed with the purpose of allowing housewives to control the gas they have at home, by monitoring the consumption of Propane gas in real time through a load force sensor, which allows to visualize the exact value of the content and send an alert when it is about to run out. A system has been built consisting of a device with the ability to monitor the propane gas content in the tank in real time and communicate them to a mobile application, this being a communication channel between the system and the user.

The project is not limited, since you can increase more modules, to order propane gas tanks online from a supplier, but this would lead to a higher cost, so this project is focused on monitoring the gas content in a domestic tank.

In the development methodology V (verification and validation) was used that allows the implementation of hardware and software. A series of technologies and tools were used, such as the use of the Arduino nano, sensors, Bluetooth modules and android studio in order to achieve a functional application for mobile phones with a friendly interface.

In this way we can conclude with the development of the application, resulting in a user-friendly system with a friendly interface application and a device that only needs to be connected to a 5v power supply to work.

Keywords: Arduino nano, Bluetooth, load force sensor.

## CHUMASCCA

Kay maytu quillqa ruwasqaypa sutimmi “llasay ccatipay” chaymi kay ruway yanapancca warmicunata yanucusccacupi. Chaykuna kunan manaña yanunkucha llantawan, cunancca yanuncu gas nisqanwan. Chaysi chay gas tukun mana pensaypi punchau warmicuna yanucuchcaptin, chay sasachacuy yanapanpacmi cay qillqa.

Kay proyecto nisqan ccatipancca balon de gas llasaqnita, chaymanta infomacion rikuricca celular inteligente nisqanpi chaymanta warmicuna qawancc celularnipy qayca gas kasccanta. Aswansi chay gas tukupayman llacckutin willacuy ricurinqa celularpy huccta rantinaapacc.

Kay llankaytaqa puririchin achka programakuna (hardware y software) mana sasachakuy kanampaq. Llapanchik sumaqta llaqtanchita puririchinapaq.

## INTRODUCCIÓN

En los últimos tiempos el masivo uso de la tecnología ha facilitado a muchas personas y empresas estar interconectadas. En este contexto la gran mayoría de los electrodomésticos están conectados a internet o entre ellos, es por ello que en la actualidad hay aplicaciones móviles que están conectadas a dispositivos y a través de ellos puedes obtener información de todo tipo mediante sensores, esto permite que la gran mayoría de personas puedan realizar acciones desde cualquier lugar sin estar necesariamente en su domicilio. Así mismos esto facilita a los usuarios ahorro de tiempo y evitar algunas molestias.

El presente trabajo de investigación, tiene por objetivo elaborar un sistema de monitoreo de consumo de gas en domicilios, donde se hace uso del gas propano en balones domésticos de 20kg para preparar alimentos y mandar una alerta de notificación al usuario, cuando este esté a punto de agotarse por completo.

El presente trabajo de investigación consta de 5 capítulos que mencionaremos a continuación:

**CAPITULO I Problema de investigación:** descripción de la realidad problemática, se describe y fundamenta el problema identificado los objetivos para dar una posible solución, la justificación, viabilidad de a investigación y algunas limitaciones de estudio.

**CAPITULO II Antecedentes:** donde se explica de manera breve a cerca de los trabajos realizados anteriormente y sacar algunas conclusiones y como te podría ayudar a cumplir con el objetivo.

**CAPITULO III Marco teórico:** donde se fundamenta todas las teorías relacionadas o involucradas con el proyecto.

**CAPITULO IV Metodología de la investigación:** donde se describe la propuesta de solución, la ejecución del desarrollo del proyecto de acuerdo a la metodología empleada.

**CAPITULO V Resultados:** describe los resultados de la investigación si se cumplieron los objetivos trazados al principio.

**CAPITULO VI Discusión:** donde se menciona algunas recomendaciones que se encontraron luego de realizar las pruebas del proyecto.

## 1. CAPITULO I: PROBLEMA DE INVESTIGACION

### 1.1. REALIDAD PROBLEMÁTICA

En los últimos 2 años se ha notado un crecimiento acelerado del consumo de gas. Ya que, en el 2013 el 10.2% de las personas apurimeñas cocinaban a gas mientras que el 2014 fueron más del 15% de personas Según el instituto nacional de estadística. (INEI, 2017). Este crecimiento acelerado del consumo de gas doméstico, conlleva a que haya una mayor demanda del producto. Por eso los proveedores aprovechan estas circunstancias y alteran el producto o reducen la cantidad que deberían de contener un balón de gas doméstico, ya que los organismos de control como el OSINERGMIN no realizan su trabajo de manera adecuada.

Por otra parte, están las amas de casa con un problema mayor, que es la incertidumbre del contenido real de gas doméstico, ya que no se sabe con precisión cuando se agotará y cuando se realizará el pedido de este. Un ama de casa al no saber la precisión, tiene problemas con la preparación de sus alimentos ya que este se agota en momentos inoportunos, ya sea cuando aún los alimentos no están bien cocidos o en horarios nocturnos, donde uno no puede tener al alcance a un proveedor de gas. Al agotarse el gas es muy difícil de reponer en el mismo instante debido a que uno tiene que llamar o solicitar al proveedor y este llegue pasado un buen tiempo a su domicilio, esto genera que dicha persona tenga que esperar tiempos prolongados, dependiendo del lugar donde se encuentra el proveedor.

El problema se agudiza más aun cuando uno no cuenta con números telefónicos de los proveedores o cuando realizan pedidos una gran cantidad de personas. El proveedor no se abastece o no localiza el domicilio donde se realizó el pedido y esto provoca que el gas llegue al domicilio en un tiempo mayor a lo estipulado generando más molestias.

Este particular caso se solucionaría con la construcción de un prototipo de medición de la capacidad de gas dentro del balón de gas, que te indique el estado actual del

gas y así evitar molestias a las amas de casa. Ya que, no solo tendrán conocimiento de la capacidad de gas que compran, si no también tendrán pleno conocimiento de la cantidad de gas remanente que ellas tienen en su balón de gas a través del aplicativo móvil que servirá como una interfaz.

Para poder realizar este prototipo se tomará en cuenta todas las normas estipuladas por el organismo de supervisión del consumidor.

➤ **Contenido neto del GLP**

Cantidad en masa de GLP, contenida en un recipiente. Este contenido corresponde a la diferencia de la masa bruta y tara del recipiente.

➤ **Contenido neto nominal de GLP**

Es el característico de cada tipo de recipiente, registrado en el protector de válvula.

➤ **Masa bruta del recipiente**

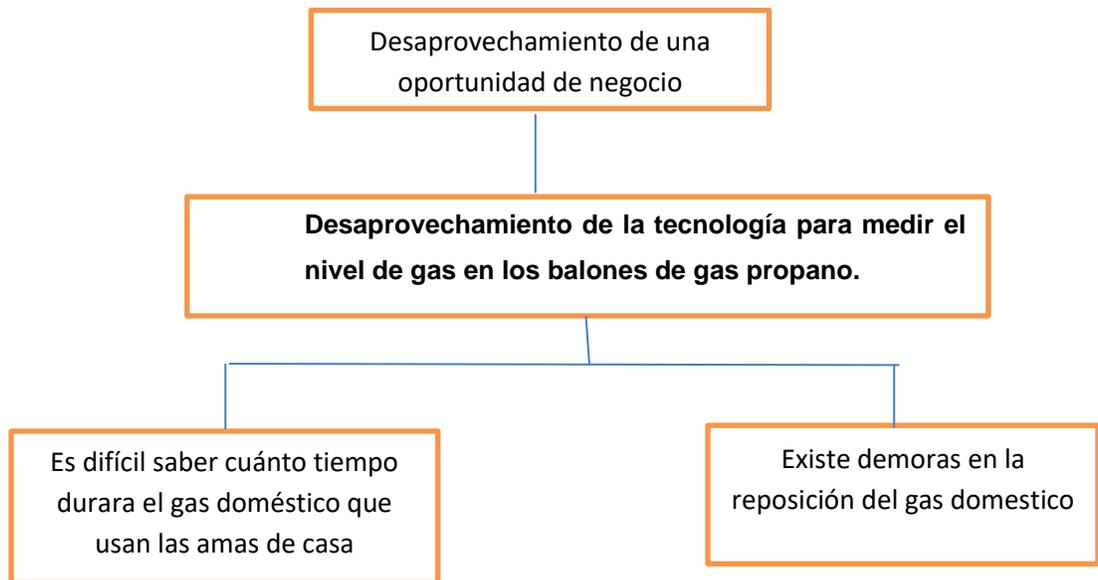
Cantidad en masa, equivalente a la suma del contenido neto de GLP más la tara del recipiente, marcado en el protector de la válvula. (SEACE, 2007)

Como antecedente a este trabajo de tesis tenemos a la empresa “Gas watch” que desarrollo un dispositivo que permite medir la cantidad de gas remanente en balones de gas para parrilleros a través de un aplicativo móvil. Este dispositivo trabaja con una batería interna y le permite al cliente conocer qué porcentaje de gas le queda. (GasWatch, 2018)

Similarmente, tenemos a la empresa “Gas tracker” de Guatemala que desarrollo un dispositivo de control digital para gas L.P de tanque estacionario. El cual te permite administrar, controlar mediante un dispositivo electrónico, basado netamente en hardware. (Gas tracker, 2016)

Los problemas de este caso se pueden visualizar de mejor manera en el siguiente árbol de problemas.

Figura 1: árbol de problemas



Fuente: elaboración propia.

## 1.2. FORMULACIÓN DEL PROBLEMA

Desaprovechamiento de la tecnología para monitorear el consumo de gas en balones de gas propano.

## 1.3. OBJETIVOS

### 1.3.1. Objetivo General

Implementar un prototipo con tecnología Android y Arduino, para monitorear el nivel de gas doméstico en un tanque de gas propano.

### 1.3.2. Objetivos Específicos

- Analizar una arquitectura con Arduino que permita conectar sensores de peso y dispositivos móviles por medio de sus puertos analógicos y digitales

- Diseñar una aplicación móvil en Android que a través de este se monitoree el nivel de gas propano.
- Realizar las pruebas de funcionamiento del monitoreo.

#### **1.4. JUSTIFICACIÓN**

El mundo cada vez más esta interconectado y computarizado, permitiendo que celulares, objetos y hasta personas estén conectadas a internet e interactuando entre ellas, dentro de ello el internet de las cosas permite que cualquier objeto pueda brindar información a un celular o una computadora siempre que esté conectado a internet y así el usuario ya no tendrá que estar pendiente de cada objeto ya que se puede controlar a través de su Smartphone desde el lugar donde se encuentre.

En el Perú la gran mayoría de la población prepara sus alimentos con gas doméstico (gas propano), que les llega a sus domicilios en balones de poca capacidad que abastece como máximo un mes, de los cuales uno no sabe con precisión el día en que este se agotara.

Gracias a la utilización de la tecnología y el internet de las cosas se podrá monitorear, controlar y gestionar el pedido de gas doméstico desde cualquier lugar.

Permitiendo resolver problemas de distancia tiempo y sobre todo brindando un servicio cómodo y de calidad para las amas de casa, que ya no tendrán que perder su tiempo en estar llamado y esperando que el proveedor reponga el vagón de gas, ya que la aplicación se encargara de la verificación, control y realización del pedido en el momento oportuno.

Con esta aplicación móvil permitirá realizar los pedidos de manera segura y oportuna ya que la ama de casa tendrá la facilidad, tranquilidad y no tendrá que preocuparse más por la falta de gas doméstico.

Teniendo en cuenta todas estas oportunidades del mundo moderno, viendo que la gran mayoría de la población hace uso del gas doméstico y viendo que esta propuesta soluciona prácticamente el problema, se ve un desaprovechamiento de un modelo de negocio, el cual se puede desarrollar de manera masiva.

## **1.5. VIABILIDAD DE LA INVESTIGACIÓN**

### **1.5.1. Viabilidad Técnica**

El proyecto a desarrollar si es viable técnicamente porque se dispone de las herramientas para el desarrollo del Aplicativo móvil, las herramientas de software están disponibles como las de software libre y con licencia, también se cuenta con los equipos como son la computadora, un dispositivo móvil con Android y otros, también se cuenta los servicios indispensables como el internet, telefonía móvil, luz eléctrica, etc.

También se cuenta con los materiales necesarios para desarrollar el detector de agotamiento entre ellas la placa de Arduino, sensores y demás accesorios que intervendrán el desarrollo del prototipo.

### **1.5.2. Viabilidad Económica**

Es viable económicamente porque se cuenta con los recursos económicos para desarrollar el aplicativo móvil, es decir para la adquisición de los bienes que son equipos mencionados con los que trabajaremos tanto hardware y software, por otro lado, también se cuenta con recursos financieros para pagar los servicios diversos que son necesarios y por último para pagar a los involucrados del proyecto (Recursos Humanos).

Con estas consideraciones el costo de la ejecución del presente proyecto es de s/. 4,787 y será financiado por la tesista. Como se muestra en el (anexo 1).

### **1.5.3. Viabilidad operativa**

Es viable operativamente porque la población hace uso de dispositivos móviles los cuales tiene el sistema operativo, así mismos tienen conocimientos de usuario básico y por ende manejan aplicaciones variadas en el entorno, de la misma manera la aplicación será fácil uso y amigable con el usuario.

### **1.5.4. Viabilidad legal**

En este proyecto de investigación se utilizara software open source los cuales no cuenta con restricciones legales por ejemplo.

- Arduino nano: con software open source denominada arduino.
- Android studio: libre y de código abierto bajo la licencia de Android.

#### **1.5.5. Viabilidad social**

Cabe mencionar el impacto positivo que causara este proyecto de tesis en la sociedad, debido que este proyecto permitirá a las amas de casa que cuentan con un teléfono inteligente aprovechar todas las utilidades de este y facilitara el proceso de pedido de gas.

Otro beneficio que traerá esta aplicación es insertar a las amas de casa al uso adecuado de la tecnología y así disminuir la brecha digital existente en nuestro país, de una u otra forma esta aplicación ara interactuar a los usuarios con el internet, aplicaciones y con una herramienta electrónica.

Es beneficiosa ya que los usuarios tendrán la certeza y confianza en esta aplicación porque se tendrá pleno conocimiento de toda la transacción realizada.

#### **1.6. LIMITACIÓN DEL ESTUDIO**

Actualmente la tecnología viene siendo usada por la gran mayoría de las personas, ya sea en el trabajo, comercio e interacción de persona a persona o grupo, porque este es un medio de comunicación fácil y rápida.

En este contexto muchas personas aprovechan el avance de la tecnología, pero lamentablemente no todas las personas cuentan con los mismos ingresos económicos especialmente las personas de las zonas rurales, quienes se ven afectados de una u otra manera porque no tienen acceso a la tecnología, uno por la lejanía del lugar donde el internet aun no llega o no tiene cobertura, otro por los elevados costos de los teléfonos inteligentes. Esto repercute a que no todos puedan tener acceso a esta aplicación ya que es necesario contar con un teléfono inteligente, acceso a internet y un hardware.

Entones especificaremos algunas limitantes:

Conocimientos tecnológicos mínimos o nulos por parte de las amas de casa.

Algunos domicilios no cuentan con servicio de internet.

Algunos usuarios no cuentan con teléfonos inteligentes ya sea (smartphones o tables).

Existe dificultad para manejar algunos dispositivos.

Algunas amas de casa no tienen idea de transacciones que se realiza mediante las aplicaciones.

No obstante, estas limitaciones se pueden superar rápidamente gracias a que las personas son capaces de aprender rápidamente por medio de amigos o familiares, etc. Quienes gustosamente les podrían ayudar a empezar a hacer uso de la tecnología actual.

## 2. CAPITULO II: ANTECEDENTES

Para poder desarrollar este trabajo de tesis, se realizó una previa investigación acerca de los trabajos que se relacionan con el problema, desarrollados previamente.

### 2.1. ANTECEDENTES INTERNACIONALES

- **TITULO:** “DESARROLLO DE UNA APLICACIÓN DOMÓTICA CON COMUNICACIÓN INALÁMBRICA BLUETOOTH”. (Sotelo Martínez, Olivo Florez, & Rodriguez Ortiz, 2015)

**Año:** 2015.

**Autor:** Sotelo Martínez Samuel, Olivo Flores Marco Antonio y Rodríguez Ortiz Juan Gabriel.

**Arquitectura usada:**

- Arduino Mega.
- App inventor para desarrollar la aplicación
- Módulo HC-06: módulo de bluetooth que utilizo para la recepción de datos enviados desde una aplicación desarrollada en “App inventor”

**Conclusiones:**

Menciona que los avances en la tecnología electrónica, de control y comunicaciones, nos van proporcionado cada vez más elementos disponibles y accesibles para desarrollar aplicaciones que puedan beneficiar ambas partes de la sociedad y generar conocimiento para estudiantes y docentes. La integración de distintos elementos en hardware y software de la aplicación mostrada se puede extraer más elementos para generar un sistema de más capacidad que proporciona las distintas ventajas que se pretenden con la domótica.

Sus resultados demostrados que para cierta clase de aplicaciones domóticas de bajos recursos se tiene un sistema suficientemente estable y confiable. Se describió una aplicación que puede potencializarse, pero presenta una alternativa económica. Se puede mejorar significativamente al tener elementos de mayor calidad, como en este caso el alcance de la señal de

comunicación y la seguridad en el sistema de control que son áreas de oportunidad a mejorar en trabajos futuros en este mismo tema.

- **TITULO: DISEÑO DE UNA APLICACIÓN MOVIL PARA LA COMUNICACIÓN INALAMBRICA DE SEÑALES AUDIOVISUALES.** (Perez, 2013)

**AÑO:** 2013.

**AUTOR:** Alberto Esteban Pérez.

**ARQUITECTURA USADA:** módulo de control (Arduino uno), Módulo Bluetooth Serial, Comunicación (Transmisión y Recepción por el puerto serial del hyperterminal), módulo Bluetooth, USB o PC (con bluetooth ya incorporado), módulo XBEE (proporciona una conectividad serie IEEE 802.11. Ofreciendo una relación consumo/coste muy baja.), Wireless shield (modulo con zócalo y adoptadores) y módulo wifi.

#### **CONCLUSIONES:**

El proyecto consiste en el diseño de una aplicación Android que mediante un módulo Arduino conectado por Ethernet a un router, permita la transmisión de archivos audiovisuales inalámbricamente. El sistema contara con la aplicación Android jugando el roll de cliente y el módulo Arduino de servidor.

Después de la implementación se puede observar que el módulo construido realiza la transmisión de datos, pero de manera lenta, esto es debido a que el Arduino uno genera cuellos de botella principalmente por que tiene la memoria SRAM de pequeña capacidad.

El presente proyecto también se puede enfocar en muchos ámbitos como turismo publicidad o incluso la seguridad. Las ventajas que se obtendrían sería tener un sistema privado de transferencia de archivos, seguro y a un bajo coste, ya que, hoy en día, las placas Arduino están a un precio muy atractivo.

Para poder realizar trabajos posteriores se deberá tomar en cuenta este detalle, yaqué si desean transferir información de gran capacidad se tendrá que analizar otro tipo de palca que tenga soporte mayor capacidad de transferencia.

- **TITULO: GAS TRACKER.** (Gas tracker, 2016).

**AÑO:** 2016

**AUTOR:** COMPAÑIA GAS TRACKER

**ARQUITECTURA USADA:**

Los materiales principales son: indicador remoto (digital), sensor electrónico con cable.

**CONCLUSIONES:**

Concluye que el dispositivo cumple los objetivos que es de medir la cantidad restante del gas en tanques estacionarios y mandar actualizaciones cada 24 horas a un indicador remoto. A parte de eso el indicador remoto tiene la capacidad de transformar la cantidad de gas en litros y al realizar esa transformación hay una diferencia del 5% que no favorece al consumidor.

