

**UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**IMPLEMENTACIÓN DE UN PROTOTIPO CON  
TECNOLOGÍA ARDUINO Y ANDROID, PARA LA  
ECOEFICIENCIA EN EL USO DEL AGUA POTABLE  
EN LOS PREDIOS DE TALAVERA.**

Presentado por

Bach. DAVID CENTENO CÁCERES

**TRABAJO TESIS PARA OBTENER EL TÍTULO  
PROFESIONAL DE INGENIERO DE SISTEMAS**

**ANDAHUAYLAS – APURÍMAC – PERÚ**

**2017**

**UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



Presentado por

**Bach. DAVID CENTENO CÁCERES**

**IMPLEMENTACIÓN DE UN PROTOTIPO CON  
TECNOLOGÍA ARDUINO Y ANDROID, PARA LA  
ECOEFICIENCIA EN EL USO DEL AGUA POTABLE  
EN LOS PREDIOS DE TALAVERA.**

**Asesor:**

M.sc. Edwin Roque Tito

**Co – Asesor:**

M.sc. Carlos Yinmel Castro Buleje

**ANDAHUAYLAS – APURÍMAC – PERÚ**

**2017**

## **DEDICATORIA**

Quiero dedicarle este trabajo a Dios que me ha dado la vida y fortaleza para concluir con este proyecto de investigación, a mis Padres por el apoyo incondicional que me brindaron y estar presentes cuando más los necesité; en especial a mi madre por su ayuda y constante cooperación durante mi formación profesional y a todas aquellas personas que me dieron aliento para seguir adelante y no desmayar en el camino.

## **AGRADECIMIENTO**

Primero y como más importante, agradezco a mi asesor M.sc. Edwin Roque Tito y co-asesor M.sc. Carlos Yinmel Castro Buleje, por su esfuerzo y dedicación.

Sus conocimientos, sus orientaciones, su paciencia y su motivación has sido fundamentales para mi formación como investigador. Ellos han inculcado en mí un sentido de seriedad, responsabilidad y rigor académico sin los cuales no podría tener una formación completa como profesional.

A la ingeniera M.sc. Cecilia Edith García Rivas Plata, Dr. Angel Fernando Navarro Raymundo, por la asesoría de mi proyecto cuando era estudiante universitario.

Agradezco a la vida por darme la oportunidad de conocer a ingenieros(as) que me apoyaron en mi proyecto, ingenieros(as) que me enseñaron a superar dificultades y así superarme profesionalmente.

A mi abuelita que siempre me aliento ser el mejor profesional y marca la diferencia del resto, me supo guiar por el buen camino demostrar la humildad donde me encuentre y ser persona de bien y nunca perder los valores inculcados.

Agradezco a Dios por darme la salud que tengo, por tener una familia tan unida que siempre estuvo en las buenas y malas junto a mí.

## INDICE

<b>1.CAPITULO I.....</b>	<b>11</b>
1.1.TÍTULO DEL PROYECTO .....	11
1.2.AUTOR DEL PROYECTO .....	11
1.3.ASESOR DEL PROYECTO .....	11
1.4.LÍNEA DE INVESTIGACIÓN.....	11
1.5.ÁREA PRIORIZADA DEL PROYECTO.....	11
1.6.INSTITUCIÓN Y LUGAR DE EJECUCIÓN DEL PROYECTO .....	11
1.7.DURACIÓN DEL PROYECTO.....	11
1.8.CRONOGRAMA DE EJECUCIÓN DEL PROYECTO .....	11
1.9.PRESUPUESTO.....	13
1.10.FINANCIAMIENTO .....	14
<b>2.CAPITULO II: PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>15</b>
<b>2.1.REALIDAD PROBLEMÁTICA.....</b>	<b>15</b>
<b>2.2.FORMULACIÓN DEL PROBLEMA.....</b>	<b>20</b>
<b>2.3.OBJETIVOS .....</b>	<b>20</b>
2.3.1.OBJETIVO GENERAL.....	20
2.3.2.OBJETIVOS ESPECÍFICOS .....	20
<b>2.4.JUSTIFICACIÓN .....</b>	<b>20</b>
<b>2.5.VIABILIDAD DE LA INVESTIGACIÓN.....</b>	<b>21</b>
2.5.1.Viabilidad Técnica .....	21
2.5.2.Viabilidad Operativa .....	21
2.5.3.Viabilidad Económica .....	21
<b>2.6.LIMITACIÓN DEL ESTUDIO .....</b>	<b>22</b>
<b>3.CAPITULO III: MARCO TEÓRICO.....</b>	<b>23</b>
3.1.Marco conceptual .....	23
3.2.Sistema.....	23
3.2.1.Clasificación de sistemas .....	23
3.3.Domótica .....	24
3.3.1.Definición .....	24
3.3.2.Definición de (domo-tic-a) .....	24
3.4.Tecnología.....	24
3.4.1.Uso de la tecnología en las viviendas.....	25
3.4.2.Viviendas inteligentes.....	26
3.5.Internet de las cosas.....	26
3.5.1.El Internet de las cosas como una red de redes .....	27

3.6.Arduino .....	27
3.6.1.Tipos de Arduinos .....	27
3.6.2.Arduino mega .....	28
3.6.3.Modulo bluetooth HC-06.....	28
3.7.Teléfonos inteligentes .....	29
3.7.1.Características .....	29
3.8.Android .....	31
3.8.1.Características de Android .....	31
3.8.2.Versiones de Android .....	32
3.8.3.Uso de smartphones en Perú .....	33
3.9.Sensor .....	33
3.9.1.Sensor YF-S201 .....	34
3.9.2.Sensor Efecto hall .....	34
3.9.3.Sensor de infrarrojo .....	35
3.10.Bluetooth .....	36
3.10.1.Descripción de la tecnología Bluetooth.....	36
3.10.2.Características del bluetooth .....	37
3.11.Teclado numérico .....	38
3.12.Ecoeficiencia.....	39
3.13.Metodología de desarrollo.....	40
3.14.Sistemas embebidos.....	40
3.14.1.Características de un sistema embebido .....	41
3.14.2.Sistemas embebidos en Arduino .....	41
3.14.3.Sistemas inteligentes.....	42
3.15.Autoridad nacional del agua.....	42
<b>3.16.ESTADO DEL ARTE .....</b>	<b>43</b>
3.16.1.Ámbito nacional.....	43
3.16.2.Ámbito internacional.....	43
<b>4.CAPITULO IV: INGENIERÍA DE PROYECTOS.....</b>	<b>45</b>
<b>4.1.METODOLOGÍA Y PLANIFICACIÓN DEL PROYECTO ECOEFICIENTE .....</b>	<b>45</b>
<b>4.2.PUESTA EN MARCHA DE LA METODOLOGÍA V .....</b>	<b>45</b>
4.2.1.REQUERIMIENTO DE HARDWARE Y SOFTWARE .....	45
4.2.1.1.Requerimientos funcionales .....	46
4.2.1.2.Arduino y Raspberry Pi.....	46
4.2.1.3.Requerimientos no funcionales.....	47
<b>4.2.2.DISEÑO DE ALTO NIVEL .....</b>	<b>48</b>

4.2.2.1. Estableciendo actores .....	48
4.2.2.2. Estableciendo Usuarios .....	49
4.2.2.3. Información clave del proyecto .....	49
4.2.2.4. Diagrama de caso de uso general .....	50
<b>4.2.3. DISEÑO DETALLADO .....</b>	<b>50</b>
<b>4.2.4. IMPLEMENTACION .....</b>	<b>56</b>
4.2.4.1. Implementación en ambos entornos de desarrollo .....	56
<b>4.2.5. FASE DE PRUEBAS DEL SISTEMA .....</b>	<b>91</b>
4.2.5.1. VERIFICACIÓN UNITARIA .....	92
4.2.5.2. TEST DE INTEGRACIÓN .....	92
4.2.5.3. TEST DE CAMPO .....	98
4.2.5.4. ENTREVISTA USUARIOS .....	112
<b>RESULTADOS .....</b>	<b>115</b>
<b>CONCLUSIONES .....</b>	<b>119</b>
<b>RECOMENDACIONES .....</b>	<b>120</b>
<b>REFERENCIAS BIBLIOGRAFICAS: .....</b>	<b>121</b>
<b>ANEXOS .....</b>	<b>123</b>

## LISTA DE TABLAS

Tabla 1: <i>Tabla de presupuesto.</i> .....	13
Tabla 2: <i>Detalle de Arduino Mega.</i> .....	28
Tabla 3: <i>Características de modulo bluetooth hc-06.</i> .....	29
Tabla 4: <i>Sistemas operativos más usados.</i> .....	29
Tabla 5: <i>Especificación del sensor hall.</i> .....	35
Tabla 6: <i>Información del proyecto.</i> .....	49
Tabla 7: <i>Especificacion de caso de uso abrir válvula lava manos.</i> .....	51
Tabla 8: <i>Especificacion de caso de uso Cerrar válvula lava manos.</i> .....	51
Tabla 9: <i>Especificación de casos de uso abrir válvula de descarga inodoro.</i> .....	52
Tabla 10: <i>Especificación de casos de uso Cerrar válvula de descarga inodoro.</i> .....	53
Tabla 11: <i>Especificación de casos de uso abrir válvula de la ducha.</i> .....	54
Tabla 12: <i>Especificación de casos de uso cerrar válvula de la ducha.</i> .....	55
Tabla 13: <i>Especificación de casos de uso visualizar consumo de agua.</i> .....	56
Tabla 14: <i>Muestra recolectada.</i> .....	74
Tabla 15: <i>Encuesta del funcionamiento del sistema.</i> .....	114
Tabla 16: <i>Datos pregunta N° 1 - N° 7.</i> .....	115
Tabla 17: <i>Datos pregunta N° 2 – N° 8.</i> .....	115
Tabla 18: <i>Datos pregunta N° 3 – N° 9.</i> .....	115
Tabla 19: <i>Datos pregunta N° 4 – N° 10.</i> .....	116
Tabla 20: <i>Datos pregunta N° 5.</i> .....	116
Tabla 21: <i>Datos pregunta N° 6.</i> .....	117
Tabla 22: <i>Datos pregunta N° 1 – N° 7.</i> .....	117
Tabla 23: <i>Datos pregunta N° 2 – N° 8.</i> .....	117
Tabla 24: <i>Datos pregunta N° 3 – N° 9.</i> .....	118
Tabla 25: <i>Datos pregunta N° 4 – N° 10.</i> .....	118
Tabla 26: <i>Datos pregunta N° 5.</i> .....	118
Tabla 27: <i>Datos pregunta N° 6.</i> .....	118

## LISTA DE FIGURAS

Figura 1: <i>Cronograma de actividades.</i> .....	12
Figura 2: <i>Disponibilidad y distribución del agua en el mundo.</i> .....	17
Figura 3: <i>desglaciación del nevado pastoruri.</i> .....	18
Figura 4: <i>Posesión y utilización de smartphones por país.</i> .....	33
Figura 5: <i>Sensor de efecto Hall.</i> .....	34
Figura 6: <i>Sensor infrarrojo.</i> .....	35
Figura 7: <i>Modelo de conectividad de Bluetooth.</i> .....	36
Figura 8: <i>Interfaz del teclado numérico.</i> .....	39
Figura 9: <i>Modelo V.</i> .....	40
Figura 10: <i>Ciclo de vida del proyecto.</i> .....	45
Figura 11: <i>Caso de uso general del sistema ecoeficiente.</i> .....	50
Figura 12: <i>Caso de uso abrir lava mano.</i> .....	50
Figura 13: <i>Caso de uso cerrar lava manos.</i> .....	51



Figura 14: Caso de uso abrir inodoro.....	52
Figura 15: Caso de uso Cerrar inodoro. ....	53
Figura 16: Caso de uso abrir ducha.....	54
Figura 17: Caso de uso cerrar ducha.....	55
Figura 18: Caso de uso consumo de agua. ....	55
Figura 19: Diseño conceptual.....	57
Figura 20: Aplicativo móvil de control de válvulas. ....	69
Figura 21: Aplicación de control de flujo.....	73
Figura 22: Formula de factor de conversión. ....	74
Figura 23: Formula de volumen. ....	76
Figura 24: Datos enviados por el puerto serial.....	77
Figura 25: Placa virgen. ....	89
Figura 26: Circuito de Válvulas Solenoide. ....	89
Figura 27: Circuito control de sensores y válvulas. ....	90
Figura 28: Circuito control de sensores y válvulas. ....	91
Figura 29: Diseño de la placa quemada.....	91
Figura 30: Diseño de la placa quemada.....	91
Figura 31: Verificación y validación circuito. ....	93
Figura 32: Verificación y validación sensor de flujo.....	93
Figura 33: Verificación y validación circuito. ....	94
Figura 34: Verificación y validación aplicación consumo.....	94
Figura 35: Verificación y validación servo motor inodoro. ....	95
Figura 36: Verificación y validación aplicación inodoro.....	95
Figura 37: Verificación y validación válvula ducha. ....	96
Figura 38: Verificación y validación teclado numérico y LCD. ....	96
Figura 39: Verificación y validación aplicación inodoro.....	97
Figura 40: Verificación y validación lava mano. ....	97
Figura 41: Verificación y validación aplicativo ducha. ....	97
Figura 42: APK del aplicativo diseñado.....	99
Figura 43: Iconos en escritorio de móvil.....	99
Figura 44: Emparejamiento de dispositivos. ....	100
Figura 45: Contraseña para emparejar.....	100
Figura 46: Interfaz de presentación.....	101
Figura 47: Control de válvulas.....	101
Figura 48: Información del consumo de agua. ....	102
Figura 49: Grafico comparativos por días. ....	103
Figura 50: Grafico comparativos por días. ....	104
Figura 51: Grafico comparativos por días. ....	105
Figura 52: Consumo Enero sensor efecto Hall.....	106
Figura 53: Consumo Febrero sensor efecto Hall. ....	106
Figura 54: Consumo Marzo sensor efecto Hall. ....	107
Figura 55: Grafico comparativos por días. ....	108
Figura 56: Grafico comparativos por días. ....	109
Figura 57: Consumo Febrero sensor efecto Hall. ....	110
Figura 58: Consumo Marzo sensor efecto Hall. ....	110

Figura 59: Grafico comparativos por meses.....	111
Figura 60: Grafico comparativos por meses.....	112

## LISTA DE CODIGO

Código 1: Variables de tipo público y privado.....	58
Código 2: Dispositivos emparejados.....	59
Código 3: Conectar según dirección MAC.....	59
Código 4: Conexión a Bluetooth.....	60
Código 5: Clase principal.....	60
Código 6: Conexión de modelo Bluetooth.....	60
Código 7: Envío de datos a Bluetooth.....	65
Código 8: Confirmar la dirección MAC.....	66
Código 9: Conexión activa o desactiva de Bluetooth.....	68
Código 10: Clase principal.....	69
Código 11: Conexión de modelo Bluetooth.....	70
Código 12: Datos recibidos por Bluetooth.....	70
Código 13: Confirmar la dirección MAC.....	71
Código 14: Conexión activa o desactiva de Bluetooth.....	73
Código 15: Conexión puerto serial.....	75
Código 16: Número de pulsos.....	75
Código 17: Conexión puerto serial Arduino.....	76
Código 18: Calcular de volumen.....	77
Código 19: Variables de inicialización.....	78
Código 20: Calculo del caudal.....	79
Código 21: Impresión de datos.....	79
Código 22: Lectura del sensor.....	80
Código 23: Detección de estado.....	81
Código 24: Tiempo del Lcd.....	81
Código 25: Estado de válvula.....	82
Código 26: Mensaje LCD.....	82
Código 27: Mensaje display.....	82
Código 28: Limpiar Screen.....	84
Código 29: Función recursiva.....	85
Código 30: Tipo de control del sistema.....	85
Código 31: Control automático del sistema.....	87
Código 32: Control manual del sistema.....	88

## RESUMEN

El presente trabajo de investigación se realizó en el distrito de Talavera, provincia Andahuaylas, como plan de sustentar y experimentar la distribución de agua potable en los predios instalados.

Actualmente el distrito cuenta con distribución de agua potable sectorizada, esto conlleva que la población no tenga agua por varias horas del día dependiendo a la escases que se presente, una de las causas es por la inexistencia en el control de su distribución de manera uniforme por cada uno de los diferentes sectores, como también los predios no cuentan con un sistema de control que permita optimizar el consumo de agua como es en el caso de lava manos, ducha e inodoro, razón por la cual se generan desperdicios innecesarios y trae como consecuencia la falta de agua, baja presión, etc. Estos hechos se pudieron apreciar en diferentes predios del distrito es por ello que se realizó esta investigación de un sistema con tecnología Arduino y Android, para la eficiencia en el uso del agua potable con el fin de optimizar el uso de este recurso en cada uno de los predios del distrito, también poner en disposición de la población de Talavera y de otros distritos que atraviesan por una situación semejante a la que se inclina el estudio del proyecto.

Para la elaboración del sistema y el cumplimiento de los objetivos planteados del proyecto se siguió de manera correcta cada paso, combinando la metodología "V" con cada uno de sus fases de desarrollo; Para el desarrollo del sistema automatizado y aplicación móvil se utilizaron diversas tecnologías como: sensores de infrarrojo, sensor Pir, sensor de efecto hall, servo motor, teclado numérico y válvulas electrónicas en el uso de Software tenemos Android Studio, Arduino y Eagle.

El proyecto estuvo enmarcado en el tipo de investigación proyectiva, se emplearon una serie de técnicas e instrumentos de recolección de datos, específicamente el análisis de fuentes documentales, la observación directa y las entrevistas. De esta manera se pudo concluir con la implementación del sistema un mejor control del flujo de agua en cada uno de los puntos de distribución en los predios del distrito de Talavera, optimizando este recurso hídrico con el uso de sensores instalados en el (lava mano, ducha e inodoro) de esta manera la realización de cada proceso de forma automatizada.

## **ABSTRACT**

The present research work was carried out in the district of Talavera, province of Andahuaylas, as a plan to sustain and experiment with the distribution of drinking water in the facilities.

Currently the district has distribution of potable water sector, this means that the population does not have water for several hours of the day depending on the shortages that is present, one of the causes is the lack of control in its distribution evenly by each of the different sectors, as well as the properties do not have a control system that allows to optimize the consumption of water as it is in the case of hands, shower and toilet, which is why waste is generated unnecessary and brings as a consequence lack of water, low pressure, etc. These facts could be seen in different areas of the district is why this research was carried out on a system with Arduino and Android technology for the eco-efficiency in the use of drinking water in order to optimize the use of this resource in each of the estates of the district, also make available to the population of Talavera and other districts that go through a situation similar to that which inclines the study of the project.

For the elaboration of the system and the fulfillment of the objectives of the project, each step was followed correctly, combining the "V" methodology with each of its development phases; For the development of the automated system and mobile application we used various technologies such as: infrared sensors, Pir sensor, hall effect sensor, servo motor, numeric keypad and electronic valves in the use of Software we have Android Studio, Arduino and Enagle.

The project was framed in the type of projective research, a series of techniques and instruments of data collection were used, specifically the analysis of documentary sources, the direct observation and the interviews. In this way, it was possible to conclude with the implementation of the system a better control of water flow in each of the distribution points in the Talavera district, optimizing this water resource with the use of sensors installed in the (hand wash, shower and toilet) in this way the completion of each process in an automated way.

## **1. CAPITULO I**

---

### **1.1. TÍTULO DEL PROYECTO**

Implementación de un prototipo con tecnología Arduino y Android, para la ecoeficiencia en el uso del agua potable en los predios de Talavera.

### **1.2. AUTOR DEL PROYECTO**

Nombres y apellidos: David Centeno Cáceres

Escuela Profesional : Ingeniería de Sistemas

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

### **1.3. ASESOR DEL PROYECTO**

Nombres y apellidos: M.sc. Edwin Roque Tito

Departamento Académico: Ingeniería y Tecnología Informática

Categoría docente: Auxiliar

Modalidad: Dedicación Exclusiva

E-mail: edwinroque2020@gmail.com

### **1.4. LÍNEA DE INVESTIGACIÓN**

Redes, telemática y sistemas distribuidos.

### **1.5. ÁREA PRIORIZADA DEL PROYECTO**

04030202. Desarrollo de aplicaciones en sistemas de comunicación (hardware y software).

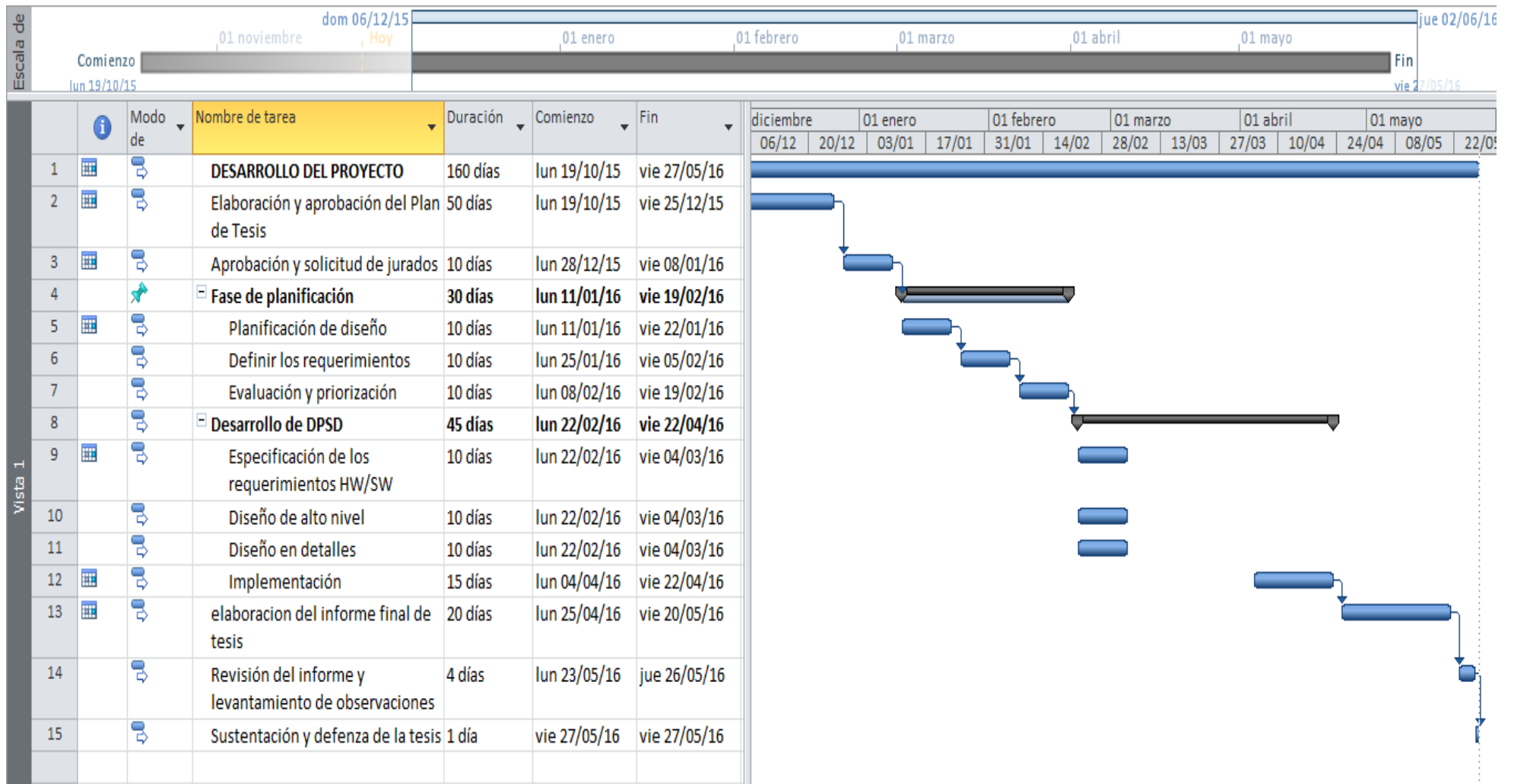
### **1.6. INSTITUCIÓN Y LUGAR DE EJECUCIÓN DEL PROYECTO**

Predios de Talavera

### **1.7. DURACIÓN DEL PROYECTO**

8 meses.

### **1.8. CRONOGRAMA DE EJECUCIÓN DEL PROYECTO**



**Figura 1: Cronograma de actividades.**  
**Fuente: Elaboración propia.**

## 1.9. PRESUPUESTO

Tabla 1: Tabla de presupuesto.

ITEM	DESCRIPCION	Cantidad	UNID. MEDIDA	Precio Unit	Precio parcial
<b>1.00.00</b>	Bienes				<b>3239.00</b>
<b>1.01.00</b>	Materiales de escritorio				<b>2,760.00</b>
1.01.01	Laptop Toshiba i5	1	unidad	2,000.00	2,000.00
1.01.02	USB HP 8 GB	1	unidad	60.00	60.00
1.01.03	Impresora Canon	1	unidad	400.00	400.00
1.01.05	Celular Android	1	unidad	300.00	300.00
<b>1.02.00</b>	Materiales consumibles				<b>479.00</b>
1.02.01	Lapicero	10	unidad	0.50	5.00
1.02.02	Tóner	2	unidad	220.00	440.00
1.02.03	Papel Bond A4	4	ciento	8.50	34.00
<b>2.00.00</b>	Servicios				<b>1,820.00</b>
<b>2.01.00</b>	servicios diversos				<b>1,820.00</b>
2.01.01	Internet	6	meses	50.00	300.00
2.01.02	Fotocopias	1000	unidad	0.10	100.00
2.01.03	Viáticos	6	meses	150.00	900.00
2.01.04	Comunicación	6	meses	30.00	180.00
2.01.05	Impresión	1000	unidad	0.10	100.00
2.01.06	Anillados de informe	10	unidad	6.00	60.00
2.01.07	Luz eléctrica	6	meses	30.00	180.00
<b>3.00.00</b>	Herramientas para el desarrollo				<b>3,744.00</b>
<b>3.01.00</b>	Software				<b>200.00</b>
3.01.01	Plataforma de desarrollo	1	unidad	-	-
3.01.02	Android Studio	1	unidad	200.00	200.00
<b>3.02.00</b>	Hardware				<b>3,544.00</b>
3.02.01	Modulo bluetooth	3	unidad	50.00	150.00
3.02.02	Arduino Mega	3	unidad	90.00	270.00
3.02.03	sensor de infrarrojo	3	unidad	190.00	570.00
3.02.04	Tubos de agua	6	unidad	15.00	90.00
3.02.04	Codos	8	unidad	4.00	32.00
3.02.04	Tubo T	8	unidad	4.00	32.00
3.02.04	Válvulas solenoide electrónicas	6	Unid	100.00	600.00
3.02.04	Material para construcción de prototipo de la casa	1	unidad	500.00	500.00
3.02.04	Sensor de nivel	4	unidad	100.00	400.00

<b>3.02.04</b>	Tanque de recepción de agua	1	unidad	450.00	450.00
<b>3.02.04</b>	Sensor de flujo de agua	3	unidad	150.00	450.00
<b>4.00.00</b>	Recursos Humanos				<b>10,000.00</b>
<b>4.01.00</b>	Personal de asesoría				<b>3,000.00</b>
<b>4.01.01</b>	Asesores	1	Persona	2,000.00	3,000.00
<b>4.02.00</b>	Desarrollo del sistema				<b>7,000.00</b>
<b>4.01.01</b>	Analista	1	Persona	2,000.00	2,000.00
<b>4.01.02</b>	Diseñador	1	Persona	2,500.00	2,500.00
<b>4.01.02</b>	Programador	1	Persona	2,500.00	2,500.00
<b>5.00.00</b>	GASTOS GENERALES				<b>3,000.00</b>
<b>5.00.01</b>	<b>Imprevistos</b>				
<b>5.00.02</b>	Imprevistos	1	Unidad	3,000.00	3,000.00
<b>COSTO TOTAL DE LA TESIS</b>					<b>21,803.00</b>

**Fuente:** *Elaboración propia.*

#### 1.10. FINANCIAMIENTO

Autofinanciamiento.....	S/.21,80
3.00	
Financiamiento de la institución.....	S/.0.00
Financiamiento de la universidad .....	S/.0.00
<b>TOTAL.....</b>	<b>S/.21,80</b>
3.00	



## **2. CAPITULO II: PLANTEAMIENTO DEL PROBLEMA**

---

### **2.1. REALIDAD PROBLEMÁTICA**

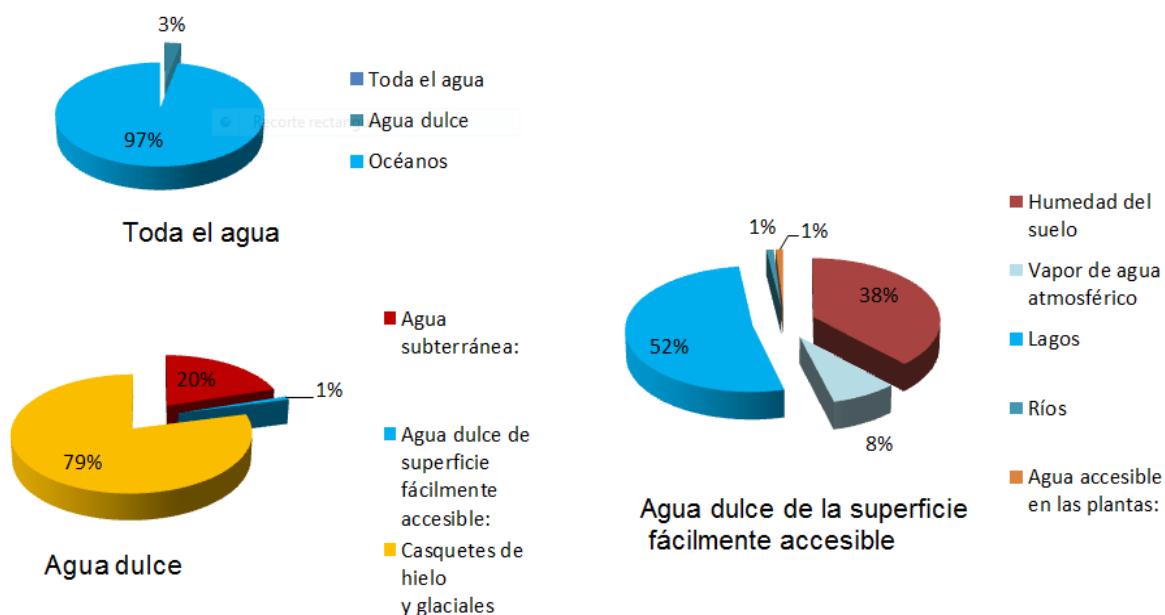
Andahuaylas está localizada en la zona noroeste del departamento de Apurímac, tiene una población aproximada de 34,087 habitantes, según el Instituto Nacional de Estadística e Informática (INEI). Es la ciudad más poblada del departamento de Apurímac, en el año 2015 albergaba una población de 63,654 habitantes. La ciudad abarca los distritos de Andahuaylas, San Jerónimo, Talavera y otros. El servicio de agua y alcantarillado está a cargo de Empresa Municipal de Saneamiento y Agua Potable "AGUAS TALAVERA" que abastece al 52% de la población urbana. Asimismo, 65 litros de aguas no tratadas son arrojados al río Chumbao, según investigación de (INEI), Apurímac es el departamento que tiene un 19.1 % de la población con acceso a internet.

Los países desarrollados son aquellos que han logrado un progreso mayor en desarrollo tecnológico ó área de la tecnología y por ende los de mayor influencia en la comercialización de la misma, que están relacionados con protocolos de control en dispositivos Domóticos, centrados en los países como: Estados Unidos, Israel, Corea, Japón, China, Francia, Italia, y Suecia; Estados Unidos es líder indiscutible en publicaciones (74.7%), Israel (6.7%), Corea del Sur (2.4%). Se conoce también la posición de los principales desarrolladores Asiáticos, donde Israel es el mejor posicionado, seguido muy de cerca por Corea, aunque se destaca un número bajo de depósitos proveniente de estos países. Los Estados Unidos se destacan en el número de depósitos debido a que muchas de las grandes empresas líderes en protocolos de control de dispositivos Domótico, tienen sede en ese país. Además, por ser el mayor mercado mundial según (Neida Boscán, 2009).

La domótica en los países desarrollados se remota a la década de los setenta, con los primeros dispositivos de automatización de edificios basado en tecnología X-10. Durante los años siguientes comenzaron diversos ensayos con avanzados electrodomésticos y dispositivos automáticos para el hogar. Los primeros sistemas comerciales fueron instalados en Estados Unidos. Más tarde tras el auge de los PC, a fines de la década de los 80 y principio de los 90 se empezaron a incorporar sistemas de cableado estructurado para facilitar la conexión de todo tipo de terminal. Posteriormente, los automatismos

destinados a edificios de oficina, junto con otras especificaciones, se han ido aplicando también a las viviendas dando origen a la vivienda domótica según (Jimeno, 2007).

La domótica avanza de una manera muy acelerada esto con el fin de proporcionarles mejor calidad de vida al ser humano y tener un mayor control de su domicilio por ejemplo control de la luz eléctrica y la distribución del agua potable, el uso de agua potable en el predio es uno de los problemas en nuestra sociedad contemporánea debido a que el planeta no tiene agua que hace 2.000 años; cuando estaba habitada por poco menos de 3% de la población actual, según (Agudelo & Marina, 2005). La demanda creciente de agua indispensable para la agricultura, la industria y el consumo doméstico ha creado una enorme competencia por el escaso recurso hídrico. El 70% de la superficie de la Tierra es agua; pero la mayor parte de esta es oceánica y solo 3% de ella es dulce, la cual se encuentra en su mayor parte en la forma de casquetes de hielo y glaciares; y solo 1% es agua dulce superficial fácilmente accesible. Esta es el agua que se encuentra en lagos, ríos y a poca profundidad en el suelo. Se considera que mundialmente se dispone de 9.000 m<sup>3</sup> por persona/año, pero para el año 2025, la disponibilidad global de agua dulce podría descender a un valor de 5.100 m<sup>3</sup> por persona/año. El agua dulce disponible no está equitativamente distribuida en el mundo, ni en la misma cantidad en las estaciones; varía de un año a otro, no se encuentra en todos los sitios donde se necesita ni en la misma cantidad, y contradictoriamente, en otros casos tenemos demasiada agua en el lugar equivocado y cuando no hace falta. Para el año 2030 en el mundo se necesitarán 4.400.000 m<sup>3</sup> de agua al año. Actualmente se consumen 3.200.000 m<sup>3</sup> de agua. La competencia por el agua entre la agricultura, la industria y el abastecimiento a las ciudades está limitando el desarrollo económico de muchos países en vías de desarrollo.