Para futuras investigaciones se tiene que tener en cuenta el presupuesto que se maneja ya que se podría utilizar otro tipo de tecnología que sea aún más barata y fácil de usar ya que el modelo no se adapta cualquier tipo de sistema de gas doméstico.

## 2.2. ANTECEDENTES NACIONALES

- **TITULO: DISEÑO DE UN SISTEMA DE CONTROL DOMOTICO Y VIDEO VIGILANCIA SUPERVISADO POR TELEFONO MOVIL.** (Fabricio, 2013)

**AÑO:** 2013

**AUTOR:** Fabricio Guerra

**ARQUITECTURA USADA:**

Arduino, módulo de wifi y sensores de movimiento.

**CONCLUSIONES:**

El diseño del sistema en que consiste esta tesis es factible en el ámbito de una red local, debido a que sus requerimientos de ancho de banda son muy bajos, además no requiere un servidor con grandes capacidades de procesamiento para su funcionamiento ni la instalación de muchos componentes complejos.

Debido al reducido tamaño del microcontrolador de dispositivos, que funciona también como servidor web y central telefónica, la instalación del sistema será

sencilla y puede darse en cualquier lugar de la casa. Debido al bajo costo del sistema y a utilizarse hardware y software de libre distribución, se presenta como una muy buena propuesta de negocio. Gracias a las alertas instantáneas que genera el sistema puede ayudar al usuario a prevenir que se complete un robo.

### 3. CAPITULO III: MARCO TEÓRICO

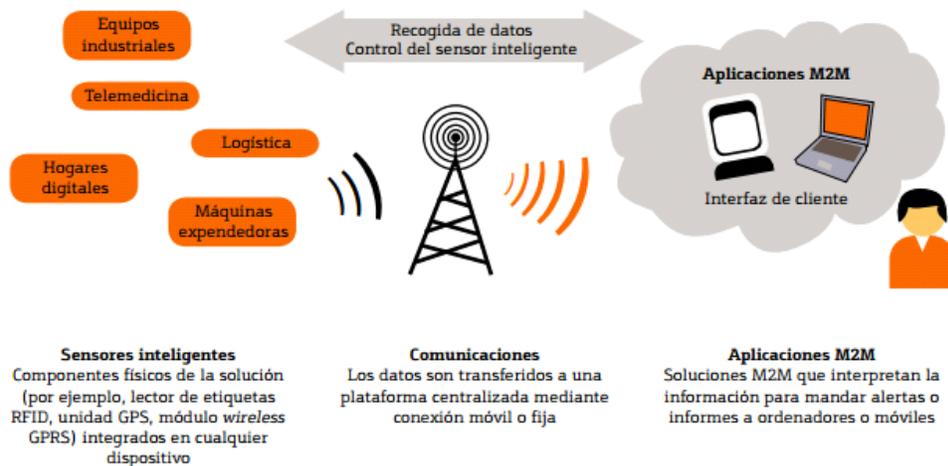
#### 3.1. INTERNET DE LAS COSAS

“El internet de las cosas permitira que podamos integrar objetos inteligentes de todo tipo y funcion, redes de sensores, y recursos de la internet actual con las personas con el fin de compartir informacion que sea util para aumentar nuestro conocimiento y tomar decisiones que mejoren nuestra calidad de vida en cualquier aspecto posible: social, economico, cultural, ambiental, etc”. (Alcaras, 2014)

internet de las cosas es una compilación del uso tecnificado e implementación de la tecnología al beneficio de las personas, convirtiendo los bits de datos en conocimiento para emplearlos de mejor manera a todos los equipos electrónicos.

El internet también es llamado red de sensores, ya que en esencia se trata de distintas cosas interconectadas mediante un mismo protocolo de comunicación. Cada una de los partes relacionados con sistemas de interconexión, como es el caso de las redes neuronales. (Evans, 2011).

Figura 2: Funcionalidades del internet de las cosas.



Fuente: (Evans, 2011).

## **3.2. DISPOSITIVO MÓVIL**

“Los dispositivos móviles (también conocidos como computadora de mano) son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, móviles o no, con conexión permanente o intermitente a una red, con memoria limitada, están diseñados para funciones específicas, pero que pueden llevar a cabo otras funciones más generales” (Molina Rivera, Sandoval Cardona, & Toledo Franco, 2012).

### **3.2.1. TIPOS DE DISPOSITIVOS MOVILES**

Para competir en el mercado cada fabricante equipa a sus dispositivos con diversas características esperando atraer al mayor número de usuarios disponibles. El dispositivo más común actualmente es el teléfono móvil, seguido de los conocidos PDAs (Personal Data Assistant). Todos los dispositivos disponen en mayor o menor medida de posibilidades de desarrollo. Los fabricantes permiten que terceras partes puedan desarrollar aplicaciones que los usuarios puedan instalar y disfrutar en los dispositivos. La aplicación más común encontrada en estos dispositivos suele ser un navegador Web, adaptado por supuesto al tamaño y forma del aparato en cuestión. (Chiliqinga, 2010)

Entre los tipos de dispositivos móviles tenemos:

#### **A. PDAs**

Una PDA (del inglés Personal Digital Assistant (asistente digital personal)), es una computadora de mano originalmente diseñada como agenda electrónica con un sistema de reconocimiento de escritura. Hoy día (2011) estos dispositivos pueden realizar muchas de las funciones de una computadora de escritorio, pero con la ventaja de ser portátil. (Martinez, 2011)

#### **B. Smartphone o teléfonos inteligentes**

(Burgos Rodas & Echeverry Aguirry, 2012) consideran que el teléfono inteligente también denominado “Smartphone” en inglés es un

dispositivo electrónico de mano con capacidades parecidas a la de un ordenador porque le permite instalar programas como Apps el cual hace que incremente el procesamiento de datos y la conectividad además su tamaño es reducido adecuado al uso personal. Y también se puede comunicar con wi-fi, bluetooth, conexión a internet y otros.

### **Características de los dispositivos móviles**

**(Chilinquina, 2010)** Algunas de las características que hacen que estos dispositivos sean diferentes de los ordenadores de sobremesa son los siguientes:

- funcionabilidad limitada.
- No necesariamente extensible y actualizable.
- Menos complicado en su manejo.
- Fácil de aprender sus operación.

### **3.3. SISTEMAS OPERATIVOS**

(Wolf & et-al, 2014) Manifiesta que “el sistema operativo es el principal programa que se ejecuta en toda computadora de propósito general. Los hay de todo tipo, desde muy simples hasta complejos, y entre más casos de uso hay para el cómputo en la vida diaria, más variedad habrá en ellos”.

### **3.4. SISTEMAS OPERATIVOS MOVILES**

Un sistema operativo móvil es un sistema que controla un dispositivo móvil al igual que las computadoras más grandes utilizan Windows, Linux o Mac OS entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos

(Pedrozo Petrazzini, 2012) Indica que los componentes de un sistema operativo móvil son los siguientes:

- **Capas:** Al igual que los sistemas operativos presentes en los componentes informáticos más grandes, un sistema operativo móvil también se encuentra compuesto por varias capas.
- **Kernel:** Una de las más importantes y esenciales piezas que componen cualquier sistema operativo, sea el de nuestro móvil, o el de la PC, es el denominado núcleo o Kernel, el cual es la capa de software que permite el acceso a los diferentes elementos de hardware que conforman nuestro móvil.  
También es el encargado de brindar diferentes servicios a las capas superiores como los controladores de hardware, gestión de procesos, sistemas de archivos, además del acceso y administración de la memoria del sistema.
- **Middleware:** Esta capa es el conjunto de módulos que permite que las aplicaciones diseñadas y escritas para tales plataformas puedan ser ejecutadas.

Su funcionamiento es totalmente transparente para el usuario, no debiendo realizar ninguna acción ni configurar alguna para su correcto desenvolvimiento.

El Middleware brinda la posibilidad de ejecutar servicios muy importantes para que otras aplicaciones, en capas superiores de la jerarquía, puedan ejecutarse.

- **Entorno de ejecución de aplicaciones:** Esta capa provee de todos los elementos necesarios para la creación y desarrollo de software es decir contiene elementos que serán de gran ayuda a los mismos, en el momento de escribir aplicaciones compatibles con ese sistema operativo.
- **Interfaz de Usuario:** La interfaz de usuario es el elemento del teléfono que usualmente utilizamos para interactuar con el aparato.

Sin esta capa no sería posible utilizar nuestro dispositivo, ya que presenta todos los elementos necesarios para facilitar cualquier tipo de tarea que deseemos realizar en nuestra terminal. Además, incluye todos los elementos

gráficos que harán posible el uso cómodo y sencillo del móvil: botones, menús, pantallas y listas, entre otros.

- **Tipos de sistemas operativos**

Se cuentan con una gran cantidad de sistemas operativos móviles y para el presente proyecto usaremos el SO android, los sistemas operativos más importantes son:

- **Android:** (Pedrozo Petrazzini, 2012) Es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google. Este sistema por lo general maneja aplicaciones como Google Play. Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.
- **iOS:** (Wolf & et-al, 2014)El sistema operativo de Apple, y diseñado exclusivamente para el hardware producido por dicha compañía. Fue el primero en implementar el interfaz usuario multitouch y, en buena medida, se puede ver como el responsable de la explosión y universalización en el uso de dispositivos móviles. Al igual que el sistema operativo que emplean para sus equipos de escritorio, MacOS X, iOS está basado en el núcleo Darwin, derivado de FreeBSD, un sistema libre tipo Unix.
- **Windows Phone:** (Gomez, 2013)es un sistema operativo desarrollado por la compañía Microsoft, como sucesor de la plataforma Windows Mobile. Está enfocado en el mercado de consumo generalista, en lugar del mercado empresarial al que estaba destinado el Windows Mobile.

Windows Phone y Windows Mobile no son compatibles, con lo que todas las aplicaciones que existían de Windows Mobile se quedarán obsoletas con el paso del tiempo.

Los sistemas operativos más usados en el Perú son:

**Tabla 1: Sistemas Operativos en el Perú**

Sistema Operativo	Usuario
Android	76%
Apple iOS	8.0%
Windows Phone	6.3%
Otros	13.1%

*Fuente: ComScore, CNN Expansion.*

### **3.5. APLICATIVO MÓVIL**

La aplicación móvil es un programa que se puede descargar y acceder directamente de un smartphone conectado o desde algún otro dispositivo móvil con acceso a internet.

Los aplicativos móviles están diseñados para ser ejecutados en dispositivos heterogéneos como teléfonos inteligentes (Smartphone), tabletas y entre otros dispositivos móviles.

#### **3.5.1. Aplicaciones nativas**

Según Leadership y Paper (2012) las aplicaciones nativas es aquella que se descarga e instala en el propio dispositivo porque tiene archivos ejecutables los cuales se almacenan de manera local. Las aplicaciones nativas son instaladas por el usuario y pueden ser descargada desde tiendas virtuales de aplicativos como a App Store de Apple, Marketplace de Android o App World de BlackBerry.

“Por otra parte, las aplicaciones móviles nativas son desarrolladas para una plataforma específica, no puede implementar en otra plataforma. Y también este tipo de aplicaciones tiene un buen rendimiento” (Zambrano, 2014).

### **3.6. ANDROID**

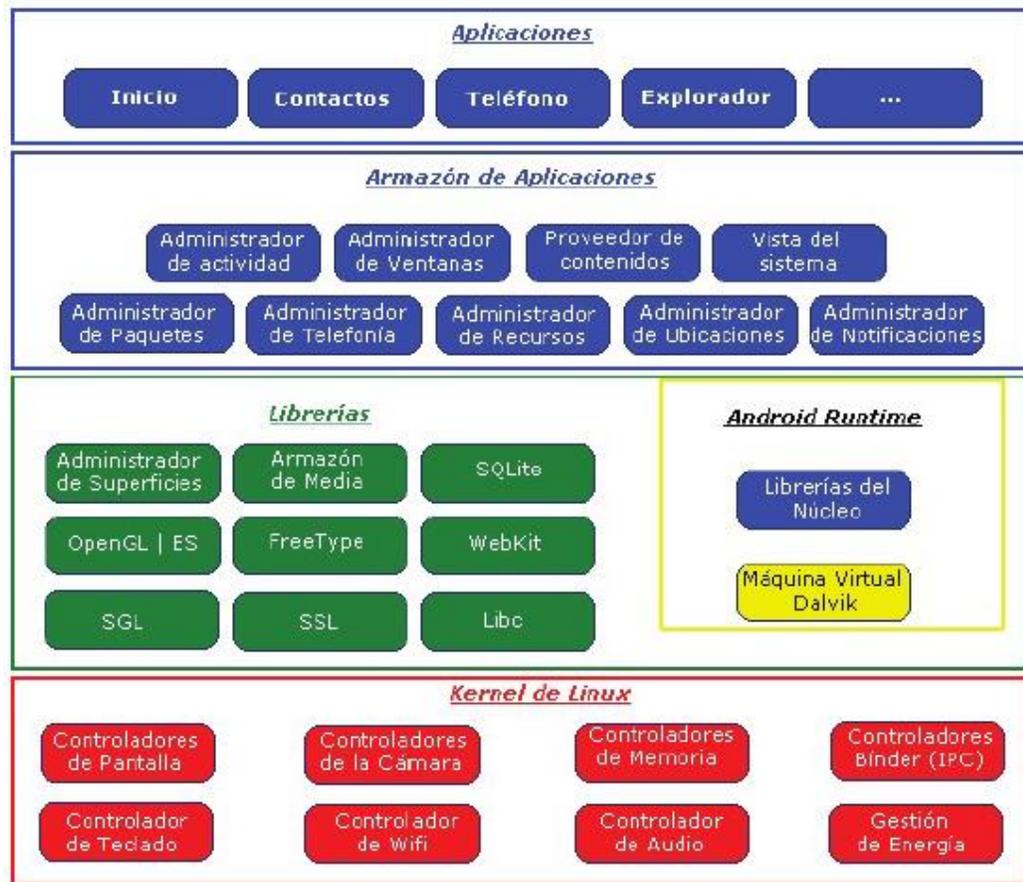
En los últimos años los teléfonos móviles han experimentado una gran evolución, desde los primeros terminales, grandes y pesados, pensados sólo para hablar por teléfono en cualquier parte, a los últimos modelos, con los que el término “medio de comunicación” se queda bastante pequeño.

Es así como nace Android. Android es un sistema operativo y una plataforma software, basado en Linux para teléfonos móviles. Además, también usan este sistema operativo (aunque no es muy habitual), tablets, netbooks, reproductores de música e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). Además, lo que le diferencia de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, widgets, o incluso, modificar el propio sistema operativo, dado que Android es de código libre, por lo que sabiendo programar en lenguaje Java, va a ser muy fácil comenzar a programar en esta plataforma.

Fue desarrollado por Android Inc., empresa que en 2005 fue comprada por Google, aunque no fue hasta 2008 cuando se popularizó, gracias a la unión al proyecto de Open Handset Alliance, un consorcio formado por 48 empresas de desarrollo hardware, software y telecomunicaciones, que decidieron promocionar el software libre. Pero ha sido Google quien ha publicado la mayor parte del código fuente del sistema operativo, gracias al software Apache, que es una fundación que da soporte a proyectos software de código abierto.

Dado que Android está basado en el núcleo de Linux, tiene acceso a sus recursos, pudiendo gestionarlo, gracias a que se encuentra en una capa por encima del Kernel, accediendo así a recursos como los controladores de pantalla, cámara, memoria flash. (Baez, Borrego, & Cordero, 2012).

Figura 3: Sistema de Capas de Android



Fuente: (Baez, Borrego, & Cordero, 2012).

### 3.7. ANDROID STUDIO

**Android Studio** es el entorno de desarrollo integrado oficial para la plataforma Android. Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android.

Estuvo en etapa de vista previa de acceso temprano a partir de la versión 0.1, en mayo de 2013, y luego entró en etapa beta a partir de la versión 0.8, lanzada en

junio de 2014. La primera compilación estable, la versión 1.0, fue lanzada en diciembre de 2014. La última versión estable es la 3.0, y fue lanzada en octubre de 2017.

- **Plataformas**

Android Studio está disponible para Windows 2003, Vista, 7, 8, y 10, tanto plataformas de 32 como de 64 bits, GNU/Linux, Linux con GNOME o KDE y 2 GB de memoria RAM mínimo y macOS, desde 10.8.5 en adelante. (wikipedia, 2019)

*Figura 4: requisitos del sistema*

	Windows	OS X/macOS	Linux
<b>OS versión</b>	Windows 10/8/7 (32- o 64-bit)	Mac OS X 10.10 (Yosemite) o superior, hasta 10.13 (macOS High Sierra)	GNOME o KDE desktop
<b>RAM</b>	3 GB RAM mínimo, 16 GB RAM recomendado más 1GB adicional para el emulador de Android		
<b>Espacio en disco</b>	2 GB de espacio en disco para Android Studio, 4GB recomendados (500MB para la IDE y al menos 1.5 GB para Android SDK, imágenes de sistema de emulador y cachés)		
<b>Java versión</b>	Java Development Kit (JDK) 8		
<b>Resolución de pantalla</b>	1280x800 mínimo, 1440x900 recomendado		

*Fuente: (wikipedia, 2019).*

### 3.8. ARDUINO

Arduino es una plataforma electrónica de código abierto basado en hardware y software. Placas Arduino son capaces de leer las entradas - la luz en un sensor, un dedo sobre un botón o un mensaje de Twitter - y lo convierten en una salida - la activación de un motor, encender un LED, publicar algo en línea. Se puede decir que su tablero qué hacer mediante el envío de un conjunto de instrucciones al microcontrolador en el tablero. Para ello se utiliza el lenguaje de programación de Arduino (basado en el cableado), y el software de Arduino (IDE), sobre la base de procesamiento. (Arduino, 2016).

Arduino es definida como una plataforma de electrónica abierta para la creación de prototipos, basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquier interesado en crear entornos u objetos interactivos. (Morales Hernades, 2012).

En términos generales, puede decirse que Arduino es un pequeño sistema embebido, ya que cuenta con todos los elementos de este tipo de sistemas; su lenguaje de programación es una derivación de "C", con funciones predefinidas para facilitar la programación, aunque es posible incorporar código C del compilador AVR (ATMEGA320 es un microcontrolador AVR). Esto es especialmente útil cuando se requiere un mayor control de los módulos inter- nos del microcontrolador.

La estructura de Arduino fue concebida para proveer entradas y salidas digitales, entradas analógicas e interrupciones externas. Se programa por vía USB, aunque también dispone de un conector ICSP para utilizar un programador externo. Al ser una plataforma de hardware abierto, tanto el diagrama esquemático como el firmware es de acceso público, de manera que puede ser tomado y modificado de acuerdo a necesidades específicas de diseño; es así como pueden encontrarse otras plataformas con la misma filosofía de Arduino, como Funduino, Pingüino, etc. (Morales Hernades, 2012).

Figura 5: Placa Arduino uno.



Fuente: (Morales Hernades, 2012).

### 3.8.1. Características

#### **Entradas y salidas:**

Cada uno de los 14 pines digitales se puede usar como entrada o como salida. Funcionan a 5V, cada pin puede suministrar hasta 40 mA. La intensidad máxima de entrada también es de 40 mA.

Cada uno de los pines digitales dispone de una resistencia de pull-up interna de entre 20K $\Omega$  y 50 K $\Omega$  que está desconectada, salvo que nosotros indiquemos lo contrario.

Arduino también dispone de 6 pines de entrada analógicos que trasladan las señales a un conversor analógico/digital de 10 bits.