**Figura 2:** Disponibilidad y distribución del agua en el mundo.  
**Fuente:** (Agudelo & Marina, 2005)

En Perú en la ciudad de Lima el abastecimiento de agua En el 2011, el 76,2 por ciento de los hogares se abastecían de agua para consumo humano proveniente del sistema de conexión a red pública y pilón de uso público. Sin embargo, aún existe un 23,8 por ciento de hogares que consumen agua proveniente de camión cisterna, pozo, agua de río u otra modalidad. De acuerdo con el área de residencia y región natural, los hogares de Lima Metropolitana (93,2%), del área urbana (90,3%) y de la costa (89,4 %) son los que acceden en mayor medida al agua proveniente por red pública.

En cambio, los hogares del área rural (38,6 %) y de la selva (55 %) presentan los menores porcentajes de acceso al agua proveniente por red pública. A pesar de la riqueza hídrica peruana, el agua cada vez se está convirtiendo en un recurso escaso, debido a diferentes factores como la deforestación, el mal uso del agua y el calentamiento global. Se prevé que en el 2030 el Perú empezará a sentir seriamente los estragos de la falta de agua mientras tanto las ciudades siguen creciendo, la agricultura se ve potenciada con Proyectos Agroexportadores y la expansión industrial, prevista en los tratados de libre comercio indican una demanda de agua cada vez mayor.

Por otro lado, Sedapal informó que en Lima 720 mil personas carecen de agua potable en sus hogares y sin embargo pocos toman conciencia del despilfarro

que se hace de este recurso. Cada día un limeño gasta o consume 251 litros de agua, exactamente el doble que lo que un francés o un Suizo usa en el mismo periodo.



**Figura 3:** *desglaciación del nevado pastoruri.*  
**Fuente:** *Elaborado por SERVINDI (2016).*

Para (Gil Mateus, 2011) el agua potable es un bien económico especial que presenta características de bien público parcial pues expresa una rivalidad en el consumo. Incluso se puede catalogar como un recurso de uso común, pues existen fuentes, canales, inodoros, entre otras, sobre las cuales no siempre se puede impedir el acceso a los ciudadanos, por lo que presenta la característica de no exclusión. Un caso más común es que algunos consumidores que pagan el cargo fijo legal por la conexión al sistema, pueden percibir que el precio es más bajo de lo que estarían dispuestos a pagar por lo que tienen fuertes incentivos a consumir y malgastar el agua potable. En el distrito de Talavera el agua potable no tiene una adecuada distribución por factores geográficos a causa de ello no se tiene acceso permanente a este recurso hídrico, el acceso es solo por horas y suceden casos en que no se tiene agua durante todo el día por diversos problemas que tiene la empresa AGUAS TALAVERA. Como las redes de distribución, escases de agua en los reservorios y desperdicio innecesario de este recurso en los predios todo ello origina malestares en los usuarios afectados, esto genera a que los predios tengan tiempo limitado de este recurso hace que se tome diversas medidas de prevención los cuales son reciclar agua en baldes, tinas, etc. Para luego poder hacer uso de este recurso hídrico en actividades permanentes como es lavar la ropa, lavarse la mano, ducha o el inodoro, que son necesidades primordiales que no se puede obviar

en una casa por lo tanto se tiene que distribuir de manera uniforme el agua reciclada de manera que no agüe falta este recurso vital en una vivienda.

En los predios las familias desperdician el agua de manera innecesaria al momento de cepillarse los dientes una persona consume 250 ml de agua por minuto (sí, se hace de la manera recomendada cerrando el grifo), pero por el contrario se deja abierta la llave por cada minuto se desperdician 5 litros de agua, también se genera desperdicio al lavarse la mano, al darse un inodoro por 10 minutos gasta aproximadamente 100 litros de agua, el uso inconsciente de este recurso perjudica que el agua pueda llegar a todo los predios de manera equitativa. Las consecuencias que produce el desperdicio del agua potable genera conflictos sociales entre conciudadanos y actividades agropecuarias, según los investigadores en un futuro no muy lejano se fundirán las enormes masas de hielo de los polos, provocando una elevación de los niveles del mar, peligrando seriamente la posibilidad de vida en las costas, por el mal uso de este recurso importante en un futuro este tendrá un valor muy alto más que el oro y solo pocos tendrán acceso a este recurso debido a la escases. Cuando una persona se baña tiende a desperdiciar agua porque no existe un control adecuado en su suministro a la hora de darse un inodoro, esto genera perdida de este líquido vital y como también un incrementó en el pago del consumo. Las personas a la hora de ir a miccionar o defecar se tienen que lavar las manos debido a que a la hora de hacer dicha necesidades las manos están expenso a patógenos, que pueden provocar enfermedades como: infecciones gastrointestinales, fiebre, cólera o infecciones respiratorias, también puede ser responsables del contagio de infecciones cutáneas y otros; debido a ello es importante el lavado de las mano para realizar esta higiene constante de todo los días los caños comunes no son los adecuado para poder ahorrar agua .

Estos problemas son atravesados en cada uno de los predios de Talavera, esta problemática en general ocasiona el gran desperdicio de agua potable que debería ser utilizado de manera proporcional sin desperdiciar este líquido vital que es muy escasa en nuestro distrito, debe ser contribuir con el ahorro de ella. En tal sentido no hay un manejo adecuado en el monitoreo y optimización de este líquido en los domicilios que son los problemas que arrastra la empresa AGUAS TALAVERA.

## **2.2. FORMULACIÓN DEL PROBLEMA**

El ineficiente mecanismo de control e inexistencia de un medio para la información del flujo del agua potable trae como consecuencia el desperdicio en su consumo en los predios de Talavera.

## **2.3. OBJETIVOS**

### **2.3.1. OBJETIVO GENERAL**

Implementar un prototipo con tecnología Arduino y Android, para mejorar el uso ecoeficiencia del agua potable en los predios de Talavera 2016.

### **2.3.2. OBJETIVOS ESPECÍFICOS**

- Analizar una arquitectura con Arduino que permite conectar sensores, válvulas electrónicas y dispositivos móviles mediante sus diferentes puertos analógicos y digitales.
- Desarrollar una aplicación Movil que permita monitorear los puntos de acceso.
- Demostrar la reducción de consumo de agua potable con el implementar del prototipo con tecnología Arduino y Android, en los predios de Talavera.

## **2.4. JUSTIFICACIÓN**

La escases del agua dulce en nuestra planeta es un tema de inquietud para todo los países del mundo, estos factores que causa la escases del agua dulce es el calentamiento global y el mal uso de ella, que genera una gran preocupación de los consumidores razón por la cual el proyecto de ecoeficiencia conlleva a mejorar el uso optimizado del agua potable, haciendo uso de las tecnologías informáticas y las comunicaciones, esto repercutiría en mejora de un estilo de vida, conocido como sistemas Domoticos para una solución alternativa a la exigencia de nuestros predios.

Viendo la necesidad de implementar las plataformas ya implementadas que se pueden adquirir en el mercado, para transmitir datos que estos permiten el funcionamiento de un sistema de comunicaciones, con estas tecnologías de información, se pretende lograr los objetivos planificados para automatizar un predio, para el uso del agua potable de manera ecoeficiente. Integrando el uso de dispositivos móviles para poder monitorearlos y de tal forma genere mayor beneficio para un servicio social;

con esta implementación del sistema ecoeficiente se tendrá un mejor control del uso del agua potable y de esta forma poder optimizar el consumo de agua en un dominio y evitar desperdicio innecesarios, el dispositivo móvil nos ayudara a tener datos del flujo de agua que se distribuyen en el predio, también controlar desde el dispositivo móvil de los diferentes puntos de acceso como: el inodoro, ducha y lavamanos.

## **2.5. VIABILIDAD DE LA INVESTIGACIÓN**

### **2.5.1. Viabilidad Técnica**

Las herramientas necesarias para implementar el diseño del prototipo son los siguientes:

#### **Hardware:**

- ✓ 1 laptop Toshiba i5, Sistema operativo de 64 bits Windows 7, procesador 2.50 GHZ.
- ✓ 1 celular Huawei resolución 720 x 1280, sistema operativo Android 5.0, modelo ALE-L23.
- ✓ 3 valvulas solenoide, voltaje 12 V, color blanco, diámetro del conector exterior 20mm diámetro interior 14mm.
- ✓ 3 sensores de flujo de agua, voltaje de 5V, temperatura de funcionamiento -25 a 80 °c, pulsos por litro 450.
- ✓ 2 Arduinos mega Atmega 2560, voltaje recomendado 7-12 V.
- ✓ 1 Bluetooth hc-05, chip BC417143, nivel TTL.

#### **Software:**

- ✓ Android Studio 2.2.3
- ✓ Eagle 7.6.0.
- ✓ Lenguaje Arduino, versión 1.6.9.

### **2.5.2. Viabilidad Operativa**

El desarrollo de este proyecto tiene la finalidad de contribuir en el uso ecoeficiente del agua potable en los predios en sus puntos de acceso el inodoro, ducha, lava manos. Además, tomar esta investigación para otras investigaciones de esta misma índole.

### **2.5.3. Viabilidad Económica**

Es viable económicamente porque se cuenta con los recursos económicos, para el diseño del prototipo de sistema ecoeficiente, 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 solventar los diversos servicios que son necesarios, Recursos Humanos.

## **2.6. LIMITACIÓN DEL ESTUDIO**

El estudio fue realizado con pruebas fehacientes en dos viviendas por el tema de no contar con la accesibilidad de otros hogares y costo económico que esta genera en su implementación, estos fueron factores para evitar la implementación del sistema ecoeficiente en más de dos viviendas. Como otro de los factores limitantes que se presentó en el proyecto es realizar las pruebas en diferentes sectores del distrito de Talavera, el cual se optó solo realizarlo en el barrio de Masuraccra, de las viviendas donde se realizaron las pruebas se tomó solo los puntos estratégicos como: el inodoro, ducha y la lava mano por el tema del coste económico, son los lugares que más consumo de agua realiza un predio.

Otra de las limitaciones es el almacenamiento de datos del consumo de cada mes que registraba esto no fue posible almacenarlos por tener problemas en la adquisición de hardware, en tal sentido se tomó la alternativa de almacenar la data en Excel manualmente, según sea el caso del consumo por cada mes; otro de los factores limitantes que se nos presento fue el manejo del sistema ecoeficiente desde puntos más distantes porque en el proyecto se plante una solución con el modulo Bluetooth y no con wifi.



### 3. CAPITULO III: MARCO TEÓRICO

---

#### 3.1. Marco conceptual

#### 3.2. Sistema

(Bertoglio & Johansen, 1982) define que un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben (entrada) datos, energía o materia del ambiente y proveen (salida) información, energía o materia. Un sistema puede ser físico o concreto (una computadora, un televisor, un humano) o puede ser abstracto o conceptual (un software). Cada sistema existe dentro de otro más grande, por lo tanto, un sistema puede estar formado por subsistemas y partes, y a la vez puede ser parte de un supersistema.

Para (Ulloa, 2010) El pensamiento de sistemas es el “estudio de las relaciones entre las partes de un ente integrado (abstracto o concreto) y de su comportamiento como un todo respecto a su entorno”.

##### 3.2.1. Clasificación de sistemas

De acuerdo con la clasificación de sistemas que hace (Ulloa, 2010) de los sistemas, estos pueden ser:

1. **Sistemas naturales.** Aquellos sistemas que han sido elaborados por la naturaleza, desde el nivel de estructuras atómicas hasta sistemas vivos, los sistemas solares y el universo.
2. **Sistemas diseñados.** Aquellos que han sido diseñados por el hombre y son parte del mundo real. Pueden ser de dos tipos: abstractos y concretos. Ejemplos de sistemas diseñados abstractos: la filosofía, las matemáticas, las ideologías, la religión, el lenguaje. De sistemas diseñados concretos: un computador, una casa, un auto, etc.
3. **Sistemas de actividad humana.** Son sistemas que describen al ser humano epistemológicamente, a través de lo que hace. Se basan en la apreciación de lo que en el mundo real una persona o un grupo de personas podrían estar haciendo, es decir, en la intencionalidad que tiene el sistema humano que se observe.
4. **Sistemas culturales.** Sistemas formados por la agrupación de personas (por ejemplo, la empresa, la familia, el grupo de estudiantes de una universidad, etc.).

### 3.3. Domótica

#### 3.3.1. Definición

Definido según (Herrera Quintero, 2005) Domótica es el conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. La domótica es una tecnología que tiene como objetivo fundamental proporcionar un estilo de vida más cómodo en el hogar. Puede definirse como la integración y aplicación de tecnologías como:

- Dispositivos electrónicos: Sensores, actuadores, etc.
- Tecnologías de Telecomunicaciones: Ethernet, Wi-Fi, Bluetooth, Zigbee, Fibra, etc.
- Sistemas de procesamiento central: Computadoras, Sistemas embebidos, Microcontroladores, etc.

**Sus principales características son:**

- Interacción.
- Facilidad de uso.
- Monitoreo a distancia.
- Fiabilidad.

#### 3.3.2. Definición de (domo-tic-a)

Definición según (Domínguez, 2012):

El análisis de acrónimos de la palabra **domo-tic-a**. Por "Domo" entenderemos casa, o vivienda, de acuerdo a su origen en latín. Tomaremos "TIC" por Tecnologías de la Información y las Comunicaciones, mientras que la "A" final denotará automatización. Nótese que esta descomposición no atiende al significado etimológico de la palabra Domótica, ya analizado previamente, sino a un ingenioso y sorprendente juego de palabras.

### 3.4. Tecnología

(Wiebe, 2005) Tecnología se refiere a un conjunto de objetos físicos o artefactos, tales como computadoras, autos, o máquinas para votar. En el siguiente nivel, también se incluyen actividades humanas, tales como en "la tecnología de voto electrónico", donde también se hace referencia al diseño, la

fabricación y el manejo de este tipo de máquinas. Finalmente, y más cercano a su origen griego, “tecnología” refiere a conocimiento: se trata tanto de aquello que la gente conoce como de lo que hace con las máquinas y los procesos de producción relacionados. Usar “tecnología” en estos tres sentidos permite ser más específico que cuando se lo emplea como un concepto contenedor en un nivel macro, como por ejemplo “la modernización política incluye los cambios actuales en las políticas y los gobiernos, en los estados y países individuales, derivados de cambios fundamentales en la tecnología”.

#### **3.4.1. Uso de la tecnología en las viviendas**

En los últimos años según (Quintero, 2005) el avance de la telecomunicaciones a través de internet permite hablar de integración a nivel de redes. Numerosas redes funcionan con éxito y han sido fundamentalmente para las diversas áreas en la medida en que la automatización de los datos permite a investigadores y profesionales tener una visión más amplia de la producción en los variados sectores. Desde hace mucho tiempo el control a distancia viene desarrollándose gracias a la innovación tecnológica con que se cuenta hoy en día, y con ello se van haciendo tangibles cada vez más entornos de integración humana basados en sistemas de telecomunicaciones y control.

- **Arquitectura centralizada:** En este tipo de arquitectura se tiene una topología de interconexión tipo estrella. Así, el sistema posee un elemento de control central que es el encargado de manejar todas las señales de control de los diversos dispositivos, y a su vez todos los dispositivos están conectados hacia él, por lo tanto, si este elemento central falla o simplemente deja de funcionar todo el sistema de control colapsa.
- **Arquitectura distribuida:** Para esta arquitectura, el sistema de control se sitúa próximo al elemento a control, dado al sistema gran flexibilidad, porque si uno de los dispositivos no puede ser controlado no significa que el resto tampoco. Los factores más influyentes para la utilización de este tipo de arquitectura son los medios de transmisión, la velocidad en las comunicaciones, el tipo de protocolo; por lo tanto, estas son algunas características a tener en cuenta si se quiere implantar una arquitectura de esta índole.

### **3.4.2. Viviendas inteligentes**

(Herrera Quintero, 2005) define que en la domótica las aplicaciones son muy variadas basadas en el concepto de vivienda inteligente, en ellas se pueden hallar dispositivos de autonomía general que realizan sus procedimientos sin la necesidad de intervención humana, como refrigeradoras inteligentes que avisan al usuario la ausencia de alimentos y/o toman decisión de solicitar a domicilio vía internet, acaso por el requerimiento del el horno microondas, que eligió preparar una receta especial para el almuerzo y otros.

Algunas de las ventajas de vivir en un hogar domótica y conectado a la red Internet son:

- Programación del encendido y apagado de sistemas de luz, gas, agua, etc. Ya sea por el usuario o por un evento en específico.
- Contadores de consumo energético y de agua potable con reportes, alarmas, recordatorios, etc.
- Control de dispositivos electrónicos.
- Monitoreo de cámaras de seguridad y micrófonos.

### **3.5. Internet de las cosas**

Para (Vega, Marcela, Santamaría, & Rivas, 2014) El Internet de los objetos, se perfila como una de las principales tendencias que dan forma al desarrollo de las tecnologías de las comunicaciones. Es el paso del Internet utilizado para la interconexión de objetos físicos que se comunican entre sí, con los seres humanos, para ofrecer un determinado servicio. Se basa en tres pilares principales: ser identificables, para comunicarse y para interactuar.

El correcto empleo del Internet de las cosas es por tres fenómenos recientes:

- La miniaturización de dispositivos electrónicos y de comunicación, lo que facilita que se pueda conectar prácticamente cualquier cosa en cualquier momento.
- La superación de la limitación de la infraestructura de telefonía móvil e Internet.
- La proliferación de las aplicaciones y los servicios que ponen en uso la gran cantidad de información creada a partir del Internet de las cosas (plataformas como Xively, Open Sen.se, EMMONS, etc.).

### **3.5.1. El Internet de las cosas como una red de redes**

En la actualidad según (Evans, 2011), el IoT se compone de un conjunto disperso de redes dispares diseñadas a medida. Los coches de hoy en día, por ejemplo, cuentan con diversas redes para controlar el funcionamiento del motor, las funciones de seguridad, los sistemas de comunicaciones, etc. Los edificios comerciales y residenciales también tienen varios sistemas de control para la calefacción, la ventilación y el aire acondicionado (HVAC); el servicio telefónico; la seguridad, y la iluminación. A medida que evoluciona el IoT, estas redes y muchas otras, se conectarán y contarán con mayores funciones de seguridad, análisis y gestión (véase la imagen). Esto permitirá que el IoT pueda contribuir más y mejor a que las personas consigan sus objetivos.

## **3.6. Arduino**

Arduino según (M. Alberto, 2017) es una plataforma de prototipos de código abierto basado en hardware fácil de usar y el software. Placas Arduino son capaces de leer los insumos, mensaje de Twitter, activación de un motor, encender un LED, publicar algo en línea. Usted puede decirle a su tablero qué hacer mediante el envío de un conjunto de instrucciones para el microcontrolador en el tablero. Para ello se utiliza el lenguaje de programación de Arduino (basado en Wiring), y la plataforma de desarrollo de Arduino (IDE), basado en Processing según página oficial de Arduino.

### **3.6.1. Tipos de Arduinos**

Existen varios tipos de Arduinos según su utilidad y uso que se les dé los cuales describiremos:

- Arduino UNO
- Arduino Pro
- Arduino Genuino Micro
- Arduino Pro Mini
- Arduino Nano
- Motor Escudo Arduino
- Arduino mega
- Arduino shield

### 3.6.2. Arduino mega

El Arduino Mega 2560, es una placa electrónica basada en el ATmega2560 (hoja de datos). Tiene 54 pines digitales de entrada/salida, 16 entradas analógicas, cuatro UARTs (puertos seriales), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reinicio. La Mega es compatible con la mayoría de los protectores diseñados para el Arduino Diecimila. El Arduino / Genuino Mega 2560 puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

**Tabla 2:** *Detalle de Arduino Mega.*

Microcontrolador	ATMega2560
Voltaje	5V
Salida de voltaje (recomendada)	7-12V
Salida de voltaje (limite)	6-20V
Digital I/O Pins	54 (which 15 provide PWM output)
Pines de salida analógica	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Memoria Flash	256 KB which 8 KB used Bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

Fuente: (M. Alberto, 2017).

### 3.6.3. Modulo bluetooth HC-06

El módulo de bluetooth HC-06 es muy popular para aplicaciones con microcontroladores PIC y Arduino. Se trata de un dispositivo relativamente económicos y que habitualmente se venden en un formato que permite insertarlos en un protoboard y cablearlo directamente a cualquier microcontrolador, incluso sin realizar soldaduras.

**Tabla 3:** Características de modulo bluetooth hc-06.

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

Fuente: (M. Alberto, 2017).

### 3.7. teléfonos inteligentes

Definición (Laura Grimaldos, 2012) un teléfono inteligente es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de computación y conectividad que un teléfono móvil convencional. El término «inteligente» hace referencia a la capacidad de usarse como un computador de bolsillo, llegando incluso a remplazar a un computador personal en algunos casos. La mayoría de los teléfonos inteligentes permiten la instalación de diversas aplicaciones, también poseen la característica de ser táctiles, por ello no necesitan de un teclado físico, aunque algunos modelos si lo poseen.

#### 3.7.1. Características

##### a) Sistema operativo

En el sitio de web de tecnología Gartner, Van der Meulen público en el 2013 el resultado de la estadística de uso de los sistemas operativos móviles, en la cual el líder es el sistema operativo de Google (Android), luego le sigue el de la empresa Apple (iOS), después el de Microsoft (Windows Phone), quedando relegados los de a compañía BlackBerry (BlackBerry OS), así como Nokia (Symbian) y los demás.

**Tabla 4:** Sistemas operativos más usados.

SO móvil	Porcentaje (%)
Android	81,9

<b>iOS</b>	12,1
<b>Windows Phone</b>	3,6
<b>BlackBerry OS</b>	1,8
<b>Bada</b>	0,3
<b>Symbian OS</b>	0,2
<b>Otros</b>	0,2

Fuente: *Newsroom (2015)*.

## b) Hardware

Los dispositivos móviles tienen unas características especiales que les permiten entrar en la categoría de equipos de computación móvil, estas características de los equipos son definidas por los fabricantes, cada fabricante desarrolla nuevas tecnologías que implementa en los diferentes modelos que producen.

En general un dispositivo móvil tiene como función procesar información, teniendo como característica principal la movilidad del usuario, además puede ayudar a realizar llamadas telefónicas, servir de asistente personal, funcionar como tableta, reloj, televisor entre los muchos dispositivos móviles que se encuentran en la actualidad en el mercado mundial.

Al destapar un dispositivo móvil se encuentran generalmente las siguientes partes básicas que permiten el funcionamiento del dispositivo:

- **Placa Base:** Es un circuito integrado que contiene el cerebro y todos los componentes electrónicos del teléfono celular.
- **Antena:** La antena permite la recepción y envío de las señales del dispositivo móvil.
- **Antena WiFi:** La antena permite la recepción y envío de las señales del estándar 802.11 a, b, g y n.
- **Antena NFC:** La antena permite el envío y recepción de las señales del dispositivo móvil a otros dispositivos en distancias cortas.
- **Pantalla:** Las pantallas o display generalmente de cristal líquido LCD, son las encargadas de servir de interfaz entre el usuario y el dispositivo móvil, actualmente las pantallas son táctiles y permiten la interacción del usuario con el dispositivo móvil.



- **Teclado:** El teclado es la característica del teléfono móvil que le permite al usuario ingresar información como datos o texto al teléfono, el teclado más utilizado es el del formato QWERTY.
- **Micrófono:** El micrófono permite es el encargado de traducir la voz del usuario en energía eléctrica para ser comprimida y enviada por el teléfono móvil a su destino.
- **Bocina o Altavoz:** El altavoz es el encargado de reproducir los sonidos del teléfono para que el usuario pueda escuchar las llamadas u otro tipo de sonidos.
- **Batería:** La batería es la encargada de almacenar y mantener la energía necesaria para el funcionamiento del teléfono móvil.
- **Puerto de Carga de Energía:** Este puerto permite realizar la carga de energía de la batería del dispositivo, en la actualidad el más utilizado es el puerto USB mini.

### 3.8. Android

Definición según (UCM, 2013).

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.

#### 3.8.1. Características de Android

- **Framework de aplicaciones:** Habilitando para la reutilización y el reemplazo de componentes.
- **La máquina virtual Dalvik:** Optimizada para dispositivos móviles.
- **Navegador integrado:** Basado en el motor del proyecto abierto WebKit.

- **Gráficos optimizados:** Suministrados por una librería de gráficos 2D. Los gráficos 3D están basados en la especificación OpenGL ES 1.0, con soporte para aceleración gráfica por hardware (opcional).
- **SQLite:** Para estructurar el almacenamiento de datos.
- **Soporte multimedia:** Común para audio, video, imágenes, soportando varios formatos (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- **Telefonía GSM** (Si el hardware lo soporta).
- **Bluetooth, EDGE, 3G, y WiFi** (Si el hardware lo soporta).
- **Camera, GPS, compass y accelerometer** (Si el hardware lo soporta).
- **Completo entorno de desarrollo:** Incluye un dispositivo emulador, herramientas de depuración, y un plugin para el IDE Eclipse.

### 3.8.2. Versiones de Android

- **Cupcake: Android Versión 1.5**  
Widgets, teclado QWERTY virtual, copy & paste, captura de vídeos y poder subirlos a Youtube directamente.
- **Donut: Android Versión 1.6**  
Añade a la anterior la mejoría de la interfaz de la cámara, búsqueda por voz, y navegación en Google Maps.
- **Eclair: Android Versión 2.0/2.1**  
Mejoras en Google Maps, salvapantallas animado, incluye zoom digital para la cámara, y un nuevo navegador de internet.
- **Froyo: Android Versión 2.2**  
Incluye hotspot Wifi, mejora de la memoria, más veloz, Microsoft Exchange y video-llamada.
- **Ginger Bread: Android Versión 2.3**  
Mejoras del consumo de batería, el soporte de vídeo online y el teclado virtual, e incluye soporte para pagos mediante NFC.
- **Honey Comb: Android Versión 3.0/3.4**  
Mejoras para tablets, soporte Flash y Divx, integra Dolphin, multitarea pudiendo cambiar de aplicación dejando las demás en espera en una columna, widgets y homepage personalizable.
- **Ice Cream Sandwich: Android Versión 4.0**

Multiplataforma (tablets, teléfonos móviles y netbooks), barras de estado, pantalla principal con soporte para 3D, widgets redimensionables, soporte usb para teclados, reconocimiento facial y controles para PS3.

### 3.8.3. Uso de smartphones en Perú

El uso y consumo de información a través de dispositivos móviles va en aumento a nivel mundial y en el Perú se da de manera acelerada. Según el último estudio de IMS y Comscore, el 88% de peruanos participantes en la investigación accede a internet a través de un smartphone o una Tablet según (IMS, 2015).

	Total	Brasil		México		Argentina		Colombia		Peru		Chile	
	%	%	Index	%	Index	%	Index	%	Index	%	Index	%	Index
Android	78	82	105	74	95	75	96	79	101	79	101	70	90
iPhone	19	16	84	26	137	11	58	22	116	26	137	36	189
Blackberry	7	3	43	6	86	13	186	12	171	19	271	9	129
Otro	8	8	100	7	88	15	188	8	100	7	88	6	75
No Sabe	1	1	100	1	100	1	100	1	100	1	100	<1	100
Wifi	80	81	101	82	103	84	105	74	93	74	93	74	93
3G	65	65	100	57	88	84	129	56	86	60	92	71	109
4G	23	19	83	31	135	9	39	41	178	36	157	27	117
Otro	1	1	100	1	100	2	200	1	100	1	100	1	100
No Sabe	1	<1	100	1	100	1	100	1	100	1	100	<1	100
Tiene aplicaciones en el teléfono	98	98	100	99	101	96	98	99	101	99	101	99	101
Número promedio de aplicaciones en el dispositivo	18	16	89	20	111	18	100	19	106	18	100	19	106

Figura 4: Posesión y utilización de smartphones por país.  
Fuente: (IMS, 2015).

### 3.9. Sensor

Un sensor o captador, como prefiera llamársele, no es más que un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que seamos capaces de cuantificar y manipular.

Normalmente estos dispositivos se encuentran realizados mediante la utilización de componentes pasivos (resistencias variables, PTC, NTC, LDR,

etc... todos aquellos componentes que varían su magnitud en función de alguna variable), y la utilización de componentes activos.

### 3.9.1. sensor YF-S201

Según (Edison Fabricio, 2014) tiene un cuerpo plástico un motor de agua y un sensor efecto hall cuando el agua fluye a través de rotor el rotor rueda, su velocidad cambia con diferente tasa de flujo el sensor defecto hall da salida al impulso correspondiente a la señal, tiene una precisión aproximada del 30% de uno a 60 litros por minuto.

### 3.9.2. Sensor Efecto hall

El efecto Hall es la medición del voltaje transversal en un conductor cuando es puesto en un campo magnético. Mediante esta medición es posible determinar el tipo, concentración y movilidad de portadores. Es decir, el sensor de efecto Hall crea un voltaje saliente proporcional al producto de la fuerza del campo magnético y de la corriente.

Este tipo de medidores tienen un sensor que funciona a modo de switch ON/OFF. Cuando un imán pasa cerca del sensor se produce un voltaje en sus terminales, si el imán se retira el voltaje cae a 0V. Este imán está integrado en el rotor. Se muestra un sensor de la empresa SEED STUDIO que dispone de un cuerpo plástico, un rotor de agua, y un sensor de Efecto Hall. Cuando el agua fluye a través del rotor, el rotor rueda. Su velocidad cambia con diferente tasa de flujo.



**Figura 5:** *Sensor de efecto Hall.*  
**Fuente:** (Edison Fabricio, 2014).

**Tabla 5:** Especificación del sensor hall.

Especificaciones del sensor
Min. Voltaje de trabajo: DC 4.5V
Max. Trabajo actual: 15 m.A (DC 5V)
Voltaje de funcionamiento: DC 5V ~ 24V
Flujo Rango de Tarifas: 1 ~ 30litros/min
Capacidad de carga: ≤ 10 mA (DC 5V)
Temperatura de funcionamiento: ≤ 80 °C
Temperatura del líquido: ≤ 120°C
Humedad de funcionamiento: 35% ~ 90% RH
Presión del agua: ≤ 1.75MPa
Temperatura de almacenamiento: -25 ~ + 80°C
Humedad de almacenamiento: 25% ~ 95% RH

**Fuente:** (Edison Fabricio, 2014).

### 3.9.3. Sensor de infrarrojo

Según (C. Alberto, Angela, & Aguirre, 2007) este sensor infrarrojo es un dispositivo capaz de medir la radiación electromagnética infrarroja de los cuerpos en su campo de visión. Todos los cuerpos emiten una cierta cantidad de radiación, esta resulta invisible para nuestros ojos, pero no para estos aparatos electrónicos, ya que se encuentran en el rango del espectro justo por debajo de la luz visible el sensor consta de un led infrarrojo y fototransistor que se usa para captar objetos de hasta 10cm posee una salida digital.



**Figura 6:** Sensor infrarrojo.

**Fuente:** (C. Alberto et al., 2007).

### 3.10. Bluetooth

Según (Sparacino Lorefice, 2003). Es básicamente un estándar para comunicaciones inalámbricas. Su Tecnología elimina la necesidad de utilizar los numerosos e incómodos cables que habitualmente conectamos a nuestros PCs, dispositivo móvil, laptops y a todo tipo de equipos de mano.

Un pequeño microchip Bluetooth, que incorpora un radio transmisor, es introducido en los dispositivos digitales, entonces, la tecnología Bluetooth se encarga de realizar todas las conexiones de forma inmediata, sin utilizar ni un solo centímetro de cable; es decir, describe cómo pueden interconectarse todo tipo de dispositivos inalámbricos, ya sea en el hogar o en la oficina, utilizando una conexión de corto alcance.



**Figura 7:** Modelo de conectividad de Bluetooth.  
**Fuente:** (Sparacino Lorefice, 2003).

#### 3.10.1. Descripción de la tecnología Bluetooth

La especificación Bluetooth viene definida por rangos de frecuencia de radio que pueden ser bajos (del orden de 10 metros) y ocasionalmente medios (sobre los 100 metros), con capacidad para la transmisión tanto de datos como de voz a más de 720 Kbps por canal.

El esfuerzo está orientado a la creación y al diseño de un chip que pueda implementarse en los circuitos de la CMO, pero teniendo en cuenta los ratios de consumo de la máquina. Una tecnología que ha de reducir por

tanto el costo, el consumo, y que también ha de velar porque el tamaño del chip sea el adecuado para su instalación en cualquier dispositivo móvil e inclusive en dispositivos estáticos.

- **En cuanto a la voz:** los requerimientos iniciales definen la necesidad de disponer de hasta tres canales sincrónicos de voz al tiempo, o de un canal que soporte al tiempo a sincronía de datos y sincronía de voz. Tal y como se define en el estándar Bluetooth, cada canal de voz soportaría 64 Kbps sincrónicos (para voz) por canal en cada dirección.
- **En cuanto a los datos:** El canal asíncrono para los datos podría soportar un máximo de 723.2 Kbps asimétricos (que podrían ser de 57.6 Kbps en sentido contrario), o de 433.9 Kbps simétricos.