#### **Alimentación de un Arduino**

Puede alimentarse directamente a través del propio cable USB o mediante una fuente de alimentación externa, como puede ser un pequeño transformador o, por ejemplo, una pila de 9V. Los límites están entre los 6 y los 12 V. Como única restricción hay que saber que, si la placa se alimenta con menos de 7V,

la salida del regulador de tensión a 5V puede dar menos que este voltaje y si sobrepasamos los 12V, probablemente dañaremos la placa.

La alimentación puede conectarse mediante un conector de 2,1mm con el positivo en el centro o directamente a los pines Vin y GND marcados sobre la placa.

Hay que tener en cuenta que podemos medir el voltaje presente en el jack directamente desde Vin. En el caso de que el Arduino esté siendo alimentado mediante el cable USB, ese voltaje no podrá monitorizarse desde aquí.

### 3.8.2. Ventajas

Según (Arduino, 2016).

- **Asequible:** Las placas Arduino son relativamente baratos en comparación con otras plataformas de microcontroladores. La versión menos cara del módulo Arduino puede ser ensamblado a nano.
- **Multiplataforma:** El software de Arduino (IDE) se ejecuta en Windows, Macintosh OS X, y Linux. La mayoría de los sistemas de microcontrolador se limitan a Windows.
- **Simple, entorno de programación clara:** El software de Arduino (IDE) es fácil de usar para los principiantes, pero lo suficientemente flexible para los usuarios avanzados que aprovechan también. Para los profesores, se basa convenientemente en el entorno de programación Processing, para que los estudiantes aprenden a programar en ese entorno estarán familiarizados con cómo funciona el IDE de Arduino.
- **El código abierto y el software extensible:** El software de Arduino se publica como herramientas de código abierto, disponible para la extensión por los programadores experimentados. El idioma se

puede ampliar a través de bibliotecas de C ++, y la gente que quiere entender los detalles técnicos pueden dar el salto de Arduino para el lenguaje de programación C AVR en la que se basa. Del mismo modo, puede agregar código AVR-C directamente en sus programas de Arduino, si así lo desea.

- **El código abierto y hardware ampliable:** Los planes de las placas Arduino se publican bajo una licencia de Creative Commons, por lo que los diseñadores de circuitos experimentados pueden hacer su propia versión del módulo, ampliándolo y mejorándolo. Incluso los usuarios con poca experiencia pueden construir la versión de tablero del módulo con el fin de entender cómo funciona y ahorrar dinero.

### 3.8.3. Tipos de Arduino

Según (Arduino, Arduino: tecnología para todos, s.f.) Hay una gran variedad de Arduino, según su utilidad y el uso que se les vaya a dar, algunos de ellos son:

#### A. Arduino Duemilanove

El Arduino Duemilanove ("2009") es una placa con microcontrolador basada en el ATmega168 ([datasheet](#)) o el ATmega328 ([datasheet](#))., Tiene 14 pines con entradas/salidas digitales (6 de las cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, un cristal oscilador a 16Mhz, conexión USB, entrada de alimentación, una cabecera ISCP, y un botón de reset. Contiene todo lo necesario para utilizar el microcontrolador; simplemente conéctalo a tu ordenador a través del cable USB o aliméntalo con un transformador o una batería para empezar a trabajar con él.

El Arduino Duemilanove puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente. Las fuentes de alimentación externas (no-USB) pueden ser tanto un transformador o una batería. La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V el pin de 5V puede proporcionar menos de

5 Voltios y la placa puede volverse inestable, si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

- **Memoria:** El ATmega328 tiene 32KB (el ATmega168 tiene 16 KB) de memoria flash para almacenar código (2KB son usados para el arranque del sistema (bootloader). El ATmega328 tiene 2 KB (Atmega168 1 KB) de memoria SRAM. El ATmega328 tiene 1KB (ATmega168 512 bytes) de EEPROM, que puede a la cual se puede acceder para leer o escribir con la [Reference/EEPROM |librería EEPROM]].
- **Comunicación:** EL Arduino Duemilanove facilita en varios aspectos la comunicación con el ordenador, otro Arduino u otros microcontroladores. Tanto el ATmega328 como el Atmega168 proporciona comunicación vía serie UART TTL (5V), disponible a través de los pines digitales 0(RX) y 1(TX). Un chip FTDI FT232RL integrado en la placa canaliza esta comunicación serie a través del USB y los drivers FTDI (incluidos en el software de Arduino) proporcionan un puerto serie virtual en el ordenador. El software incluye un monitor de puerto serie que permite enviar y recibir información textual de la placa Arduino. Los LEDs RX y TX de la placa parpadearán cuando se detecte comunicación transmitida a través del chip FTDI y la conexión USB (no parpadearán si se usa la comunicación serie a través de los pines 0 y 1).
- **Programación:** El ATmega328 y el ATmega168 en las placas Arduino Duemilanove viene precargado con un gestor de arranque (bootloader) que permite cargar nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500 original (archivo de cabecera C).

## B. Arduino mega

El Arduino Mega es una placa microcontrolador basada ATmeg1280 (datasheet). Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset. Contiene todo lo necesario para hacer funcionar el microcontrolador; simplemente conéctalo al ordenador con el cable USB o aliméntalo con un transformador o batería para empezar.

- **Alimentación:** El Arduino Mega puede ser alimentado vía la conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente. La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7V el pin de 5V puede proporcionar menos de 5 Voltios y la placa puede volverse inestable, si se usan más de 12V los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.
- **Memoria:** El ATmega1280 tiene 128KB de memoria flash para almacenar código (4KB son usados para el arranque del sistema (bootloader). El ATmega1280 tiene 8 KB de memoria SRAM. El ATmega1280 tiene 4KB de EEPROM, que puede a la cual se puede acceder para leer o escribir con la [Reference/EEPROM |librería EEPROM]].
- **Programación:** El ATmega1280 en el Arduino Mega viene precargado con un gestor de arranque (bootloader) que permite cargar nuevo código sin necesidad de un programador por hardware externo. Se comunica utilizando el protocolo STK500 original (archivo de cabecera C). También te puedes saltar el gestor de arranque y programar directamente el microcontrolador a través del puerto ISCP (In Circuit Serial Programming).

- **Características físicas:** La longitud y amplitud máxima de la placa Duemilanove es de 4 y 2.1 pulgadas respectivamente, con el conector USB y la conexión de alimentación sobresaliendo de estas dimensiones. Tres agujeros para fijación con tornillos permiten colocar la placa en superficies y cajas. Ten en cuenta que la distancia entre los pines digitales 7 y 8 es 160 mil (0,16"), no es múltiple de la separación de 100 mil entre los otros pines.

### C. Arduino nano

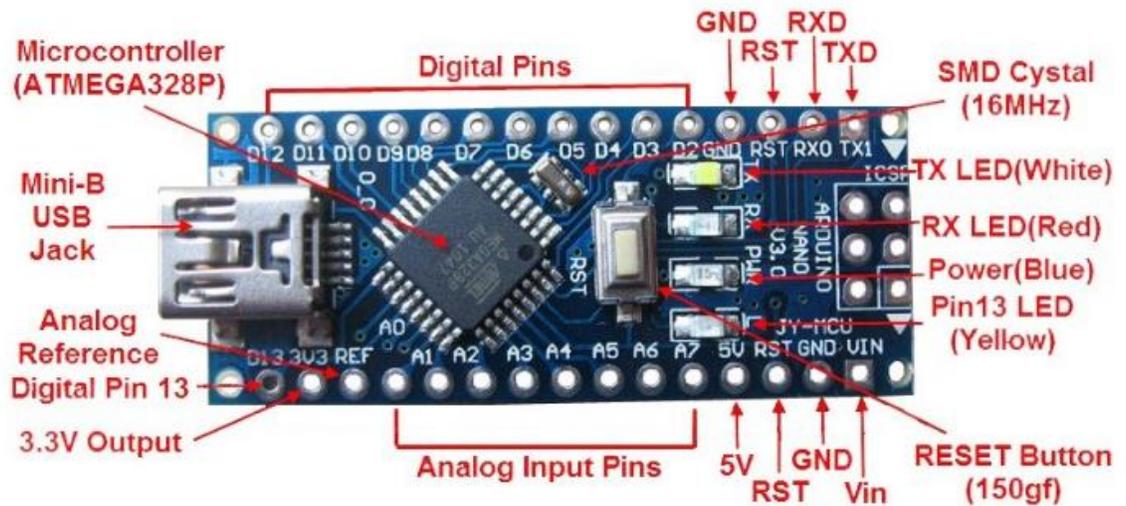
El Arduino Nano es una pequeña y completa placa basada en el ATmega328 (Arduino Nano 3.0) o ATmega168 (Arduino Nano 2.x) que se usa conectándola a una protoboard. Tiene más o menos la misma funcionalidad que el Arduino Duemilanove, pero con una presentación diferente. No posee conector para alimentación externa, y funciona con un cable USB Mini-B en vez del cable estandar. El nano fue diseñado y está siendo producido por Gravitech.

- **Alimentación:** El Arduino Nano puede ser alimentado usando el cable USB Mini-B, con una fuente externa no regulada de 6-20V (pin 30), o con una fuente externa regulada de 5V (pin 27). La fuente de alimentación es seleccionada automáticamente a aquella con mayor tensión.

El chip FTDI FT232RL que posee el Nano solo es alimentado si la placa está siendo alimentada usando el cable USB. como resultado, cuando se utiliza una fuente externa (no USB), la salida de 3.3V (la cual es proporcionada por el chip FTDI) no está disponible y los pines 1 y 0 parpadearán si los pines digitales 0 o 1 están a nivel alto.

- **Memoria:** El ATmega168 posee 16KB de memoria flash para almacenar el código (de los cuales 2KB son usados por el bootloader); el ATmega 328 posee 32KB, (también con 2 KB usados por el bootloader). El ATmega168 posee 1KB de SRAM y 512 bytes de EEPROM (la cual puede ser leída y escrita con la librería EEPROM); el ATmega328 posee 2 KB de SRAM y 1KB de EEPROM.
- **Comunicación:** El Arduino Nano tiene algunos métodos para la comunicación con un PC, otro Arduino, u otros microcontroladores. El ATmega168 y el ATmega328 poseen un módulo UART que funciona con TTL (5V) el cual permite una comunicación vía serie, la cual está disponible usando los pines 0 (RX) y 1 (TX). El chip FTDI FT232RL en la placa hace de puente a través de USB para la comunicación serial y los controladores FTDI (incluidos con el software de Arduino) provee al PC de un puerto com virtual para el software en el PC.
- **Programación:** El ATmega168 o ATmega328 del Arduino Nano vienen pre programados con un bootloader que te permite subir tu código al Arduino sin la necesidad de un programador externo. Se comunica usando el protocolo STK500 original (Archivos cabecera C).

Figura 6: características del arduino nano



Fuente: (Arduino, Arduino: tecnología para todos, s.f.)

### 3.9. ARDUINO IDE

El Entorno de Desarrollo Integrado de Arduino - o Arduino Software (IDE) contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino y Genuino para cargar programas y comunicarse con ellos. (Arduino, 2017).

#### 3.9.1. Escritura de bocetos

Los programas escritos con el software Arduino (IDE) se denominan bocetos. Estos bocetos están escritos en el editor de texto y se guardan con la extensión de archivo.ino. El editor tiene características para cortar / pegar y para buscar / reemplazar texto. El área de mensajes da retroalimentación al guardar y exportar y también muestra errores. La consola muestra el texto generado por el software Arduino (IDE), incluyendo mensajes de error completos y otra información. La esquina inferior derecha de la ventana muestra la tarjeta configurada y el puerto serie. Los botones de la barra de herramientas le

permiten verificar y cargar programas, crear, abrir y guardar bocetos y abrir el monitor en serie. (Arduino, 2017).

Nota: Las versiones del software Arduino (IDE) anteriores a 1.0 salvaron bocetos con la extensión. pde. Es posible abrir estos archivos con la versión 1.0, se le pedirá que guarde el boceto con la extensión.ino al guardar.

Tabla 2: Herramientas de Arduino IDE.

	<p><b>Verificar</b> Comprueba si el código contiene errores al compilarlo.</p>
	<p><b>Subir</b> Compila su código y lo sube al tablero configurado.</p> <p>Nota: Si está utilizando un programador externo con su tarjeta, puede mantener presionada la tecla "shift" en su computadora cuando use este icono. El texto cambiará a "Cargar usando el Programador"</p>
	<p><b>Nuevo</b> Crea un nuevo boceto.</p>
	<p><b>Abrir</b> Presenta un menú de todos los bocetos de su cuaderno de bocetos. Al hacer clic en uno, se abrirá dentro de la ventana actual sobrescribiendo su contenido.</p> <p>Nota: debido a un error en Java, este menú no se desplaza; si necesita abrir un boceto más tarde en la lista, utilice la ficha Archivo   Sketchbook en su lugar.</p>
	<p><b>Guardar</b> Guarda el boceto.</p>
	<p><b>Serial</b> <span style="float: right;"><i>Monitor</i></span> Abre el monitor en serie.</p>

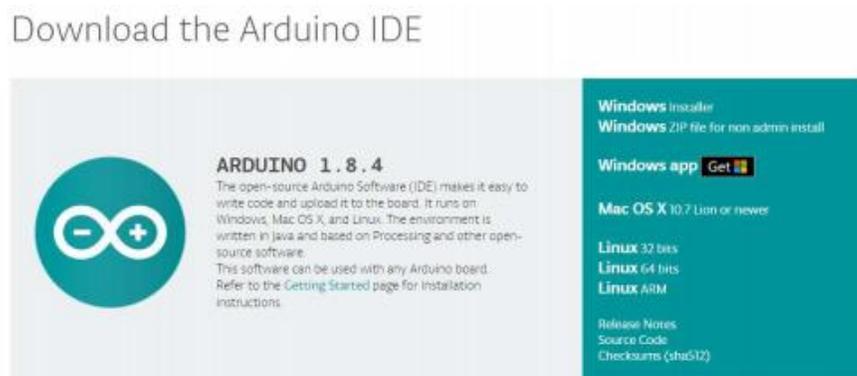
Fuente: (Arduino, 2017)

Los comandos adicionales se encuentran dentro de los cinco menús: Archivo, Editar, Bosquejo, Herramientas, Ayuda. Los menús son sensibles al

contexto, lo que significa que sólo están disponibles los elementos relevantes para el trabajo que se está llevando a cabo.

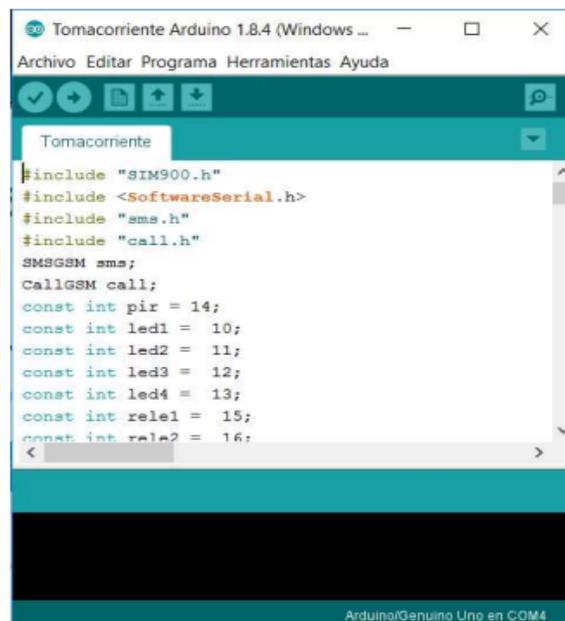
Actualmente se encuentra la versión 1.8.4 descargable desde <https://www.arduino.cc/en/Main/Software>

Figura 7: versión Arduino IDE.



Fuente: (Arduino, 2017).

Figura 8: Entorno de programación Arduino IDE 1.8.5



Fuente: (Arduino, 2017).

### 3.10. MODULO BLUETOOTH

El módulo Bluetooth HC-05 nos permite conectar nuestros proyectos con Arduino a un smartphone, celular o PC de forma inalámbrica (Bluetooth), con la facilidad de operación de un puerto serial. La transmisión se realiza totalmente en forma transparente al programador, por lo que se conecta en forma directa a los pines seriales de nuestro microcontrolador preferido (respetando los niveles de voltaje, ya que el módulo se alimenta con 3.3V). Todos los parámetros del módulo se pueden configurar mediante comandos AT. La placa también incluye un regulador de 3.3V, que permite alimentar el módulo con un voltaje entre 3.6V - 6V. Este módulo es el complemento ideal para nuestros proyectos de robótica, domótica y control remoto con Arduino, PIC, Raspberry PI, ESP8266, ESP32, STM32, etc.

La comunicación Bluetooth se da entre dos tipos de dispositivos: un maestro y un esclavo. Si nuestro objetivo es conectar nuestro proyecto a un smartphone android podemos utilizar tanto un módulo HC-06 o un HC-05 configurado como esclavo. El módulo Bluetooth HC-05 viene configurado de fábrica para trabajar como **esclavo**, es decir, preparado para escuchar peticiones de conexión, pero podemos configurarlo para trabajar con **Maestro** utilizando comandos AT. Por otra parte si nuestro objetivo es conectar dos proyectos, necesitaremos utilizar un módulo HC-05 configurado como maestro y un HC-06 (esclavo) o un HC-05 configurado como esclavo. Este módulo cumple con las especificaciones del estándar Bluetooth 2.0 que es perfectamente compatible con celulares o smartphones Android, más no con los Iphone. Para trabajar con Iphone recomendamos utilizar el Módulo Bluetooth 4.0 BLE HM-10, que también es compatible con los celulares Android modernos. (*Mechatronic*, 2018)

Tabla 3: características del módulo bluetooth hc-05

Características	
Compatible con el protocolo Bluetooth	V2.0.
Voltaje de alimentación:	3.3VDC – 6VDC.
Voltaje de operación	3.3VDC.
Baud rate ajustable	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.
Tamaño	1.73 in x 0.63 in x 0.28 in (4.4 cm x 1.6 cm x 0.7 cm)
Corriente de operación	< 40 mA
Corriente modo sleep	< 1mA

Fuente: (Mechatronic, 2018).

### 3.11. CELDA DE CARGA CON TRANSMISOR HX711

El módulo HX711 es un transmisor entre las celdas de carga y un microcontrolador como Arduino, permitiendo leer el peso en la celda de manera sencilla. Es compatible con las celdas de carga de 5kg, 20kg y 50kg. Utilizado en sistemas de medición automatizada, procesos industriales, industria médica.

El chip HX711 posee internamente la electrónica para la lectura del puente de Wheatstone formado por la celda de carga y también un conversor ADC de 24 bits. Se comunica con el microcontrolador por medio de un protocolo de tipo serial mediante 2 pines (Clock y Data).

Las celdas de carga están formadas por galgas extensiométricas en configuración de puente Wheatstone. Para conectar la celda al módulo HX711 son necesarios 4 cables, los colores utilizados habitualmente son Rojo, Negro, Blanco y Verde. (Mechatronic, 2018)

Cada color corresponde a una señal como se muestra a continuación:

Rojo: Voltaje de excitación +, E+, VCC.

Negro: Voltaje de excitación -, E-, GND.

Blanco: Amplificador +, Señal +, A+.

Verde: Amplificador -, Señal -, A-.

Figura 9: celda de carga.



Fuente: (Mechatronic, 2018)

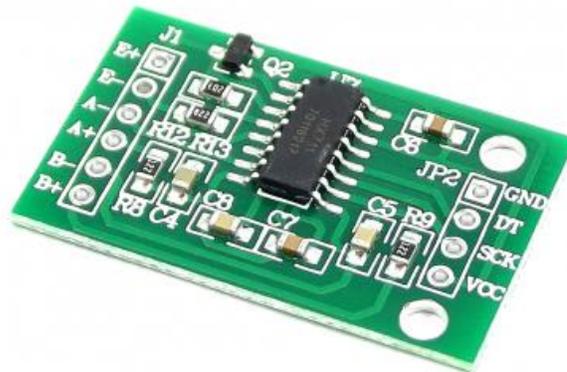
### 3.11.1. Transmisor de celda de carga HX711

Este módulo es una interface entre las celdas de carga y el microcontrolador, permitiendo poder leer el peso de manera sencilla. Internamente se encarga de la lectura del puente wheatstone formado por la celda de carga, convirtiendo la lectura analógica a digital con su conversor A/D interno de 24 bits.