### 3.10.2. Características del bluetooth

Características definidas según (Sparacino Lorefice, 2003):

- Permite rápidas y seguras transmisiones tanto de voz como de datos, incluso cuando los dispositivos no se encuentran en su radio de acción.
- Al estar orientado al uso personal, las distancias de comunicación son del orden de la decena de metros, y sólo con adecuados amplificadores y antenas se pueden alcanzar distancias de aproximadamente 100 m.
- Bluetooth opera en una banda de frecuencias que va de 2400 MHz a los 2483 MHz. Esta banda se encuentra liberalizada en gran parte del mundo, a excepción de Francia, España y Japón. Sólo existe un, pero, y es que, al existir variaciones en el ancho de banda en diferentes países, los aparatos vendidos en éstos no están muy claros que sean compatibles.
- Bluetooth es una tecnología que nos permite conectar nuestros dispositivos digitales sin utilizar cable.
- El microchip Bluetooth, incorpora un radio transmisor, es introducido en los dispositivos digitales. Bluetooth se encarga de realizar todas las conexiones de forma inmediata, sin utilizar ni un solo centímetro de cable.
- Para la transmisión tanto de voz como de datos, Bluetooth emplea una combinación de conmutación de circuitos y de paquetes. Cada canal de voz soporta un enlace síncrono a 64 Kbps.

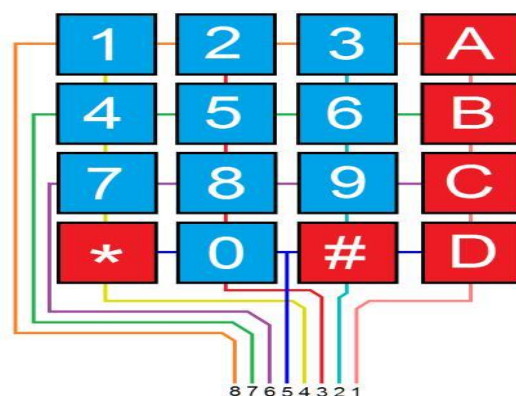
- Para la transmisión de datos se emplean canales síncronos, cada uno puede soportar un enlace asimétrico a 721 Kbps en un sentido y 57.6 Kbps en el otro, o bien un enlace simétrico a 432.6 Kbps en ambos sentidos.
- El alcance del sistema es de 10 metros, ampliable a 100 metros aumentando la potencia transmitida con un amplificador adecuado.
- A diferencia de otros sistemas de comunicaciones inalámbricos como los basados en infrarrojos, Bluetooth no requiere que haya línea de visión directa entre los dispositivos.
- Las conexiones son instantáneas y se mantienen incluso cuando los dispositivos no están dentro de su radio de acción. Esta es quizás, una de las características más relevantes que tiene bluetooth.
- Para comunicarse con otros dispositivos Bluetooth, se requiere un hardware específico para Bluetooth, que incluye un módulo de banda base, así como otro módulo de radio y una antena. Además, deberá haber un software encargado de controlar la conexión entre dos dispositivos Bluetooth; este software por lo general correrá en un microprocesador dedicado. Los Link Managers de diferentes dispositivos Bluetooth se comunicarán mediante el protocolo LMP.

### **3.11. Teclado numérico**

Un teclado es una colección de botones, a cada uno de los cuales le asignamos un símbolo o una función determinada. La lectura de los botones es algo sencillo, pero si conectáramos cada tecla a un pin digital de nuestro Arduino, pronto estaríamos en apuros.

Para que nuestro Arduino pueda saber que tecla se pulsa, basta con poner tensión en las filas de forma secuencial y luego leer las columnas para ver cuál de ellas tiene HIGH=ALTO. Los teclados matriciales usan una combinación de filas y columnas para conocer el estado de los botones. Cada tecla es un pulsador conectado a una fila y a una columna. Cuando se pulsa una de las teclas, se cierra una conexión única entre una fila y una columna.





**Figura 8:** *Interfaz del teclado numérico.*  
**Fuente:** NAYLAMP (2015).

### 3.12. Ecoeficiencia

Definido según (Orlando Advíncula Zeballos, Samantha García Junco, Juvenal García Armas Katerin Toribio Tomayo, 2014). La ecoeficiencia es el uso eficiente y racional de la energía y los recursos naturales con beneficios ecológicos y económicos, con menos desechos y residuos, logrando disminuir la contaminación ambiental. Para fines de este estudio es considerada como una estrategia administrativa que permite mejorar el desempeño ambiental y al mismo tiempo generar ahorros económicos significativos.

En 1991, cuando el WBCSD (Consejo Empresarial Mundial para el Desarrollo Sostenible) usó por primera vez el término ecoeficiencia, era difícil prever qué tan importante se volvería, administrativamente, se había generado la estrategia para unir los logros ambientales y el éxito empresarial (Fussler & James, 1999). Los gobiernos pueden implementar una política que fomente el crecimiento económico, que favorezca la reducción del uso de recursos y que evite la contaminación, con incentivos para la eco-innovación.

Según el Decreto Supremo N° 009-2009-MINAM, Medidas de ecoeficiencia para el Sector Público, en sus Artículos 3° y 4° indican que se debe realizar un control de fugas, disponer de avisos sobre el buen uso de los servicios, observar las averías sanitarias, criterios para el riego de los jardines y la implementación progresiva de dispositivos ahorradores de agua en los servicios higiénicos (Diario oficial El Peruano, 2009). Una experiencia en ecoeficiencia del consumo de agua potable desarrollada por el MINAM en el 2009, en catorce entidades del sector público determinó un consumo promedio de 29.97 m<sup>3</sup>/año/trabajador, donde el consumo mínimo promedio fue registrado

en el Ministerio de Economía y Finanzas con 12.8 m<sup>3</sup>/año/trabajador y el máximo fue para el Ministerio de Transportes y Telecomunicaciones con 69.80 m<sup>3</sup>/año/trabajador (Legales, 2009).

### 3.13. Metodología de desarrollo

Según (Perez et al., 2006) define.

El modelo en V se desarrolló para terminar con algunos de los problemas que se vieron utilizando el enfoque de cascada tradicional.

Este tipo de modelo se enfoca al desarrollo de software en un componente embebido para el proceso de desarrollo. El modelo en V hace más explícita parte de las iteraciones y repeticiones de trabajo que están ocultas en el modelo en cascada. Mientras el foco del modelo en cascada se sitúa en los documentos y productos desarrollados, el modelo en V se centra en las actividades y la corrección según Martha E Rojas Vera (2010).

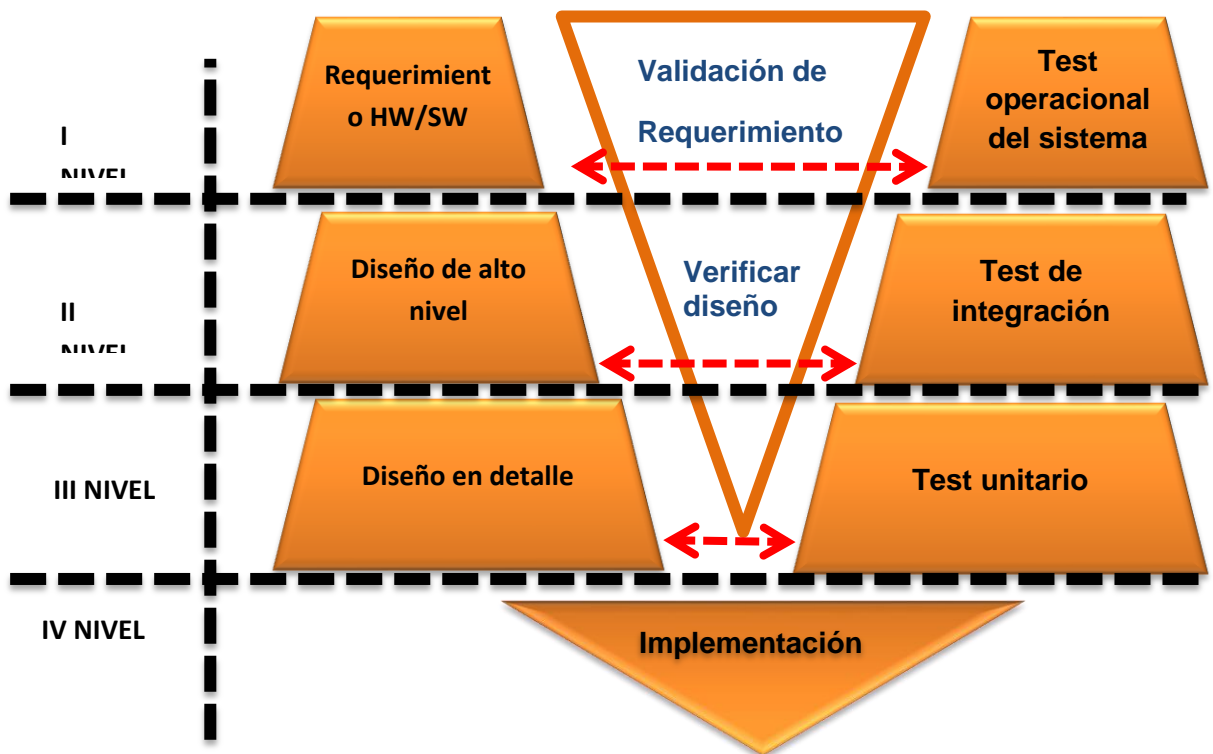


Figura 9: Modelo V.

Fuente: (Perez et al., 2006)

### 3.14. Sistemas embebidos

Según (Víctor, Juan, Sergio, & Carlos, 2013) un sistema embebido (SE) o sistema empotrado lo vamos a definir como un sistema electrónico diseñado específicamente para realizar unas determinadas funciones, habitualmente

formando parte de un sistema de mayor entidad. La característica principal es que emplea para ello uno o varios procesadores digitales (CPUs) en formato microprocesador, microcontrolador o DSP lo que le permite aportar 'inteligencia' al sistema anfitrión al que ayuda a gobernar y del que forma parte.

Para (Llinares, 2013) Se entiende por sistemas embebidos a una combinación de hardware y software de computadora, sumado tal vez a algunas piezas mecánicas o de otro tipo, diseñado para tener una función específica. Es común el uso de estos dispositivos, pero pocos se dan cuenta que hay un procesador y un programa ejecutándose que les permite funcionar.

Esto ofrece un contraste con la computadora personal, que si bien también está formada por una combinación de hardware y software más algunas piezas mecánicas (discos rígidos, por ejemplo). Sin embargo, la computadora personal no es diseñada para un uso específico. Si no que es posible darle muchos usos diferentes.

#### **3.14.1. Características de un sistema embebido**

Las principales características de un Sistema Embebido son el bajo costo y consumo de potencia. Dado que muchos sistemas embebidos son concebidos para ser producidos en miles o millones de unidades, el costo por unidad es un aspecto importante a tener en cuenta en la etapa de diseño.

Un Sistema Embebido está conformado por un microprocesador y un software que se ejecuta sobre él mismo. Sin embargo, este software necesita un lugar donde pueda guardarse para luego ser ejecutado por el procesador. Esto podría tomar la forma de memoria RAM o ROM, la cual cierta cantidad es utilizada por el Sistema Embebido.

#### **3.14.2. Sistemas embebidos en Arduino**

Arduino es 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 cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces y motores.

Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta, lo que da libertad a cualquiera para adaptarlos a sus necesidades.

### **3.14.3. Sistemas inteligentes**

En la actualidad se está planteando la evolución de los “Sistemas Embebidos” a “Sistemas Inteligentes”, en donde la principal diferencia para considerarlos como inteligentes es que deben estar conectados a otro dispositivo (M2M, Comunicación Maquina a Maquina) o en especial a Internet.

### **3.15. Autoridad nacional del agua**

El agua es un recurso necesario para la pervivencia del ser humano, en la medida que se incrementa la población en el mundo será necesario el acceso a mayor cantidad de agua de calidad, sin embargo, la cantidad de agua que hay en el mundo no se incrementa.

- El agua dulce disponible no alcanza ni el 0,5% de la totalidad del agua existente.
- El agua dulce se renueva por lluvia: 40.000 a 50.000 kilométricos cúbicos al año.
- El consumo mundial de agua se dobla cada 20 años (un ritmo dos veces mayor que el crecimiento humano).
- Según información del Programa Naciones Unidas para el Desarrollo (PNUD), más de 1.100 millones de personas en el mundo carecen agua potable y 31 países padecen escasez de ella y dos de cada cinco personas no cuentan con instalaciones adecuadas de saneamiento.

El Perú es un país privilegiado, cuenta con 1.89 % de la disponibilidad de agua dulce del mundo, por ello, debemos cuidarla y administrarla con justicia y equidad para todos.

Nuestra geografía ha determinado la existencia de 159 cuencas hidrográficas en nuestro territorio, cada una de ellas tiene sus singularidades y necesidades

de gestión de recursos hídricos adecuados, por ello, la Autoridad Nacional del Agua, a través del Proyecto de Modernización de la Gestión de los Recursos Hídricos viene promoviendo la creación, instalación y gestión de los consejos de recursos hídricos por cuencas como uno de los modelos de gestión del agua más eficientes y adecuados para el país.

### **3.16. ESTADO DEL ARTE**

En las presentes investigaciones realizadas de sistemas domóticos se pudo rescatar las que se mencionará en el siguiente apartado según a la conclusión de cada investigación.

#### **3.16.1. Ámbito nacional**

- **Primera investigación**

(Tamayo & Contreras, 2014) La Universidad Nacional Agraria la Molina, 14 de marzo del 2014 realizó un estudio de línea base para establecer un plan de Ecoeficiencia para el uso del Agua potable en las instalaciones de las áreas académicas y administrativas de la Universidad, el presente estudio se elaboró con el objetivo de que la UNALM esté a la par con otras instituciones que ya cuentan con una línea base, la cual incluye un inventario de los equipos sanitarios, la identificación de prácticas inadecuadas y la evaluación del consumo de agua en las áreas administrativas y académicas.

- **Segunda investigación**

El ministerio del ambiente en el 2009, estableció una guía para el uso ecoeficiente de los recursos, esto implica el uso eficiente de los recursos, que conlleva menor producción de residuos y contaminación, a la vez que se reduce los costos operativos, contribuyendo así a la sostenibilidad económica general de la institución.

#### **3.16.2. Ámbito internacional**

- **Primera investigación**

(Edison Fabricio, 2014) La Universidad Nacional de Loja Ecuador en el año 2014, realizó una investigación este con el objetivo de poder medir el consumo de agua potable en un hogar a través de internet en tiempo real y con reportes de correo electrónico hacia el usuario. Se estudia e

implementa la ingeniería de hardware y software que el prototipo necesitara para su correcto funcionamiento.

- **Segunda investigación**

(ANDRÉS, 2011) La Escuela Politécnica del Ejército de Ecuador en el año 2011, realizó la investigación de un proyecto que trata de diseñar e implementar un medidor digital del consumo de agua potable en lugares residenciales; así como su envío de datos vía SMS hasta una central la cual es la encargada de guardar y detallar la información receptada de cada uno de los clientes mes a mes.

## 4. CAPITULO IV: INGENIERÍA DE PROYECTOS

### 4.1. METODOLOGÍA Y PLANIFICACIÓN DEL PROYECTO ECOEFICIENTE

Para elaboración de presente proyecto se utilizó el método V (**Verificar y Validar**). El método V es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida del proyecto, para este proyecto constituye la metodología más adecuada el cual se evaluó las cuatro etapas de la metodología para lograr el objetivo principal del proyecto.

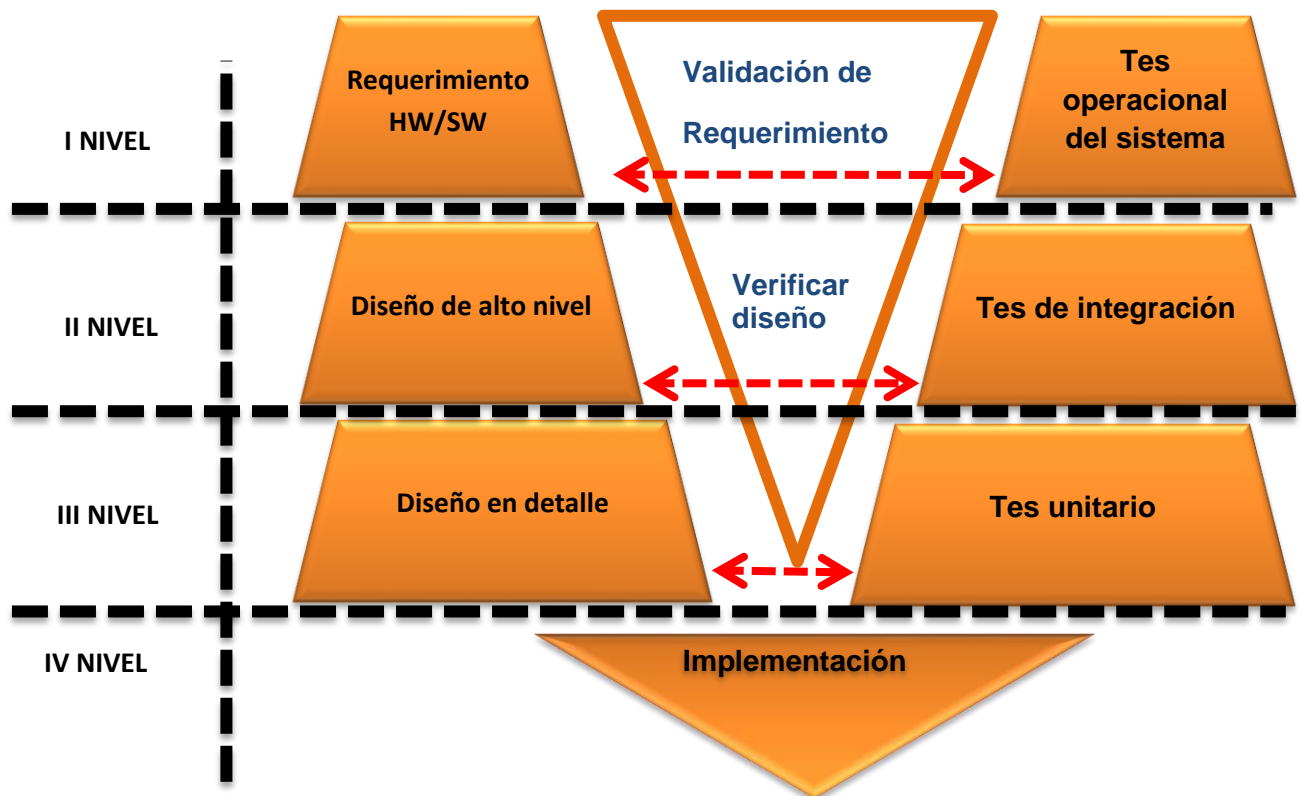


Figura 10: Ciclo de vida del proyecto.  
Fuente: (Perez et al., 2006).

A continuación, se detalla los niveles de la metodología que debe tener el sistema.

### 4.2. PUESTA EN MARCHA DE LA METODOLOGÍA V

#### 4.2.1. REQUERIMIENTO DE HARDWARE Y SOFTWARE

El requerimiento de hardware y software se realizó según a los requerimientos funcionales y no funcionales, para el diseño de prototipo con tecnología Arduino y Android; En este nivel se fue analizó un test operacional del sistema (verificación y validación), del hardware y

software a usarse todo ello con la finalidad de no tener riesgos en el proyecto.

#### **4.2.1.1. Requerimientos funcionales**

- ✓ El sistema tiene incorporado el sensor de efecto hall el cual mide el flujo de agua.
- ✓ El móvil recibe la información de los litros por minuto, hora y días que envía el sensor de efecto hall.
- ✓ Se tiene un servo motor que realiza un giro de 90 grados para el abierto y cerrado del inodoro.
- ✓ El sensor Pir detecta la señal del movimiento de la mano del usuario para enviar un pulso de energía al servo motor.
- ✓ El sistema permite programar el tiempo de activo de la ducha, mediante un teclado numérico que tiene incorporado.
- ✓ Al programarse el tiempo en el sistema tiene a hacer un conteo reductivo esto como sistema de información para el usuario de cuánto tiempo le resta, esta información se visualiza en una pantalla LCD.
- ✓ El sistema permite abrir y cerrar lava mano mediante el sensor de infrarrojo.
- ✓ El usuario tiene la información de consumo de agua en el dispositivo móvil.
- ✓ El aplicativo móvil tiene selectores de abierto y cerrado el cual es ejecutado en casos que sea necesario, como también tiene la opción de automático y manual.
- ✓ El usuario tiene la posibilidad de acceder al código fuente del sistema.
- ✓ El sistema es controlado desde un dispositivo móvil que cuenta con la interfaz diseñada.

#### **4.2.1.2. Arduino y Raspberry Pi**

El proyecto titulado implementación de un prototipo con tecnología arduino y android, para la ecoeficiencia en el uso del agua potable en los predios de Talavera. Para el proyecto mencionado se utilizó Arduino por las características que se detalla en la tabla, previo análisis del costo y las características para el proyecto.



**Tabla 6:** Diferencias entre tecnologías

<b>HARDWARE Y SOFTWARE</b>		
<b>Característica</b>	Arduino	Raspberry pi
<b>Precio en dólares</b>	S/. 60	S/. 90
<b>Tamaño</b>	7.6 x 1.9 x 6.4 cm	8.6cm x 5.4cm x 1.7cm
<b>Memoria</b>	0.002MB	512MB
<b>Velocidad de reloj</b>	16 MHz	700 MHz
<b>On Board Network</b>	Ninguna	10/100 wired Ethernet RJ45
<b>Multitarea</b>	No	Sí
<b>Voltaje de entrada</b>	7 a 12 V	5 V
<b>Memoria Flash</b>	32KB	Tarjeta SD (2 a 16G)
<b>Puertos USB</b>	Uno	Dos
<b>Sistema operativo</b>	Ninguno	Distribuciones de Linux
<b>Entorno de desarrollo integrado (IDE)</b>	Arduino	Scratch, IDLE, cualquiera con soporte Linux

**Fuente:** (M. Alberto, 2017).

#### **4.2.1.3. Requerimientos no funcionales**

##### **Eficiencia:**

- ✓ La funcionalidad del sistema responde al usuario en menos de 2 segundos.
- ✓ El sistema es capaz de operar adecuadamente 24 x 7 sin ningún tipo de dificultad.
- ✓ El sistema es capaz de hacer varios procesos a la vez sin dificultad.

##### **Seguridad:**

- ✓ Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador o encargado del acceso a código fuente.

- ✓ El sistema es seguro por el tema de que solo es permitido acceder mediante computadora.
- ✓ Todo el sistema debe ser actualizada cada mes para un buen funcionamiento eficaz y seguro, este permitirá trabajar de una manera más eficiente.
- ✓ El sistema no continuara operando en caso de que no exista energía eléctrica.
- ✓ El sistema no continuara operando en caso de incendio.

**Usabilidad:**

- ✓ El tiempo de aprendizaje del sistema por un usuario es sencillo.
- ✓ Cuenta con un manual de usuario estructurados adecuadamente.
- ✓ Posee interfaz gráfica de fácil manejo para cualquier usuario.

**4.2.2. DISEÑO DE ALTO NIVEL**

El proyecto se constituyó con la plataforma de desarrollo (Arduino, Android, etc.) y hardware (válvulas, sensores, temporizador, Arduino, dispositivo móvil, etc.). Bajo estas tecnologías se dio la solución al problema que se planteó para reducir el consumo de agua potable y de la misma manera poder contar con la cantidad de consumo que realiza un predio, así como también se realizó un test de validación y verificación para no tener errores en el desarrollo.

**4.2.2.1. Estableciendo actores**

Se identificaron los actores y el equipo de desarrollo para el proyecto, a continuación, se lista la relación de los actores que participaron de forma activa durante el desarrollo del sistema:

- **Director del proyecto:** Encargado de brindar la buena viabilidad y disponibilidad de recursos que se necesita en el proyecto.
- **Analista de sistemas:** Encargado en diseñar la infraestructura de la aplicación, teniendo en cuenta los requisitos funcionales, alcance y limitaciones que la aplicación debe tener.

- **Programador:** Encargado de la implementación y pruebas de la aplicación donde la infraestructura de esta, es diseñada anteriormente por el analista de sistemas.

El presente proyecto fue desarrollado por una sola persona, en el periodo establecido, cumpliendo éste las tres responsabilidades mencionadas anteriormente.

#### 4.2.2.2. Estableciendo Usuarios

Identificamos los beneficiarios, que serán las personas que harán uso del sistema para el uso ecoeficiente del agua potable:

- **Usuarios directos:** Familias que viven en los predios de Talavera.
- **Usuarios indirectos:** Familias interesadas en un uso ecoeficiente del agua potable en los predios.

#### 4.2.2.3. Información clave del proyecto

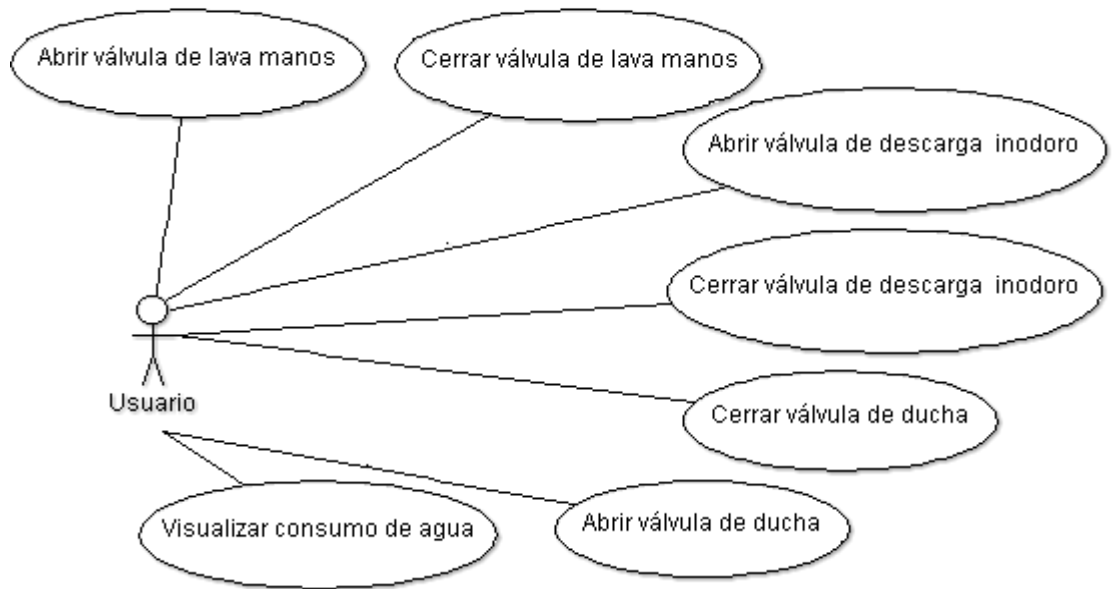
A continuación, se define el objetivo general del proyecto, tipo de aplicación y la plataforma con la cual se desarrolló el proyecto.

**Tabla 7:** Información del proyecto.

INFORMACION GENERAL DEL PROYECTO	
<b>OBJETIVO GENERAL DEL PROYECTO</b>	Implementación de un prototipo con tecnología Arduino y Android, para la ecoeficiencia en el uso del agua potable en los predios de Talavera 2016.
<b>LUGAR DE IMPLEMENTACION</b>	Predios de Talavera – Andahuaylas.
<b>PLATAFORMA</b>	Arduino-Android

**Fuente:** *Elaboración propia.*

#### 4.2.2.4. Diagrama de caso de uso general



**Figura 11:** Caso de uso general del sistema ecoeficiente.  
**Fuente:** Elaboración propia.

#### 4.2.3. DISEÑO DETALLADO

El hardware y software que se utilizó para construir el prototipo con tecnología Arduino y Android se mencionan a continuación de forma detallada.

##### Herramientas de desarrollo (Software)

- Arduino 1.6.9
- Android Studio 2.3.3
- Eagle 7.6.0

##### Herramientas físicas (Hardware)

- 1 laptop Toshiba core i5, sistema operativo Windows 7, procesador 2.50 GHz
- 1 tablet de 9' pulgadas, sistema operativo Android 4.4.2
- 1 Smartphone con sistema operativo Android 5.0

##### Casos de uso del sistema a detalle

- **Circuito del lavamanos:** En esta parte del circuito se observa el funcionamiento del sensor de infrarrojo y la válvula solenoide.

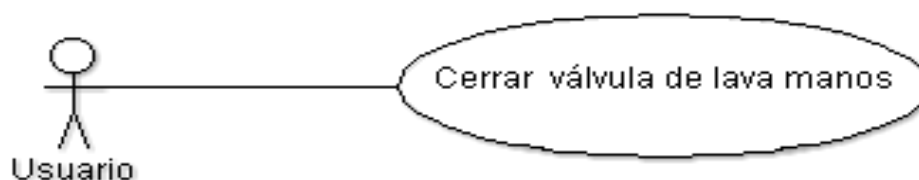


**Figura 12:** Caso de uso abrir lava mano.  
**Fuente:** Elaboración propia.

**Tabla 8:** Especificación de caso de uso abrir válvula lava manos.

CASO DE USO:	<b>Abrir válvula lava manos.</b>
ACTORES:	Usuario
DESCRIPCIÓN:	El caso de uso del sistema inicia cuando la mano del usuario es detectado por el sensor del lavamanos.
PRECONDICIONES:	<ul style="list-style-type: none"> <li>✓ La válvula debe estar cerrada.</li> <li>✓ El sensor tiene que estar en estado de espera.</li> </ul>
FLUJO BÁSICO:	<ul style="list-style-type: none"> <li>✓ El sensor detecta la mano del usuario y este emite un dato al Arduino.</li> <li>✓ El sistema activa el led de encendido del sensor.</li> <li>✓ El Arduino recibe el dato y este emtime un pulso de energía 12V para abrir la válvula.</li> <li>✓ Mientras la válvula se encuentre abierta el sistema muestra dos estados con led`s de encendido y apagado de la válvula.</li> </ul>
FLUJO ALTERNATIVO:	<ul style="list-style-type: none"> <li>✓ Mientras el sensor no detecte presencias de calor, la válvula permanece cerrado.</li> </ul>

**Fuente:** Elaboración propia.



**Figura 13:** Caso de uso cerrar lava manos.

**Fuente:** Elaboración propia.

**Tabla 9:** Especificación de caso de uso Cerrar válvula lava manos.

CASO DE USO:	<b>Cerrar válvula lava manos.</b>
ACTORES:	Usuario
DESCRIPCIÓN:	El caso de uso del sistema inicia cuando el usuario retira la mano del rango de detección del sensor de lava manos.
PRECONDICIONES:	<ul style="list-style-type: none"> <li>✓ La válvula debe estar abierta.</li> <li>✓ El sensor tiene que estar en estado de espera.</li> </ul>

FLUJO BÁSICO:	<ul style="list-style-type: none"> <li>✓ La válvula estará abierta mientras el usuario no retire la mano del rango de detección del sensor.</li> <li>✓ El arduino actualiza constantemente los datos recibidos del sensor.</li> <li>✓ El usuario aleja la mano del sensor el dato es enviado al Arduino.</li> <li>✓ El arduino recibe los datos del sensor y este cierra la válvula.</li> </ul>
FLUJO ALTERNATIVO:	<ul style="list-style-type: none"> <li>✓ Mientras el sensor detecte presencias de calor, la válvula permanece abierta.</li> </ul>

Fuente: *Elaboración propia.*

- **Circuito del inodoro:** En este apartado se tiene el funcionamiento de un servo motor y un sensor de presencia ambos hacen el trabajo.



Figura 14: *Caso de uso abrir inodoro.*

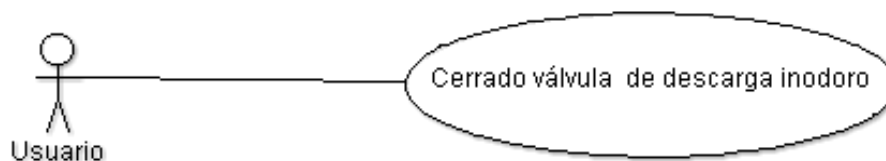
Fuente: *Elaboración propia.*

Tabla 10: *Especificación de casos de uso abrir válvula de descarga inodoro.*

CASO DE USO:	<b>Abrir válvula de descarga inodoro</b>
ACTORES:	Usuario
DESCRIPCIÓN:	El caso de uso del sistema inicia cuando la mano del usuario es detectada por el sensor del inodoro.
PRECONDICIONES:	<ul style="list-style-type: none"> <li>✓ La válvula debe estar cerrada.</li> <li>✓ El sensor tiene que estar en estado de espera.</li> </ul>
FLUJO BÁSICO:	<ul style="list-style-type: none"> <li>✓ El sensor detecta la mano del usuario y este emite un dato al Arduino.</li> <li>✓ El sistema activa el led de encendido del sensor.</li> <li>✓ El Arduino recibe el dato y este emite un pulso de energía 5V para girar el servo motor en un ángulo de 120 grado.</li> <li>✓ Mientras la válvula se encuentre abierta el sistema muestra dos estados con led's de encendido y apagado de la válvula.</li> </ul>

FLUJO ALTERNATIVO:	✓ Mientras el sensor no detecte presencias de movimiento, la válvula permanece cerrado.
--------------------	---

Fuente: *Elaboración propia.*



**Figura 15:** *Caso de uso Cerrar inodoro.*

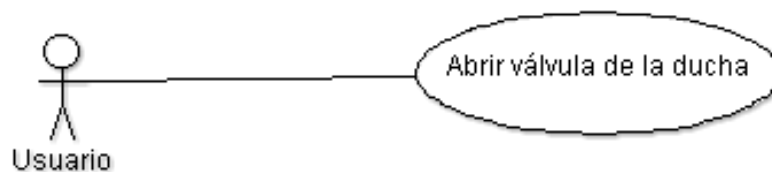
Fuente: *Elaboración propia.*

**Tabla 11:** *Especificación de casos de uso Cerrar válvula de descarga inodoro.*

CASO DE USO:	<b>Cerrar válvula de descarga inodoro</b>
ACTORES:	Usuario
DESCRIPCIÓN:	El caso de uso del sistema inicia cuando el usuario retira la mano del rango de detección del sensor de lava manos.
PRECONDICIONES:	<ul style="list-style-type: none"> <li>✓ La válvula debe estar abierta.</li> <li>✓ El sensor tiene que estar en estado de espera.</li> </ul>
FLUJO BÁSICO:	<ul style="list-style-type: none"> <li>✓ La válvula de descarga estará abierta mientras el usuario no retire la mano del rango de detección del sensor.</li> <li>✓ El Arduino actualiza constantemente los datos recibidos del sensor.</li> <li>✓ El usuario aleja la mano del sensor el dato es enviado al Arduino.</li> <li>✓ El servo motor retorna a su estado inicial en un ángulo 0.</li> <li>✓ El Arduino recibe los datos del sensor y este cierra la válvula de descarga.</li> </ul>
FLUJO ALTERNATIVO:	✓ Mientras el sensor detecte presencias de movimiento, la válvula permanece abierta.

Fuente: *Elaboración propia.*

- **Circuito de la ducha:** Este armado de circuito en donde se trabajará con tres componentes muy importantes la válvula solenoide, teclado numérico y LCD.



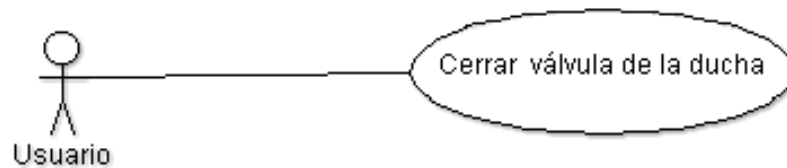
**Figura 16:** Caso de uso abrir ducha.  
**Fuente:** Elaboración propia.

**Tabla 12:** Especificación de casos de uso abrir válvula de la ducha.

CASO DE USO:	<b>Abrir válvula de la ducha</b>
ACTORES:	Usuario
DESCRIPCIÓN:	El caso de uso del sistema inicia cuando el usuario programa el tiempo de encendido de la ducha.
PRECONDICIONES:	<ul style="list-style-type: none"> <li>✓ La válvula debe estar cerrada</li> <li>✓ El tiempo de inicio tiene que estar en cero (00:00:00).</li> </ul>
FLUJO BÁSICO:	<ul style="list-style-type: none"> <li>✓ Inicia cuando el usuario programa el tiempo del abierto de la válvula.</li> <li>✓ El tiempo programado es enviado al Arduino y este almacenado en un buffet.</li> <li>✓ El tiempo programado es almacenado para hacer el conteo de manera descendiente.</li> <li>✓ El Arduino contando con el tiempo programado emite un pulso de energía de 12V para abrir la válvula.</li> <li>✓ Mientras la válvula se encuentre abierta el sistema muestra dos estados con led's de encendido y apagado de la válvula.</li> </ul>
FLUJO ALTERNATIVO:	<ul style="list-style-type: none"> <li>✓ En el caso que el usuario pretenda cancelar el tiempo programado solo tiene que pulsar el teclado de reinicio.</li> <li>✓ Mientras no se envíe el tiempo mediante el teclado no se abrirá la válvula.</li> </ul>

**Fuente:** Elaboración propia.





**Figura 17:** Caso de uso cerrar ducha.

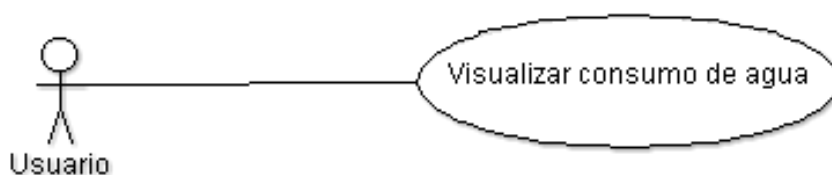
**Fuente:** *Elaboración propia.*

**Tabla 13:** Especificación de casos de uso cerrar válvula de la ducha.

CASO DE USO:	<b>Cerrar válvula de la ducha</b>
ACTORES:	Usuario
DESCRIPCIÓN:	El caso de uso del sistema inicia cuando el usuario programa el tiempo de encendido de la ducha.
PRECONDICIONES:	<ul style="list-style-type: none"> <li>✓ La válvula debe estar abierta.</li> <li>✓ El tiempo programado no tiene que ser incremental.</li> </ul>
FLUJO BÁSICO:	<ul style="list-style-type: none"> <li>✓ Inicia cuando el sistema está en espera del tiempo programado.</li> <li>✓ El tiempo programado por el usuario se muestra en una pantalla LCD.</li> <li>✓ Cuando finaliza el tiempo el Arduino deja de suministrar de energía a la válvula.</li> </ul>
FLUJO ALTERNATIVO:	<ul style="list-style-type: none"> <li>✓ En el caso que el usuario pretenda cancelar el tiempo programado solo tiene que pulsar el teclado de reinicio.</li> <li>✓ Mientras no culmine el tiempo programado mediante el teclado no se cerrará la válvula.</li> </ul>

**Fuente:** *Elaboración propia.*

- **Consumo del agua:** Se muestra el consumo por día y mes en el teléfono móvil el cual ayuda al usuario poder llevar la cifra exacta del consumo.



**Figura 18:** Caso de uso consumo de agua.

**Fuente:** *Elaboración propia.*

**Tabla 14:** Especificación de casos de uso visualizar consumo de agua.

CASO DE USO:	<b>Vizualisar consumo de agua</b>
ACTORES:	Usuario
DESCRIPCIÓN:	El caso de uso del sistema inicia cuando el usuario abre el aplicativo móvil y se conecta al módulo Bluetooth.
PRECONDICIONES:	<ul style="list-style-type: none"> <li>✓ El modulo Bluetooth del dispositivo móvil tiene que estar encendido.</li> <li>✓ El aplicativo móvil tiene que estar instalado en el móvil.</li> </ul>
FLUJO BÁSICO:	<ul style="list-style-type: none"> <li>✓ Inicia cuando usuario inicia la aplicación en el dispositivo.</li> <li>✓ El usuario selecciona el modulo Bluetooth del Arduino.</li> </ul>
FLUJO ALTERNATIVO:	<ul style="list-style-type: none"> <li>✓ Mientras no esté encendido el modulo Bluetooth no se podrá mostrar el consumo de agua.</li> </ul>

**Fuente:** *Elaboración propia.*

#### 4.2.4.IMPLEMENTACION

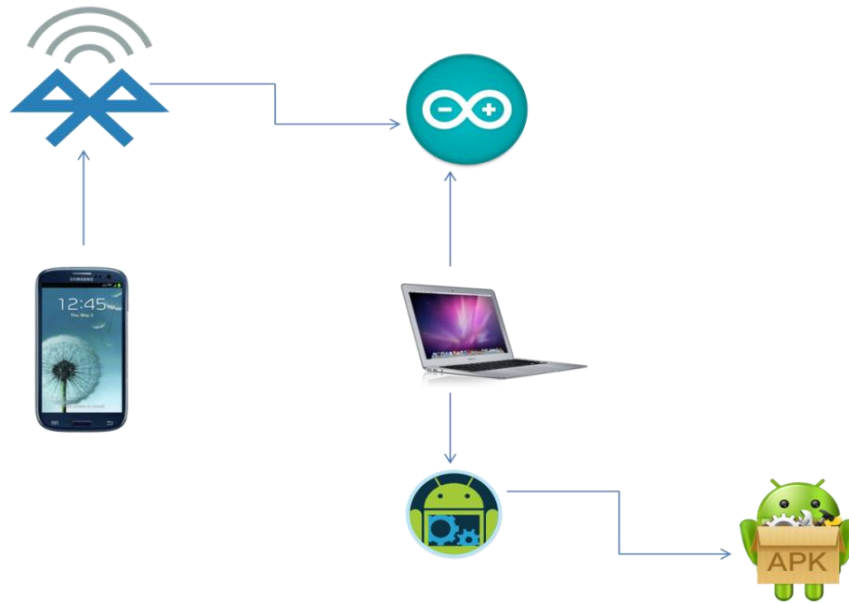
Se realizó la programación del sistema el cual dio la funcionalidad de cada uno de los componentes que conforma el diseño de prototipo.

En cada una de estas etapas del desarrollo se realizó una prueba de verificación y validación del desarrollo para cumplir con los requerimientos establecidos. Cada fase del desarrollo tendrá que pasar por ser verificado y valido para que el producto no tenga fallas en el futuro.

##### 4.2.4.1. Implementación en ambos entornos de desarrollo

Tomando el análisis realizado con los casos de uso se procedió a realizar la implantación del aplicativo móvil, que hará posible control del sistema.

Es uno de los apartados, con mayor importancia es donde el usuario aprenderá a manipular el sistema desde el móvil cuando este inicia, esto empieza desde que la aplicación es ejecutada y posteriormente conectada a Bluetooth del sistema ecoeficiente, posteriormente realizar ya sea el cerrado, apagado y tener la información de consumo de agua potable.



**Figura 19:** *Diseño conceptual.*

**Fuente:** *Elaboración propia.*

### **Diseño de la aplicación móvil**

Para el diseño de la aplicación móvil se consideró lo siguiente:

- ✓ **Emparejamiento de dispositivos Bluetooth - control válvulas**
- ✓ **Menú de control de aplicativo - control válvulas**
- ✓ **Emparejamiento de dispositivo Bluetooth - consumo de agua**
- ✓ **Menú de visualización - consumo de agua**

### **Emparejamiento de dispositivo Bluetooth - control válvulas**

Se procedió a emparejamiento de los dispositivos Bluetooth, con la finalidad de hacer una conexión y compartir datos.

#### **Librerías para Bluetooth**

- ✓ **util.Set** : Conjunto de los objetos a crearse.
- ✓ **app.Activity**: Clase activity que es llamada por deviceListActivity.
- ✓ **bluetooth.BluetoothAdapter**: Iniciar la búsqueda de dispositivos, consultar una lista de dispositivos enlazados (emparejados).
- ✓ **bluetooth.BluetoothDevice**: permite crear una conexión con el dispositivo respectivo o consulta de información sobre el mismo, tales como el nombre, dirección.
- ✓ **Intent**: descripción abstracta de una operación a realizar.
- ✓ **os.Bundle**: asignación de claves de cadena a diferentes.

- ✓ **util.Log:** envío de salida del registro.
- ✓ **view.View:** Crea cuadro de dialogo de forma rectangular emitiendo un mensaje.
- ✓ **widget.AdapterView:** Forma parte principal del View.
- ✓ **widget.ArrayAdapter:** Respaldado por una gran variedad de objetos arbitrarios.
- ✓ **widget.Button:** Boton de permitir o rechazar el mensaje del View.
- ✓ **widget.ListView:** Muestra elementos de una lista de desplazamiento vertical.
- ✓ **widget.TextView:** Muestra una caja de texto emergente.
- ✓ **widget.Toast:** Mensaje de texto que se muestra para el usuario.
- ✓ **widget.AdapterView.OnItemClickListener:** Envía la data recibida pulsado por el usuario.

Se configuro la creación de un Textview con la finalidad de mostrar un mensaje de información que acepte o rechazar la conexión a Bluetooth.

```
TextView textView1;
public static String EXTRA_DEVICE_ADDRESS = "device_address";
private BluetoothAdapter mBtAdapter;
private ArrayAdapter<String> mPairedDevicesArrayAdapter;
```

**Código 1:** Variables de tipo público y privado.

**Fuente:** Autor utilizando Recortes de Windows 7.

Con la variable **noDevices** se confirma el emparejamiento de algún dispositivo en el móvil, en caso contrario no existe emparejamiento con ningún dispositivo.

```

// agregar dispositivo emparejado
if (pairedDevices.size() > 0) {
    findViewById(R.id.title_paired_devices).setVisibility(View.VISIBLE);
    for (BluetoothDevice device : pairedDevices) {
        mPairedDevicesArrayAdapter.add(device.getName() + "\n" + device.getAddress());
    }
} else {
    String noDevices = "Ningun dispositivo pudo ser emparejado".toString();
    mPairedDevicesArrayAdapter.add(noDevices);
}
}
}

```

**Código 2:** *Dispositivos emparejados.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

Con la línea del código anterior se comprueba la conexión del dispositivo emparejado, se emite un mensaje al usuario “conectando...” el dispositivo.

```

private OnItemClickListener mDeviceClickListener = new OnItemClickListener() {
    public void onItemClick(AdapterView<?> av, View v, int arg2, long arg3) {

        textView1.setText("Conectando...");
        // direccion Mac del dispositivo
        String info = ((TextView) v).getText().toString();
        String address = info.substring(info.length() - 17);
        // iniciar la actividad de reconocer la MAC.
        Intent i = new Intent(DeviceListActivity.this, MainActivity.class);
        i.putExtra(EXTRA_DEVICE_ADDRESS, address);
        startActivity(i);
    }
};

```

**Código 3:** *Conectar según dirección MAC.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

Una vez que carga el proceso de conexión, las líneas de código anterior comprueban si es posible la conexión, en el caso que no sea posible se verifica en la nueva función **checkBTState**, posteriormente se emite los mensajes correspondientes.

```

private void checkBTState() {
    // Condición Bluetooth activo o desactivo
    mBluetoothAdapter=BluetoothAdapter.getDefaultAdapter();
    if(mBluetoothAdapter==null) {
        Toast.makeText(getApplicationContext(), "El dispositivo no soporta Bluetooth", Toast.LENGTH_SHORT).show();
    } else {
        if (mBluetoothAdapter.isEnabled()) {
            Log.d(TAG, "...Bluetooth Activado...");
        } else {
            //Solicitar la activación del Bluetooth
            Intent enableBluetoothIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBluetoothIntent, 1);
        }
    }
}
}

```

**Código 4:** Conexión a Bluetooth.

**Fuente:** Autor utilizando Recortes de Windows 7.

### Menú de control de dispositivo - control válvulas

Dentro de la carpeta **Java – MainActivity** procedemos darle el control del dispositivo y enviar datos al Bluetooth emparejado del Arduino.

```

public class MainActivity extends Activity {
}

```

**Código 5:** Clase principal.

**Fuente:** Autor utilizando Recortes de Windows 7.

En la clase **MainActivity** se crea un identificador UUID, el cual es para conectar dispositivos Bluetooth, como el módulo BT RN42 String, para MAC address.

```

private static final UUID BTMODULEUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
private static String address = null;

```

**Código 6:** Conexión de modelo Bluetooth.

**Fuente:** Autor utilizando Recortes de Windows 7.

Con el código anterior se hace posible la identificación del dispositivo Bluetooth, este código permite el envío y recepción de datos del Arduino, para poder enviar los datos se usa ya sea carácter o números los cuales serán enviados al módulo Arduino, este a su vez responde según a la data recibida.

```

btAdapter = BluetoothAdapter.getDefaultAdapter();
checkBTState();

s1.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1,int arg2, long arg3)
    {
        int index = arg0.getSelectedItemPosition();
        if(index==0){
            mConnectedThread.write("1");
            Toast.makeText(getApplicationContext(),"Lavamanos cerrado : " + presidents[index],Toast.LENGTH_SHORT).show();
        }
        if(index==1){
            mConnectedThread.write("2");
            Toast.makeText(getApplicationContext(),"Lavamanos abierto: " + presidents[index],Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {}
});

```

```

s2.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1,int arg2, long arg3)
    {
        int index = arg0.getSelectedItemPosition();
        if(index==0){
            mConnectedThread.write("b");
            Toast.makeText(getApplicationContext(),"baño cerrado: " + presidents2[index],Toast.LENGTH_SHORT).show();
        }
        if(index==1){
            mConnectedThread.write("B");
            Toast.makeText(getApplicationContext(),"baño abierto: " + presidents2[index],Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {}
});

```



```

s3.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1,int arg2, long arg3)
    {
        int index = arg0.getSelectedItemPosition();
        if(index==0){
            mConnectedThread.write("d");
            Toast.makeText(getApplicationContext(),"ducha cerrado: " + presidents3[index],Toast.LENGTH_SHORT).show();
        }
        if(index==1){
            mConnectedThread.write("D");
            Toast.makeText(getApplicationContext(),"ducha abierto: " + presidents3[index],Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {}
});

```

```

s4.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1,int arg2, long arg3)
    {
        int index = arg0.getSelectedItemPosition();
        if(index==0){
            mConnectedThread.write("p");
            Toast.makeText(getApplicationContext(),"valvula principal cerrado: " + presidents4[index],Toast.LENGTH_SHORT).show();
        }
        if(index==1){
            mConnectedThread.write("P");
            Toast.makeText(getApplicationContext(),"valvula principal abierto: " + presidents4[index],Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {}
});

```

```

s5.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1,int arg2, long arg3)
    {
        int index = arg0.getSelectedItemPosition();
        if(index==0){
            mConnectedThread.write("A");
            Toast.makeText(getApplicationContext(),"modo automatico activo: " + presidents5[index],Toast.LENGTH_SHORT).show();
        }
        if(index==1){
            mConnectedThread.write("M");
            Toast.makeText(getApplicationContext(),"modo manual activo: " + presidents5[index],Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {}
});
}

```

**Código 7:** *Envío de datos a Bluetooth.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

La función **onResume** obtiene la dirección MAC, que selecciona el usuario para la creación de un nuevo MAC y configurar la dirección MAC.

```
@Override
public void onResume() {
    super.onResume();
    Intent intent = getIntent();
    address = intent.getStringExtra(DeviceListActivity.EXTRA_DEVICE_ADDRESS);
    BluetoothDevice device = btAdapter.getRemoteDevice(address);
    try {
        btSocket = createBluetoothSocket(device);
    } catch (IOException e) {
        Toast.makeText(getBaseContext(), "La creación del Socket fallo", Toast.LENGTH_LONG).show();
    }
    try
    {
        btSocket.connect();
    } catch (IOException e) {
        try
        {
            btSocket.close();
        } catch (IOException e2)
        {
        }
    }
    mConnectedThread = new ConnectedThread(btSocket);
    mConnectedThread.start();
    mConnectedThread.write("x");
}
```

**Código 8:** Confirmar la dirección MAC.

**Fuente:** Autor utilizando Recortes de Windows 7.

Una vez agregado la dirección MAC del código anterior, se emite el mensaje **“no soporta una conexión a Bluetooth”**. El dispositivo almacena los datos ingresados en el buffer en caso que no existiera conexión con el dispositivo seguidamente se emite un mensaje de fallo de conexión.

```

@Override
public void onPause()
{
    super.onPause();
    try
    {
        btSocket.close();
    } catch (IOException e2) {
    }
}
private void checkBTState() {

    if(btAdapter==null) {
        Toast.makeText(getBaseContext(), "El dispositivo no soporta bluetooth", Toast.LENGTH_LONG).s
    } else {
        if (btAdapter.isEnabled()) {
        } else {
            Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBtIntent, 1);
        }
    }
}
}

```

```

private class ConnectedThread extends Thread {
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;
    public ConnectedThread(BluetoothSocket socket) {
        InputStream tmpIn = null;
        OutputStream tmpOut = null;

        try {
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException e) { }

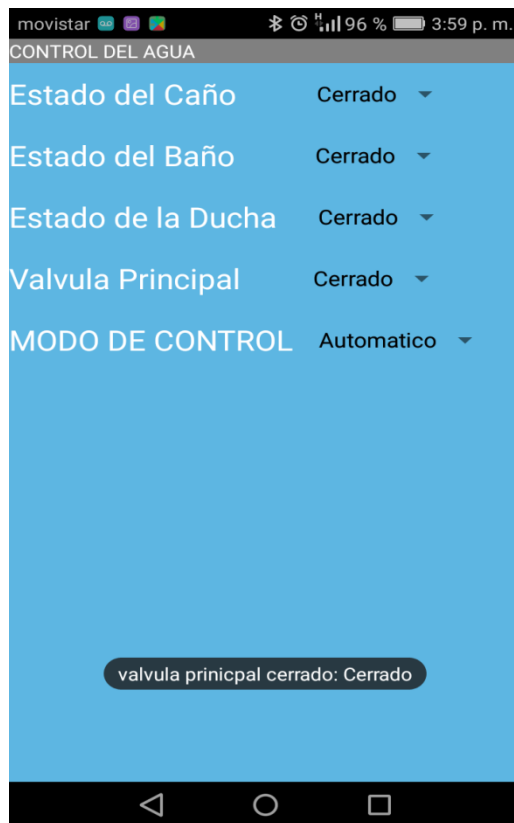
        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }
    public void run() {
        byte[] buffer = new byte[256];
        int bytes;
        while (true) {
            try {
                bytes = mmInStream.read(buffer);
                String readMessage = new String(buffer, 0, bytes);
                bluetoothIn.obtainMessage(handlerState, bytes, -1, readMessage).sendToTarget()
            } catch (IOException e) {
                break;
            }
        }
    }

    public void write(String input) {
        byte[] msgBuffer = input.getBytes();
        try {
            mmOutStream.write(msgBuffer);
        } catch (IOException e) {
            Toast.makeText(getApplicationContext(), "La Conexión fallo", Toast.LENGTH_LONG).show();
            finish();
        }
    }
}

```

**Código 9:** *Conexión activa o desactiva de Bluetooth.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*



**Figura 20:** *Aplicativo móvil de control de válvulas.*

**Fuente:** *Captura del dispositivo móvil.*


### **Emparejamiento de dispositivo Bluetooth - consumo de agua**

En este caso para el consumo de agua utilizaremos el mismo código de **emparejamiento** que se utilizó para el control de las válvulas, el código se detalla en páginas anteriores.

### **Menú de visualización - consumo de agua**

Dentro de la carpeta **Java – MainActivity** procedemos darle el control del dispositivo y enviar datos al Bluetooth emparejado del Arduino; para ello hacemos el llamado de algunas librerías el cual nos serán muy útiles en este apartado.

Procedemos a crear una clase general y este a su vez hace un llamado de la clase Activity.

```
public class MainActivity extends Activity {
    }
```

**Código 10:** *Clase principal.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

En la clase **MainActivity** se crea un identificador UUID el cual es para conectar dispositivos Bluetooth como el módulo BT RN42 String para MAC address.

```
private static final UUID BTMODULEUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
private static String address = null;
```

**Código 11:** *Conexión de modelo Bluetooth.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

Como se puede observar en el código anterior se realiza la conexión o empalme a un dispositivo Bluetooth, con este empalme procedemos a recibir los datos del sensor de flujo en el móvil.

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

            recDataString.append(readMessage);
            int endOfLineIndex = recDataString.indexOf("~");
            if (endOfLineIndex > 0) {
                String dataInPrint = recDataString.substring(0, endOfLineIndex);
                txtString.setText("Resumen" + dataInPrint);
                int dataLength = dataInPrint.length();
                txtStringLength.setText("Datos enviados : " + String.valueOf(dataLength));
                recDataString.delete(0, recDataString.length());
            }
        }
    }
};

btAdapter = BluetoothAdapter.getDefaultAdapter();
checkBTState();
```

**Código 12:** *Datos recibidos por Bluetooth.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

La función **onResume** obtiene la dirección MAC que selecciona el usuario, para la creación de un nuevo MAC y configurar la dirección MAC.



```

@Override
public void onResume() {
    super.onResume();
    Intent intent = getIntent();
    address = intent.getStringExtra(DeviceListActivity.EXTRA_DEVICE_ADDRESS);
    BluetoothDevice device = btAdapter.getRemoteDevice(address);
    try {
        btSocket = createBluetoothSocket(device);
    } catch (IOException e) {
        Toast.makeText(getBaseContext(), "La creación del Socket fallo", Toast.LENGTH_LONG).show()
    }
    try
    {
        btSocket.connect();
    } catch (IOException e) {
        try
        {
            btSocket.close();
        } catch (IOException e2)
        {
        }
    }
    mConnectedThread = new ConnectedThread(btSocket);
    mConnectedThread.start();
    mConnectedThread.write("x");
}
}

```

**Código 13:** Confirmar la dirección MAC.

**Fuente:** Autor utilizando Recortes de Windows 7.

Una vez agregado la dirección MAC del código anterior, se emite el mensaje “no soporta una conexión a Bluetooth”. El dispositivo almacena los datos ingresados en el buffer en caso que no existiera conexión con el dispositivo seguidamente se emite un mensaje de fallo de conexión.

```

@Override
public void onPause()
{
    super.onPause();
    try
    {
        btSocket.close();
    } catch (IOException e2) {
    }
}

private void checkBTState() {

    if(btAdapter==null) {
        Toast.makeText(getApplicationContext(), "El dispositivo no soporta bluetooth", Toast.LENGTH_LONG).sh
    } else {
        if (btAdapter.isEnabled()) {
        } else {
            Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBtIntent, 1);
        }
    }
}

private class ConnectedThread extends Thread {
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;
    public ConnectedThread(BluetoothSocket socket) {
        InputStream tmpIn = null;
        OutputStream tmpOut = null;

        try {
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException e) { }

        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }
    public void run() {
        byte[] buffer = new byte[256];
        int bytes;
        while (true) {
            try {
                bytes = mmInStream.read(buffer);
                String readMessage = new String(buffer, 0, bytes);
                bluetoothIn.obtainMessage(handlerState, bytes, -1, readMessage).sendToTarget(
            } catch (IOException e) {
                break;
            }
        }
    }
}

```

```

public void write(String input) {
    byte[] msgBuffer = input.getBytes();
    try {
        mmOutputStream.write(msgBuffer);
    } catch (IOException e) {
        Toast.makeText(getApplicationContext(), "La Conexión fallo", Toast.LENGTH_LONG).show();
        finish();
    }
}
}

```

**Código 14:** *Conexión activa o desactiva de Bluetooth.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*



**Figura 21:** *Aplicación de control de flujo.*

**Fuente:** *Captura dispositivo móvil.*

### **Implementación del circuito**

Para empezar con Arduino lo primero que se realiza es reconocer la placa a usarse como podría ser un Arduino nano, mega, mini, micro u otros.

- ✓ **Calibración del sensor de flujo de agua YF-201**
- ✓ **Implementación Arduino nano**
- ✓ **Implementación Arduino mega**

A continuación, detallamos cada uno de ellos.

### Calibración de sensor de flujo agua YF-201

Para poder tener una medición exacta del flujo de agua se trabajo en calibrar el sensor para ello se realizó varias muestras con 1L y 2L para lo cual se hace la medición de pulsos que envía el sensor, a continuación, se muestra en la siguiente tabla.

**Tabla 15:** Muestra recolectada.

	Muestras de sensor de flujo (pulsos)						
	1 LITRO	1 LITRO	1 LITRO	2 LITROS	2 LITROS	2 LITROS	
	Pulsos						
<b>1</b>	385	415	379	734	774	776	
<b>2</b>	413	412	374	764	777	789	
<b>3</b>	414	414	381	770	777	788	
<b>4</b>	406	413	409	750	778	764	
<b>5</b>	413	412	405	765	780	786	
<b>6</b>	412	409	415	770	780	784	
<b>7</b>	404	405	411	770	775	773	
<b>8</b>	394	412	407	764	778	767	
<b>9</b>	404	416	404	768	782	777	
<b>10</b>	413	404	396	771	776	784	
<b>Promedio</b>	405.8	411.2	398.1	762.6	777.7	778.8	
<b>K</b>	6.763333	6.853333	6.635	6.355	6.480833	6.49	
<b>PROMEDIO TOTAL</b>							6.59625

Fuente: Elaboración propia. Microsoft 2010.

$$K = \frac{N^{\circ} \text{pulsos}}{\text{volumen} * 60}$$

**Figura 22:** Formula de factor de conversión.

Fuente: Autor utilizando Recortes de Windows 7.

Para poder obtener los resultados que se muestra en la tabla se implementó el siguiente código el cual hace posible la medición de los pulsos que envía el sensor de flujo de agua. Lo primero que se hace es declarar las variables que nos ayudara a capturar los pulsos, procedemos a inicializar las variables en **void setup()** ya sea de entrada o salida y como también las interrupciones con el que trabaja nuestro Arduino y la conexión del puerto serial.

- ✓ **Serial.begin():** Conexión con el puerto serial
- ✓ **pinMode():** Función del pin ya sea salida o entrada
- ✓ **attachInterrupt():** Interrupción para los pulsos

```
void setup()
{
  Serial.begin(9600);
  pinMode(PinSensor, INPUT);
  attachInterrupt(0, ContarPulsos, RISING);
  interrupts();
}
```

**Código 15:** *Conexión puerto serial.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

Luego dentro del bucle de **void loop()** se realizamos el envío de datos por el puerto serial **9600** el número de pulsos del sensor de flujo de agua.

```
void loop ()
{
  //-----Enviamos por el puerto serie-----
  Serial.print ("Numerode Pulsos = ");
  Serial.println (NumPulsos);
  delay(100);
}
```

**Código 16:** *Número de pulsos.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

### **Implementación de la medición del volumen de consumo de agua**

Teniendo el caudal que se midió con los pulsos recibidos del sensor de flujo de agua, podemos medir el volumen de agua, puesto que el

caudal es la variación del volumen con respecto al tiempo, de esto podemos calcular el volumen:

Luego en el **void setup()** Procedemos a hacer la conexión del puerto **seril.begin()** y poner el pin cero como interrupción y crear el millis para los segundo.

```
void setup()
{

    Serial.begin(9600);
    pinMode(PinSensor, INPUT);
    attachInterrupt (0,ContarPulsos,RISING);
    Serial.println ("Envie 'r' para restablecer el volumen a 0 Litros");
    t0=millis();
}
```

**Código 17:** *Conexión puerto serial Arduino.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

Para el cálculo del caudal se usa la fórmula de la siguiente figura, posteriormente el cálculo del volumen del agua que circula por el tubo, este es enviado por consola.

$$Q = \frac{\Delta V}{\Delta t}$$
$$V = V_0 + Q\Delta t$$

**Figura 23:** *Formula de volumen.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

```

void loop ()
{
  if (Serial.available()) {
    if(Serial.read()=='r')volumen=0;
  }
  float frecuencia=ObtenerFrecuecia();
  float caudal_L_m=frecuencia/factor_conversion;
  dt=millis()-t0;
  t0=millis();
  volumen=volumen+(caudal_L_m/60)*(dt/1000);

  //-----Enviamos por el puerto serie-----
  Serial.print ("Caudal: ");
  Serial.print (caudal_L_m,3);
  Serial.print ("L/mintVolumen: ");
  Serial.print (volumen,3);
  Serial.println (" L");
}

```

**Código 18:** *Calcular de volumen.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

```

Caudal: 0.000L/mintVolumen: 0.000 L
Caudal: 0.000L/mintVolumen: 0.000 L
Caudal: 0.000L/mintVolumen: 0.000 L
Caudal: 0.000L/mintVolumen: 0.000 L
Caudal: 0.000L/mintVolumen: 0.000 L
Caudal: 0.000L/mintVolumen: 0.000 L
Caudal: 0.890L/mintVolumen: 0.015 L
Caudal: 14.540L/mintVolumen: 0.257 L
Caudal: 16.024L/mintVolumen: 0.524 L
Caudal: 13.205L/mintVolumen: 0.744 L
Caudal: 10.237L/mintVolumen: 0.915 L
Caudal: 8.309L/mintVolumen: 1.053 L
Caudal: 3.709L/mintVolumen: 1.115 L
Caudal: 11.424L/mintVolumen: 1.306 L
Caudal: 15.282L/mintVolumen: 1.560 L
Caudal: 3.412L/mintVolumen: 1.617 L
Caudal: 5.341L/mintVolumen: 1.706 L
Caudal: 12.018L/mintVolumen: 1.907 L
Caudal: 10.831L/mintVolumen: 2.087 L
Caudal: 8.605L/mintVolumen: 2.230 L

```

**Figura 24:** *Datos enviados por el puerto serial.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

### Implementación Arduino nano

En esta fase de la implementación se tendrá la recepción de los datos del sensor de flujo de agua y él envió de datos del sensor mediante el módulo Bluetooth hacia el dispositivo móvil, seguidamente se procede a la declaración de variables, crear la

función de pulso y obtener la frecuencia a trabajar con la función **void setup ()** se inicializan las variables.

- ✓ **Serial.begin():** Conexión con el puerto serial
- ✓ **pinMode():** Función del pin ya sea salida o entrada
- ✓ **attachInterrupt():** Interrupción para los pulsos
- ✓ **millis():** Tiempo en milisegundos

```
void setup()
{

  Serial.begin(9600);
  pinMode(PinSensor, INPUT);
  attachInterrupt(0, ContarPulsos, RISING);
  interrupts();

  Serial.println (" sensor de flujo de agua ");
  t0=millis();
}
```

**Código 19:** Variables de inicialización.

**Fuente:** Autor utilizando Recortes de Windows 7.

Luego de inicializar las variables en **void setup()** procedemos a hacer uso de las variables inicializadas en la función **void loop ()** donde se hace bucles consecutivos, en el que envía la data al dispositivo móvil y cada interacción cambia el valor.

En este apartado se tendrá el cálculo de flujo de agua en litros por minuto, hora y día por el consumo realizado, para lo cual se hace uso de las variables que se inicializo con anterioridad **millis()**, **obtenerfrecuencia()**.



```

void loop ()
{
  if (Serial.available())
  {
    if(Serial.read()=='r')
      volumen=0;
  }
  float frecuencia=ObtenerFrecuencia();
  float caudal_L_m=frecuencia/factor_conversion;
  float caudal_L_h=caudal_L_m*60;
  float caudal_L_d=caudal_L_h*24;
  dt=millis()-t0;
  t0=millis();
  volumen=volumen+(caudal_L_m/60)*(dt/1000);
}

```

**Código 20:** *Calculo del caudal.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

Los datos que son imprimidos en el dispositivo móvil, son enviados por el módulo Bluetooth.