Es muy utilizado en procesos industriales, sistemas de medición automatizada e industria médica.

Se comunica con el microcontrolador mediante 2 pines (Clock y Data) de forma serial.

Figura 10: celda de carga HX711



Fuente: (Mechatronic, 2018)

### **3.12. GAS PROPANO**

El gas propano es un combustible derivado de la refinación del petróleo, que a la temperatura ambiente y a presión atmosférica se encuentra en estado gaseoso.

Características:

- formula:  $C_3H_8$
- punto de ebullición 0 °C
- densidad de vapor (aire 1)
- temperatura de ignición en aire a 580 °C
- límites de inflamabilidad en mezcla con el aire:
- inferior: 2.4 %
- superior: 9.6%

Conocimiento del cilindro: el cilindro está construido en lamina de hierro de diferentes calibres, según la necesidad [ lbs. (hogar)]; [ lbs. (industrial)]. en la parte inferior tiene una base en lamina, para protección del cilindro.

En la parte superior tiene una brida en lamina figurada para protección de la válvula. En la parte superior tiene una válvula para llenado y servicio, donde se instala a la estufa.

En la parte posterior a la válvula de servicio tiene una válvula cheque de seguridad para cuando el cilindro es sometido a presiones. Los cilindros por lo regular tienen dentro de su cuerpo 85% de material líquido y 15% de gas. (Escovedo, 2013)

En cuanto al contenido y al envase, el gas viene en cuatro presentaciones comerciales que van desde 5 KG, 10 KG, 15 KG y 45 KG dirigido a cada tipo de cliente o empresa. (Lima Gas, 2018)

### **3.13. METODOLOGÍA**

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que

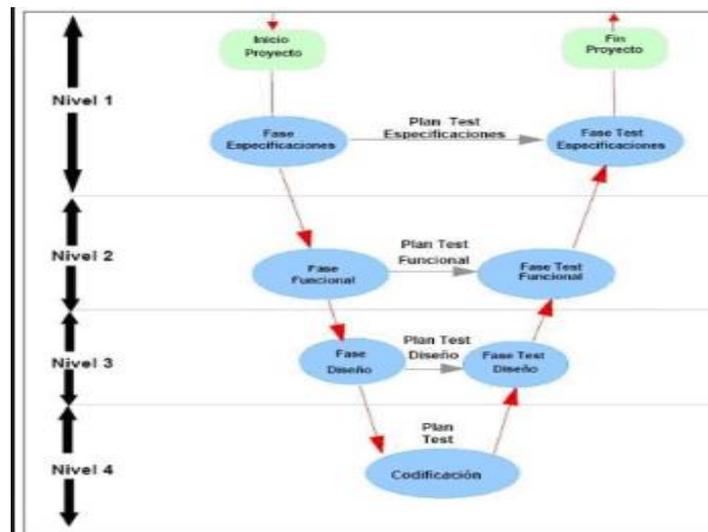
guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo” (Avison & Fitzgerald, 2006).

### 3.13.1. Metodología V

(A. Perez, 2006) La metodología V es un modelo para desarrollo de software embebido, que logra la combinación del software y hardware de confiabilidad realizada, y dirigido a diversos campos de aplicación industrial donde priman las limitaciones de coste, tiempo y confiabilidad.

En el siguiente grafico se muestra el ciclo de vida del modelo V:

Figura 11: Modelo en V



Fuente: metodología de desarrollo de software.

#### Niveles:

- **El nivel 1:** Está orientado al “cliente”. El inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Se compone del

análisis de requisitos y especificaciones, se traduce en un documento de requisitos y especificaciones.

- **El nivel 2:** Se dedica a las características funcionales del sistema propuesto. Puede considerarse el sistema como una caja negra, y caracterizarla únicamente con aquellas funciones que son directa o indirectamente visibles por el usuario final, se traduce en un documento de análisis funcional.
- **El nivel 3:** Define los componentes hardware y software del sistema final, a cuyo conjunto se denomina arquitectura del sistema.
- **El nivel 4:** Es la fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.

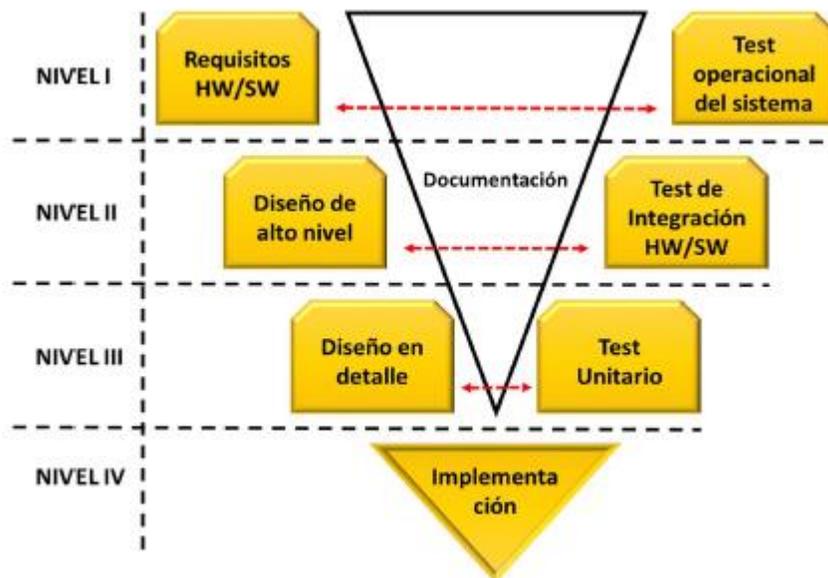
### **Fases del desarrollo**

- (Fase 1) **DEFINICIÓN DE ESPECIFICACIONES:** Se deben definir y documentar los diferentes requisitos del sistema a desarrollar, identificando los valores numéricos más concretos posibles. Entre ellos debe estar la especificación del nivel de integridad, o SIL, en caso de ser requerido.
- (Fase 2) **DISEÑO GLOBAL:** También llamado diseño de alto nivel. Su objetivo es obtener un diseño y visión general del sistema.
- (Fase 3) **DISEÑO EN DETALLE:** Consiste en detallar cada bloque de la fase anterior.
- (Fase 4) **IMPLEMENTACIÓN:** Es la fase en la que se materializa el diseño en detalle.
- (Fase 5) **TEST UNITARIO:** En esta fase se verifica cada módulo HW y SW de forma unitaria, comprobando su funcionamiento adecuado.
- (Fase 6) **INTEGRACIÓN:** En esta fase se integran los distintos módulos que forman el sistema. Como en el caso anterior, ha de generarse un documento de pruebas. Por una parte, se debe comprobar en todo el sistema el funcionamiento correcto, y por otra, en caso de tratarse con un sistema tolerante a fallos, debe verificarse

que ante la presencia de un fallo persiste el funcionamiento correcto. Se comprueba el cumplimiento de los requisitos establecidos.

- (Fase 7) TEST OPERACIONAL DEL SISTEMA: Se realizan las últimas pruebas, pero sobre un escenario real, en su ubicación final, anotando una vez más las pruebas realizadas y los resultados obtenidos.

Figura 12: modelo en v del ciclo de vida.



Fuente: (A. Perez, 2006)

#### 4. CAPITULO IV: METODOLOGÍA DE LA INVESTIGACIÓN

##### 4.1. METODOLOGÍA DE DESARROLLO DEL PROYECTO

Para la ejecución del proyecto se empleará la metodología V, que como principal característica propone la combinación de software y hardware de confiabilidad realizada y dirigido a diversos campos de aplicación industrial donde prevalecen las limitaciones de coste, tiempo y confiabilidad. Por ende, esta metodología es la más adecuada para poder desarrollar este proyecto y cumplir con los objetivos trazados.

A continuación, se detallará las acciones y técnicas a desarrollarse en cada fase de la metodología V.

*Tabla 4: fases, técnicas y acciones del ciclo de vida*

ACCION	TECNICAS	FASE
<b>Gestión del proyecto</b>	Definición de tareas y responsables.	Todas.
	Gestión del estado de las tareas.	Todas.
<b>Definición de requisitos, plan de seguridad y garantía de calidad.</b>	Definición con UML, estructurando los requisitos.	1
	Realizar revisiones tras cada cambio en cada etapa.	Todas.
<b>Documentación</b>	Metodología documentada.	Todas.
	Estructura UML documentada.	Todas.
<b>Recursos</b>	Personal calificado.	Todas.
	Componentes de hardware.	Todas.
	Componentes de software.	Todas.
<b>Codificación</b>	Código fuente, programación en el ID Arduino.	4
<b>Implementación</b>	Guía del diseño del layout.	4
<b>Verificación y validación</b>	Simulaciones.	2y3
	Revisiones.	Todas.
	Autodiagnóstico.	2,3y 4
	Pruebas con usuarios.	5

<b>Instalación y mantenimiento</b>	Personal calificado. Plan definido y amigable.	Posterior. 2y3
<b>Modificaciones</b>	Repetir todos los anteriores a los que afecte el cambio. Control de modificaciones y documentación de los mismos.	Todas. Todas.

*Fuente: elaboración propia.*

#### **4.1.1. NIVEL1 DESARROLLO: FASE DE ESPECIFICACIONES**

Después de analizar la realidad problemática y ver las necesidades de los usuarios se debe tener en cuenta los siguientes requerimientos.

##### **A. REQUERIMIENTOS FUNCIONALES**

- El sistema será capaz de monitorear el consumo de gas doméstico en tiempo real.
- El sistema será capaz de enviar notificaciones al usuario cuando el balón de gas tenga la capacidad del 50% y 10% restante.
- El sistema permitirá al usuario saber que su contenido de gas esta por consumirse en su totalidad.

##### **B. REQUERIMIENTOS NO FUNCIONALES**

###### **Eficiencia:**

- El sistema funcionara con una latencia no mayor a los 5 minutos.
- El sistema funcionara adecuadamente en todo momento sin ningún problema.
- El sistema será solo para plataforma de celular.

**Seguridad:**

- Los datos de la aplicación solo serán modificados por usuarios autorizados.
- La licencia de uso de software donde se aloje y con la que se realice la aplicación debe ser lo menos restrictiva, preferentemente software libre.

**Usabilidad:**

- la App será de interfaz amigable y fácil de usar para el usuario.
- Contará con un manual de usuario.

**C. ARDUINO NANO**

**Arduino:** es una plataforma electrónica de hardware libre basada en una placa con un microcontrolador. 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 costo que permite el desarrollo de múltiples diseños. Por otro lado, el software consiste en un entorno de desarrollo que implementa el lenguaje de programación procesing / wiring y el cargador de arranque que es ejecutado en la placa.

**Microcontrolador:** un microcontrolador es un circuito integrado diseñado con el propósito de tareas específicas. Es principalmente usado en productos que requieren un grado de control impuesto por el usuario.

**Microcontrolador Atmega328:** Arduino Nano es una placa de desarrollo de tamaño compacto, completo y compatible con protoboards, basada en el microcontrolador ATmega328P. Tiene 14 pines de entrada/salida digital (de los cuales 6 pueden ser usando con PWM), 6 entradas analógicas, un cristal de 16Mhz, conexión Mini-USB, terminales para conexión ICSP y un botón de reseteo.

Posee las mismas capacidades que un Arduino UNO, tanto en potencia del microcontrolador como en conectividad, solo se ve recortado en su

conector USB, conector jack de alimentación y los pines cambia un formato de pines header.

*Tabla 5: Características arduino nano*

características	Arduino Nano
Core	Atmega328
Reloj principal	16Mhz
Tamaño de registros	8 bits
RAM	2.5Kb
Flash	32Kb
GPIOs	20 IO +9(Power+ Aux)
Tamaño (laxAnxALmm)	68,6 X 53,3X 10
Real time	SI
Consumo	0.125 W
Precio	S/.99.00
Sistema operativo	Posible

*Fuente: elaboración propia.*

#### **4.1.2. NIVEL 2 DESARROLLO: DISEÑO DE ALTO NIVEL**

El proyecto se desarrolló bajo la plataforma de desarrollo android studio y arduino nano con el circuito integrado.

Para un mejor entendimiento se busca identificar a los interesados del proyecto, los cuales son.

##### **A. ESTABLECIMIENTO DE ACTORES**

Se identifican los actores y el equipo encargado de desarrollar el proyecto, quienes participaron de forma activa durante el desarrollo de la aplicación móvil:

- **Analista de sistemas:** encargado de diseñar la infraestructura de la aplicación móvil, considerando los requerimientos funcionales, alcance y las limitaciones que la aplicación tiene.
- **Programador:** encargado de implementar la aplicación considerando la infraestructura diseñada por el analista del sistema.

## B. ESTABLECIENDO USUARIOS

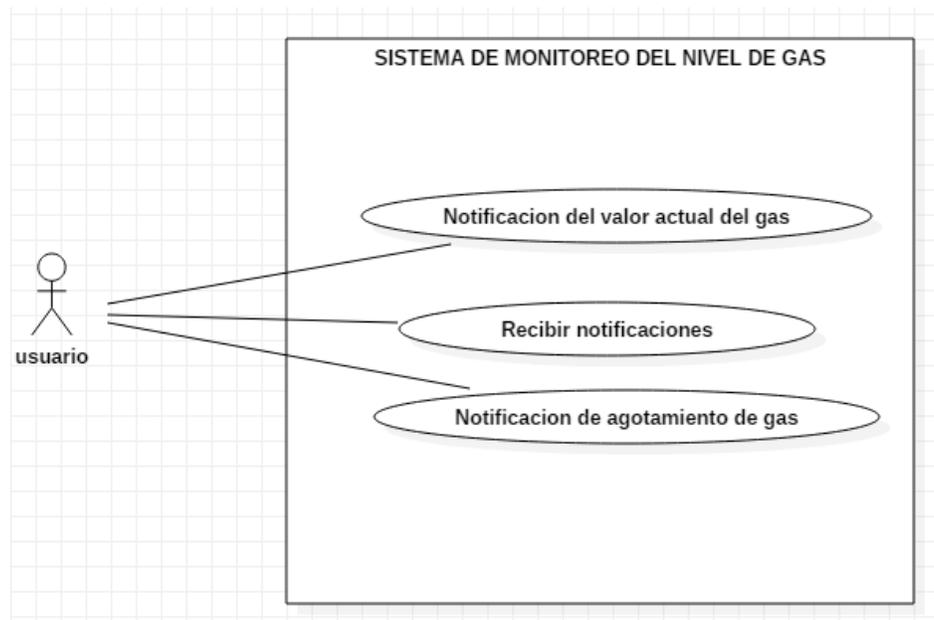
Identificando a los usuarios, quienes serán las personas que harán uso de la aplicación son:

- **Usuarios directos:** amas de casa que vivan en el barrio de Curibamba.
- **Usuarios indirectos:** amas de casa de las diferentes regiones del país.

## C. DIAGRAMA DE CASO DE USO GENERAL

El diagrama de casos de uso, es la descripción de las actividades que se realizan para definir un conjunto de funcionalidades que el sistema debe de cumplir para satisfacer los requerimientos, como se muestra en el siguiente diagrama.

Figura 13: Diagrama de caso de uso general.



Fuente: elaboración propia.

#### D. SUB CASO DE USO 1

En el siguiente caso de uso se realiza la descomposición del caso de uso general para especificar el comportamiento del sistema.

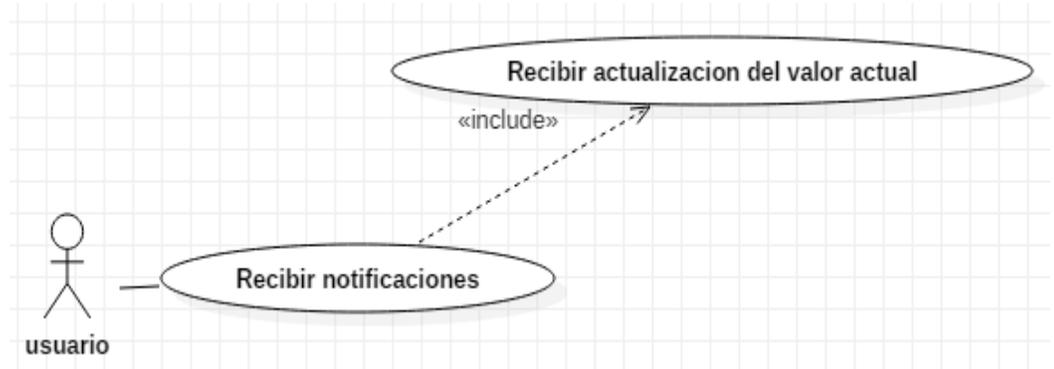
Figura 14: diagrama de sub caso de uso1



Fuente: elaboración propia.

## E. SUB CASO DE USO 2

Figura 15: diagrama de sub caso de uso2

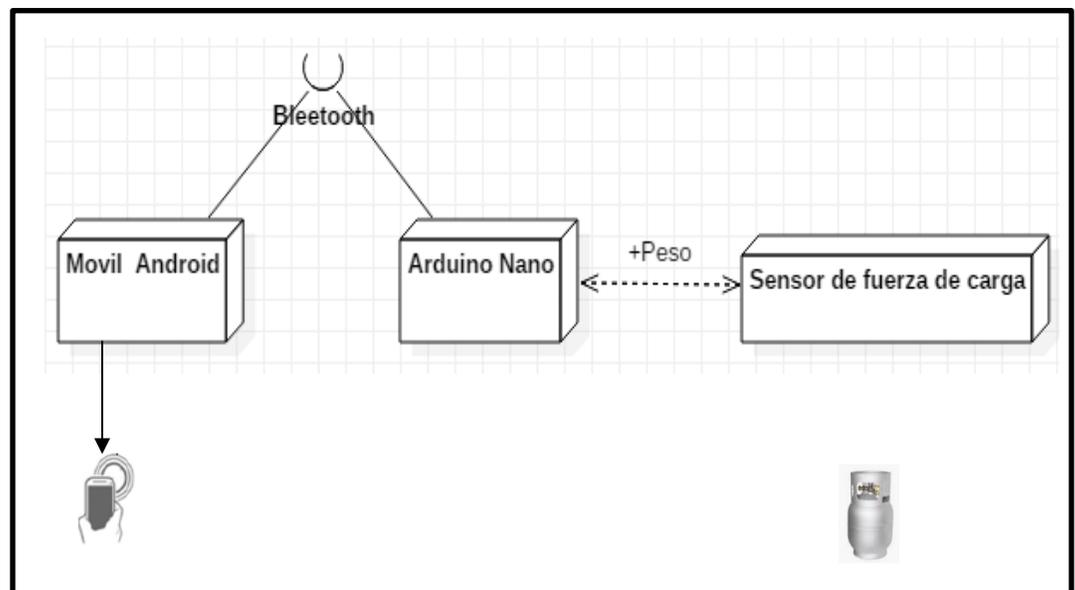


Fuente: elaboración propia.

## F. DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue permite mostrar la arquitectura en tiempo de ejecución del sistema respecto al hardware y software, con este diagrama se muestra básicamente la estructura del sistema, modelando los aspectos físicos.

Figura 16: diagrama de despliegue



Fuente: elaboración propia.