```

Serial.println (" Caudal: ");
Serial.print (caudal_L_m);
Serial.println (" L/m ");
Serial.print (caudal_L_h);
Serial.println (" L/h ");
Serial.print (caudal_L_d);
Serial.println (" L/d ");
Serial.println (" Volumen(consumo): ");
Serial.print (volumen);
Serial.println ("L");
Serial.println('~');
Serial.println ();
}

```

**Código 21:** *Impresión de datos.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

### Implementación del Arduino mega

Para la implementación del código del Arduino mega y tener el control de los componentes se usó algunas librerías los cuales son indispensables en la implementación del código.

#### Librería

- ✓ **#include <Wire.h>**: Librería para el temp.

- ✓ **#include <LiquidCrystal\_I2C.h>**: Librería para el LCD 20x4
- ✓ **#include <Keypad.h>**: Librería para el teclado 4x4
- ✓ **#include <infrarrojo.h>**: Librería del sensor infrarrojo
- ✓ **#include <Servo.h>** : Librería del servomotor
- ✓ **#include <Stepper.h>**: Librería para controlar motores paso a paso.

Procedemos a declarar todas las variables y llamado a las librerías lo que se procede es a crear un función donde se detecte la señal de movimiento que nos envía el sensores en este caso del sensor pir si llega a detectaremos poner un **LOW** o **HIGH** (0-1).

```
int sensorpir()
{
    val = digitalRead(PIRPin);
    if (val == HIGH)
    {
        digitalWrite(LEDPin, HIGH);
        if (pirState == LOW)
        {
            seguro.write(ABRIR);
            //stepper.step(33);
            pirState = HIGH;
        }
    }
    else
    {
        digitalWrite(LEDPin, LOW);
        if (pirState == HIGH)
        {
            seguro.write(CERRAR);
            //stepper.step(-33);
            pirState = LOW;
        }
    }
}
```

**Código 22:** Lectura del sensor.

**Fuente:** Autor utilizando Recortes de Windows 7.

En este caso se procede a crear una función, el cual es del sensor infrarrojo, en caso que detecte algún movimiento este tiende a encender un led y posteriormente abrir una válvula o cerrar.

```

int SensorInfrarojoA ()
{
    infrarojo_estadoA = estadoA.lectura (VALORA);
    if (infrarojo_estadoA == 1)
    {
        digitalWrite(ledhd, LOW);
        digitalWrite (VALVULA1, VALVULA_OFF);
    }
    else
    {
        digitalWrite(ledhd, HIGH);
        digitalWrite (VALVULA1, VALVULA_ON);
        //elay(200);
    }
}

```

**Código 23:** *Detección de estado.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

A continuación se tiene la función showEnteredTime() del LCD que es de 20x4, con sus respectivas ubicaciones el que hace posible que se muestre el tiempo trascurrido.

```

void showEnteredTime ()
{
    lcd.setCursor (14, 3);
    lcd.print (currentTimeValue [0]);
    lcd.print (currentTimeValue [1]);
    lcd.print (":");
    lcd.print (currentTimeValue [2]);
    lcd.print (currentTimeValue [3]);
}

```

**Código 24:** *Tiempo del Lcd.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

Luego se procede a crear un función **relayStatus()** esto para poder verificar en qué estado se encuentra la válvula.

```

void relayStatus (bool state)
{
    if (state)
        digitalWrite (VALVULA2, VALVULA_ON);
    else
        digitalWrite (VALVULA2, VALVULA_OFF);
}

```

**Código 25:** Estado de válvula.

**Fuente:** Autor utilizando Recortes de Windows 7.

Ya verificado el estado, se procede a imprimir los datos en el LCD el tiempo restante y con un mensaje de información para detener.

```
void showCountdown()
{
  char timest[6];\

  lcd.setCursor(0,0);
  lcd.print("*****");
  lcd.setCursor(0,1);
  lcd.print("***cuenta atras* **");
  lcd.setCursor(0,2);
  lcd.print("*** ");
  sprintf(timest, "%d:%.2d", (timerSeconds/60), (timerSeconds - ((timerSeconds/60)*60));
  lcd.print(timest);
  lcd.print(" **");
  lcd.setCursor(0,3);
  lcd.print("pres.(*)para PARAR ");
}
}
```

**Código 26:** Mensaje LCD.

**Fuente:** Autor utilizando Recortes de Windows 7.

Seguidamente se tiene el temporizador en el display, este nos ayuda a tener la regresión del tiempo según lo programado por el teclado numérico.

```
void displayCodeEntryScreen()
{
  clearScreen();
  lcd.setCursor(0,0);
  lcd.print("****TEMPORIZADOR****");
  lcd.setCursor(0,1);
  lcd.print("****VALVULA 001****");
  lcd.setCursor(0,2);
  lcd.print("tiempo en mm:ss");
  lcd.setCursor(0,3);
  lcd.print("INICIAR->(#) ");
}
}
```

**Código 27:** Mensaje display.

**Fuente:** Autor utilizando Recortes de Windows 7.

Luego creamos la función limpiar pantalla del LCD esto para mostrar un estado en cero.

```
void clearScreen()
{
    int l;
    char tempVal[3];
    char key = keypad.getKey();

    //key pressed and state is 1
    if (int(key) != 0 and currentState == 1)
    {
        switch (key)
        {
            case '*':
                relayStatus(false);
                currentTimeValue[0]='0';
                currentTimeValue[1]='0';
                currentTimeValue[2]='0';
                currentTimeValue[3]='0';
                showEnteredTime();
                currentState = 1;
                lpcnt = 0;
                timerSeconds = 0;
                break;

            case '#':
                tempVal[0] = currentTimeValue[0];
                tempVal[1] = currentTimeValue[1];
                tempVal[2] = 0;

                timerSeconds = atol(tempVal) * 60;

                tempVal[0] = currentTimeValue[2];
                tempVal[1] = currentTimeValue[3];
                tempVal[2] = 0;

                timerSeconds = timerSeconds + atol(tempVal);
                currentState = 2;
                break;

            default:
                currentTimeValue[0] = currentTimeValue[1];
                currentTimeValue[1] = currentTimeValue[2];
                currentTimeValue[2] = currentTimeValue[3];
                currentTimeValue[3] = key;
                showEnteredTime();
                break;
        }
    }
}
```

```

if (currentState == 2)
{
    if (int(key) != 0)
    {
        if (key == '*')
        {
            relayStatus(false);
            displayCodeEntryScreen();
            currentTimeValue[0]='0';
            currentTimeValue[1]='0';
            currentTimeValue[2]='0';
            currentTimeValue[3]='0';
            showEnteredTime();
            currentState = 1;
            lpcnt = 0;
            timerSeconds = 0;
        }
    }
    else
    {
        if (lpcnt > 9)
        {
            lpcnt = 0;
            --timerSeconds;
            showCountdown();
            if (timerSeconds <= 0)
            {
                currentState = 1;
                relayStatus(false);
                displayCodeEntryScreen();
                showEnteredTime();
            }
            else
            {
                relayStatus(true);
            }
        }
    }

    ++lpcnt;
    //delay(100);
}
}

```

**Código 28:** Limpiar Screen.

**Fuente:** Autor utilizando Recortes de Windows 7.

A continuación se crea una función recursiva de código Void loop() conjuntamente con el condigo anterior.

```
void loop()  
{
```

**Código 29:** *Función recursiva.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

Procedemos escoger el modo manual o modo automático, para que pueda ser controlado por el sistema.

```
if(Serial.available())  
{  
    in = Serial.read();  
    if (in == 'A'){mode = 'a'; }  
    if (in == 'M'){mode = 'm'; }  
    digitalWrite(VALVULA1, VALVULA_ON);  
    digitalWrite(VALVULA2, VALVULA_ON);  
    digitalWrite(VALVULA3, VALVULA_ON);  
    digitalWrite(VALVULA4, VALVULA_ON);  
    seguro.write(CERRAR);  
}  
}
```

**Código 30:** *Tipo de control del sistema.*

**Fuente:** *Autor utilizando Recortes de Windows 7.*

Luego de escoger el modo de control automático el sistema es autónomo no necesita la intervención del usuario.

```

if (mode == 'a')
{
    int l;
    char tempVal[3];
    char key = keypad.getKey();

    //key pressed and state is 1
    if (int(key) != 0 and currentState == 1)
    {
        switch (key)
        {
            case '*':
                relayStatus(false);
                currentTimeValue[0]='0';
                currentTimeValue[1]='0';
                currentTimeValue[2]='0';
                currentTimeValue[3]='0';
                showEnteredTime();
                currentState = 1;

                lpCnt = 0;
                timerSeconds = 0;
                break;

            case '#':
                tempVal[0] = currentTimeValue[0];
                tempVal[1] = currentTimeValue[1];
                tempVal[2] = 0;

                timerSeconds = atol(tempVal) * 60;

                tempVal[0] = currentTimeValue[2];
                tempVal[1] = currentTimeValue[3];
                tempVal[2] = 0;

                timerSeconds = timerSeconds + atol(tempVal);
                currentState = 2;
                break;

            default:
                currentTimeValue[0] = currentTimeValue[1];
                currentTimeValue[1] = currentTimeValue[2];
                currentTimeValue[2] = currentTimeValue[3];
                currentTimeValue[3] = key;
                showEnteredTime();
                break;
        }
    }
}

```



```

if (currentState == 2)
{
  if (int(key) != 0)
  {
    if (key == '*')
    {
      relayStatus(false);
      displayCodeEntryScreen();
      currentTimeValue[0]='0';
      currentTimeValue[1]='0';
      currentTimeValue[2]='0';
      currentTimeValue[3]='0';
      showEnteredTime();
      currentState = 1;
      lpcnt = 0;
      timerSeconds = 0;
    }
  }
}
else
{
  if (lpcnt > 9)
  {
    lpcnt = 0;
    --timerSeconds;
    showCountdown();

    if (timerSeconds <= 0)
    {
      currentState = 1;
      relayStatus(false);
      displayCodeEntryScreen();
      showEnteredTime();
    }
  }
  else
  {
    relayStatus(true);
  }
}

++lpcnt;
delay(100);
}
}
SensorInfrarojoA();
sensorpir();
}

```

**Código 31:** Control automático del sistema.

**Fuente:** Autor utilizando Recortes de Windows 7.

Luego que active el modo manual el usuario interviene para poder hacer algunas manipulaciones de ella.

```
else if (mode == 'm')
{
  if (in == 'C'){
    digitalWrite(ledhd, LOW);
    digitalWrite(VALVULA1, VALVULA_ON);
  }
  else(in == 'c'){
    digitalWrite(ledhd, HIGH);
    digitalWrite(VALVULA1, VALVULA_OFF);
  }

  if (in == 'B'){
    seguro.write(ABRIR);
    //stepper.step(33);
    pirState = HIGH;
  }
  else if(in == 'b'){
    seguro.write(CERRAR);
    //stepper.step(-33);
    pirState = LOW;
  }

  if (in == 'D'){
    digitalWrite(VALVULA2, VALVULA_ON);
  }
  else if(in == 'd'){
    digitalWrite(VALVULA2, VALVULA_OFF);
  }
  if (in == 'D'){
    digitalWrite(VALVULA3, VALVULA_ON);
  }
  else if(in == 'p'){
    digitalWrite(VALVULA3, VALVULA_OFF);
  }
}
}
```

**Código 32:** Control manual del sistema.

**Fuente:** Autor utilizando Recortes de Windows 7.

### Diseño de la placa del circuito

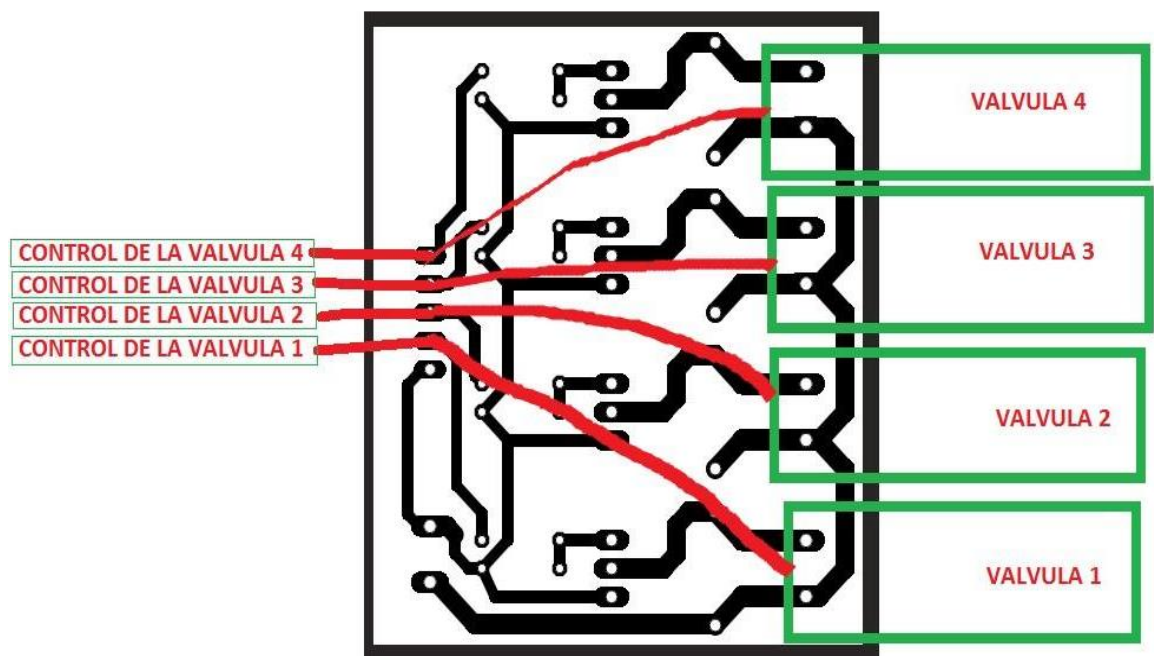
- Se adquirió una placa virgen para el quemado de la placa donde está diseñado el circuito del sistema esto externo a la placa Arduino.



**Figura 25:** *Placa virgen.*

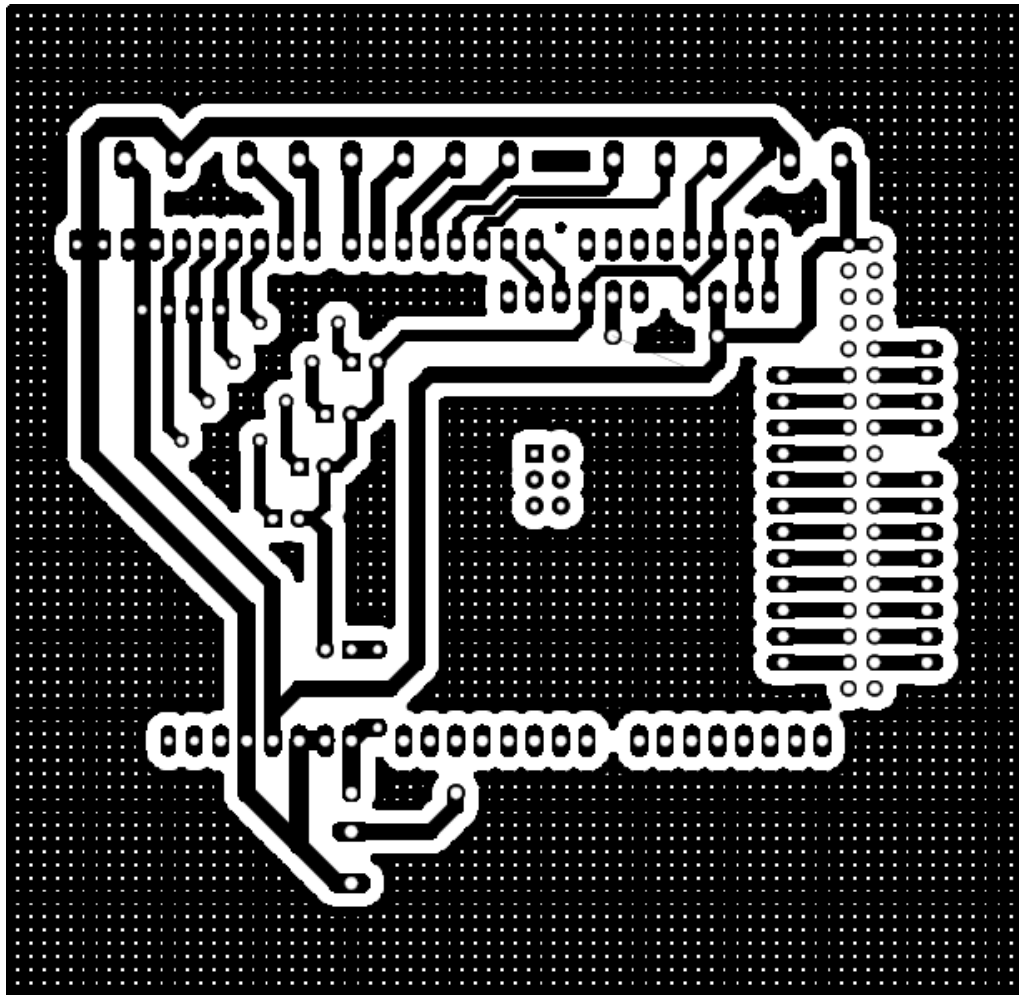
**Fuente:** *Captura de imagen.*

- Se Escogió una la placa virgen con el fin de obtener más brillo, la placa no tiene que tener ninguna mancha para su uso. Se procede a dibujar el circuito del control de válvulas, con el programa Eagle.



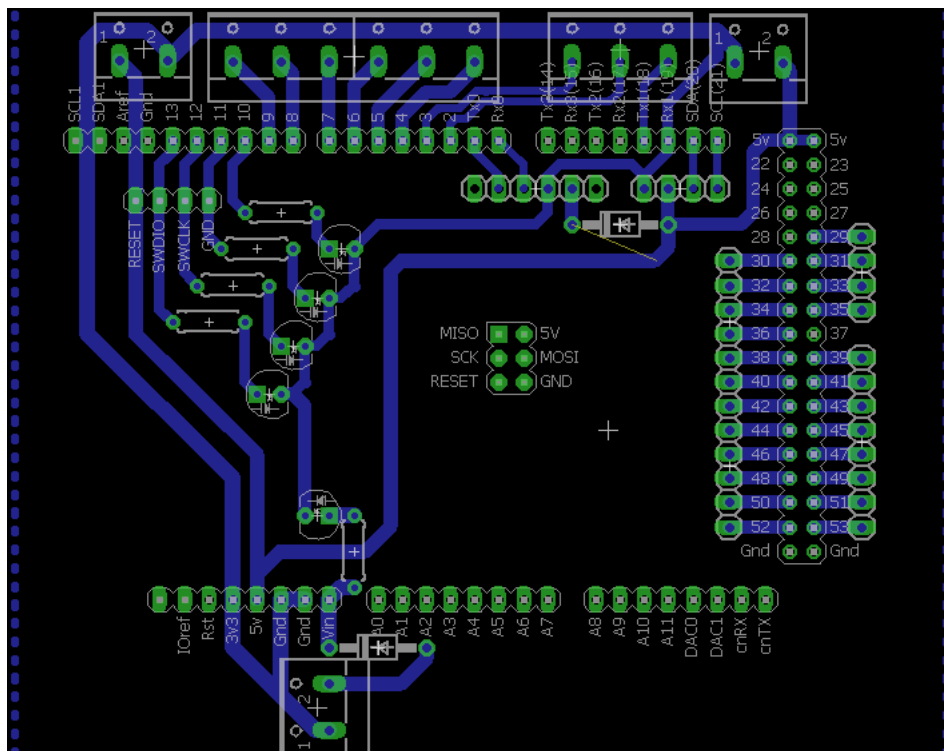
**Figura 26:** *Circuito de Válvulas Solenoide.*

**Fuente:** *Elaboración con Eagle.*



**Figura 27:** Circuito control de sensores y válvulas.

Fuente: Elaboración con Eagle.



**Figura 28:** *Circuito control de sensores y válvulas.*

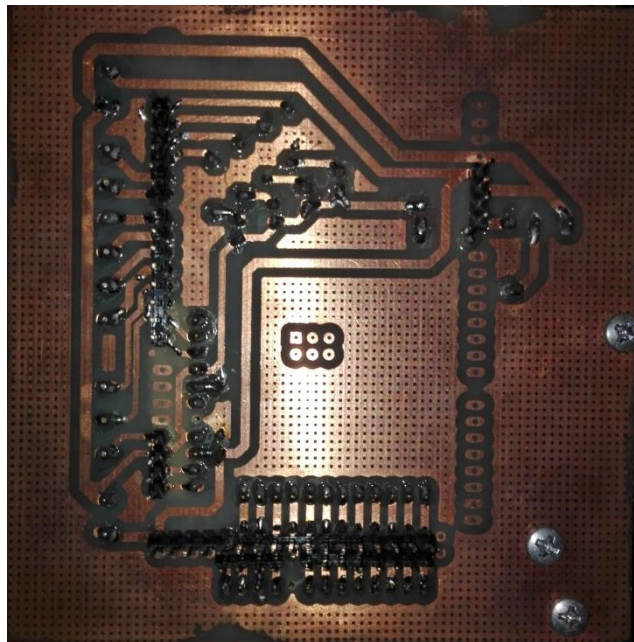
**Fuente:** *Elaboración con Eagle.*

- Como consiguiente se imprimió el diseño del circuito, con estas placas impresas se redujo el cableado o puentes.



**Figura 29:** *Diseño de la placa quemada.*

**Fuente:** *Elaboración propia.*



**Figura 30:** *Diseño de la placa quemada.*

**Fuente:** *Elaboración propia.*

#### **4.2.5. FASE DE PRUEBAS DEL SISTEMA**

Realizaremos la prueba general del sistema desarrollado desde la prueba de verificación unitaria, test de integración y test de campo estas pruebas nos ayudaron a eliminar errores en el sistema desarrollado.

#### **4.2.5.1. VERIFICACIÓN UNITARIA**

Se realizó las pruebas del Hardware y Software del sistema, en el cual se pudo detectar el error en los sensores infrarrojo y movimiento, los cuales no tenían una exactitud en la detección, en ese sentido se tuvo que trabajar mucho para resolver dicho problema analizando en ambos campos tanto código como el funcionamiento del Hardware, luego de una revisión exhaustiva se llega a la conclusión de que el problema era el posicionamiento del sensor se verifica y posteriormente se valida el error detectado, ambos sensores trabajaron de forma satisfactoria en ambos domicilios. En caso del resto de los componentes no se tuvo ningún inconveniente la lógica que se utilizó para la programación respondió de manera muy grata.

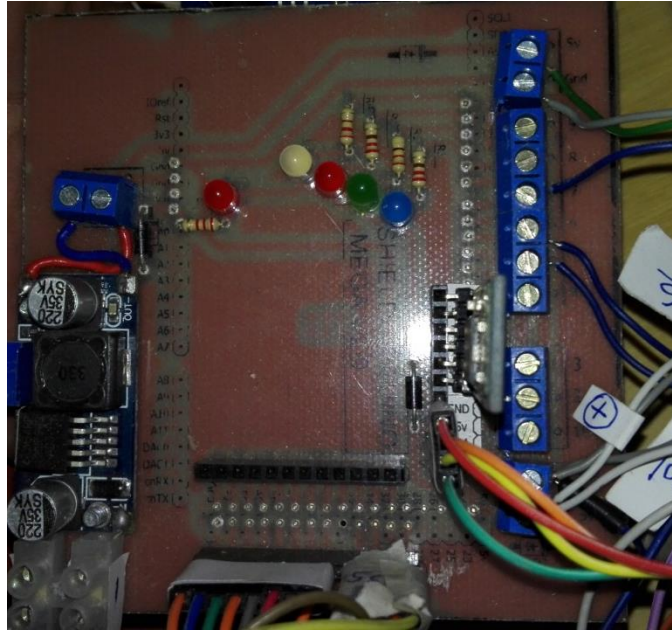
#### **4.2.5.2. TEST DE INTEGRACIÓN**

Se integró todo el sistema que está compuesto (Hardware, Software) esto en la prueba física previa instalación en los domicilios, el emparejamiento del móvil y el modulo Bluetooth fueron satisfactorias como las instrucciones enviadas desde el móvil a cada uno de los sensores, uno de los inconvenientes que se encontró fue el almacenamiento de datos este se solucionó con un almacenamiento temporal en el Arduino, también se verifico y se validó dicho error haciendo el cambio en el código del Arduino como resultado se tuvo excelentes resultados.

#### **INSTALACIÓN PARA LA VERIFICACIÓN DE HW Y SW**

Para la prueba de integración se procedió a la instalación del sistema y se probó que funciona HW Y SW.

1. Se realizo la verificación de los circuitos y el cableado a cada punto (Inodoro, Ducha y Lava mano). Como resultado se validó en cada una de estas etapas.



**Figura 31:** Verificación y validación circuito.

**Fuente:** Elaboración propia.

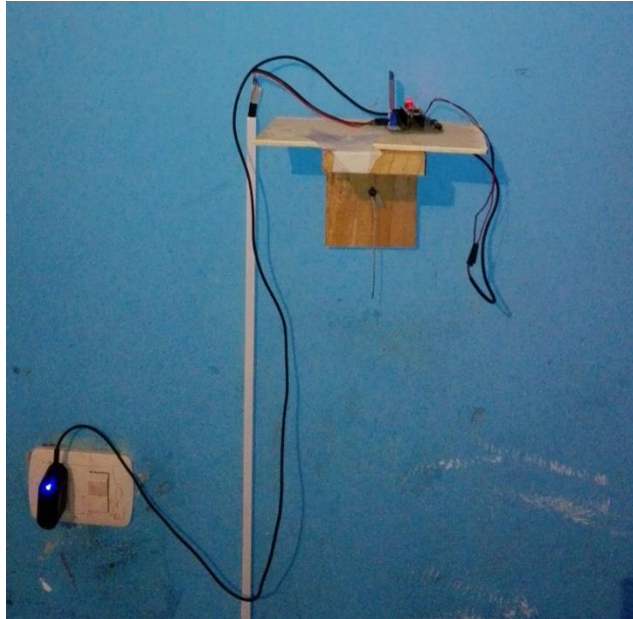
2. Se realizó la verificación del sensor de flujo de agua que este en un correcto funcionamiento, Como resultado se validó que el sensor no tuvo ningún problema en el funcionamiento.



**Figura 32:** Verificación y validación sensor de flujo.

**Fuente:** Elaboración propia.

3. Se realizó las pruebas del sensor de flujo de agua con el modulo Bluetooth que incorpora la placa del Arduino nano.



**Figura 33:** Verificación y validación circuito.

**Fuente:** Elaboración propia.



**Figura 34:** Verificación y validación aplicación consumo.

**Fuente:** Elaboración propia.

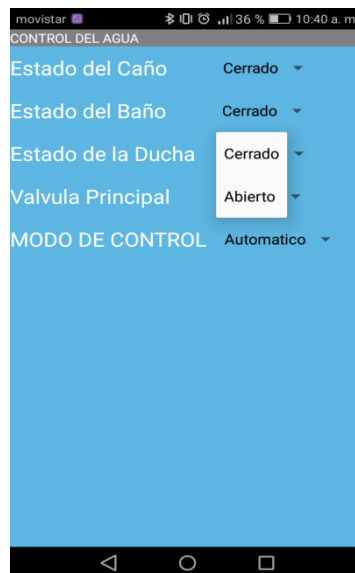
4. Seguidamente hacemos la verificación de la instalación del Inodoro tanto del sensor como del aplicativo.





**Figura 35:** Verificación y validación servo motor inodoro.

**Fuente:** Elaboración propia.



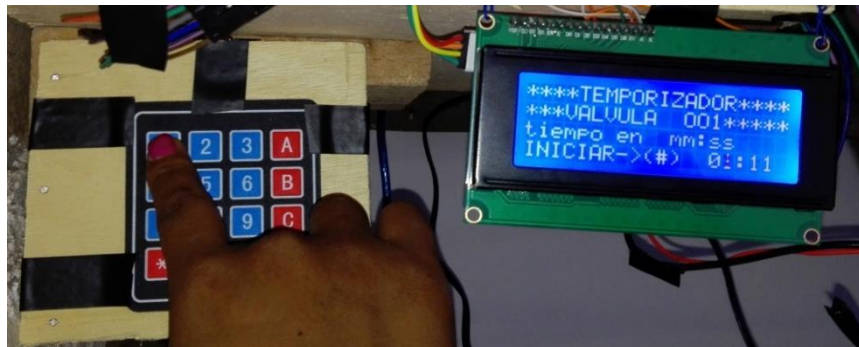
**Figura 36:** Verificación y validación aplicación inodoro.

**Fuente:** Elaboración propia.

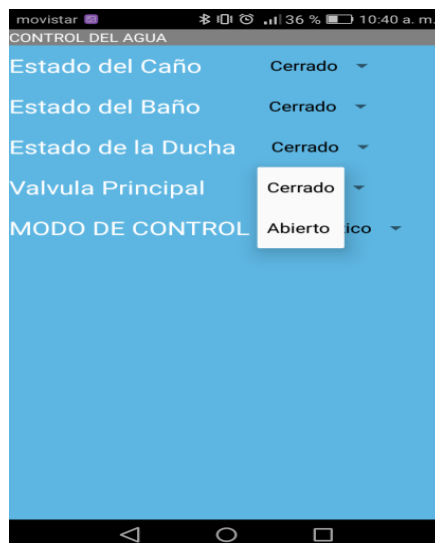
5. A continuación, se hace la verificación de la ducha y se realizó la validación tanto del teclado numérico, LCD y aplicativo móvil.



**Figura 37:** Verificación y validación válvula ducha.  
**Fuente:** Elaboración propia.



**Figura 38:** Verificación y validación teclado numérico y LCD.  
**Fuente:** Elaboración propia.



**Figura 39:** Verificación y validación aplicación inodoro.

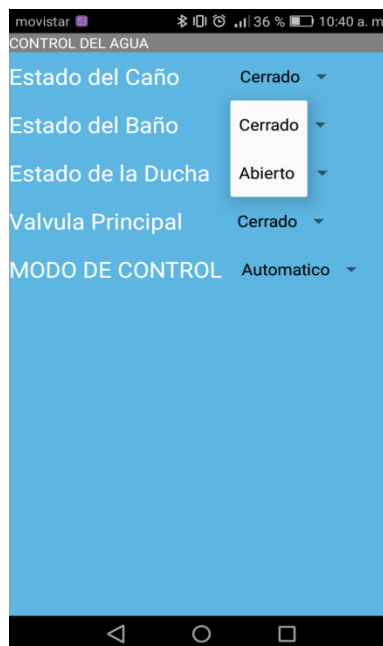
**Fuente:** Elaboración propia.

6. Seguidamente procedimos a verificar el sensor de lava mano juntamente con el aplicativo y validar el funcionamiento.



**Figura 40:** Verificación y validación lava mano.

**Fuente:** Elaboración propia.



**Figura 41:** Verificación y validación aplicativo ducha.

**Fuente:** Elaboración propia.

#### **4.2.5.3. TEST DE CAMPO**

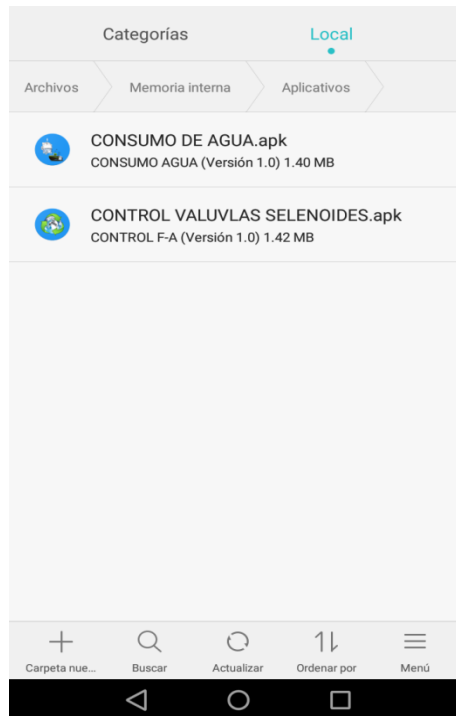
Luego de la verificación en el sistema construido se procedió a recolectar datos del consumo diario, en horas especificadas que fue a las 5:00 a.m – 6:00 a.m, se tomo las lecturas diarias del medidor de los dos predios con la finalidad de comparar y almacenar los datos diarios, el sensor de flujo fue el que cumplió uno de los papeles más importantes en registrar el consumo de todo un mes, el cual nos ayudó a poder contrastar con los datos del medidor de la vivienda. En conclusión, toda la información almacenada en la hoja de Excel fueron los que nos ayudaron a concluir la reducción del consumo de agua diario y mensual, cada uno de los componentes del sistema cumplió un papel muy importante para poder conseguir el objetivo planteado.

Para hacer uso del sistema se capacitó a todos los miembros de la familia para que no exista ninguna dificultad al momento de hacer uso de este sistema ecoeficiente, como también se estuvo una constante supervisión en caso que existiera alguna falla o duda por parte de los usuarios.

- **CAPACITACION DE LOS MIEBROS DE LA FAMILIA**

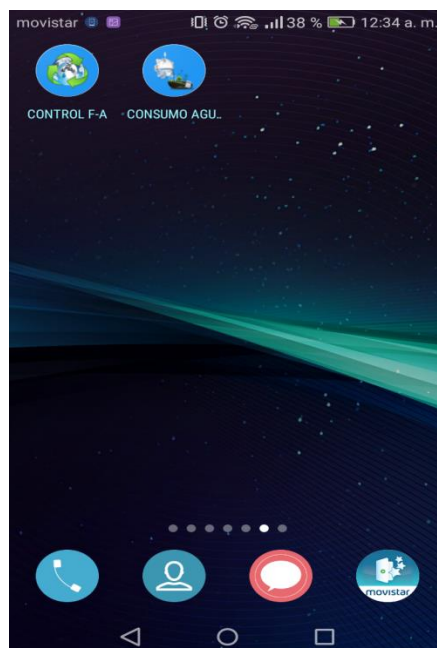
Para que el sistema entre en funcionamiento se le enseñó a cada miembro de la familia a usar el sistema, con la finalidad de que puedan hacer el mejor uso de ella sin ningún problema o dificultad.

Como paso inicial se les enseñó a instalar el aplicativo siguiendo los pasos correspondientes.



**Figura 42:** APK del aplicativo diseñado.

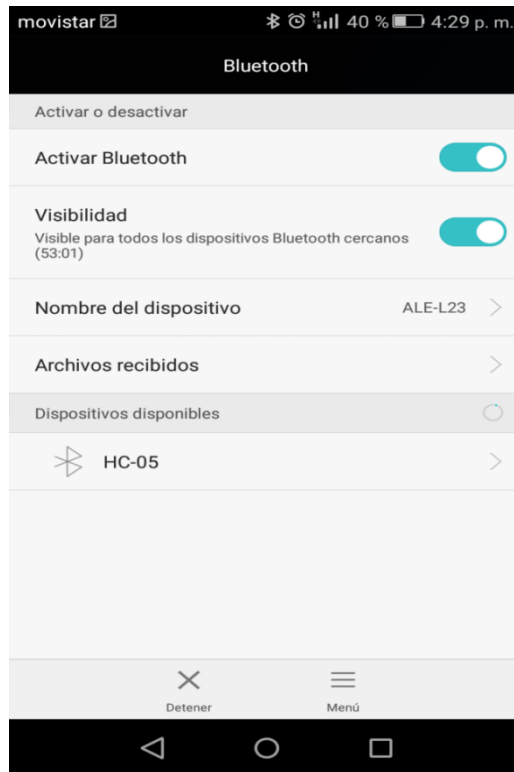
**Fuente:** Captura Android.



**Figura 43:** Iconos en escritorio de móvil.

**Fuente:** Recorte de móvil.

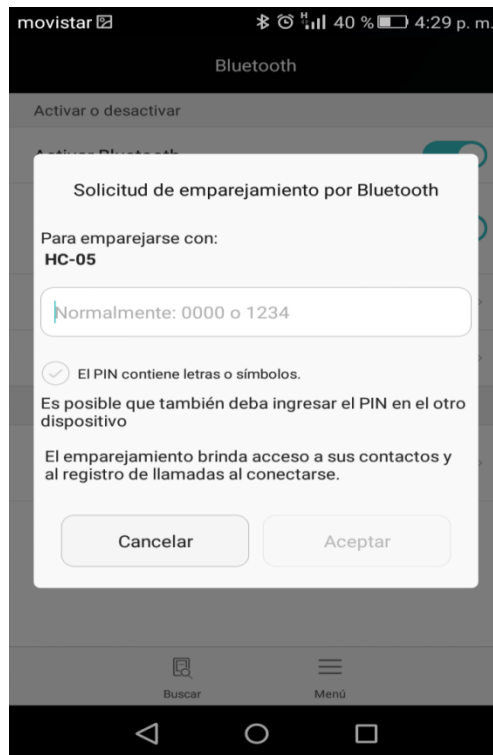
Una vez instalada la aplicación se procede a emparejar el dispositivo con fin de poder intercambiar información con los sensores.



**Figura 44:** *Emparejamiento de dispositivos.*

**Fuente:** *Captura Android.*

Seguidamente nos pedirá la contraseña del 1234 del módulo Bluetooth es con la finalidad de autenticar y tener un enlace entre ambos dispositivos.



**Figura 45:** *Contraseña para emparejar.*

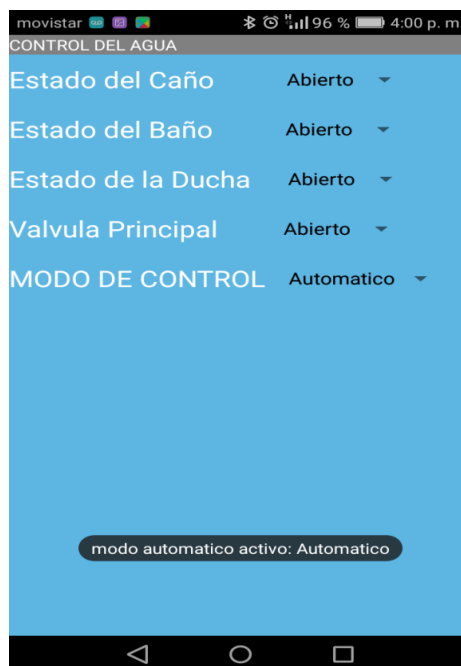
**Fuente:** *Captura Android.*

Seguidamente se procedió a acceder al panel de presentación del dispositivo móvil el cual nos brindó la facilidad de escoger el modulo Bluetooth.



**Figura 46:** *Interfaz de presentación.*

**Fuente:** *Captura Android.*



**Figura 47:** *Control de válvulas.*

**Fuente:** *Captura Android.*



**Figura 48:** Información del consumo de agua.

**Fuente:** Captura Android.

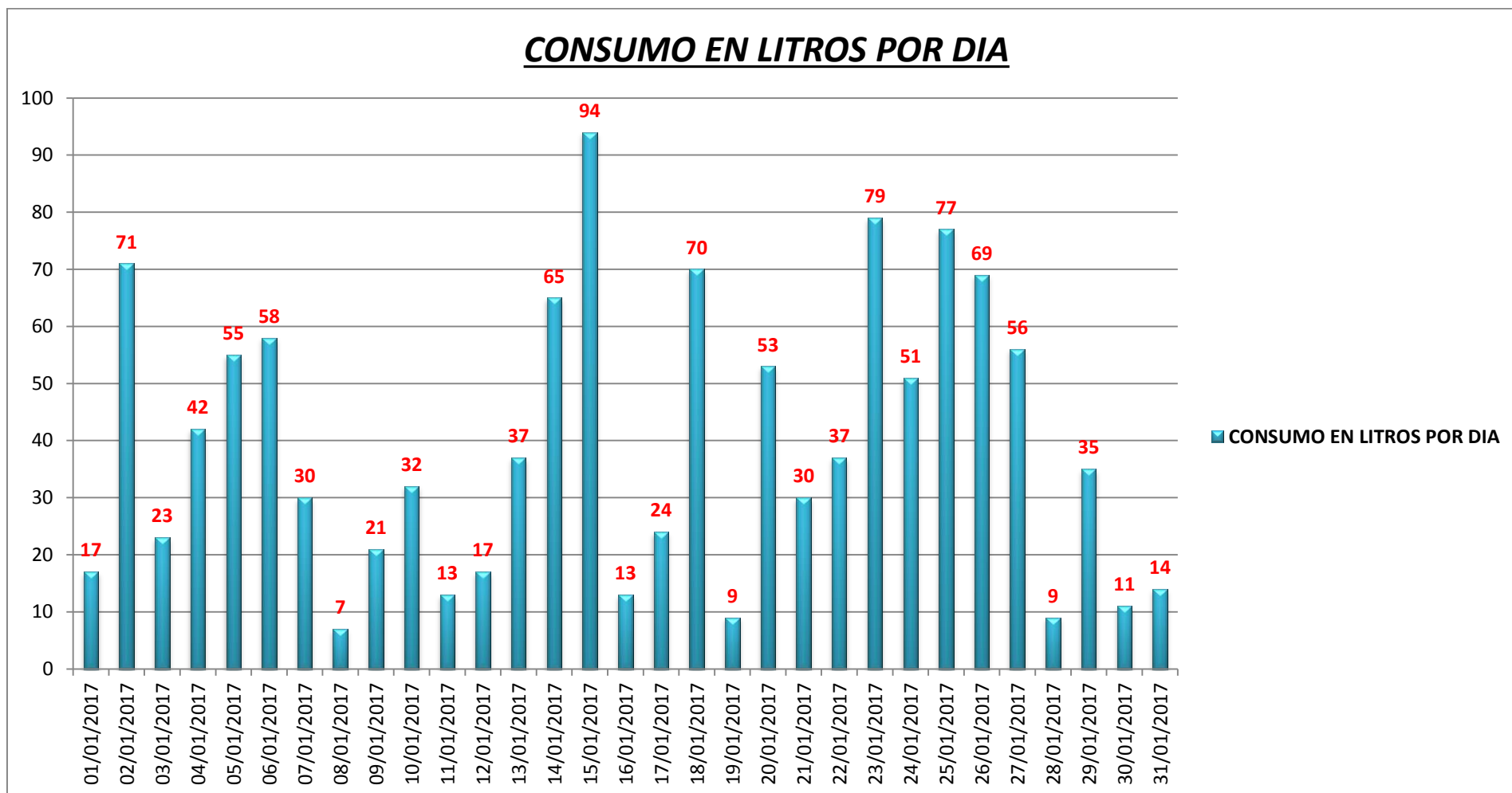
- **RECOLECIÓN DE DATOS POR DÍAS**

**VIVIENDA LINO CENTENO ROJAS**

Vivienda ubicada en Jr. Bélgica 565 – Talavera (Masuraccra). La vivienda donde se realizó las pruebas durante tres meses iniciando en **Enero - Marzo** con el medidor N° 585020 esto con el fin de poder recolectar datos del consumo de agua potable en dicha vivienda y comprobar que el sistema es funcional.



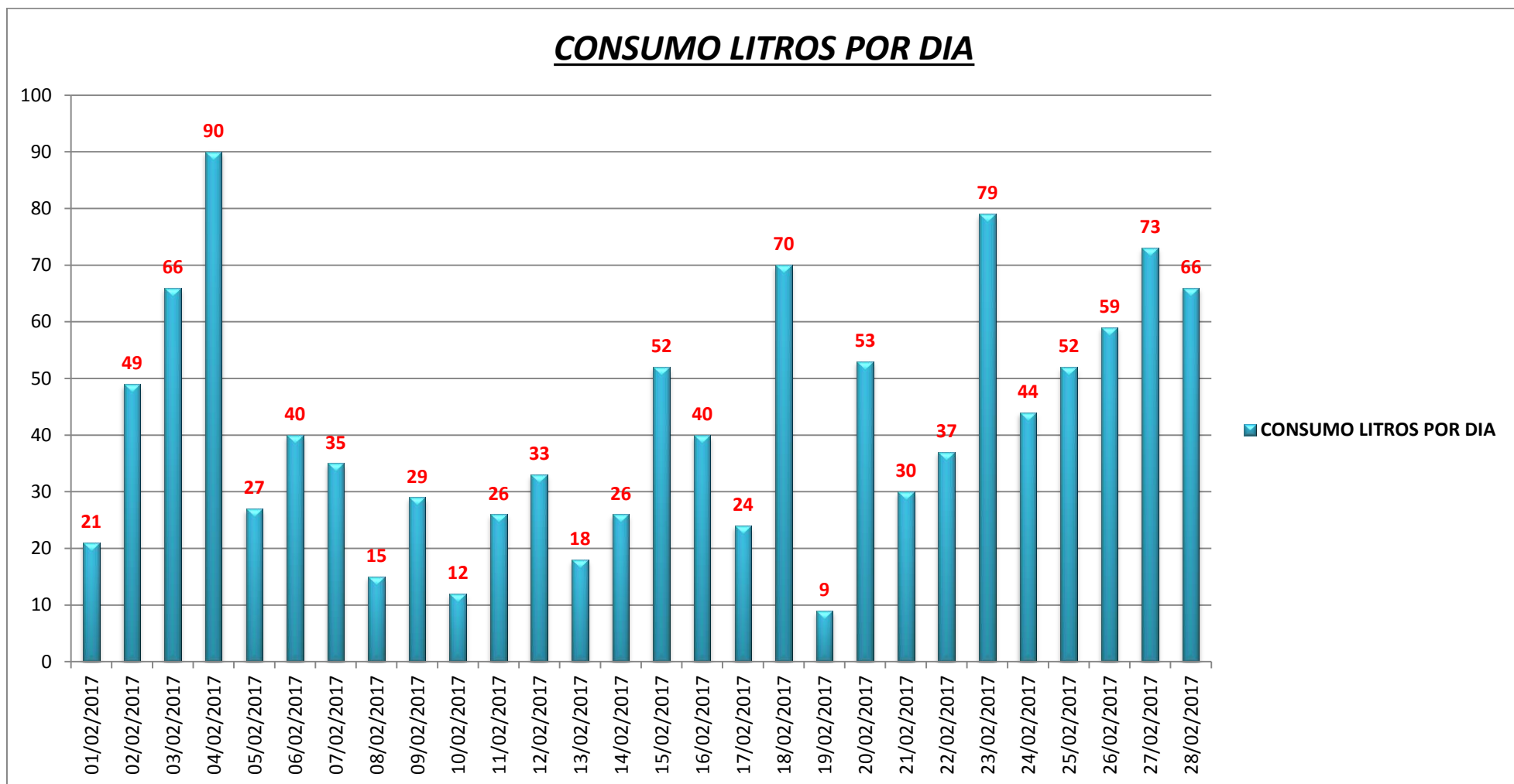
## ENERO



**Figura 49:** Grafico comparativos por días.

**Fuente:** Elaboración en Excel.

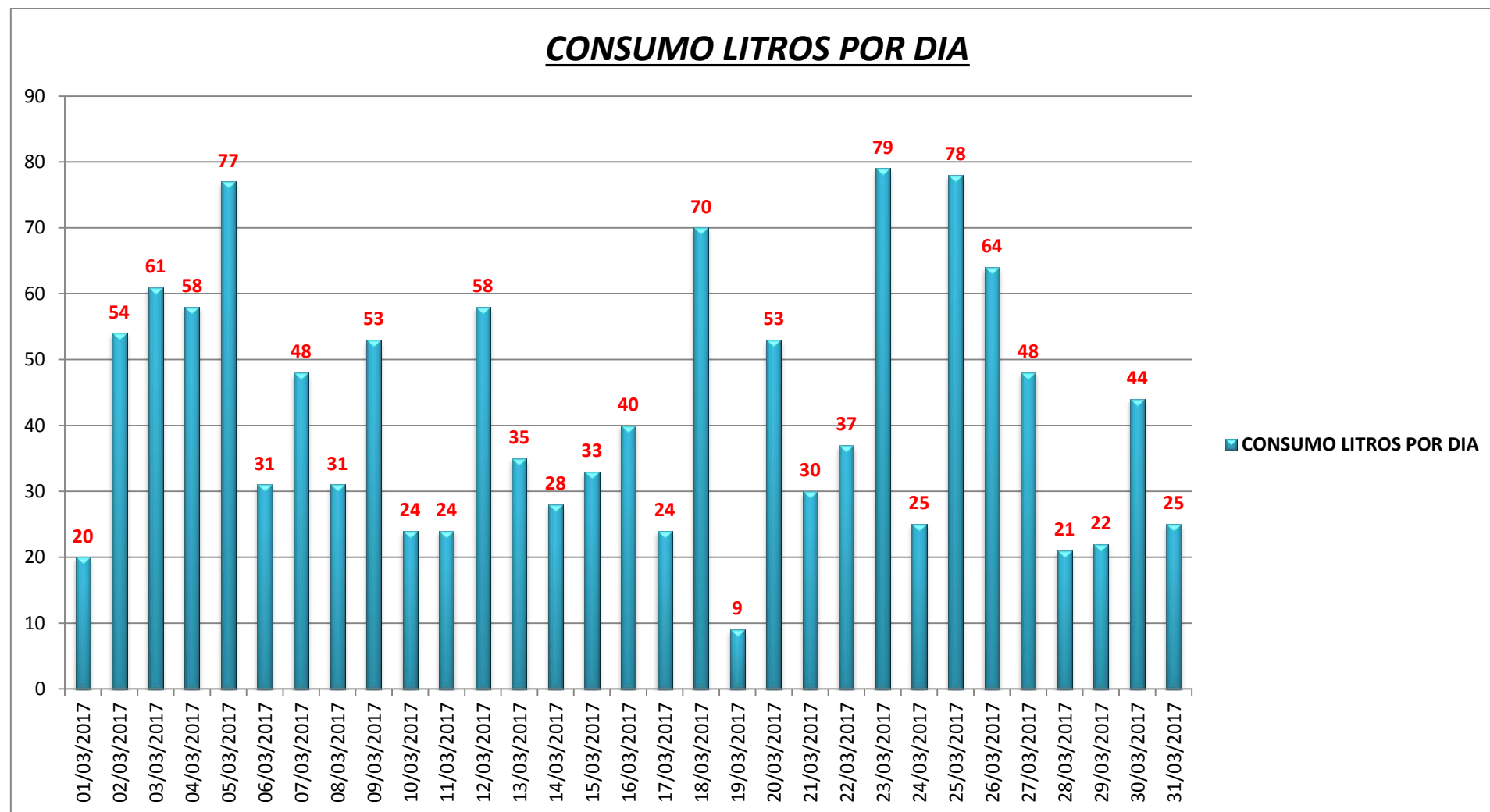
## FEBRERO



**Figura 50:** Grafico comparativos por días.

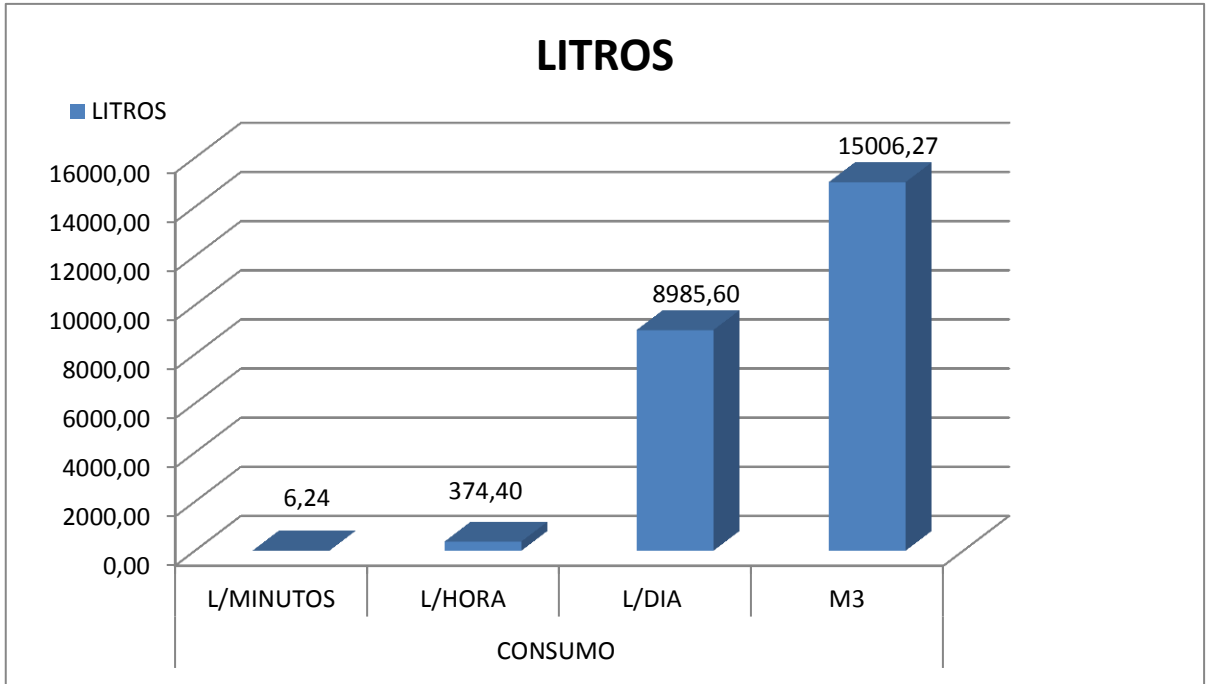
**Fuente:** Elaboración en Excel.

## MARZO



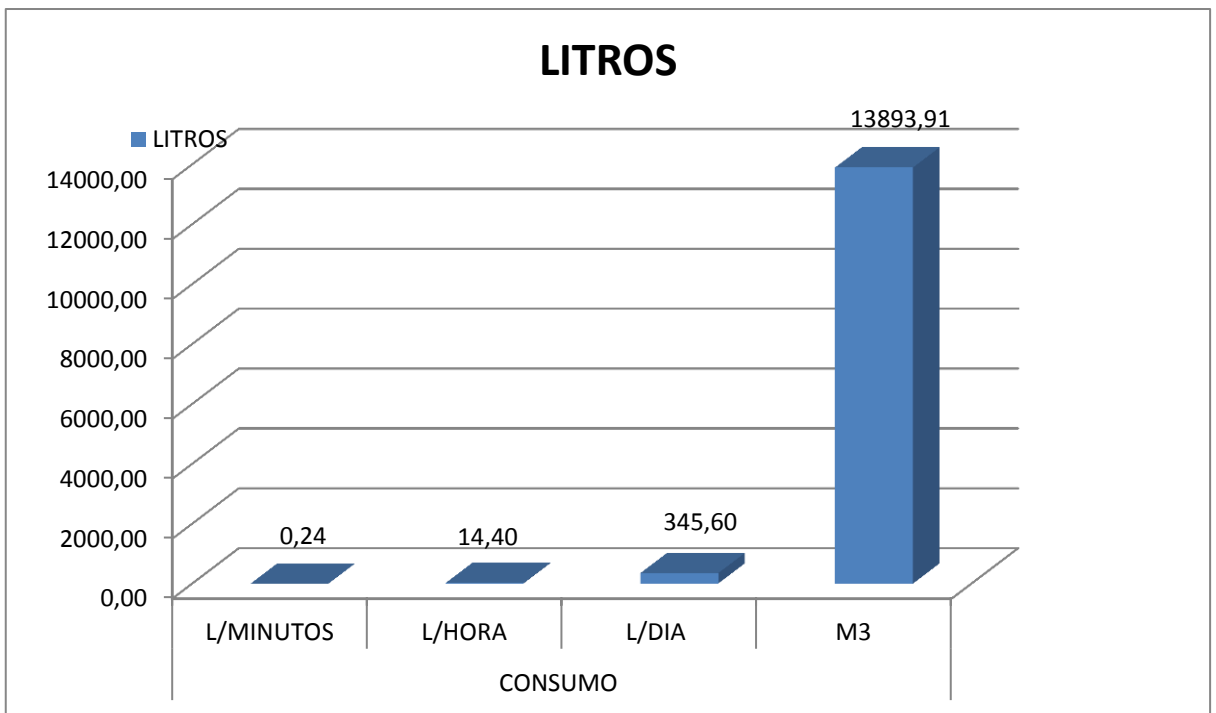
**Figura 51:** Grafico comparativos por días.

**Fuente:** Elaboración en Excel.



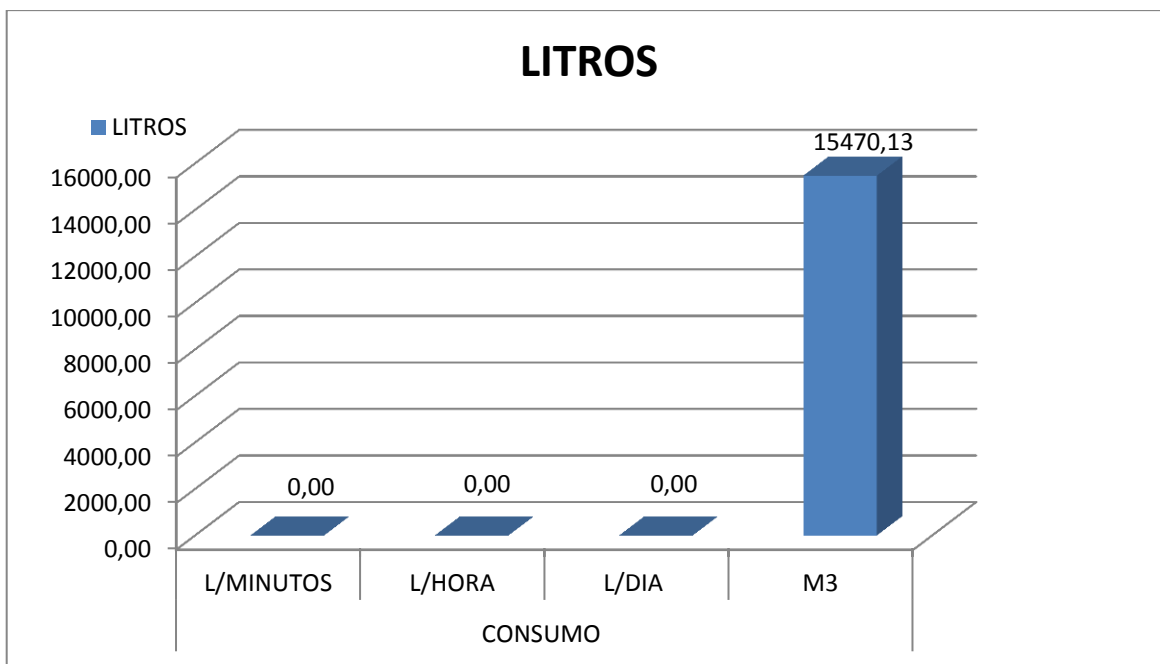
**Figura 52:** Consumo Enero sensor efecto Hall.

**Fuente:** Datos del sensor.



**Figura 53:** Consumo Febrero sensor efecto Hall.

**Fuente:** Datos del sensor.



**Figura 54:** Consumo Marzo sensor efecto Hall.

**Fuente:** Datos del sensor.

**VIVIENDA ADA VANESA ALCCA HUILLCAHUAYA**

Vivienda ubicada en Jr. Masusaccra – Talavera. La vivienda donde se realizó las pruebas durante tres meses iniciando en **Enero - Marzo** con el medidor N° 5584 esto con el fin de poder recolectar datos del consumo de agua potable en dicha vivienda y comprobar que el sistema es funcional.

## FEBRERO

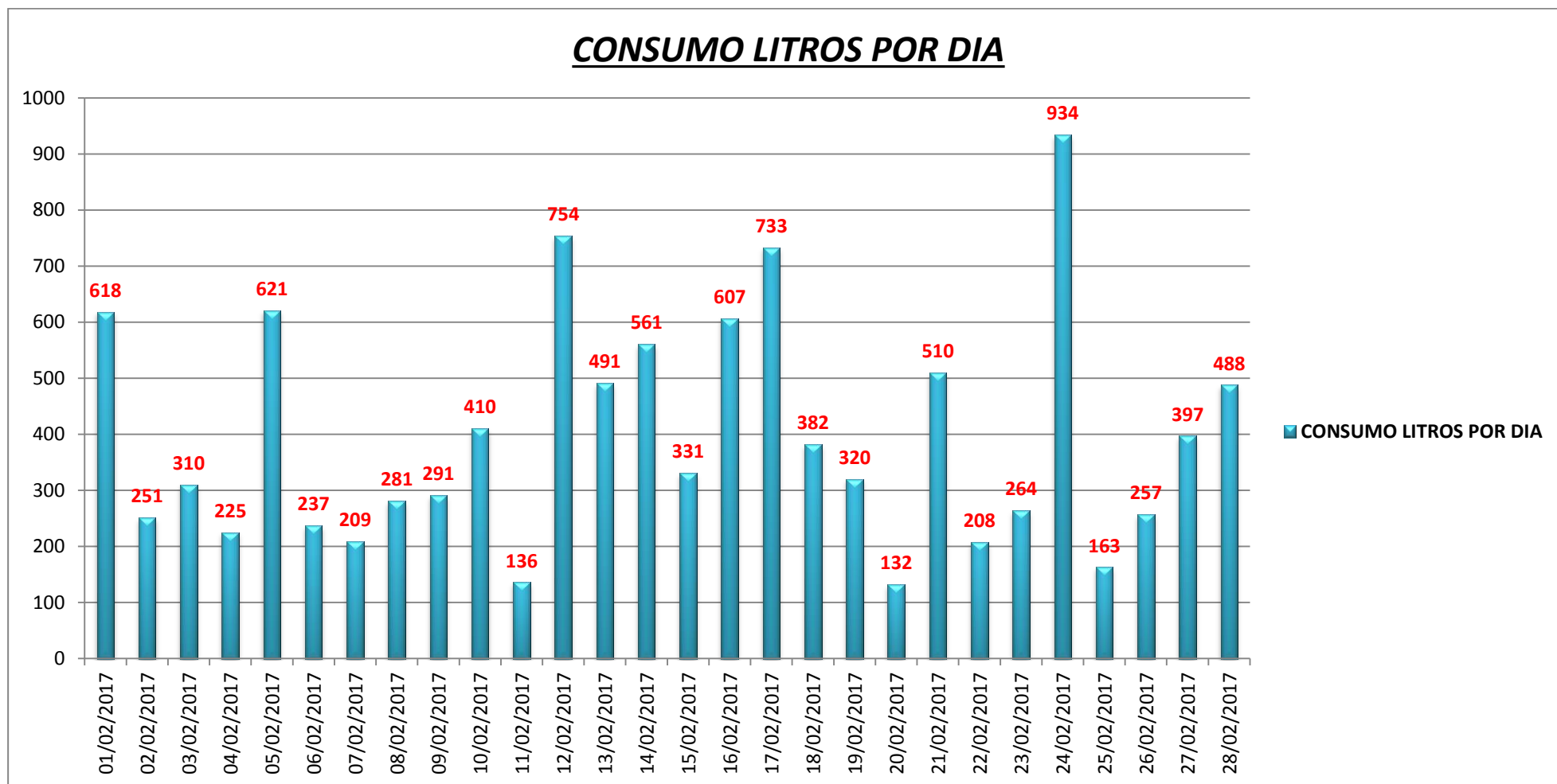


Figura 55: Grafico comparativos por días.

Fuente: Elaboración en Excel.

MARZO

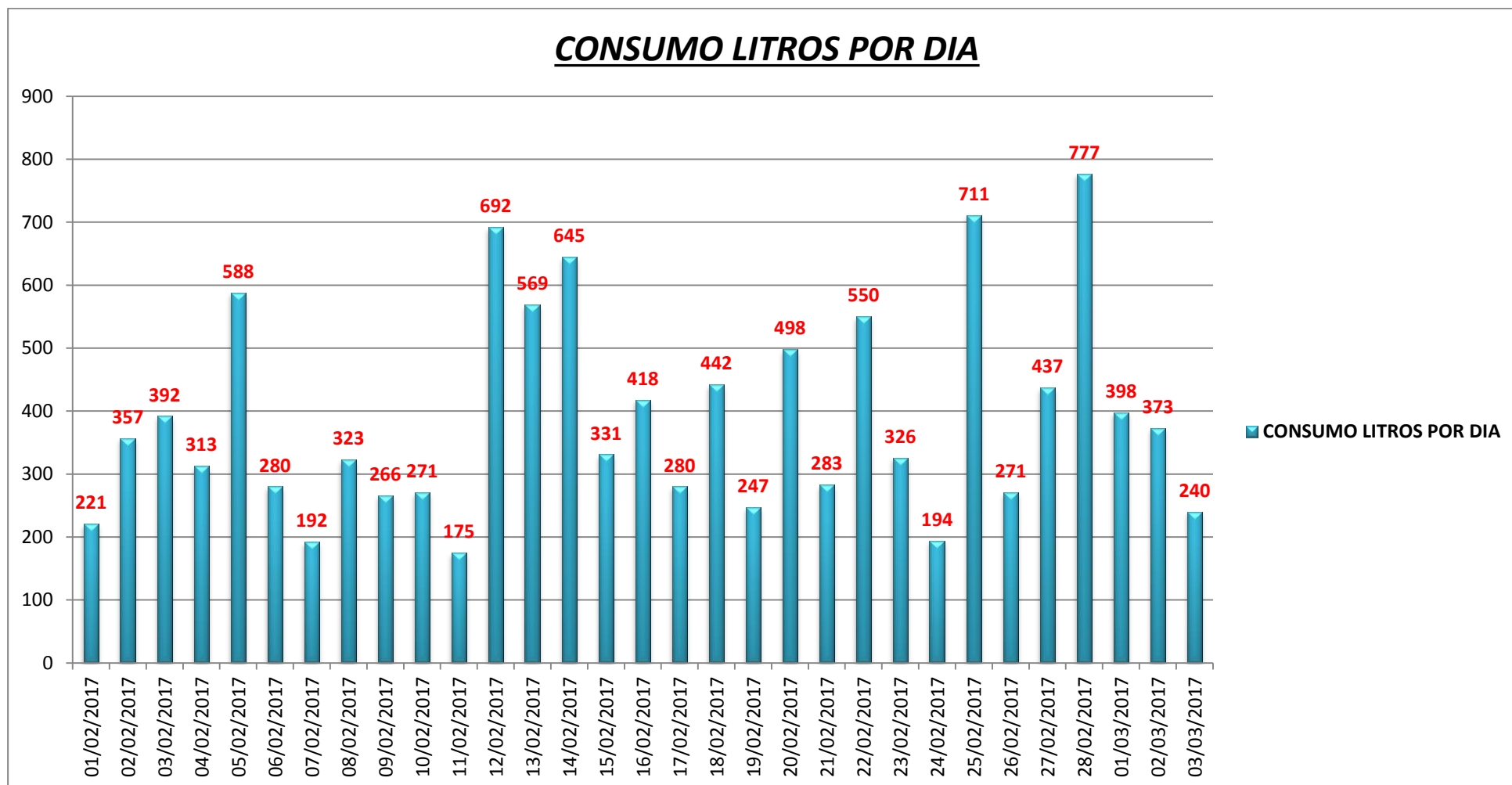
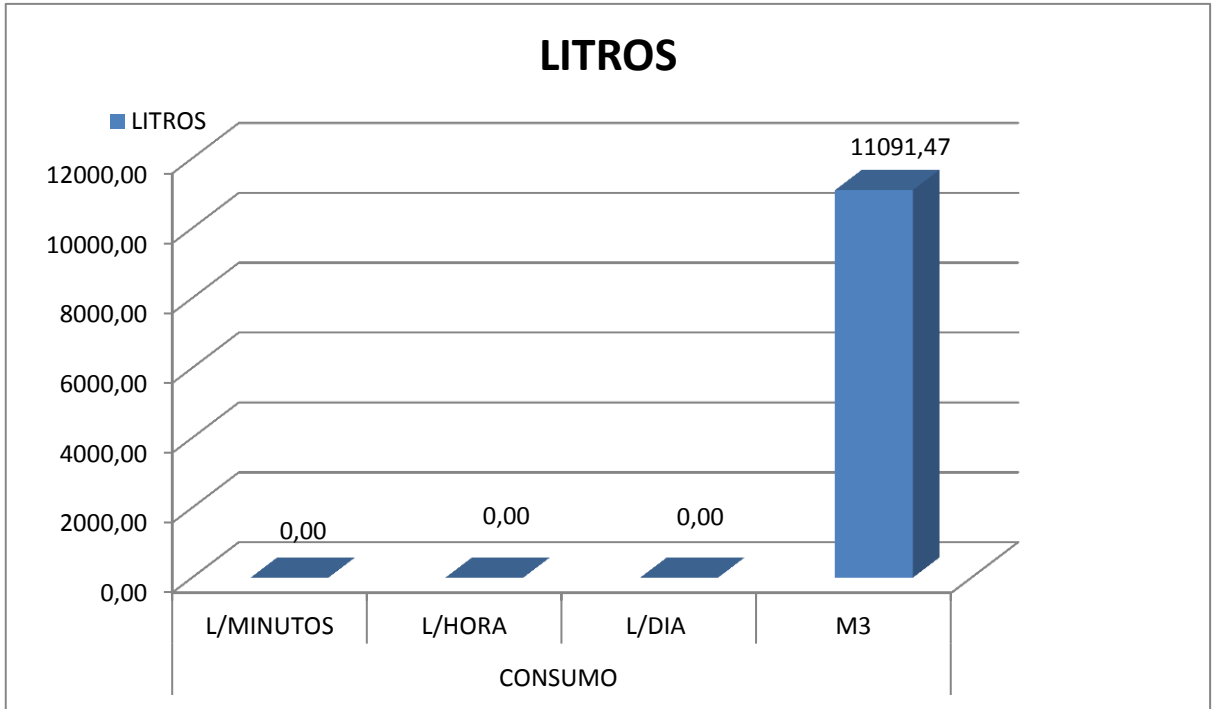


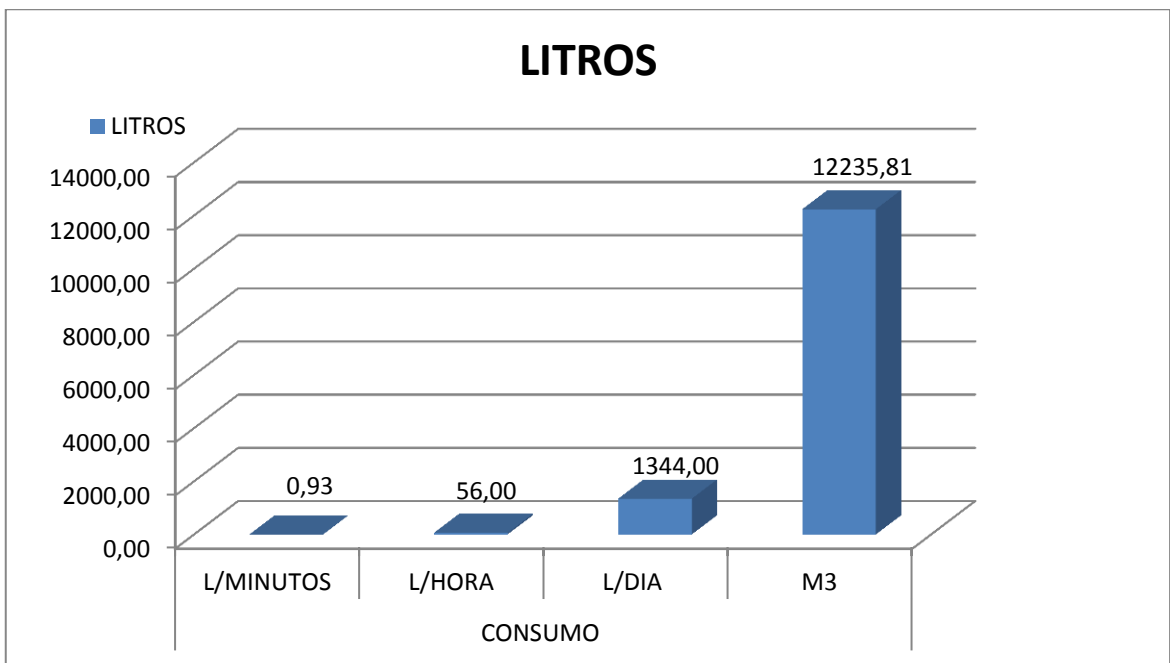
Figura 56: Grafico comparativos por días.

Fuente: Elaboración en Excel.



**Figura 57:** Consumo Febrero sensor efecto Hall.

**Fuente:** Datos del sensor.



**Figura 58:** Consumo Marzo sensor efecto Hall.

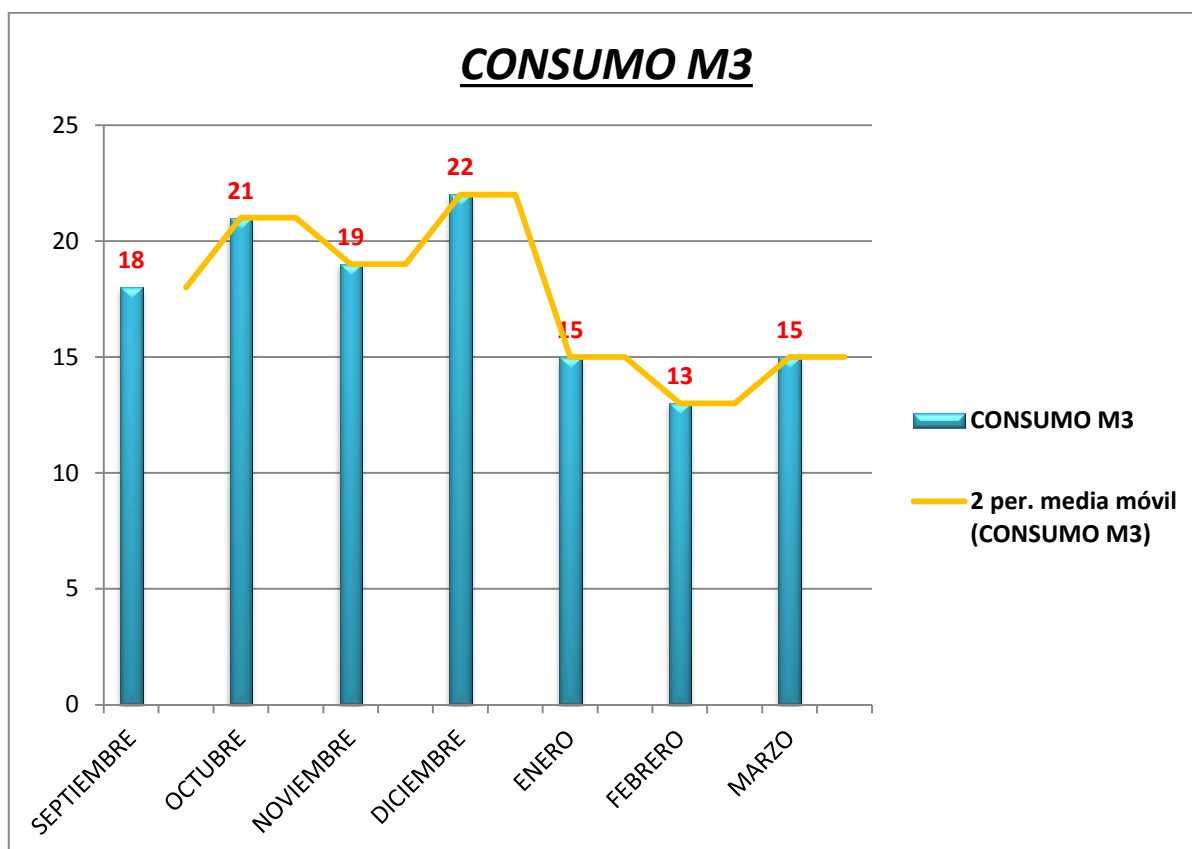
**Fuente:** Datos del sensor.



- **COMPARACION DE DATOS DE LOS MESES**  
**LINO CENTENO ROJAS**

Después de la prueba por días que se realizó entre los meses de **ENERO Y MARZO** se hace una comparación con meses anteriores para verificar los resultados y dar por visto bueno que el sistema con plataforma Arduino y Android optimiza el consumo del agua en la vivienda.

**SETIEMBRE – MARZO**



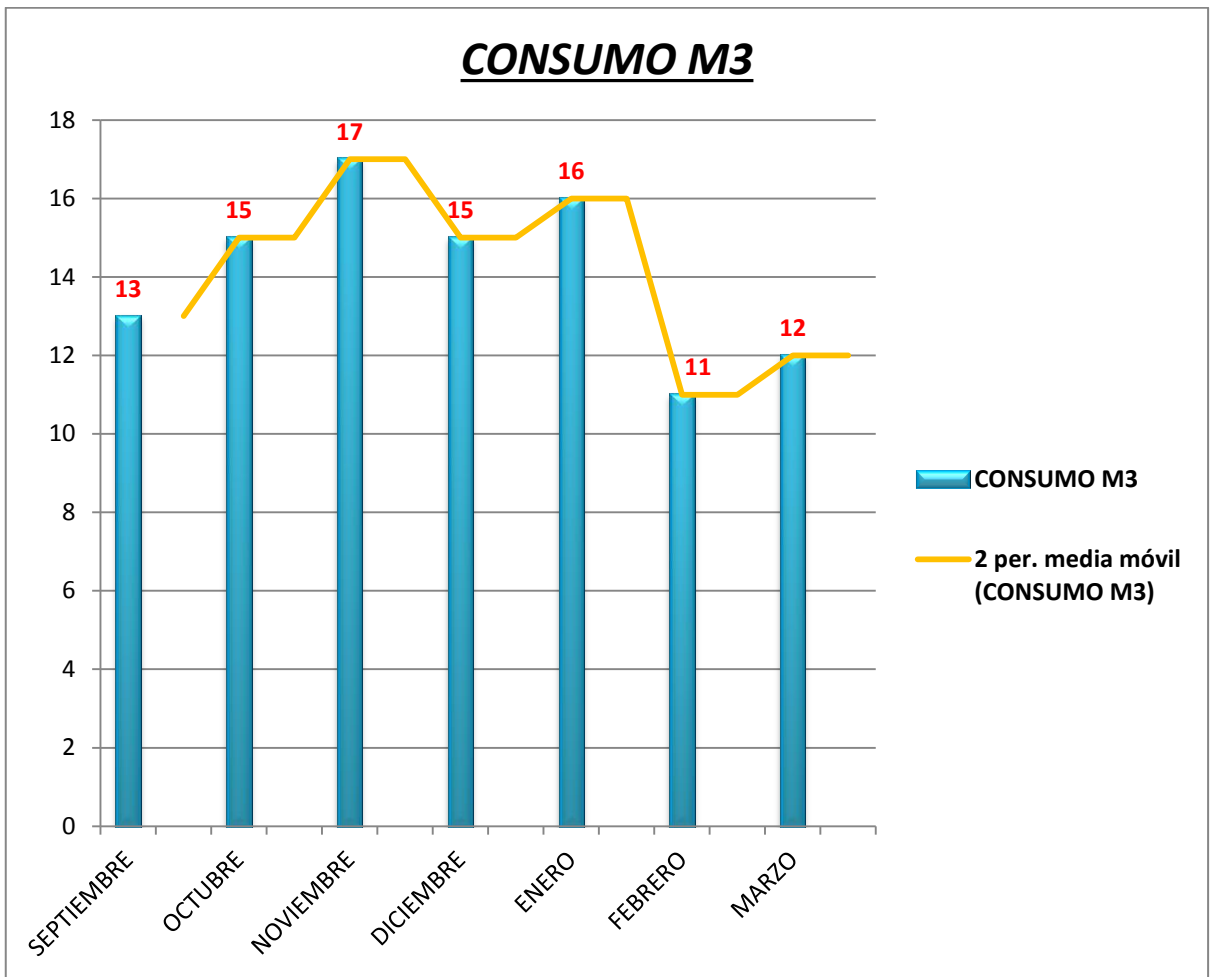
**Figura 59:** Grafico comparativos por meses.

**Fuente:** Empresa y datos pruebas.

**ADA VANESA ALCCA HUILLCAHUAYA**

Después de la prueba por días que se realizó entre los meses de **ENERO Y MARZO** se hace una comparación con meses anteriores para verificar los resultados y dar por visto bueno que el sistema con plataforma Arduino y Android optimiza el consumo del agua en la vivienda.

## SEPTIEMBRE - MARZO



**Figura 60:** Grafico comparativos por meses.

**Fuente:** Empresa y datos recolectados.

### 4.2.5.4. ENTREVISTA USUARIOS

Luego de la instalación y recoger los datos del consumo de agua se realizó una prueba de entrevista, que tiene como finalidad la usabilidad de la aplicación.

Se realizó un encuentro a 6 y 3 miembros por vivienda, donde se instaló el sistema y funcionó por tres meses. Para poder realizar esta prueba se le entregó a cada uno de los aplicativos del consumo como de control de válvulas.

## **INSTRUCCIONES DEL USO DEL SISTEMA ECOEFICIENTE CON PLATAFORMA ARDUINO Y ANDROID**

1. Ubique en el lugar adecuado el circuito del sistema Arduino.
2. Cerciórese de que el circuito no este expenso a la humedad.
3. Distribuya los cables por a cada uno de los puntos de acceso (Ducha, Lava mano y Inodoro).
4. Conecte la válvula solenoide a la Ducha.
5. Brinde los niveles de seguridad a la válvula con cinta teflón.
6. Seguidamente ubique el teclado numérico en una parte de la ducha para su fácil manipulación.
7. Localiza la pantalla LCD en un lugar visible para dar lectura del tiempo.
8. Instale la válvula solenoide en el lava manos asegurando con cinta teflón.
9. Ubique el sensor de infrarrojo en un lugar donde esté al alcance de la mano del usuario de recomendación pegado entre la perilla del caño.
10. Colocar el servo motor para el abierto y cerrado del tanque.
11. Ubicar el sensor de presencia junto al Inodoro.
12. Colocamos en la llave principal el sensor de flujo de agua.
13. Conectamos con el Arduino nano.
14. Seguidamente instalamos el aplicativo.
15. Ubicamos el archivo. APK tanto de control de las válvulas como la apk de información del consumo de agua.
16. Instalar el APK en el dispositivo móvil ya se Tablet o Smarfont en este caso primo se tiene que desbloquear en ambos dispositivos APLICACIONES DE ORIGEN DESCONOCIDO.
17. Emparejamos el Bluetooth del dispositivo y el Arduino.
18. Autenticamos con la contraseña por defecto 1234.
19. Iniciamos la aplicación en el dispositivo.
20. Seleccionamos el nombre del módulo Bluetooth.
21. Controlamos el abierto y el cerrado del sistema con cada botón de cada uno de los puntos.
22. Para salir pulsar el botón atrás.
23. Seguidamente para el consumo nos conectamos con el aplicativo de consumo.
24. Iniciamos la aplicación pulsamos en el Bluetooth correspondiente.
25. Accedemos a la información de lectura del sensor de flujo de agua.

**Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.**

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

**Tabla 16:** Encuesta del funcionamiento del sistema.

<b>Muy difícil</b>	<b>Difícil</b>	<b>Más o menos</b>	<b>Fácil</b>	<b>Muy fácil</b>
1	2	3	4	5

**Pregunta N° 1:**

- ¿Cómo fue navegar por toda las interfaces del aplicación?\_\_\_\_\_

**Pregunta N° 2:**

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema?\_\_\_\_\_

**Pregunta N° 3:**

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual?\_\_\_\_\_

**Pregunta N° 4:**

- ¿Cómo fue para usted interactuar con cada uno de los sensores?\_\_\_\_\_

**Pregunta N° 5:**

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema?\_\_\_\_\_

**Pregunta N° 6:**

- ¿Cómo le pareció el sistema en general?\_\_\_\_\_

<b>Malo</b>	<b>Buena</b>	<b>Muy buena</b>
1	2	3

**Pregunta N° 7:**

- ¿Cómo respondió la aplicación para usted?\_\_\_\_\_

**Pregunta N° 8:**

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria?\_\_\_\_\_

**Pregunta N° 9:**

- ¿Qué cree usted de la reducción del agua?\_\_\_\_\_

**Pregunta N° 10:**

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable?\_\_\_\_\_

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI\_\_\_\_\_ NO\_\_\_\_\_

**Fuente:** *Elaboración propia.*

## RESULTADOS

La prueba de usabilidad del aplicativo y el sistema Arduino fue realizada en los tres meses de prueba desde la instalación del sistema ecoeficiente, la encuesta fue realizada con los miembros de la familia que conformaron entre 6 y 3 entre 8 y 42 años.

La calificación por cada aspa es de **16.67 %** esto significa por las 6 personas hace un total de **100 %** por calificación.

### VIVIENDA DE 6 PERSONAS

**Tabla 17:** Datos pregunta N° 1 - N° 7.

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil		Malo	Bueno	Muy bueno
1		X						X	
2					X			X	
3				X					X
4					X			X	
5					X				X
6				X					X
		<b>16.66%</b>		<b>18.32 %</b>	<b>49.67%</b>			<b>49.67%</b>	<b>49.67%</b>

**Fuente:** Elaboración propia.

**Tabla 18:** Datos pregunta N° 2 – N° 8.

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil		Malo	Bueno	Muy bueno
1				X					X
2				X					X
3				X					X
4				X					X
5					X				X
6				X					X
				<b>83.35 %</b>	<b>16.67 %</b>				<b>100 %</b>

**Fuente:** Elaboración propia.

**Tabla 19:** Datos pregunta N° 3 – N° 9.

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil		Malo	Bueno	Muy bueno
---------	-------------	---------	-------------	-------	-----------	--	------	-------	-----------

1					X				X
2					X				X
3		X							X
4			X					X	
5					X			X	
6					X				X
		16.67%	16.67%		66.68%			33.34%	66.68%

Fuente: *Elaboración propia.*

**Tabla 20:** Datos pregunta N° 4 – N° 10.

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil		Malo	Bueno	Muy bueno
1				X					X
2				X					X
3				X				X	
4				X					X
5				X				X	
6				X					X
				100%				33.34%	66.68%

Fuente: *Elaboración propia.*

**Tabla 21:** Datos pregunta N° 5.

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1			X		
2				X	
3			X		
4		X			
5		X			
6				X	
		33.34%	33.34%	33.34%	

Fuente: *Elaboración propia.*

**Tabla 22: Datos pregunta N° 6.**

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1			X		
2			X		
3			X		
4				X	
5					X
6					X
			<b>50.01%</b>	<b>16.67%</b>	<b>33.34%</b>

**Fuente:** *Elaboración propia.*

Luego de la encuesta realizada se llegó a la conclusión, que el manejo del sistema ecoeficiente en la vivienda ubicada en Jr. Bélgica 565 – Talavera, los usuarios no tuvieron problemas graves en el uso del sistema tanto con los sensores y el aplicativo móvil, en este sentido se concluye que el sistema es funcional.

### VIVIENDA DE 3 PERSONAS

**Tabla 23: Datos pregunta N° 1 – N° 7.**

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil		Malo	Bueno	Muy bueno
1				X				X	
2				X				X	
3					X			X	
				<b>66.68%</b>	<b>33.34%</b>			<b>100%</b>	

**Fuente:** *Elaboración propia.*

**Tabla 24: Datos pregunta N° 2 – N° 8.**

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil		Malo	Bueno	Muy bueno
1				X					X
2				X					X
3				X					X
				<b>100%</b>					<b>100%</b>

**Fuente:** *Elaboración propia.*

**Tabla 25: Datos pregunta N° 3 – N° 9.**

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil		Malo	Bueno	Muy bueno
1			X						X
2				X					X
3				X					X
			<b>33.34%</b>	<b>66.68%</b>					<b>100%</b>

**Fuente:** *Elaboración propia.*

**Tabla 26: Datos pregunta N° 4 – N° 10.**

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil		Malo	Bueno	Muy bueno
1					X				X
2					X				X
3					X				X
					<b>100%</b>				<b>100%</b>

**Fuente:** *Elaboración propia.*

**Tabla 27: Datos pregunta N° 5.**

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1			X		
2				X	
3					X
			<b>33.34%</b>	<b>33.34%</b>	<b>33.34%</b>

**Fuente:** *Elaboración propia.*

**Tabla 28: Datos pregunta N° 6.**

USUARIO	Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1					X
2					X
3					X
					<b>100%</b>

**Fuente:** *Elaboración propia.*

Luego de la encuesta realizada se llegó a la conclusión, que el manejo del sistema ecoeficiente en la vivienda ubicada en Jr. Masusaccra – Talavera. Los usuarios no tuvieron problemas en el uso del sistema tanto con los sensores y el aplicativo móvil, en este sentido se concluye que el sistema es funcional.



## CONCLUSIONES

- La implementación del sistema ecoeficiente con tecnología Arduino y Android incidió positivamente en la automatización de cada uno de los procesos de control del consumo del agua tanto en la ducha, inodoro y lava manos, cada Hardware utilizado como son los sensores y las válvulas solenoides fueron los principales actores.
- Con el análisis realizado de las diferentes arquitecturas del arduino se llegó a concluir que el arduino Mega es el adecuado para poner realizar conexiones con sensores, válvulas electrónicas y dispositivos móviles.
- Con el desarrollo del aplicativo móvil se pudo realizar el adecuado control de los diferentes puntos de acceso al agua potable de las dos viviendas en donde se implemento el proyecto.
- Con la implantación del sistema ecoeficiente con tecnología Arduino y Android se pudo reducir en un 30% el consumo del agua en el predio de Vanesa Alcca Huillcahuaya, un 20% en el predio de Lino Centeno Rojas.

## RECOMENDACIONES

- Incorporar baterías que ayuden contrarrestar problemas eléctricos, cuando ocurra un corte de energía eléctrica, las baterías entran en funcionamiento y alimentan a todo el sistema.
- Incorporar plataformas de protección para evitar la humedad, para proteger a los circuitos y no exista interferencia.
- Brindarle mayores niveles de seguridad en modulo Bluetooth, para evitar infiltración en el sistema y manipulación de ella.
- Socializar el proyecto con los ciudadanos y las municipalidades, haciendo énfasis en las ventajas que tiene el proyecto y lo económico que es para adquirirlo.
- Diseñar un aplicativo que implemente un sistema de alerta en caso de fugas de agua potable, periodo de reportes, riesgos de consumo, etc.
- Incorporar al sistema conexión a mayor alcance diferente al módulo Bluetooth, para tener mayor control en caso de riesgos.

## REFERENCIAS BIBLIOGRAFICAS:

- Agudelo, C., & Marina, R. (2005). El agua , recurso estratégico del siglo XXI. *Revista Facultad Nacional de Salud Pública, vol.23*, pg.91–102.
- Alberto, C., Angela, L., & Aguirre, T. (2007). TECNOLOGÍA INFRARROJA.
- Alberto, M. (2017). Sistema de adquisición de datos de bajo costo con la plataforma arduino \*  
Low cost data acquisition system with the arduino platform Resumen Introducción.
- ANDRÉS, L. N. R. F. M. S. C. (2011). Escuela politécnica del ejército. *Diseño de Calderas Con Regeneracion*, 54–107.
- Bertoglio, O. J., & Johansen, O. (1982). *Introducción a la teoría general de sistemas*. Limusa.
- Domínguez, H. M. (2012). *Domótica : Un enfoque sociotécnico*.
- Edison Fabricio, C. (2014). Universidad nacional de loja.
- Evans, D. (2011). Internet of Things La próxima evolución de Internet lo está cambiando todo  
Internet of Things La próxima evolución de Internet lo está cambiando todo.
- Gil Mateus, E. (2011). Demanda de agua en hogares urbanos y cambios tarifarios en Bogotá.  
*Finanzas Y Política Economica, 3*, 71–90.
- Herrera Quintero, L. F. (2005). Viviendas inteligentes ( Domótica ). *Revista Ingeniería E Investigación, 25(2)*, 47–53. Retrieved from  
<http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r101024.PDF>
- IMS. (2015). metodologías, 1–29.
- Jimeno, C. L. (2007). L a Domótica.
- Legales, N. (2009). Medidas de Ecoeficiencia para el Sector. *Revista El Peruano*, pp. 1–3.
- Llinares, A. N. G. (2013). Sistemas Embebidos. *Proyecto de Investigación*, p.g 1–30.
- Neida Boscán, R. V. (2009). Protocolo de control de dispositivos domoticos: Análisis de patentes. *Télématique, vol. 8(2)*, pp. 1–14.
- Orlando Advíncula Zeballos, Samantha García Junco, Juvenal Garcia Armas Katerin Toribio Tomayo, V. M. C. (2014). PLAN DE ECOEFICIENCIA EN EL USO DEL AGUA POTABLE Y ANÁLISIS DE UNIVERSIDAD NACIONAL AGRARIA LA MOLINA. *Departamento Académico de Biología, Universidad Agraria La Molina, 13(1)*, pp. 44–54.
- Perez, A., Berreteaga, O., Olano, A. R. De, Urkidi, A., Perez, J., Coop, I. S., ... Gipuzkoa, A. (2006). UNA METODOLOGÍA PARA EL DESARROLLO DE HARDWARE Y SOFTWARE EMBEBIDOS EN SISTEMAS CRÍTICOS DE SEGURIDAD. *Ikerlan S. Coop*, pp. 70–75.
- Quintero, L. F. H. (2005). Viviendas inteligentes ( Domótica ). *Ingeniería E Investigación, vol. 25(2)*, pg. 47–53.

- Sparacino Lorefice, G. (2003). Tecnología inalámbrica Bluetooth sobre los servicios de comunicaciones en los ámbitos social y empresarial. *Télématique*, vol.2(2), pp. 36–49.
- Tamayo, K. T., & Contreras, M. (2014). Plan De Ecoeficiencia En El Uso Del Agua Potable Y Análisis De Universidad Nacional Agraria La Molina Eco-Efficiency Plan and Human Water Quality Analysis in Academic and Administrative Areas At La Molina National, 13(1).
- UCM, V. L. y G. T. (2013). Introducción a Android. *E.M.E Editorial*, pg. 1–121.
- Ulloa, R. R. (2010). La sistémica , los sistemas blandos Y los sistemas de información. *Universidad El Pacífico*.
- Vega, E., Marcela, A., Santamaría, P., & Rivas, T. (2014). Internet de los objetos empleando arduino para la gestión eléctrica domiciliaria.
- Víctor, M., Juan, S., Sergio, I., & Carlos, A. (2013). Navegación Robótica Basada en Patrones Estáticos Utilizando el Sistema Embebido CMUcam3 Robotic Navigation based in Statics Patterns using the CMUcam3. *Revista Tecnológicas*, pp. 617–629.
- Wiebe, E. (2005). ¿Cómo y por qué es importante la tecnología? *Revista Redes*, vol. 11(21), pp. 19–53.

## ANEXOS

### IMPLEMENTACION DE CODIGO FUNTE DE APLICATIVO

#### ANEXO A: DISEÑO DE INTERFAZ – CONTROL VALVULAS

(device\_list.xml)

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo1">
    <TextView android:id="@+id/title_paired_devices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Seleccione un dispositivo btSerial "
    android:visibility="gone"
    android:background="#666"
    android:textColor="#fff"
    android:paddingLeft="5dp"
    />
    <ListView android:id="@+id/paired_devices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:stackFromBottom="false"
    android:layout_weight="1"
    />

    <TextView
    android:id="@+id/connecting"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
    android:id="@+id/infoText"
    android:layout_width="308dp"
    android:layout_height="wrap_content"
    android:text="Si no hay dispositivos en la lista, por favor enlaza tu
dispositivo en la configuración de Android"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_margin="5dp"
    android:textSize="18dp"

    android:layout_centerHorizontal="true"
    android:textColor="@android:color/holo_red_dark" />

    <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center">

    </LinearLayout>

</LinearLayout>
```