### **Descripción:**

- **Nodo Arduino Nano:** Es en este nodo donde se llevará a cabo la actualización del peso del balón de gas el cual será alimentada por una batería de 5v.
- **Nodo sensor de peso:** Este nodo es el encargado de obtener la variación del peso en porcentaje.
- **Nodo Móvil:** este nodo es necesario para poder visualizar las funcionalidades.

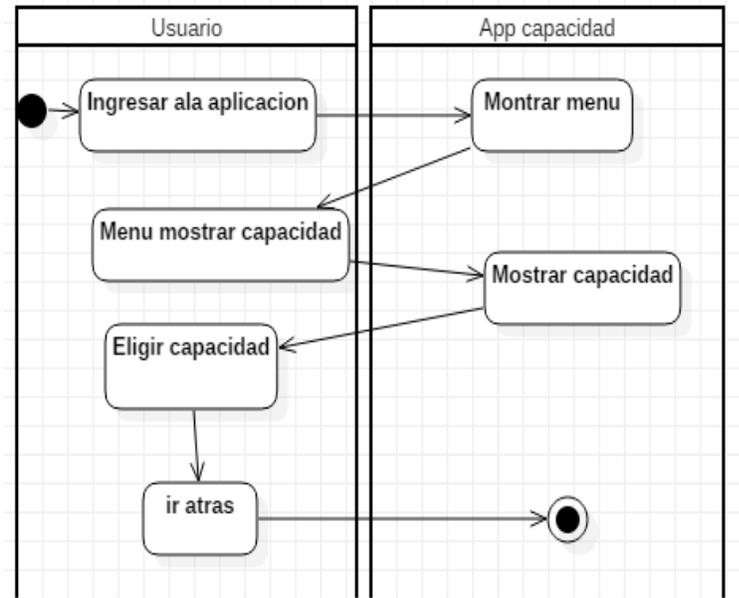
### **4.1.3. NIVEL3 DESARROLLO: DISEÑO EN DETALLE**

#### **A. DIAGRAMA DE ACTIVIDADES**

Un diagrama de actividades muestra el flujo de acciones (ejecución de proceso de nodos), generalmente secuenciales además presenta los resultados de dichas acciones.

- **Diagrama de actividades**

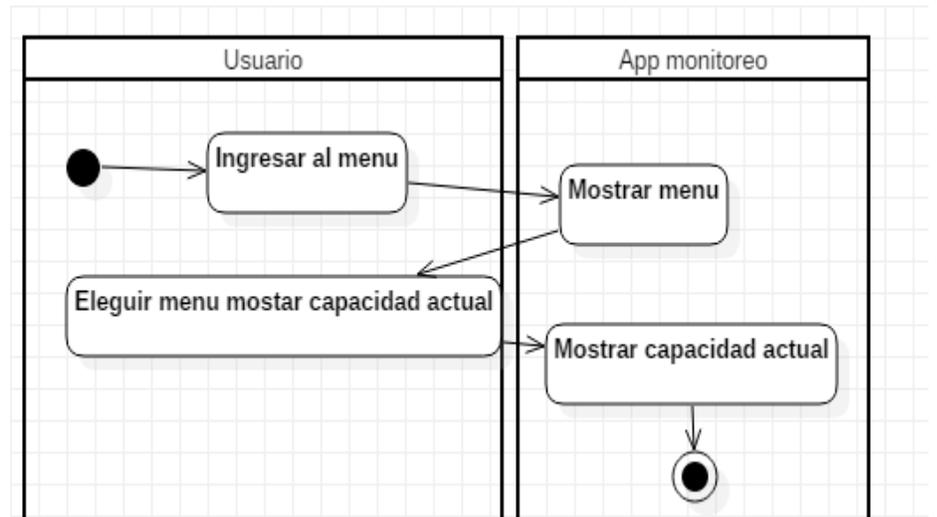
Figura 17: diagrama de actividad elegir capacidad



Fuente: elaboración propia.

- **Diagrama de actividad monitoreo**

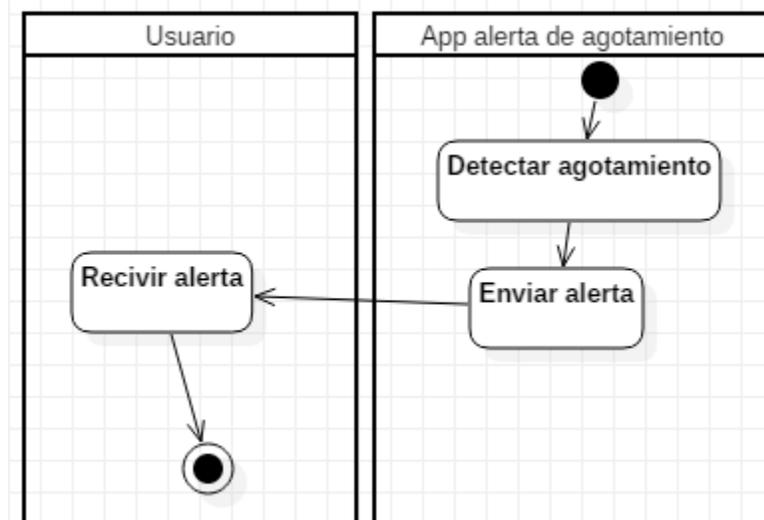
Figura 18: diagrama de actividad monitoreo



Fuente: elaboración propia.

- **Diagrama alerta de agotamiento**

*Figura 19: diagrama alerta de agotamiento.*



*Fuente: elaboración propia.*

En el diseño del proyecto se utilizó uso de los diagramas de casos de uso, diagrama de despliegue y diagrama de actividades, los cuales fueron fundamentales y esenciales para el desarrollo del proyecto. Para este fin se utilizó la herramienta de starUML.

*Tabla 6: casos de uso*

<b>PROTOTIPO DE SISTEMA DE MONITOREO DE BALON DE GAS</b>	
<b>DESCRIPCION DE CASO DE USO 1</b>	
<b>NOMBRE DEL CASO 1</b>	Elegir capacidad del balón de gas
<b>ACTORES:</b>	Usuario
<b>FUNCION:</b>	Elegir el tipo de balón de gas que utilizara
<b>DESCRIPCION:</b>	El usuario al ingresar a la aplicación podrá elegir la capacidad del balón de su domicilio

<b>PRECONDICIONES:</b>	Debe haber ingresado con su usuario y contraseña
<b>DESCRIPCION DE CASO DE USO 2</b>	
<b>NOMBRE DEL CASO 2</b>	Visualización del monitoreo
<b>ACTORES:</b>	Usuario
<b>FUNCION:</b>	Enviar actualización de la capacidad actual
<b>DESCRIPCION:</b>	El sistema podrá censar el peso en tiempo real y enviar el valor actual
<b>PRECONDICIONES:</b>	Identificarse como un usuario y contraseña
<b>DESCRIPCION DE CASO DE USO 3</b>	
<b>NOMBRE DEL CASO 2</b>	Recibir alerta de agotamiento
<b>ACTORES:</b>	Usuario
<b>FUNCION:</b>	Recibir información
<b>DESCRIPCION:</b>	El usuario recibe la notificación el cual le informa que su balón de gas esta por consumirse en su totalidad
<b>PRECONDICIONES:</b>	No cerrar sesión de la App

*Fuente: Elaboración propia.*

#### 4.1.4. NIVEL 4 DESARROLLO: IMPLEMENTACIÓN

##### A. IMPLEMENTACIÓN DEL HARDWARE

Como base del hardware se utilizó Arduino Nano CH340G el cual es una pequeña y completa placa basada en el micro controlador ATmega328P, la cual presenta varias ventajas a diferencia del Arduino mega y uno.

Los materiales necesarios: como mínimo para funcionar el Arduino nano se necesita.

- Adaptador de alimentación con micro USB.
- Teclado.
- Micro SD clase 10 de 16 GB.

- Cable HDMI.
- Mouse.
- Televisor o monitor.

## CONFIGURACIÓN DEL ARDUINO NANO

Descargamos Arduino versión 1.8.9 de la página oficial de Arduino <https://www.arduino.cc/en/Main/Software>, lo ventajoso de descargar directamente es que te dan todas las versiones para todos los SO. En este caso descargaremos el instalador para el sistema operativo de Windows.

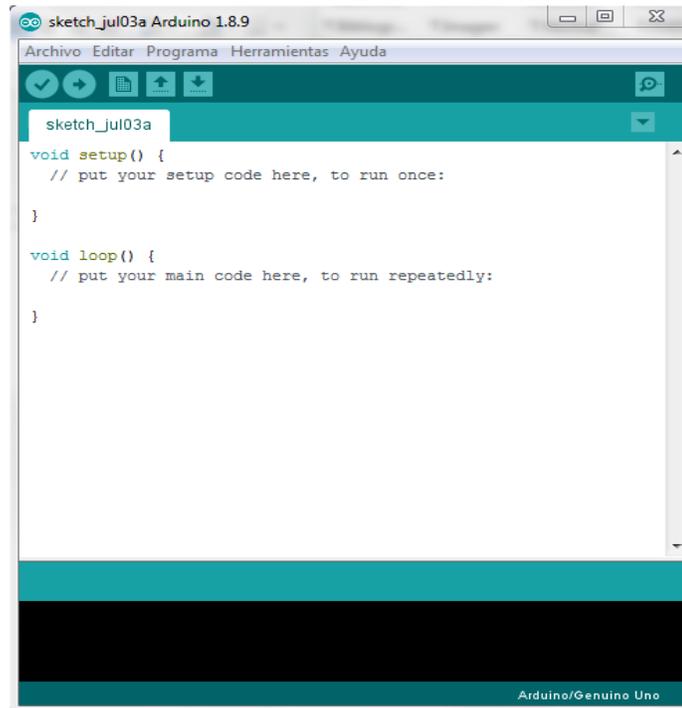
Figura 20: Arduino



Fuente: (Arduino, 2016)

**Instalar Arduino 1.8.9:** ejecutar el instalador de Arduino e instalar en la computadora.

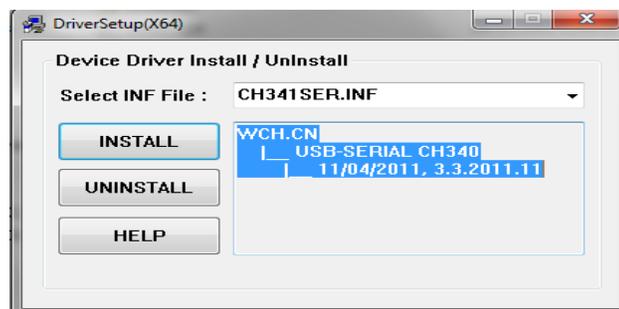
Figura 21: primera pantalla del Arduino



Fuente: (Arduino, 2016)

Luego se realizará la instalación de las librerías para poder realizar la implementación de todo el sistema. En este caso se instalará la librería CH341SER, para que el ordenador pueda reconocer el puerto USB.

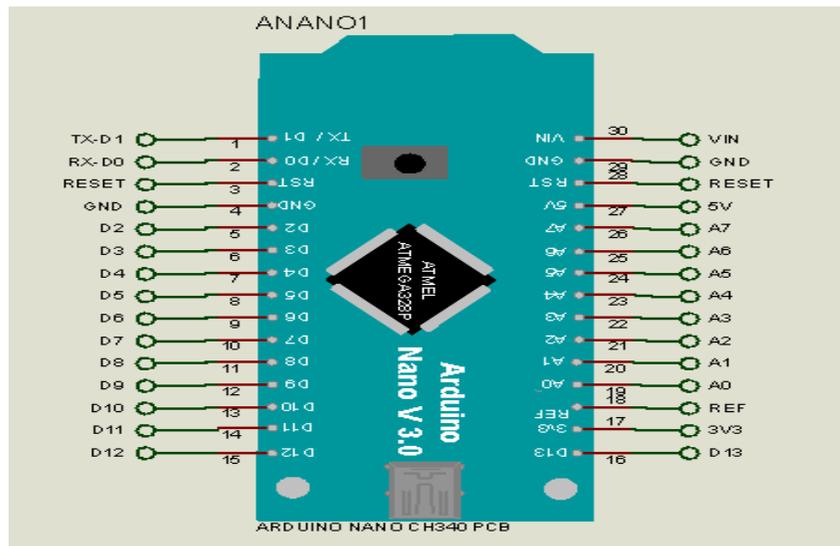
Figura 22: driver de instalación Arduino



Fuente: imagen propia, realizada con recortes de pantalla.

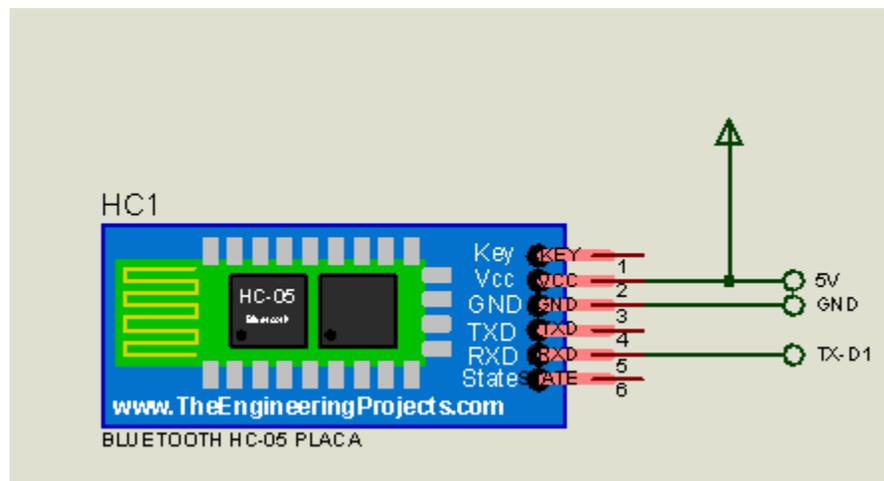
Se realiza el reconocimiento de la placa Arduino nano, el tipo de pines ( entrada, salida, conexión a tierra y fuente de alimentación) y sus respectivos componentes como el transmisor de celda de carga, el bluetooth y los pines de salida, para poder realizar las conexiones y ensamblar con los respectivos componentes que se mencionaran más adelante.

Figura 23: Arduino Nano



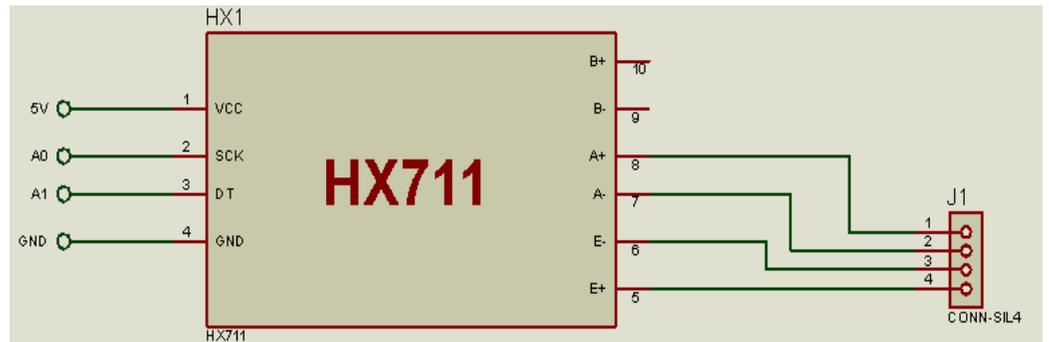
Fuente: imagen propia realizada con recorte de pantalla.

Figura 24: modulo bluetooth



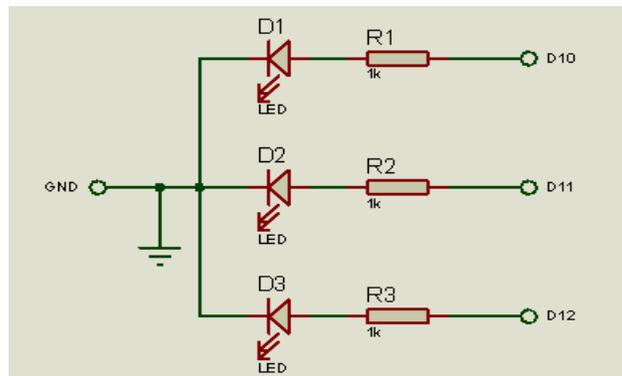
Fuente: imagen propia realizada con recorte de pantalla.

Figura 25: transmisor HX711



Fuente: imagen propia realizada con recorte de pantalla.

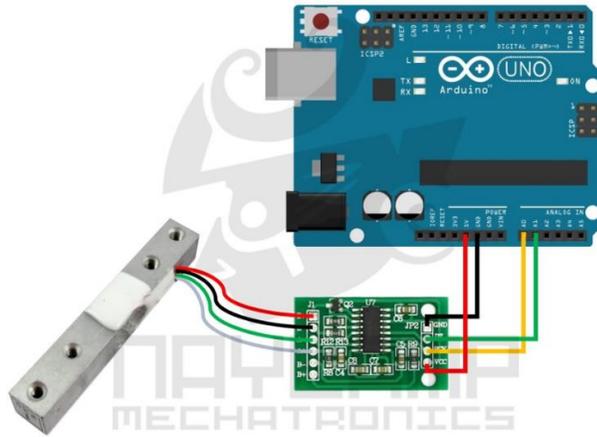
Figura 26: leds de salida



Fuente: imagen propia realizada con recorte de pantalla

**Simulación virtual en proteus:** para poder integrar los componentes primero tendremos que realizar una simulación entre la celda de carga el, el transmisor de carga HX711 y el microcontrolador Arduino nano, ya que la celda de carga solo es capaz de leer datos analógicos lo cual hay que convertir a datos digitales, para evitar futuros inconvenientes.

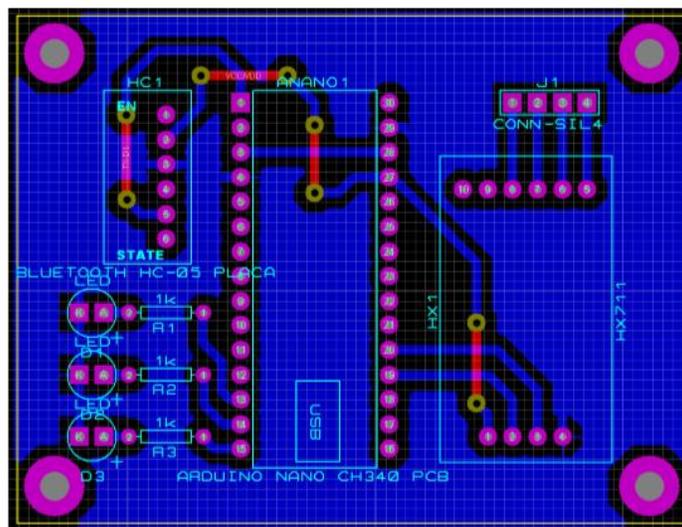
Figura 27: simulación de la celda de carga



Fuente: imagen propia, realizada con recortes de pantalla.

**Armado de la placa:** luego de ver que, si funciona en un entorno virtual, se pasa a diseñar la placa de integración con la ayuda de un ingeniero eléctrico. Esta placa lleva integrada los zócalos para cada uno de los componentes como Arduino nano, modulo bluetooth, celda de carga, transmisor de celda de carga HX711, leds y algunos resistores como parte del circuito integrado.

Figura 28: interior de la placa



Fuente: imagen propia, realizada con recortes de pantalla.

Una vez diseñada el circuito integrado se pasa a quemar en la placa y queda de la siguiente manera.