(activity\_main.xml)

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="fill_parent"
android:layout_height="fill_parent">

<ScrollView
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:fillViewport="true"
android:id="@+id/scrollView">

<RelativeLayout
android:layout_width="fill_parent"
android:layout_height="fill_parent">

<LinearLayout
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:id="@+id/Conexion"
android:background="#808080">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceSmall"
android:text="CONTROL DEL AGUA"
android:id="@+id/textView"
android:layout_toLeftOf="@+id/sensorView0"
android:layout_toStartOf="@+id/sensorView0"
android:layout_marginRight="41dp"
android:layout_marginEnd="41dp"
android:textColor="#FFFFFF"/>

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:background="#20B2AA">

<LinearLayout
android:orientation="horizontal"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/linearPruebaConexion1">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceLarge"
android:text="Estado del Caño "
android:id="@+id/tv_sendorldr1"
android:layout_below="@+id/buttonOn"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_gravity="center_vertical"
android:layout_marginRight="10dp"
android:textColor="#FFFFFF"/>

```

```

<Spinner
android:id="@+id/spinner1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:drawSelectorOnTop="true"/>
</LinearLayout>

<LinearLayout
android:orientation="horizontal"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/linearPruebaConexion2">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceLarge"
android:text="Estado del Inodoro "
android:id="@+id/tv_sendorldr2"
android:layout_below="@+id/buttonOn"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_gravity="center_vertical"
android:layout_marginRight="10dp"
android:textColor="#FFFFFF"/>

<Spinner
android:id="@+id/spinner2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:drawSelectorOnTop="true"/>
</LinearLayout>

<LinearLayout
android:orientation="horizontal"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/linearPruebaConexion3">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceLarge"
android:text="Estado de la Ducha "
android:id="@+id/tv_sendorldr3"
android:layout_below="@+id/buttonOn"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_gravity="center_vertical"
android:layout_marginRight="10dp"
android:textColor="#FFFFFF"/>

<Spinner
android:id="@+id/spinner3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:drawSelectorOnTop="true"/>
</LinearLayout>

<LinearLayout

```

```

android:orientation="horizontal"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/linearPruebaConexion4">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceLarge"
android:text="Valvula Principal "
android:id="@+id/tv_sendorldr4"
android:layout_below="@+id/buttonOn"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_gravity="center_vertical"
android:layout_marginRight="10dp"
android:textColor="#FFFFFF"/>

<Spinner
android:id="@+id/spinner4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:drawSelectorOnTop="true"/>
</LinearLayout>

<LinearLayout
android:orientation="horizontal"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/linearPruebaConexion5">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceLarge"
android:text="MODO DE CONTROL"
android:id="@+id/tv_sendorldr5"
android:layout_below="@+id/buttonOn"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_gravity="center_vertical"
android:layout_marginRight="10dp"
android:textColor="#FFFFFF"/>

<Spinner
android:id="@+id/spinner5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:drawSelectorOnTop="true"/>
</LinearLayout>
</LinearLayout>

</LinearLayout>

</RelativeLayout>

</ScrollView>

</RelativeLayout>
(values – strings.xml)

```



```

<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">CONTROL F-A</string>
  <string name="action_settings">Configuracion</string>
  <string name="hello_world">Hola Mundo</string>
  <string name="none_paired">Ningun dispositivo pudo ser
emparejado</string>
  <string-array name="presidents_array">
    <item>Cerrado</item>
    <item>Abierto</item>

  </string-array>
  <string-array name="presidents_array2">
    <item>Cerrado</item>
    <item>Abierto</item>

  </string-array>
  <string-array name="presidents_array3">
    <item>Cerrado</item>
    <item>Abierto</item>
  </string-array>
  <string-array name="presidents_array4">
    <item>Cerrado</item>
    <item>Abierto</item>
  </string-array>

  <string-array name="presidents_array5">
    <item>Automatico</item>
    <item>Manual</item>
  </string-array>
</resources>

```

## ANEXO B: EMPAREJAMIENTO DE BLUETOOTH – CONTROL DE VALVULA

```

package com.agua.controlflujo;

import java.util.Set;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class DeviceListActivity extends Activity {
  // Depuración de LOGCAT
  private static final String TAG = "DeviceListActivity";
  private static final boolean D = true;
  TextView textView1;
  // Cadena EXTRA para enviar a mainactivity
  public static String EXTRA_DEVICE_ADDRESS = "device_address";

```

```

private BluetoothAdapter mBluetoothAdapter;
private ArrayAdapter<String> mPairedDevicesArrayAdapter;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.device_list);
}

@Override
public void onResume()
{
    super.onResume();
    checkBTState();
    textView1 = (TextView) findViewById(R.id.connecting);
    textView1.setTextSize(40);
    textView1.setText(" ");
    // Inicializar adaptador de matriz para dispositivos emparejados
    mPairedDevicesArrayAdapter = new ArrayAdapter<String>(this,
R.layout.device_name);
    // Buscar y configurar ListView para dispositivos emparejados
    ListView pairedListView = (ListView)
findViewById(R.id.paired_devices);
    pairedListView.setAdapter(mPairedDevicesArrayAdapter);
    pairedListView.setOnItemClickListener(mDeviceClickListener);
    // Obtener el adaptador Bluetooth local
    mBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
    Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();
    // Agregar dispositivos previamente apareados a la matriz
    if (pairedDevices.size() > 0) {
        findViewById(R.id.title_paired_devices).setVisibility(View.VISIBLE);
        for (BluetoothDevice device : pairedDevices) {
            mPairedDevicesArrayAdapter.add(device.getName() + "\n" +
device.getAddress());
        }
    } else {
        String noDevices =
getResources().getText(R.string.none_paired).toString();
        mPairedDevicesArrayAdapter.add(noDevices);
    }
}