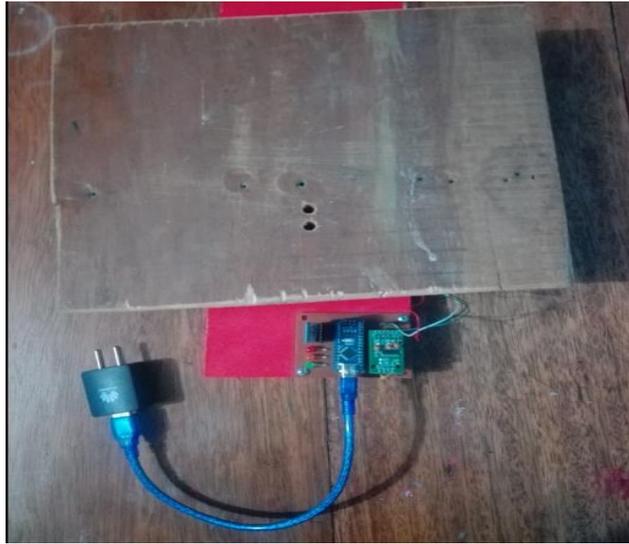
*Figura 29: circuito integrado*



*Fuente: imagen propia, realizada con recortes de pantalla.*

**Producto final:** después de tener el circuito integrado, pasamos a unir todos los componentes en este. El Arduino nano, el módulo bluetooth, la celda de carga y el trasmisor de carga quedan integrados en la placa. Además, se realiza una estructura de madera que se adecue a un prototipo, para poder simular el monitoreo. Al final el producto queda de siguiente manera.

*Figura 30: estructura del prototipo final*

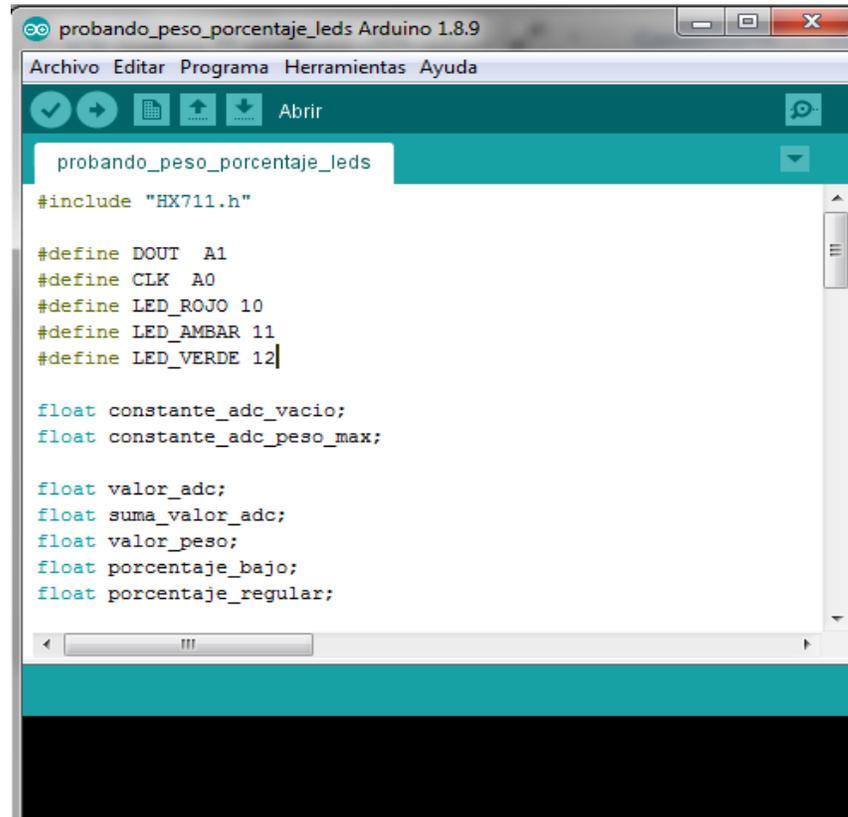


*Fuente: imagen propia, realizada con recortes de pantalla.*

**Parte lógica del circuito:** ahora pasaremos a especificar las condiciones que debe de cumplir para poder monitorear el balón de gas. Y empezamos con el primer punto como se muestra a continuación.

**Captura de los datos del Arduino nano** después de ensamblar pasamos a la fase de procesamiento de datos, los cuales son transmitir los valores actualizados del peso del balón de gas a través de bluetooth a una aplicación móvil.

Figura 31: captura de los datos del Arduino nano



```
probando_peso_porcentaje_leds Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
Abrir
probando_peso_porcentaje_leds
#include "HX711.h"

#define DOUT A1
#define CLK A0
#define LED_ROJO 10
#define LED_AMBAR 11
#define LED_VERDE 12

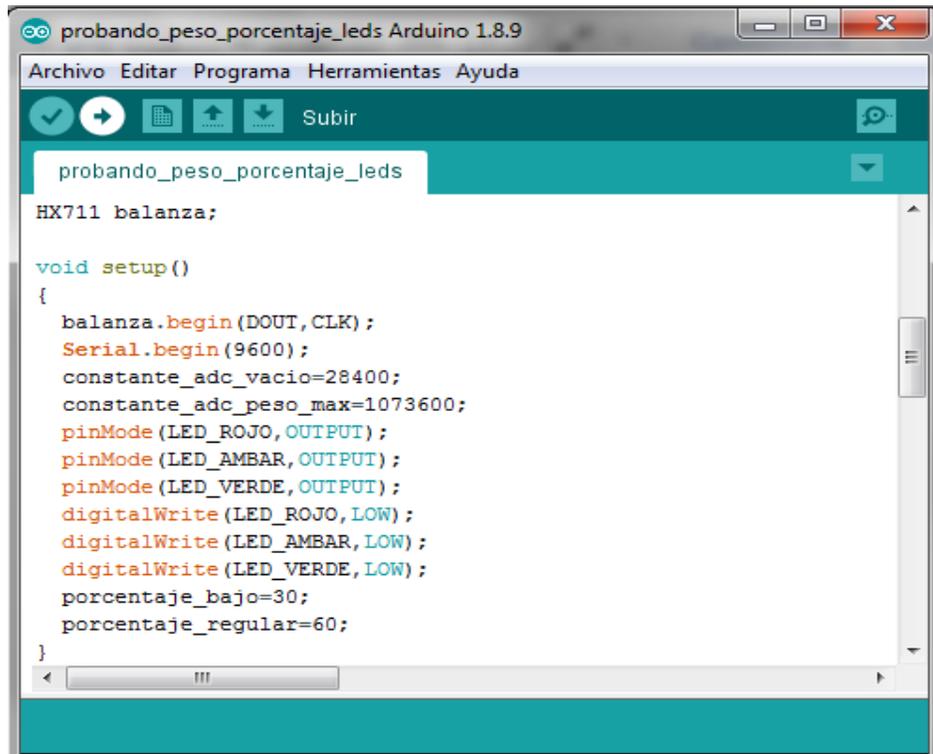
float constante_adc_vacio;
float constante_adc_peso_max;

float valor_adc;
float suma_valor_adc;
float valor_peso;
float porcentaje_bajo;
float porcentaje_regular;
```

Fuente: imagen propia, realizada con recortes de pantalla.

La siguiente imagen detalla cómo se configuran los pines del Arduino nano, para poder utilizarlos como entrada y salida del sistema. De la misma manera inserta la librería del módulo de celda de carga HX711 el cual servirá poder visualizar los datos en la aplicación móvil.

Figura 32: configuración del Arduino.



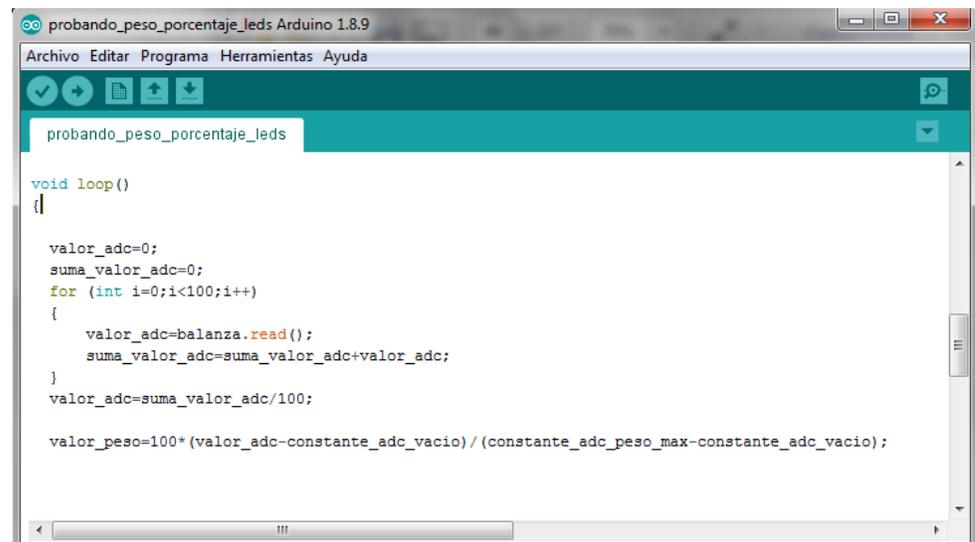
```
probando_peso_porcentaje_leds Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
Subir
probando_peso_porcentaje_leds
HX711 balanza;

void setup()
{
  balanza.begin(DOUT, CLK);
  Serial.begin(9600);
  constante_adc_vacio=28400;
  constante_adc_peso_max=1073600;
  pinMode(LED_ROJO, OUTPUT);
  pinMode(LED_AMBAR, OUTPUT);
  pinMode(LED_VERDE, OUTPUT);
  digitalWrite(LED_ROJO, LOW);
  digitalWrite(LED_AMBAR, LOW);
  digitalWrite(LED_VERDE, LOW);
  porcentaje_bajo=30;
  porcentaje_regular=60;
}
```

Fuente: imagen propia, realizada con recortes de pantalla.

El siguiente código muestra las condiciones de la captura del peso y luego los convierte en porcentaje para poder visualizar de una mejor manera y se indica el valor de “gas lleno”, “gas regular” y manda una alerta cuando el contenido este por debajo del 10%.

Figura 33: código de condiciones.

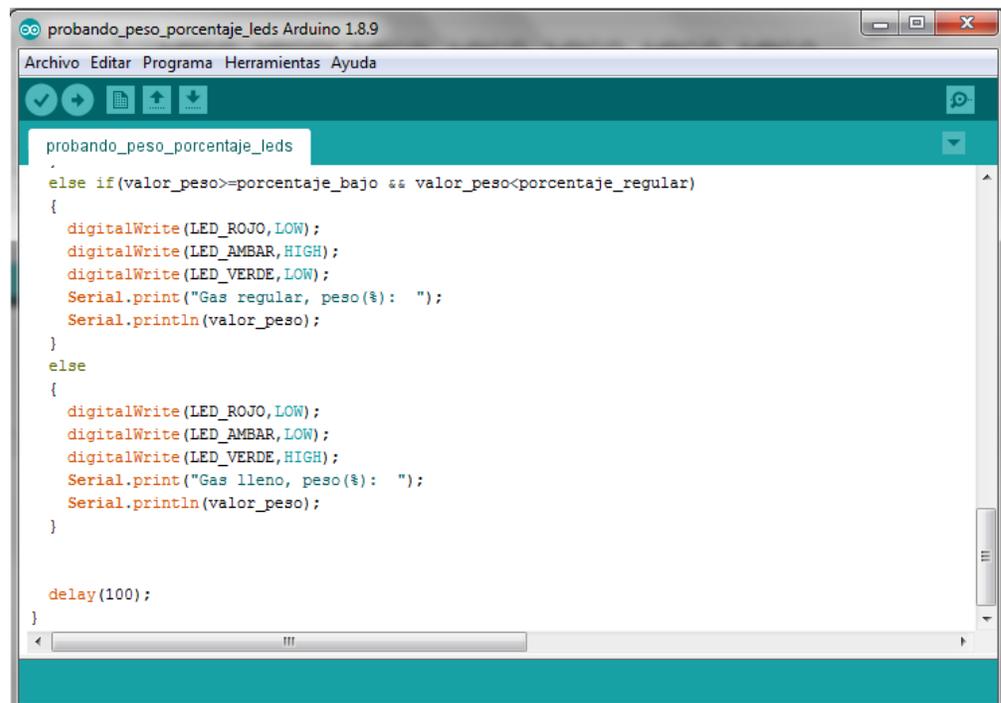


```
void loop()
{
  valor_adc=0;
  suma_valor_adc=0;
  for (int i=0;i<100;i++)
  {
    valor_adc=balanza.read();
    suma_valor_adc=suma_valor_adc+valor_adc;
  }
  valor_adc=suma_valor_adc/100;

  valor_peso=100*(valor_adc-constante_adc_vacio)/(constante_adc_peso_max-constante_adc_vacio);
```

Fuente: imagen propia, realizada con recortes de pantalla.

Figura 34: segunda parte del código



```
else if(valor_peso>=porcentaje_bajo && valor_peso<porcentaje_regular)
{
  digitalWrite(LED_ROJO,LOW);
  digitalWrite(LED_AMBAR,HIGH);
  digitalWrite(LED_VERDE,LOW);
  Serial.print("Gas regular, peso(%): ");
  Serial.println(valor_peso);
}
else
{
  digitalWrite(LED_ROJO,LOW);
  digitalWrite(LED_AMBAR,LOW);
  digitalWrite(LED_VERDE,HIGH);
  Serial.print("Gas lleno, peso(%): ");
  Serial.println(valor_peso);
}

delay(100);
}
```

Fuente: imagen propia, realizada con recortes de pantalla.

## B. IMPLEMENTACIÓN DEL SOTFWARE

El proyecto de aplicación móvil se desarrolló en Android, para lo cual el primer paso es descargar el IDE Android studio para poder programar nuestro proyecto de la siguiente manera.

*Figura 35: Android studio.*

androidstudio

Android Studio provides the fastest tools for building apps on every type of Android device.

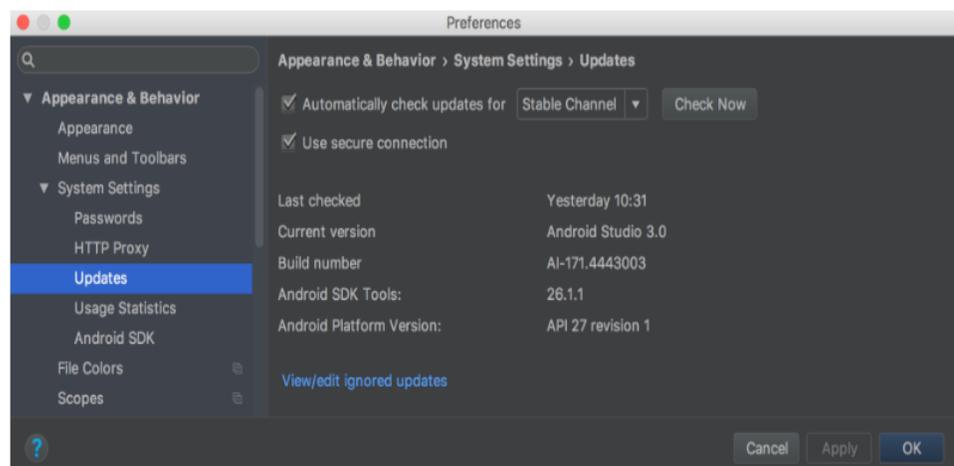
DOWNLOAD ANDROID STUDIO

3.4.1 for Windows 64-bit (971 MB)

*Fuente: (Android, 2018)*

Una vez instalado se procederá a instalar el SDK y algunas herramientas necesarias para que pueda trabajar normalmente como se muestra en la siguiente figura.

*Figura 36: imágenes de instalación de librerías*



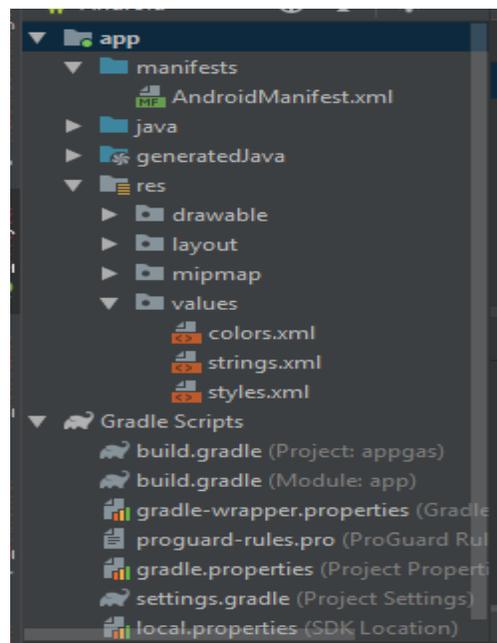
*Fuente: imagen propia, instalación del sdk.*

Una vez instalado todos los requerimientos y librerías de android studio se procede a crear un nuevo proyecto.

### Estructura de directorios y archivos del proyecto

Al crear un nuevo proyecto en el Android studio se crea una estructura definida que permite organizar el código de la siguiente manera como se muestra en el siguiente gráfico.

Figura 37: creación de un proyecto.



Fuente: imagen propia, realizada con recortes de pantalla.

- **Manifests:** aquí podemos encontrar el AndroidManifest. ya que este contiene los recursos que vamos a utilizar a medida que desarrollamos nuestra aplicación. Directorio `"/src/main/AndroidManifest.xml"`.
- **java:** en este directorio encontraremos los ficheros.java con el código fuente. Ubicación: `"/src/main/java"`.

- **Res:**
  - **anim:** archivos XML que definen las animaciones. Directorio: `"/src/main/res/anim"`.
  - **drawable:** en este directorio se encuentran las imágenes en formato PNG o JPEG (Google tiene preferencia por las imágenes PNG). Su ubicación es: `"/src/main/res/drawable"`.
  - **layout:** aquí se encuentran todos los layouts de nuestro proyecto. Ya hemos visto qué es un layout y qué tipos de layouts existen. Se ubican en el directorio `"/src/main/res/layout"`.
  - **menu:** archivos XML que definen las plantillas de los menus del proyecto `"/src/main/res/menu"`.
  - **mipmap:** contiene los iconos de la aplicación con sus diferentes resoluciones. Anteriormente se encontraban en el directorio drawable, pero en las últimas versiones de Android Studio han sido movidos al directorio mipmap. Ubicación `"/src/main/res/mipmap"`.
  - **raw:** en esta carpeta ubicamos los archivos multimedia descomprimidos, de esta manera Android sabe que no debe procesarlos más tarde. Se encuentra en el directorio `"/src/main/res/raw"`.
  - **values:** archivos XML que definen valores constantes. Ya hemos detallado [qué es y para qué sirve la carpeta Values](#), así que solo haremos un pequeño resumen. Se encuentra en: `"/src/main/res/values"`.

## Módulos de la aplicación móvil

### Login:

Se realiza el módulo de usuario y contraseña de la aplicación para esto se establecerá una pequeña base de datos en MYSQL en el local host, para poder almacenar nuestros usuarios. De la misma manera se establece una conexión una con la base de datos y luego las instrucciones en el Android studio. Este es el primer módulo que se cargara automáticamente de la aplicación.

Figura 38: App Login

```
public void onClick(View view) {

    sUsuario=txtUsuario.getText().toString();
    sContrasenia=txtContrasenia.getText().toString();

    JsonRequest JsonRequest =
    "http://192.168.6.196/RecFacPolicial/verificar_login.php";
    JsonRequest = JsonRequest +
    "?Usuario="+sUsuario+"&Contrasenia="+sContrasenia;
    RequestQueue requestQueue =
    Volley.newRequestQueue(context);

    JsonRequestRequest obrq= new
    JsonRequest(Request.Method.GET, JsonRequest, null,
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject
response) {
                try {
                    if (response.getString("estado")
== null ||
response.getString("estado").equals("2") ||
response.getString("estado").equals("-1") ||
response.getString("estado").equals("3")) {
                        Toast.makeText(context,
"Usuario no esta registrado", Toast.LENGTH_LONG).show();
                        tilUsuario.setError("Error de
ingreso Usuario");
                        tilContrasenia.setError("Error de ingreso Contraseña");

                    } else {
                        /*Intent takePictureIntent =
new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                        // Ensure that there's a
camera activity to handle the intent
                        if
(takePictureIntent.resolveActivity(getPackageManager()) !=
null) {
                            // Create the File where
the photo should go
                            File photoFile = null;
                            try {
                                photoFile =
createImageFile();
                            } catch (IOException ex)
{
                                // Error occurred
                                while creating the File
                            }
                        }
                    }
                }
            }
        });
    requestQueue.add(JsonRequestRequest);
}
```

```

// Continue only if the
File was successfully created
        if (photoFile != null) {
            Uri photoURI =
FileProvider.getUriForFile(context, "com.example.android.filep
rovider", photoFile);

takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT,
photoURI);

startActivityForResult(takePictureIntent,
REQUEST_TAKE_PHOTO);
        }
    }*/
    //Toast.makeText(context, "El
usuario si esta registrado", Toast.LENGTH_LONG).show();
    Intent intent = new
Intent(MainActivity.this, DispositivosBT.class);
    startActivity(intent);
    finish();
    }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    }
},
new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError
error) {
        Log.e("Volley", "Error");
        Toast.makeText(context, "Error de
Conexión", Toast.LENGTH_LONG).show();
    }
}
);
requestQueue.add(obrq);
}
});

```

*Fuente: imagen propia, extracto del Android studio.*

Autenticamos al usuario mediante un usuario y password y se realiza la conexión a la base de datos del servidor, mediante el IP de la computadora, que se encuentra en un local host.

Figura 39: login.ap



Fuente: imagen propia, recortes de pantalla de Android studio.

### Integración de los módulos de la página principal

Una vez autenticado el usuario se carga la ventana donde se selecciona el módulo bluetooth, en esta ventana se puede escoger el bluetooth con el que se comunica nuestro prototipo de monitoreo.

Figura 40: ventana de selección de bluetooth

```
public void onResume ()
{
    super.onResume ();
    //-----
    VerificarEstadoBT ();

    // Inicializa la array que contendra la lista de los
    dispositivos bluetooth vinculados
    mPairedDevicesArrayAdapter = new ArrayAdapter (this,
    R.layout.nombre_dispositivos); //<-<- PARTE A MODIFICAR >->->
    // Presenta los dispositivos vinculados en el ListView
    IdLista = (ListView) findViewById (R.id.IdLista);
    IdLista.setAdapter (mPairedDevicesArrayAdapter);
}
```

```

        IdLista.setOnItemClickListener(mDeviceClickListener);
        // Obtiene el adaptador local Bluetooth adapter
        mBtAdapter = BluetoothAdapter.getDefaultAdapter();

        //----- EN CASO DE ERROR -----
        -----
        //SI OBTIENES UN ERROR EN LA LINEA (BluetoothDevice
        device : pairedDevices)
        //CAMBIA LA SIGUIENTE LINEA POR
        //Set <BluetoothDevice> pairedDevices =
        mBtAdapter.getBondedDevices();
        //-----
        -----

        // Obtiene un conjunto de dispositivos actualmente
        emparejados y agrega a 'pairedDevices'
        Set<BluetoothDevice> pairedDevices =
        mBtAdapter.getBondedDevices();

        // Adiciona un dispositivos previo emparejado al array
        if (pairedDevices.size() > 0)
        {
            for (BluetoothDevice device : pairedDevices) { //EN
CASO DE ERROR LEER LA ANTERIOR EXPLICACION
                mPairedDevicesArrayAdapter.add(device.getName() +
"\n" + device.getAddress());
            }
        }
    }

// Configura un (on-click) para la lista
private AdapterView.OnItemClickListener mDeviceClickListener
= new AdapterView.OnItemClickListener() {
    public void onItemClick(AdapterView av, View v, int arg2,
long arg3) {

        // Obtener la dirección MAC del dispositivo, que son
los últimos 17 caracteres en la vista
        String info = ((TextView) v).getText().toString();
        String address = info.substring(info.length() - 17);

        // Realiza un intent para iniciar la siguiente
actividad
        // mientras toma un EXTRA_DEVICE_ADDRESS que es la
dirección MAC.
        Intent i = new Intent(DispositivosBT.this,
UserInterfas.class); //<-<- PARTE A MODIFICAR >->->
        i.putExtra(EXTRA_DEVICE_ADDRESS, address);
        startActivity(i);
    }
};

private void VerificarEstadoBT() {
    // Comprueba que el dispositivo tiene Bluetooth y que
está encendido.
    mBtAdapter= BluetoothAdapter.getDefaultAdapter();

```



```

<RadioButton
    android:id="@+id/radioButton6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="20 Kilos" />
</RadioGroup>

```

Fuente: imagen propia, realizada con recortes pantalla.

El segundo formulario es para poder visualizar el monitoreo en tiempo real del estado actual del balón de gas, para esto se crea un text botón donde se puede visualizar el estado actual en tiempo real.

El tercero es un alert donde notificara al usuario el reporte de alerta cuando el contenido del balón de gas este a un 10% de capacidad. Todo esto se desarrolla en un solo formulario.

Figura 42: formulario alerta

```

bluetoothIn = new Handler() {
    public void handleMessage(android.os.Message msg) {
        if (msg.what == handlerState) {
            String readMessage = (String) msg.obj;
            DataStringIN.append(readMessage);

            int endOfLineIndex = DataStringIN.indexOf("#");

            if (endOfLineIndex > 0) {
                String dataInPrint =
                DataStringIN.substring(0, endOfLineIndex);
                IdBufferIn.setText("N.GAS:" +
                dataInPrint);//<-<- PARTE A MODIFICAR >->->
                int cant=dataInPrint.length();
                //int datalen= dataInPrint.length();
                IdEncender.setText("Tamaño del string:"+
                String.valueOf(cant));
                if (dataInPrint.charAt(0)=='-'){
                    final AlertDialog.Builder alert= new
                AlertDialog.Builder(UserInterfas.this);
                    alert.setMessage("LE SUGERIMOS CAMBIAR DE
                BALON DE GAS ESTE TIENE UN MINIMO NIVEL DE GAS");
                    alert.setTitle(";ATENCIÓN!");
                    alert.setCancelable(true);
                    alert.setPositiveButton("SI", new

```

```

DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface
dialogInterface, int i) {
        dialogInterface.cancel();
    }
});
alert.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface
dialogInterface, int i) {
        dialogInterface.cancel();
    }
});
AlertDialog ale=alert.create();
ale.show();
//String
dijito=dataInPrint.substring(20,20);

    }
    DataStringIN.delete(0,
DataStringIN.length());
    dataInPrint="";
}
//DataStringIN.delete(0, DataStringIN.length());
}
}
};

```

*Fuente: imagen propia, elaborada con recortes de pantalla.*

## 5. CAPITULO V: RESULTADOS

### 5.1. TEST UNITARIO

#### 5.1.1. Prueba de hardware

Antes de comenzar con la implementación se realizó la revisión de cada dispositivo.

- **Arduino nano:** se realizó las pruebas de los pines, para ver si brindan una respuesta en cada uno de ellos, se utilizó algunos leds para comprobar, también se habilitó los puertos de entrada y salida para poder conectar el transmisor de la celda de carga y el módulo bluetooth.
- los demás dispositivos como cable USB, celda de carga y módulo bluetooth fueron adquiridos con garantía y se comprobó su funcionamiento mientras se iba armando el módulo.

#### 5.1.2. Pruebas de software

Mientras se desarrollaba el código fuente de la aplicación se iba realizando la compilación, si es que hubo errores quedaba en manos del programador ir corrigiéndolos.

En el desarrollo de la programación de instrucciones para el Arduino, se tuvo especial consideración con la librería del puerto USB y la compatibilidad del sistema operativo Windows, solo se puede subir a la placa Arduino nano desde el sistema operativo Windows 7. Como se muestra a continuación.

Figura 43: cargar la información a Arduino nano

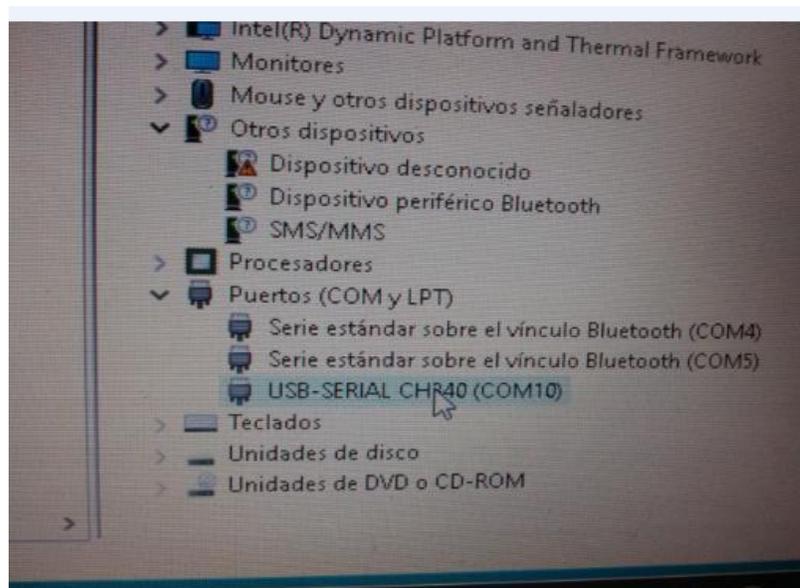
```
digitalWrite(LED_ROJO,LOW);
digitalWrite(LED_AMBAR,LOW);
digitalWrite(LED_VERDE,HIGH);
Serial.print("Gas lleno, peso( ): ");
Serial.println(valor_peso);
Serial.println("#");
}

Problema subiendo a la placa. Visita http://www.arduino.cc/en/Guide/Troubleshooting#upload para sugerencia
```

Fuente: imagen propia, realizada con recortes de pantalla.

Inmediatamente después se procede a cambiar de puerto de USB y se soluciona el problema:

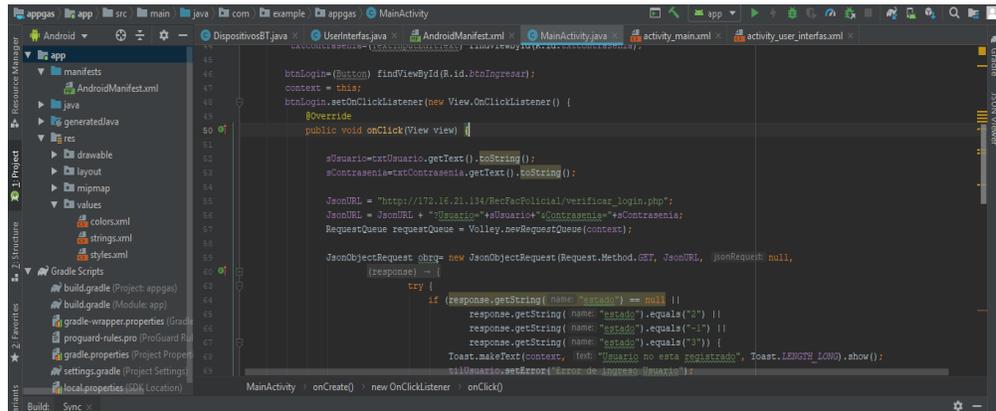
Figura 44: cambio de puerto USB



Fuente: imagen propia, realizada con recortes de pantalla.

Similarmente, las pruebas de la aplicación móvil, ya que se tiene que tener cuidado con la conexión a la base de datos, ya que el IP del ordenador es sumamente importante.

Figura 45: prueba de conexión con la base de datos



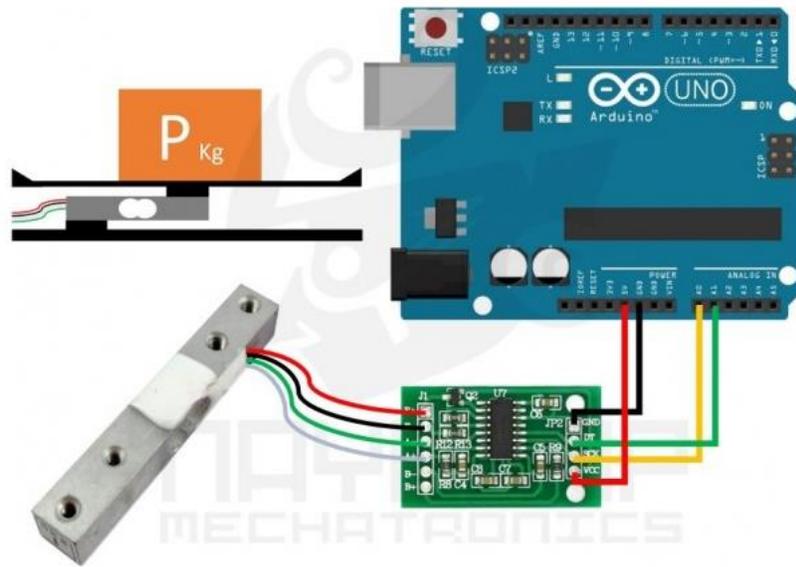
```
45 btnLogin.setOnClickListener {
46     findViewById(R.id.btnIngresar);
47     context = this;
48     btnLogin.setOnClickListener {
49         @Override
50         public void onClick(View view) {
51
52             sUsuario=txtUsuario.getText().toString();
53             sContraseña=txtContraseña.getText().toString();
54
55             JsonURL = "https://72.14.21.134/RestFulLocal/verifica_login.php";
56             JsonURL = JsonURL + "?Usuario="+sUsuario+"&Contraseña="+sContraseña;
57             RequestQueue requestQueue = Volley.newRequestQueue(context);
58
59             JSONObjectRequest objReq= new JSONObjectRequest(Request.Method.GET, JsonURL, JsonRequest: null,
60                 (response) => {
61                 try {
62                     if (response.getString(name: "estado") == null ||
63                         response.getString(name: "estado").equals("1") ||
64                         response.getString(name: "estado").equals("2") ||
65                         response.getString(name: "estado").equals("3")) {
66                         Toast.makeText(context, "El usuario no esta registrado", Toast.LENGTH_LONG).show();
67                     }
68                     nullUserLoginError("Error de acceso de usuario");
69                 }
70             });
71         }
72     }
73 }
```

Fuente: imagen propia, elaborado con recortes de pantalla.

## 5.2. TEST DE INTEGRACION

Partimos de la conexión entre el Arduino nano y el módulo de celda de carga más el transmisor de celda de carga HX711.

Figura 46: diagrama de conexión en el proteuz



Fuente: imagen propia, realizada con recortes de pantalla.

Es fundamental la conexión entre la celda de carga y el transmisor de carga HX711 ya que este se encarga de convertir la lectura analógica del puente de Winston a digital con su conversor A/D interno de 24 bits, como se muestra en el gráfico anterior.

### Transmisión del resultado por bluetooth

Con el código fuente compilado y subido al microcontrolador Arduino nano podemos visualizar el estado actual de balón de gas en tiempo real. Para esto primero se instala el módulo bluetooth hc-05 y el terminal del mismo nombre, para que sirva de interfaz. La prueba se está realizando con un Balón de gas de capacidad 5kg

Figura 47: terminal bluetooth hc-05 transmisión de datos



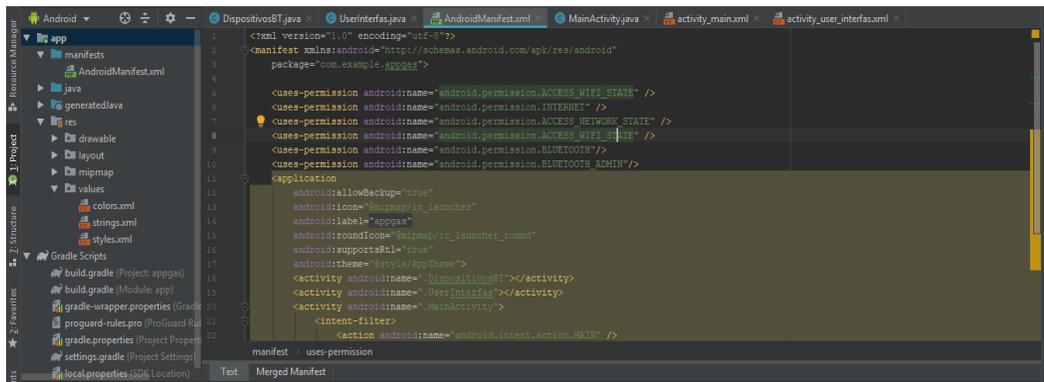
Received Log:

```
[15 jul. 2019 1:28:57 PM] ASCII:  
-Gas bajo, peso(%): 0.05  
#-Gas bajo, peso(%): 0.06  
#-Gas bajo, peso(%): 0.05  
#-Gas bajo, peso(%): 0.06  
#
```

Fuente: imagen propia, realizada con recortes de pantalla.

Una vez realizada la prueba de terminal de bluetooth y visualizar que datos está recibiendo nuestro prototipo. Se procede a integrar nuestro prototipo y la aplicación móvil que servirá de interfaz del usuario.

Figura 48: instrucciones de conexión bluetooth entre la aplicación móvil y circuito integrado



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.appgas">
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".DispositivosBT"></activity>
        <activity android:name=".UserInterfas"></activity>
        <activity android:name=".MainActivity"></activity>
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
        </intent-filter>
    </application>
</manifest>
```

Fuente: imagen propia, elaborada con recortes de pantalla.

La aplicación móvil al final cuenta con la siguiente interfaz una de registro de usuario. Para que una de ama de casa empieza a registrarse para hacer uso de la aplicación móvil.

Figura 49: interfaz de registro de usuario



Fuente: imagen propia, elaborada con recortes de pantalla.

Una vez registrado, el usuario podrá conectarse al prototipo de monitoreo o vincularse al bluetooth del prototipo.

Figura 50: conexión a bluetooth



*Fuente: imagen propia, realizada con recortes de pantalla.*

Después de conectarse al bluetooth entramos a la interfaz donde podemos elegir la capacidad del balón de gas en este caso un balón de 5kg de capacidad, así mismo el monitoreo en tiempo real del balón de gas. Similarmente se activará la alerta de agotamiento cuando el contenido de gas indique el 10% del contenido.

Las pruebas se realizaron con tres valones de gas de 5kg de capacidad.

La primera con un balón de gas completamente lleno. Como se muestra a continuación.

Figura 51: prueba final con balón de 5Kg



**Fuente:** imagen propia, realizada con recortes de pantalla.

La segunda prueba se realizó con un balón de gas de contenido medio.

Figura 52: prueba final con balón de gas de capacidad media



**Fuente:** imagen propia realizada con recortes de pantalla.

La última prueba se realiza con un balón de gas con contenido de baja cantidad.

Figura 53: prueba final balón de gas con poca cantidad



Fuente: imagen propia, realizada con recortes de pantalla.

## 6. CAPITULO VI: DISCUSIÓN

- En primer lugar, todos los materiales que se utilizaron son de bajo costo, con los cuales se pudo realizar un proyecto que sea accesible al bolsillo de las amas de casa.
- Antes de realizar las pruebas se tuvo en consideración tres cosas:
  - o El primero que el circuito tiene un peso que se le denomina tara, al final el peso total sería igual a la tara de balón de gas más el peso del contenido.
  - o Segundo el balón de gas sin contenido tiene un peso de 5kg.
  - o Tercero el contenido del balón de gas es de 10kg, 5kg.

Todas estas diferencias se trabajan en la programación porque al final, el prototipo funcione solo con el valor del contenido del balón de gas.

- Las pruebas que se desarrolló fueron para un balón de gas de 5kg el cual estaba en uso por unas dos horas y se pudo notar con claridad cómo iba variando el resultado en tiempo real a medida que este se consume.
- Después de haber realizado pruebas de cada uno de los componentes se puede observar que se cumple con el principal objetivo.

## CONCLUSIONES

El objetivo de esta tesis era implementar un prototipo de monitoreo de gas en balones domésticos, para poder saber el estado actual en tiempo real del gas doméstico y de esta manera facilitarles la vida a las amas de casa y evitándole molestias generadas por la mala gestión de sus balones de gas. Al concluir se logra cumplir todos los sub objetivos plateados inicialmente.

- Se analizó la arquitectura Arduino nano y se ha logrado implementar el prototipo y se mostró su gran funcionabilidad con otros dispositivos, así mismo la compatibilidad con la aplicación móvil ya que estos pueden comunicarse entre sí sin ningún problema.
- Se ha diseñado y desarrollado la aplicación móvil en Android con una interfaz amigable y sencilla de usar, cuya finalidad es registrar al usuario, monitorear estado del gas en el balón de gas y servir de interfaz al usuario final, en este caso la ama de casa.
- En esta tesis se desarrolló el módulo mostrar valor actual en tiempo real del contenido del balón de gas a medida que este se consume.
- Se logró realizar las pruebas del funcionamiento del monitoreo haciendo uso de la celda de carga, llegando a obtener el valor real del contenido de gas y luego convertirlos a porcentaje.

## RECOMENDACIONES

Para poder realizar la conexión entre el Arduino nano y la celda de carga, se recomienda primero establecer una conexión entre el transmisor de carga HX711 y celda de carga ya que este último emite señales analógicas por lo que es oportuno obtener señales digitales. De esta manera habrá una compatibilidad entre el Arduino y la celda de carga.

Con respecto a la aplicación móvil se recomienda realizar bien la conexión al módulo bluetooth ya que un error mínimo implica que aplicación y la placa no puedan comunicarse entre sí.

Cada módulo que se va desarrollando se tiene que ir probando y solucionando los errores que pudieran presentar, como exige la metodología V, la cual es recomendada para el desarrollo de estos sistemas embebidos.

Para futuras investigaciones se tiene que tener en cuenta que el proyecto está enfocado exclusivamente en el monitoreo del contenido del balón de gas, sin embargo, se observa la necesidad de que en un futuro se incorpore más funcionalidades. Por ejemplo, se podría desarrollar otra aplicación para un proveedor de gas y hacer pedidos en línea y así aprovechar al máximo es te proyecto.

## REFERENCIAS BIBLIOGRAFICAS

- A. Perez, O. (2006). una metodologia para el desarrollo de hardware y sistemas.
- Alcaras, M. (09 de 21 de 2014). *Internet de las cosas*. Obtenido de Internet de las cosas: <http://jeuazarru.com/wp-content/uploads/2014/10/Internet-of-Things.pdf>
- Android. (2013). *FlooMe, un alcoholímetro que funciona con tu smartphone*. Obtenido de FlooMe, un alcoholímetro que funciona con tu smartphone: <https://www.xatakandroid.com/gadgets-android/floome-un-alcoholimetro-que-funciona-con-tu-smartphone>
- Arduino. (2016). *Arduino Genuino*. Obtenido de Arduino Genuino: <https://www.arduino.cc/en/Guide/Introduction>
- Arduino. (2017). *Arduino software IDE*. Obtenido de Arduino software IDE: <https://www.arduino.cc/en/Guide/Environment>
- arduino. (2019). *Arduino*. Obtenido de Arduino: <https://www.arduino.cc/>
- Arduino. (2019). *Arduino*. Obtenido de Arduino: <https://www.arduino.cc/>
- Arduino. (s.f.). *Arduino: tecnologia para todos*. Obtenido de Arduino: tecnologia para todos: <https://arduinodhtics.weebly.com/historia.html>
- Avison, D., & Fitzgerald, G. (2006). methodologies, techniques and tools. *Information systems development*, 15-18.
- Baez, M., Borrego, A., & Cordero, J. (2012). *Introducción a Android*. Obtenido de Introducción a Android.: <http://www.itdocs.net/ddata/18.pdf>
- Burgos Rodas, D. A., & Echeverry Aguirry, H. J. (12 de 04 de 2012). *Estado del arte del uso de aplicaciones e dispositivos moviles en el area de la telemedicina*. Obtenido de Estado del arte del uso de aplicaciones e dispositivos moviles en el area de la telemedicina: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/2740/0053B957.pdf?sequence=1&isAllowed=y>
- Chilinquina. (2010). Ecuador.
- Chiliquinga, V. (2010). *Estudio comparativo de las tecnologías asp, ajax, j2ee, j2me y .net para el desarrollo de aplicaciones móviles, aplicado al desarrollo de una aplicación móvil web para*

*registro de calificaciones y asistencia de los alumnos de la escuela de ingeniería de Ecuador.*

- ELECTROTEC. (2017). *electrotec electrónica y robótica*. Obtenido de <http://electrotec.pe/tienda/shield-gsmgprs-sim900>: <http://electrotec.pe/tienda/shield-gsmgprs-sim900>
- Escovedo, G. S. (Junio de 2013). *Osinergrmin*. Obtenido de Osinegrmin: <http://www.osinergrmin.gob.pe/newweb/uploads/Publico/OficinaComunicaciones/EventoRealizados/ForoTumbes/2/2.%20Seguridad%20en%20la%20comercializacion%20de%20G LP.pdf>
- Esteban Perez , A. (06 de 2013). *Diseño de aplicación móvil para la comunicacion inalambrica de señales audiovisuales*. Obtenido de Diseño de aplicación móvil para la comunicacion inalambrica de señales audiovisuales: [https://upcommons.upc.edu/bitstream/handle/2099.1/21895/alberto.esteban.perez\\_90828.pdf?sequence=1&isAllowed=y](https://upcommons.upc.edu/bitstream/handle/2099.1/21895/alberto.esteban.perez_90828.pdf?sequence=1&isAllowed=y)
- Evans, D. (11 de 04 de 2011). *Cisco Internet Business Solution*. Obtenido de Cisco Internet Business Solution: [https://www.cisco.com/c/dam/global/es\\_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf](https://www.cisco.com/c/dam/global/es_mx/solutions/executive/assets/pdf/internet-of-things-iot-ibsg.pdf)
- Fabricio, G. (2013). *diseño de un sistema de control domotico y video vigilancia supervisado por el telefono movil*. Lima.
- Fabricio, G. (2013). *diseño de un sistema de control domotico y video vigilancia supervisado por telefono movil*. Lima.
- Forero Marin, M. M. (09 de 11 de 2009). *ALARMA MULTIMEDIA VÍA SISTEMA DE TELEFONÍA GSM/GPRS*. Obtenido de ALARMA MULTIMEDIA VÍA SISTEMA DE TELEFONÍA GSM/GPRS: <https://e-archivo.uc3m.es/bitstream/handle/10016/8705/Memoria%20Alarma%20Multimedia.pdf>
- Gas tracker. (2016). *Gas tracker*. Obtenido de Gas traker: <http://gastracker.com.mx/>
- GasWatch*. (2018). Obtenido de GasWatch: <http://www.gaswatch.com>
- Gestion, G. (08 de 01 de 2013). *Gestion*. Obtenido de Gestion: <https://gestion.pe/tendencias/16-peruanos-cuenta-smartphone-29297>
- Gomez, R. (2013). *Sistemas Operativos endispositivos móviles*. Madrid: Universidad Pontificia de Salamanca.

- IEEE. (21 de 09 de 2014). *Telecomunicaciones Network World*. Obtenido de Telecomunicaciones Network World: <http://www.networkworld.es/telecomunicaciones/ieee-disenara-un-marco-estandar-para-el-internet-de-las-cosas>
- INEI. (16 de 06 de 2017). *Instituto Nacional de Estadística e Informática*. Obtenido de Instituto Nacional de Estadística e Informática: <https://www.inei.gob.pe/prensa/noticias/el-834-de-los-hogares-del-pais-utilizan-gas-para-preparar-los-alimentos-9805/>
- ITU. (24 de 11 de 2014). *Informe sobre medición de la sociedad de la información 2014*. Obtenido de Informe sobre medición de la sociedad de la información 2014: [https://www.itu.int/dms\\_pub/itu-d/opb/ind/D-IND-ICTOI-2014-SUM-PDF-S.pdf](https://www.itu.int/dms_pub/itu-d/opb/ind/D-IND-ICTOI-2014-SUM-PDF-S.pdf)
- Lima Gas. (2018). *Lima gas*. Obtenido de lima gas : <https://www.limagas.com/cliente-domiciliario.php>
- Martinez. (2011). Aplicaciones para dispositivos móviles.
- Mechatronic, N. (2018). *Naylamp mechatronic*. Obtenido de Naylamp mechatronic: <https://naylampmechatronics.com/inalambrico/43-modulo-bluetooth-hc05.html>
- Molina Rivera, Y. J., Sandoval Cardona, J., & Toledo Franco, S. A. (12 de 03 de 2012). *Repositorio UTP*. Obtenido de Repositorio UTP: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/2687/0053M722.pdf;jsessionid=3A958CBD23DEE30AE1761C051217A036?sequence=1>
- Morales Hernades, R. A. (2012). Aplicaciones domóticas con android y arduino. *Revista tecnologica*, 6-10.
- Pedrozo Petrazzini, G. (2012). *Sistemas operativos en dispositivos móviles*. Argentina.
- Perez, A. E. (Junio de 2013). *upcommons*. Obtenido de upcommons: [https://upcommons.upc.edu/bitstream/handle/2099.1/21895/alberto.esteban.perez\\_90828.pdf?sequence=1&isAllowed=y](https://upcommons.upc.edu/bitstream/handle/2099.1/21895/alberto.esteban.perez_90828.pdf?sequence=1&isAllowed=y)
- PROMETEC. (2017). *MÓDULO GSM/GPRS*. Obtenido de MÓDULO GSM/GPRS: <http://www.prometec.net/gprs-llamar-enviar-sms/>
- SEACE. (2007). *recipientes portátiles para el gas licuado de petróleo*. Peru.
- Sotelo Martinez, S., Olivo Florez, M. A., & Rodriguez Ortiz, J. G. (2015). *La mecatronica en Mexico*. Obtenido de La mecatronica en Mexico: <http://www.mecamex.net/revistas/LMEM/>
- tracker, g. (2016). *gas tracker*. Obtenido de gastracker: <http://gastracker.com.mx/>
- wikipedia. (30 de Junio de 2019). Obtenido de [https://es.wikipedia.org/wiki/Android\\_Studio](https://es.wikipedia.org/wiki/Android_Studio)

Wolf, G., & et-al. (2014). En *Fundamentos de sistemas operativos*.

Zambrano, R. (agosto de 2014). *mobidoo*. Obtenido de Desarrollo de aplicaciones móviles: ¿nativas, multiplataforma, HTML5 o híbridas?: <http://www.mobidoo.es/desarrollo-movil/desarrollo-de-aplicaciones-moviles-nativas-multiplataforma-html5-hibridas/>

## ANEXOS

### ANEXO1: DATOS GENERALES

#### 1. TITULO DEL PROYECTO

Implementar un prototipo, para monitorear el nivel de gas en balones domésticos.

#### 2. AUTOR DEL PROYECTO

Nombres y apellidos : Estefanie Herminia Neraly Apaza Román

Escuela Profesional : Ingeniería de Sistemas

E-mail : [neralyar@gmail.com](mailto:neralyar@gmail.com)

#### 3. ASSESSOR DEL PROYECTO

Ingeniero: Juan José Ore Cerón.

#### 4. LINEA DE INVESTIGACION

Ingeniería de software.

#### 5. AREA PRIORIZADA DEL PROYECTO

04 03 02 02

#### 6. INSTITUCION Y LUGAR DE EJECUCION DEL PROYECTO

En las viviendas donde se preparen alimentos con gas propano.

#### 7. DURACION DEL PROYECTO

4 meses

## ANEXO2: CRONOGRAMA DE EJECUCION DEL PROYECTO

Tabla 7: cronograma de ejecución del proyecto

Nombre de tarea	Duración	Comienzo	Fin
<b>PROYECTO</b>	<b>111 días</b>	<b>lun 18/02/19</b>	<b>lun 22/07/19</b>
<b>EXPLORACION</b>	<b>18 días</b>	<b>lun 18/02/19</b>	<b>mié 13/03/19</b>
estudio y planteamiento del problema	2 días	lun 18/02/19	mar 19/02/19
identificar el objetivo general	2 días	mié 20/02/19	jue 21/02/19
identificar los clientes	2 días	vie 22/02/19	lun 25/02/19
recopilar toda la informacion necesaria	12 días	mar 26/02/19	mié 13/03/19
<b>INICIACION</b>	<b>28 días</b>	<b>jue 14/03/19</b>	<b>lun 22/04/19</b>
<b>Analisis</b>	<b>12 días</b>	<b>jue 14/03/19</b>	<b>vie 29/03/19</b>
analisis de requerimientos	2 días	jue 14/03/19	vie 15/03/19
analisis de requerimientos no funcionales	2 días	lun 18/03/19	mar 19/03/19
analisis de datos	8 días	mié 20/03/19	vie 29/03/19
<b>Diseño</b>	<b>16 días</b>	<b>lun 01/04/19</b>	<b>lun 22/04/19</b>
diseño de diagramas de casos de uso	3 días	lun 01/04/19	mié 03/04/19
diseño del prototipo arduino	5 días	jue 04/04/19	mié 10/04/19
diseño del prototipo amdroid	8 días	jue 11/04/19	lun 22/04/19
<b>PRODUCCION</b>	<b>43 días</b>	<b>mar 23/04/19</b>	<b>jue 20/06/19</b>
desarrollo	40 días	mar 23/04/19	lun 17/06/19
pruebas unitarias	3 días	mar 18/06/19	jue 20/06/19
<b>PRUEBAS</b>	<b>6 días</b>	<b>vie 21/06/19</b>	<b>vie 28/06/19</b>
pruebas de aceptacion	3 días	vie 21/06/19	mar 25/06/19
correccion de errores	3 días	mié 26/06/19	vie 28/06/19
<b>DOCUMENTACION</b>	<b>16 días</b>	<b>lun 01/07/19</b>	<b>lun 22/07/19</b>
elaboracion del informe final	12 días	lun 01/07/19	mar 16/07/19
revison del informe	3 días	mié 17/07/19	vie 19/07/19
sustentacion de tesis	1 día	lun 22/07/19	lun 22/07/19

Fuente: elaboración propia.

### ANEXO3: PRESUPUESTO

Tabla 8: presupuesto del proyecto

PRESUPUESTO					
ITEM	DESCRIPCION	CANTIDAD	UNID.MEDIDA	precio unitario	precio parcial
<b>1.0</b>	<b>BIENES</b>				<b>S/. 4.327,00</b>
<b>1.1</b>	<b>Equipos</b>				<b>S/. 4.300,00</b>
1.1.1	Laptop	1	unidad	S/. 3.000,00	S/. 3.000,00
1.1.2	USB	2	unidad	S/. 25,00	S/. 50,00
1.1.3	Tablet	1	unidad	S/. 400,00	S/. 400,00
1.1.4	Smartphone	1	unidad	S/. 600,00	S/. 600,00
1.1.5	Arduino	1	unidad	S/. 150,00	S/. 150,00
1.1.6	Dispositivos de implementación para la placa de Arduino	1	Unidad de cada uno	S/. 100,00	S/. 100,00
<b>1.2</b>	<b>Materiales de Escritorio</b>				<b>S/. 27,00</b>
1.2.1	papel bond	1	millar	S/. 25,00	S/. 25,00
1.2.2	lapiceros	4	unidades	S/. 0,50	S/. 2,00
<b>2.0</b>	<b>SERVICIOS</b>				<b>S/. 460,00</b>
2.1	internet	7	meses	S/. 30,00	S/. 210,00
2.2	fotocopias	1000	unidad	S/. 0,10	S/. 100,00
2.3	impresiones	1500	unidad	S/. 0,10	S/. 150,00
<b>3.0</b>	<b>SOFTWARE</b>				<b>S/. -</b>
3.1	AndoidStudio	1	unidad	S/. -	S/. -
3.2	Java	1	unidad	S/. -	S/. -
3.3	Arduino IDE	1	unidad	S/. -	S/. -
<b>COSTO TOTAL DEL PROYECTO</b>					<b>S/. 4.787,00</b>

Fuente: elaboración propia.

## MANUAL DE USUARIO

### MANUAL DE USO DEL PROTOTIPO DE MONITOREO DE GAS

#### Manual del usuario

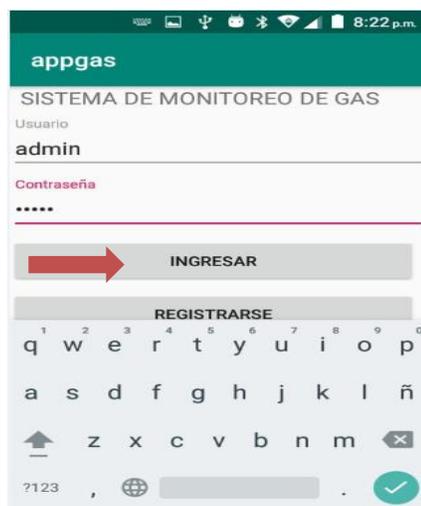
Antes utilizar el aplicativo móvil el usuario tendrá que ubicar el prototipo del arduino debajo de su balón de gas. Luego procederá a utilizar la aplicación



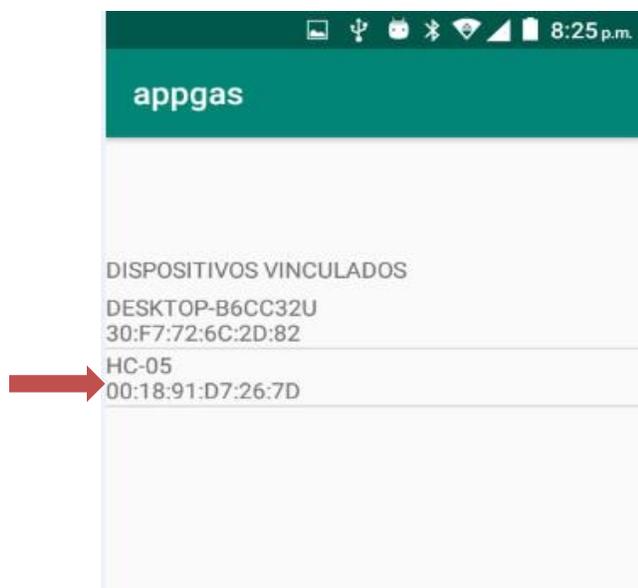
1. primero el usuario tendrá que activar el bluetooth de su teléfono luego ingresar al menú del dispositivo móvil, ejecutar la aplicación e iniciar sesión, el usuario para iniciar sesión deberá registrarse mediante usuario y contraseña.

A screenshot of a mobile application interface. At the top, there is a green header with the text 'appgas'. Below the header, the title 'SISTEMA DE MONITOREO DE GAS' is displayed. There are two input fields: 'Usuario' and 'Contraseña'. Below the input fields, there are two buttons: 'INGRESAR' and 'REGISTRARSE'. The status bar at the top shows the time as 8:32 a.m. and various icons for connectivity and battery.

2. Registro de un nuevo usuario: ingresar en el usuario: nombre y contraseña, pulse en el botón ingresar.



3. luego seleccionar el bluetooth del prototipo



4. ingresara al menú principal este menú tiene las siguientes opciones
  - Deseleccionar tipo de balón de gas de 10kg o el de 5kg
  - Reportar monitoreo: estado actual del contenido del balón de gas

- Alerta cuando el contenido del balón de gas este menor igual al 10% del contenido total.



5. para poder salir de la aplicación solo presionar el botón desconectar