// Configurar el oyente en el clic para la lista
private OnItemClickListener mDeviceClickListener = new
OnItemClickListener() {
    public void onItemClick(AdapterView<?> av, View v, int arg2, long
arg3) {
        textView1.setText("Conectando...");
        // Obtener la dirección MAC del dispositivo
        String info = ((TextView) v).getText().toString();
        String address = info.substring(info.length() - 17);
        Intent i = new Intent(DeviceListActivity.this, MainActivity.class);
        i.putExtra(EXTRA_DEVICE_ADDRESS, address);
        startActivity(i);
    }
};

private void checkBTState() {
    // Compruebe que el dispositivo tiene Bluetooth y que está encendido
    mBluetoothAdapter=BluetoothAdapter.getDefaultAdapter();
    if(mBluetoothAdapter==null) {

```

```

    Toast.makeText(getBaseContext(), "El dispositivo no soporta
Bluetooth", Toast.LENGTH_SHORT).show();
} else {
    if (mBtAdapter.isEnabled()) {
        Log.d(TAG, "...Bluetooth Activado...");
    } else {
        //Solicitar al usuario que active Bluetooth
        Intent enableBtIntent = new
        Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        startActivityForResult(enableBtIntent, 1);

    }
}
}
}
}
}
}

```

## ANEXO C: MENU DE CONTROL – CONTROL VALVULAS

```

package com.agua.controlflujo;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.UUID;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {
    String[] presidents;
    String[] presidents2;
    String[] presidents3;
    String[] presidents4;
    String[] presidents5;

    TextView sensorView0;
    Handler bluetoothIn;

    final int handlerState = 0;
    private BluetoothAdapter btAdapter = null;
    private BluetoothSocket btSocket = null;
    private StringBuilder recDataString = new StringBuilder();
    private ConnectedThread mConnectedThread;
    // Servicio UUID: conexion a los los dispositivos
    private static final UUID BTMODULEUUID = UUID.fromString("00001101-
0000-1000-8000-00805F9B34FB");
    // Cadena para la dirección MAC
    private static String address = null;

```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    presidents = getResources().getStringArray(R.array.presidents_array);
    presidents2 =
getResources().getStringArray(R.array.presidents_array2);
    presidents3 =
getResources().getStringArray(R.array.presidents_array3);
    presidents4 =
getResources().getStringArray(R.array.presidents_array4);
    presidents5 =
getResources().getStringArray(R.array.presidents_array5);

    Spinner s1 = (Spinner) findViewById(R.id.spinner1);
    Spinner s2 = (Spinner) findViewById(R.id.spinner2);
    Spinner s3 = (Spinner) findViewById(R.id.spinner3);
    Spinner s4 = (Spinner) findViewById(R.id.spinner4);
    Spinner s5 = (Spinner) findViewById(R.id.spinner5);

    ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown_ite
m, presidents);
    ArrayAdapter<String> adapter2 = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown_ite
m, presidents2);
    ArrayAdapter<String> adapter3 = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown_ite
m, presidents3);
    ArrayAdapter<String> adapter4 = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown_ite
m, presidents4);
    ArrayAdapter<String> adapter5 = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown_ite
m, presidents5);

    s1.setAdapter(adapter);
    s2.setAdapter(adapter2);
    s3.setAdapter(adapter3);
    s4.setAdapter(adapter4);
    s5.setAdapter(adapter5);
    //Vincule los botones y textViews a las vistas respectivas
    sensorView0 = (TextView) findViewById(R.id.sensorView0);
    bluetoothIn = new Handler() {
    public void handleMessage(android.os.Message msg) {

    }
    };

    btAdapter = BluetoothAdapter.getDefaultAdapter(); // Obtener
adaptador Bluetooth
    checkBTState();
    // Envio de datos al Arduino
    s1.setOnItemClickListener(new AdapterView.OnItemClickListener()
    {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,
long arg3)
    {
    // Escribir el caracter mediante nConnectedThead.write
    int index = arg0.getSelectedItemPosition();

```

```

    if(index==0){
        mConnectedThread.write("1");
        Toast.makeText(getBaseContext(),"Lavamanos cerrado : " +
presidents[index],Toast.LENGTH_SHORT).show();
    }
    if(index==1){
        mConnectedThread.write("2");
        Toast.makeText(getBaseContext(),"Lavamanos abierto: " +
presidents[index],Toast.LENGTH_SHORT).show();
    }

}
@Override
public void onNothingSelected(AdapterView<?> arg0) {}
});
s2.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()

{
    @Override
    public void onItemSelected(AdapterView<?> arg0, View arg1,int arg2,
long arg3)
    {
        // Escribir el caracter mediante nConnectedThead.write
        int index = arg0.getSelectedItemPosition();
        if(index==0){
            mConnectedThread.write("b");
            Toast.makeText(getBaseContext(),"inodoro cerrado: " +
presidents2[index],Toast.LENGTH_SHORT).show();
        }
        if(index==1){
            mConnectedThread.write("B");
            Toast.makeText(getBaseContext(),"inodoro abierto: " +
presidents2[index],Toast.LENGTH_SHORT).show();
        }
    }
}
@Override
public void onNothingSelected(AdapterView<?> arg0) {}
});

s3.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()

{
    @Override
    public void onItemSelected(AdapterView<?> arg0, View arg1,int arg2,
long arg3)
    {
        // Escribir el caracter mediante nConnectedThead.write
        int index = arg0.getSelectedItemPosition();
        if(index==0){
            mConnectedThread.write("d");
            Toast.makeText(getBaseContext(),"ducha cerrado: " +
presidents3[index],Toast.LENGTH_SHORT).show();
        }
        if(index==1){
            mConnectedThread.write("D");
            Toast.makeText(getBaseContext(),"ducha abierto: " +
presidents3[index],Toast.LENGTH_SHORT).show();
        }
    }
}
}

```

```

@Override
public void onNothingSelected(AdapterView<?> arg0) {}
});
s4.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()

{
@Override
public void onItemSelected(AdapterView<?> arg0, View arg1,int arg2,
long arg3)
{
// Escribir el caracter mediante nConnectedThead.write
int index = arg0.getSelectedItemPosition();
if(index==0){
mConnectedThread.write("p");
Toast.makeText(getBaseContext(),"valvula principal cerrado: " +
presidents4[index],Toast.LENGTH_SHORT).show();
}
if(index==1){
mConnectedThread.write("P");
Toast.makeText(getBaseContext(),"valvula principal abierto: " +
presidents4[index],Toast.LENGTH_SHORT).show();
}

}
@Override
public void onNothingSelected(AdapterView<?> arg0) {}
});
s5.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()

{
@Override
public void onItemSelected(AdapterView<?> arg0, View arg1,int arg2,
long arg3)
{
// Escribir el caracter mediante nConnectedThead.write
int index = arg0.getSelectedItemPosition();
if(index==0){
mConnectedThread.write("A");
Toast.makeText(getBaseContext(),"modo automatico activo: " +
presidents5[index],Toast.LENGTH_SHORT).show();
}
if(index==1){
mConnectedThread.write("M");
Toast.makeText(getBaseContext(),"modo manual activo: " +
presidents5[index],Toast.LENGTH_SHORT).show();
}

}
@Override
public void onNothingSelected(AdapterView<?> arg0) {}
});
}

private BluetoothSocket createBluetoothSocket(BluetoothDevice device)
throws IOException {
//Crea una conexión saliente segura con el dispositivo BT mediante
UUID
return device.createRfcommSocketToServiceRecord(BTMODULEUUID);
}
}

```

```

@Override
public void onResume() {
    super.onResume();
    //Obtener la dirección MAC de DeviceListActivity
    Intent intent = getIntent();
    //Obtenga la dirección MAC de DeviceListActivty a través de EXTRA
    address =
intent.getStringExtra(DeviceListActivity.EXTRA_DEVICE_ADDRESS);
    //Crear dispositivo y configurar la dirección MAC
    BluetoothDevice device = btAdapter.getRemoteDevice(address);
    try {
        btSocket = createBluetoothSocket(device);
    } catch (IOException e) {
        Toast.makeText(getBaseContext(), "La creacción del Socket fallo",
Toast.LENGTH_LONG).show();
    }
    // Establezca la conexión del zócalo Bluetooth.
    try
    {
        btSocket.connect();
    } catch (IOException e) {
        try
        {
            btSocket.close();
        } catch (IOException e2)
        {

        }
    }
    mConnectedThread = new ConnectedThread(btSocket);
    mConnectedThread.start();
    //Envío de un carácter al reanudar.
    mConnectedThread.write("x");
}

@Override
public void onPause()
{
    super.onPause();
    try
    {
        // Cerrar el BT despues de la actividad
        btSocket.close();
    } catch (IOException e2) {

    }
}

//Comprueba que el dispositivo Bluetooth Bluetooth está disponible y
solicita que se encienda si está desactivado
private void checkBTState() {

    if(btAdapter==null) {
        Toast.makeText(getBaseContext(), "El dispositivo no soporta
bluetooth", Toast.LENGTH_LONG).show();
    } else {
        if (btAdapter.isEnabled()) {
        } else {
            Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);

```

```

startActivityResult(enableBtIntent, 1);
}
}
}

//Crear una nueva clase para conectar el hilo
private class ConnectedThread extends Thread {
private final InputStream mmInStream;
private final OutputStream mmOutStream;
//Creación del hilo de conexión
public ConnectedThread(BluetoothSocket socket) {
InputStream tmpIn = null;
OutputStream tmpOut = null;

try {
tmpIn = socket.getInputStream();
tmpOut = socket.getOutputStream();
} catch (IOException e) { }

mmInStream = tmpIn;
mmOutStream = tmpOut;
}

public void run() {
byte[] buffer = new byte[256];
int bytes;
// bucle para escuchar los mensajes recibidos
while (true) {
try {
bytes = mmInStream.read(buffer);
//Leer los bytes del búfer de entrada
String readMessage = new String(buffer, 0, bytes);
bluetoothIn.obtainMessage(handlerState, bytes, -1,
readMessage).sendToTarget();
} catch (IOException e) {
break;
}
}
}
//método de escritura
public void write(String input) {
//Convierte la cadena introducida en bytes
byte[] msgBuffer = input.getBytes();
try {
//Escribir bytes sobre la conexión BT a través de outstream
mmOutStream.write(msgBuffer);
} catch (IOException e) {
Toast.makeText(getApplicationContext(), "La Conexión fallo",
Toast.LENGTH_LONG).show();
finish();
}
}
}
}
}

```



## ANEXO D: DISEÑO DE INTERFAZ – CONSUMO DE AGUA

### (device\_list.xml)

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondo1">
    <TextView android:id="@+id/title_paired_devices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Seleccione un dispositivo btSerial "
    android:visibility="gone"
    android:background="#666"
    android:textColor="#fff"
    android:paddingLeft="5dp"
    />
    <ListView android:id="@+id/paired_devices"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:stackFromBottom="false"
    android:layout_weight="1"
    />

    <TextView
    android:id="@+id/connecting"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
    android:id="@+id/infoText"
    android:layout_width="317dp"
    android:layout_height="wrap_content"
    android:text="Si no hay dispositivos en la lista, por favor enlaza tu
dispositivo en la configuración de Android"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_margin="5dp"
    android:textSize="18dp"
    android:textColor="@android:color/holo_red_dark" />

    <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center">

    </LinearLayout>

</LinearLayout>
```

### (activity\_main.xml)

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```

<ScrollView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:fillViewport="true"
android:id="@+id/scrollView">

<RelativeLayout
android:layout_width="fill_parent"
android:layout_height="fill_parent">

<LinearLayout
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:id="@+id/Conexion"
android:background="#20B2AA">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceSmall"
android:text="Datos a Recibir del Sensor"
android:id="@+id/textView"
android:layout_toLeftOf="@+id/sensorView0"
android:layout_toStartOf="@+id/sensorView0"
android:layout_marginRight="41dp"
android:layout_marginEnd="41dp" />
</LinearLayout>
</RelativeLayout>

</ScrollView>
<RelativeLayout
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>

<TextView
android:id="@+id/testView1"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Tamaño del String = "
android:textSize="15sp"
android:layout_alignParentTop="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_marginTop="112dp" />

<TextView
android:id="@+id/txtString"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Datos recibidos ="
android:textSize="20dp"
android:layout_below="@+id/testView1"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:gravity="center" />
</RelativeLayout>
</RelativeLayout>

```

## (values – strings.xml)

```
<resources>
  <string name="app_name">CONSUMO AGUA</string>
  <string name="action_settings">Configuracion</string>
  <string name="none_paired">Ningun dispositivo pudo ser
emparejado</string>
</resources>
```

## ANEXO E: EMPAREJAMIENTO BLUETOOTH – CONSUMO AGUA

En este caso para el consumo de agua utilizaremos el mismo código de **emparejamiento** que se utilizó para el control de las válvulas, el código se detalla con en páginas anteriores.

## ANEXO F: MENU DE VISUALIZACION- CONSUMO DE AGUA

```
package com.consumo.agua;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.UUID;
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity {
    TextView txtString, txtStringLength, sensorView0;
    Handler bluetoothIn;
    final int handlerState = 0;
    private BluetoothAdapter btAdapter = null;
    private BluetoothSocket btSocket = null;
    private StringBuilder recDataString = new StringBuilder();
    private ConnectedThread mConnectedThread;
    // Servicio UUID: conexion a los los dispositivos
    private static final UUID BTMODULEUUID = UUID.fromString("00001101-
0000-1000-8000-00805F9B34FB");
    // Cadena para la dirección MAC
    private static String address = null;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txtString = (TextView) findViewById(R.id.txtString);
        txtStringLength = (TextView) findViewById(R.id.testView1);
        sensorView0 = (TextView) findViewById(R.id.sensorView0);
        // Recepcion de datos del sensor de flujo de agua
        bluetoothIn = new Handler() {
```

```

public void handleMessage(android.os.Message msg) {
    if (msg.what == handlerState) {
        String readMessage = (String) msg.obj;

        recDataString.append(readMessage);
        int endOfLineIndex = recDataString.indexOf("~");
        if (endOfLineIndex > 0) { // Asegúrese de que haya datos antes de ~
            String dataInPrint = recDataString.substring(0, endOfLineIndex); //
            Extraer cadena
            txtString.setText("Resumen" + dataInPrint);
            int dataLength = dataInPrint.length();
            txtStringLength.setText("Datos enviados : " +
            String.valueOf(dataLength));
            recDataString.delete(0, recDataString.length());

        }

    }
};
// Obtener adaptador Bluetooth
btAdapter = BluetoothAdapter.getDefaultAdapter();
checkBTState();
}
private BluetoothSocket createBluetoothSocket(BluetoothDevice device)
throws IOException {
    return device.createRfcommSocketToServiceRecord(BTMODULEUUID);
}

@Override
public void onResume() {
    super.onResume();
    //Obtener la dirección MAC de DeviceListActivity
    Intent intent = getIntent();
    //Obtenga la dirección MAC de DeviceListActivty a través de EXTRA
    address =
intent.getStringExtra(DeviceListActivity.EXTRA_DEVICE_ADDRESS);
    //Crear dispositivo y configurar la dirección MAC
    BluetoothDevice device = btAdapter.getRemoteDevice(address);
    try {
        btSocket = createBluetoothSocket(device);
    } catch (IOException e) {
        Toast.makeText(getApplicationContext(), "La creación del Socket fallo",
        Toast.LENGTH_LONG).show();
    }
    // Establezca la conexión del zócalo Bluetooth.
    try
    {
        btSocket.connect();
    } catch (IOException e) {
        try
        {
            btSocket.close();
        } catch (IOException e2)
        {

        }

    }
    mConnectedThread = new ConnectedThread(btSocket);
    mConnectedThread.start();
    //Envío de un carácter al reanudar.

```

```

mConnectedThread.write("x");
}

@Override
public void onPause()
{
    super.onPause();
    try
    {
        // Cerrar el BT despues de la actividad
        btSocket.close();
    } catch (IOException e2) {
    }
}
//Comprueba que el dispositivo Bluetooth Bluetooth está disponible y
solicita que se encienda si está desactivado
private void checkBTState() {

    if(btAdapter==null) {
        Toast.makeText(getApplicationContext(), "El dispositivo no soporta
bluetooth", Toast.LENGTH_LONG).show();
    } else {
        if (btAdapter.isEnabled()) {
        } else {
            Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBtIntent, 1);
        }
    }
}
//Crear una nueva clase para conectar el hilo
private class ConnectedThread extends Thread {
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;
    //Creación del hilo de conexión
    public ConnectedThread(BluetoothSocket socket) {
        InputStream tmpIn = null;
        OutputStream tmpOut = null;

        try {
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        } catch (IOException e) { }

        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }
    public void run() {
        byte[] buffer = new byte[256];
        int bytes;
        // bucle para escuchar los mensajes recibidos
        while (true) {
            try {
                bytes = mmInStream.read(buffer);
                //Leer los bytes del búfer de entrada
                String readMessage = new String(buffer, 0, bytes);
                bluetoothIn.obtainMessage(handlerState, bytes, -1,
readMessage).sendToTarget();
            } catch (IOException e) {
                break;
            }
        }
    }
}

```

```

}
}
//método de escritura
public void write(String input) {
//Convierte la cadena introducida en bytes
byte[] msgBuffer = input.getBytes();
try {
//Escribir bytes sobre la conexión BT a través de outstream
mmOutputStream.write(msgBuffer);
} catch (IOException e) {
Toast.makeText(getApplicationContext(), "La Conexión fallo",
Toast.LENGTH_LONG).show();
finish();
}
}
}
}
}

```

## ANEXO G: CALIBRACION DEL SENSOR DE FLUJO DE AGUA

```

volatile long NumPulsos; //cantidad de pulsos recibidos
int PinSensor = 2; //Sensor en el pin 2
void ContarPulsos () //Función que se ejecuta en interrupción
{
NumPulsos++; //incrementamos la variable de pulsos
}
void setup() //Función para obtener frecuencia de los pulsos
{
Serial.begin(9600);
pinMode(PinSensor, INPUT); // Pin de salida del sensor
attachInterrupt(0,ContarPulsos,RISING); //(Interrupción 0(Pin2),función,Flanco de subida)
interrupts(); //Habilitamos las interrupciones
}

void loop ()
{
//Enviamos por el puerto serie
Serial.print ("Numero de Pulsos = ");
Serial.println (NumPulsos);
delay(100);
}

```

## ANEXO H: CALCULO DEL VOLUMEN DE CONSUMO DE AGUA

```
volatile int NumPulsos; //variable para la cantidad de pulsos recibidos
int PinSensor = 2; //Sensor conectado en el pin 2
float factor_conversion=12.5; //para convertir de frecuencia a caudal
float volumen=0;
long dt=0; //variación de tiempo por cada bucle
long t0=0; //millis() del bucle anterior
//Función que se ejecuta en interrupción
void ContarPulsos ()
{
  NumPulsos++;
}
//Función para obtener frecuencia de los pulsos
int ObtenerFrecuecia()
{
  int frecuencia;
  NumPulsos = 0; //Ponemos a 0 el número de pulsos
  interrupts(); //Habilitamos las interrupciones
  delay(1000); //muestra de 1 segundo
  noInterrupts(); //Deshabilitamos las interrupciones
  frecuencia=NumPulsos; //Hz(pulsos por segundo)
  return frecuencia;
}
void setup()
{
  Serial.begin(9600);
  pinMode(PinSensor, INPUT);
  attachInterrupt(0,ContarPulsos,RISING); //(Interrupción 0(Pin2),función,Flanco de subida)
  Serial.println ("Envie 'r' para restablecer el volumen a 0 Litros");
  t0=millis();
}
void loop ()
{
  if (Serial.available()) {
    if(Serial.read()=='r')volumen=0; //restablecemos el volumen si recibimos 'r'
  }
  float frecuencia=ObtenerFrecuecia(); //obtenemos la frecuencia de los pulsos en Hz
  float caudal_L_m=frecuencia/factor_conversion; //calculamos el caudal en L/m
  dt=millis()-t0; //calculamos la variación de tiempo
  t0=millis();
  volumen=volumen+(caudal_L_m/60)*(dt/1000); // volumen(L)=caudal(L/s)*tiempo(s)

  //Enviamos por el puerto serie
  Serial.print ("Caudal: ");
  Serial.print (caudal_L_m,3);
}
```

```

Serial.print ("L/mintVolumen: ");
Serial.print (volumen,3);
Serial.println (" L");
}

```

## **ANEXO I: IMPLEMENTACION ARDUINO MEGA CONTROL VALVULAS**

```

#include <Wire.h> // libreria para el temp.
#include <LiquidCrystal_I2C.h> // libreria para el LCD 20x4
#include <Keypad.h> // libreria para el teclado 4x4
LiquidCrystal_I2C lcd(0x27,20,4); // establecer la dirección LCD de 0x27 para una pantalla de 16
caracteres y 2 líneas
char currentTimeValue[4]; // caracte de 4 digitos para los min ,seg
int currentState = 1; //variables para lñas funciones siguientes
int timerSeconds = 0;
int lpcnt = 0;

//define el keypad

const byte rows = 4; // declaramos el numero de filas
const byte cols = 4; // declaramos numero de columnas
char keys[rows][cols] = {{'1','2','3','A'}, {'4','5','6','B'}, {'7','8','9','C'}, {'*','0','#','D'}}; // arreglo de
las disposiciones de las teclas del teclado
byte rowPins[rows] = {39,41,43,45}; // declaramos los pines a usar del arduino mega para las
filas
byte colPins[cols] = {47,49,51,53}; // declaramos los pines a usar del arduino mega para las
columnas
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols); // creamos
unobjeto keypad

//infrarojo//

#include <infrarrojo.h>
infrarrojo estadoA(5); //definir el numero de pin del arduino a usar
int VALORA; //variable que recibe el dato del sensor (para el caño)
int ledhd = 11; // led que nos indica si esta funcionado el sensor
int infrarojo_estadoA; // variable de encendido y apagado
int infrarojoState = LOW;

//pir
const int LEDPin = 10; // pin para el LED
const int PIRPin = 6; // pin de entrada (for PIR sensor)
int pirState = LOW; // de inicio no hay movimiento
int val = 0; // estado del pin

//servo
#include <Servo.h> //Control del servomotor ( libreria para el servo)

```



```

#define CERRAR 0 //Calibrar a gusto la pos del servomotor cerrado
#define ABRIR 120 //Calibrar a gusto la posicion en grados del servomotor abierto
Servo seguro; //servomotor

//pap
#include <Stepper.h> //Importamos la librería para controlar motores paso a paso
#define STEPS 200 //Ponemos el número de pasos que necesita para dar una vuelta. 200 en
nuestro caso
Stepper stepper(STEPS, 32, 30, 28, 26); //Stepper nombre motor (número de pasos por vuelta,
pins de control)

//MODULO RELE DE 4 CANALES//

#define VALVULA1 29 // se definen los pines donde se conectaran los reles
#define VALVULA2 31
#define VALVULA3 33
#define VALVULA4 35
#define VALVULA_ON 1
#define VALVULA_OFF 0

//bluetooth//
char in;
char mode;
int sensorpir()
{
val = digitalRead(PIRPin);
if (val == HIGH) //si está activado
{
digitalWrite(LEDpin, HIGH); //LED ON
if (pirState == LOW) //si previamente estaba apagado
{
seguro.write(ABRIR);
//stepper.step(33);
pirState = HIGH;
}
}
else //si esta desactivado
{
digitalWrite(LEDpin, LOW); // LED OFF
if (pirState == HIGH) //si previamente estaba encendido
{
seguro.write(CERRAR);
//stepper.step(-33);
pirState = LOW;
}
}
}

```

```

}
}

int SensorInfrarojoA ()
{
  infrarojo_estadoA = estadoA.lectura(VALORA); //LED QUE RECOGE EL ESTADO DEL SENSOR
  if (infrarojo_estadoA == 1) //ETAPA DE COMPARACION PARA ACTIVAR UN LED SEGUN EL
  ESTADO DEL SENSOR
  {
    digitalWrite(ledhd, LOW);
    digitalWrite(VALVULA1, VALVULA_OFF); //PRENDE UN LED SI EL SENSOR DETECTA
    OBSTACULO

  }
  else
  {
    digitalWrite(ledhd, HIGH);
    digitalWrite(VALVULA1, VALVULA_ON);
    //elay(200);

  }
}
//*****

void showEnteredTime()
{
  lcd.setCursor(14,3);
  lcd.print(currentTimeValue[0]);
  lcd.print(currentTimeValue[1]);
  lcd.print(":");
  lcd.print(currentTimeValue[2]);
  lcd.print(currentTimeValue[3]);
}
void relayStatus(bool state)
{
  if (state)
    digitalWrite(VALVULA2, VALVULA_ON);
  else
    digitalWrite(VALVULA2, VALVULA_OFF);
}
void showCountdown()
{
  char timest[6];\

  lcd.setCursor(0,0);

```

```

lcd.print("*****");
lcd.setCursor(0,1);
lcd.print("***cuenta atras* **");
lcd.setCursor(0,2);
lcd.print("** ");
sprintf(timest, "%d:%.2d", (timerSeconds/60), (timerSeconds - ((timerSeconds/60)*60));
lcd.print(timest);
lcd.print(" **");
lcd.setCursor(0,3);
lcd.print("pres.(*)para PARAR ");

}

void displayCodeEntryScreen()
{
clearScreen();
lcd.setCursor(0,0);
lcd.print("****TEMPORIZADOR****");
lcd.setCursor(0,1);
lcd.print("***VALVULA OO1****");
lcd.setCursor(0,2);
lcd.print("tiempo en mm:ss");
lcd.setCursor(0,3);
lcd.print("INICIAR->(#) ");
}

void clearScreen()
{
int i;
char tempVal[3];
char key = keypad.getKey();

//key pressed and state is 1
if (int(key) != 0 and currentState == 1)
{
switch (key)
{
case '*':
relayStatus(false);
currentTimeValue[0]='0';
currentTimeValue[1]='0';
currentTimeValue[2]='0';
currentTimeValue[3]='0';
showEnteredTime();
currentState = 1;
}
}
}

```

```

lpcnt = 0;
timerSeconds = 0;
break;

case '#':
tempVal[0] = currentTimeValue[0];
tempVal[1] = currentTimeValue[1];
tempVal[2] = 0;

timerSeconds = atol(tempVal) * 60;

tempVal[0] = currentTimeValue[2];
tempVal[1] = currentTimeValue[3];
tempVal[2] = 0;

timerSeconds = timerSeconds + atol(tempVal);
currentState = 2;
break;

default:
currentTimeValue[0] = currentTimeValue[1];
currentTimeValue[1] = currentTimeValue[2];
currentTimeValue[2] = currentTimeValue[3];
currentTimeValue[3] = key;
showEnteredTime();
break;
}

if (currentState == 2)
{
if (int(key) != 0)
{
if (key == '*')
{
relayStatus(false);
displayCodeEntryScreen();
currentTimeValue[0]='0';
currentTimeValue[1]='0';
currentTimeValue[2]='0';
currentTimeValue[3]='0';
showEnteredTime();
currentState = 1;
lpcnt = 0;
timerSeconds = 0;
}
}
}
}

```

```

}
else
{
if (lpcnt > 9)
{
lpcnt = 0;
--timerSeconds;
showCountdown();
if (timerSeconds <= 0)
{
currentState = 1;
relayStatus(false);
displayCodeEntryScreen();
showEnteredTime();
}
else
{
relayStatus(true);
}
}

++lpcnt;
//delay(100);
}
}
}
}
//*****
void setup()
{

Serial.begin(9600);
///temp//
lcd.init(); // inicializar el LCD
lcd.backlight(); //mostrar la pantalla principal
displayCodeEntryScreen(); //configuración y apague De la balvula

//tiempo por defecto de configuración para 00:00 mm:ss
currentTimeValue[0]='0';
currentTimeValue[1]='0';
currentTimeValue[2]='0';
currentTimeValue[3]='0';
showEnteredTime();

//servo//

```

```

pinMode(22, OUTPUT); // LED QUE INDICA ABIERTO(13)
pinMode(23, OUTPUT); // LED QUE INDICA CERRADO(19)
seguro.attach(7); // Pin del servomotor
seguro.write(CERRAR);
seguro.write(0); //Cerrar

//PAP////
stepper.setSpeed(30);

//pir//
pinMode(LEDpin, OUTPUT); //declaramos el LEDpin(11) como salida digital
pinMode(ledhd, OUTPUT);
pinMode(PIRpin, INPUT); //declaramos el PIRpin(5) como entrada digital

//PARA LAS VALVULAS

pinMode(VALVULA1, OUTPUT);
pinMode(VALVULA2, OUTPUT);
pinMode(VALVULA3, OUTPUT);
pinMode(VALVULA4, OUTPUT); // Se configuran los pines como salidas
digitalWrite (VALVULA1, VALVULA_OFF);
digitalWrite (VALVULA2, VALVULA_OFF);
digitalWrite (VALVULA3, VALVULA_OFF);
digitalWrite (VALVULA4, VALVULA_OFF);
//SENSOR DE FLUJO
}
//*****
void loop()
{
if(Serial.available())
{
in = Serial.read();
if (in == 'A'){mode = 'a'; }
if (in == 'M'){mode = 'm';}
digitalWrite(VALVULA1, VALVULA_ON);
digitalWrite(VALVULA2, VALVULA_ON);
digitalWrite(VALVULA3, VALVULA_ON);
digitalWrite(VALVULA4, VALVULA_ON);
seguro.write(CERRAR);
}
}

//MODO AUTONOMO

if (mode == 'a')

```

```

{
int l;
char tempVal[3];
char key = keypad.getKey();

//key pressed and state is 1
if (int(key) != 0 and currentState == 1)
{
switch (key)
{
case '*':
relayStatus(false);
currentTimeValue[0]='0';
currentTimeValue[1]='0';
currentTimeValue[2]='0';
currentTimeValue[3]='0';
showEnteredTime();
currentState = 1;

lpcnt = 0;
timerSeconds = 0;
break;

case '#':
tempVal[0] = currentTimeValue[0];
tempVal[1] = currentTimeValue[1];
tempVal[2] = 0;

timerSeconds = atol(tempVal) * 60;

tempVal[0] = currentTimeValue[2];
tempVal[1] = currentTimeValue[3];
tempVal[2] = 0;

timerSeconds = timerSeconds + atol(tempVal);
currentState = 2;
break;

default:
currentTimeValue[0] = currentTimeValue[1];
currentTimeValue[1] = currentTimeValue[2];
currentTimeValue[2] = currentTimeValue[3];
currentTimeValue[3] = key;
showEnteredTime();
break;
}
}

```

```

}
}

if (currentState == 2)
{
if (int(key) != 0)
{
if (key == '*')
{
relayStatus(false);
displayCodeEntryScreen();
currentTimeValue[0]='0';
currentTimeValue[1]='0';
currentTimeValue[2]='0';
currentTimeValue[3]='0';
showEnteredTime();
currentState = 1;
lpcnt = 0;
timerSeconds = 0;
}
}
else
{
if (lpcnt > 9)
{
lpcnt = 0;
--timerSeconds;
showCountdown();

if (timerSeconds <= 0)
{
currentState = 1;
relayStatus(false);
displayCodeEntryScreen();
showEnteredTime();
}
else
{
relayStatus(true);
}
}

++lpcnt;
delay(100);
}
}

```



```

}
SensorInfrarojoA();
sensorpir();
}
//MODO MANUAL (BLUETOOTH)

else if (mode == 'm')
{
if (in == '1'){
digitalWrite(ledhd, LOW);
digitalWrite(VALVULA1, VALVULA_ON);
}
else if (in == '2'){
digitalWrite(ledhd, HIGH);
digitalWrite(VALVULA1, VALVULA_OFF);
}

if (in == 'B'){
seguro.write(ABRIR);
//stepper.step(33);
pirState = HIGH;
}
else if(in == 'b'){
seguro.write(CERRAR);
//stepper.step(-33);
pirState = LOW;
}

if (in == 'D'){
digitalWrite(VALVULA2, VALVULA_ON);
}
else if(in == 'd'){
digitalWrite(VALVULA2, VALVULA_OFF);
}
if (in == 'P'){
digitalWrite(VALVULA3, VALVULA_ON);
}
else if(in == 'p'){
digitalWrite(VALVULA3, VALVULA_OFF);
}
}
}
}
}

```

## ANEXO J: IMPLEMENTACION ARDUINO NANO CONSUMO DE AGUA

```
volatile long NumPulsos; //variable para la cantidad de pulsos recibidos
int PinSensor = 2; //Sensor conectado en el pin 2
float factor_conversion=12.3; //para convertir de frecuencia a caudal
float volumen=0;
float costovolumen;
long dt=0; //variación de tiempo por cada bucle
long t0=0; //millis() del bucle anterior

//Función que se ejecuta en interrupción
void ContarPulsos ()
{
  NumPulsos++; //incrementamos la variable de pulsos
}
//Función para obtener frecuencia de los pulsos
int ObtenerFrecuencia()
{
  int frecuencia;
  NumPulsos = 0; //Ponemos a 0 el número de pulsos
  interrupts(); //Habilitamos las interrupciones
  delay(2000); //muestra de 1 segundo
  noInterrupts(); //Desabilitamos las interrupciones
  frecuencia=NumPulsos; //Hz(pulsos por segundo)
  return frecuencia;
}

void setup()
{
  Serial.begin(9600);
  pinMode(PinSensor, INPUT);
  attachInterrupt(0,ContarPulsos,RISING); //(Interrupcion 0(Pin2),funcion,Flanco de
  subida)
  interrupts(); //Habilitamos las interrupciones
  Serial.println (" sensor de flujo de agua ");
  t0=millis();
}
void loop ()
{
  if (Serial.available())
  {
    if(Serial.read()=='r') //restablecemos el volumen si recibimos 'r'
    volumen=0;
  }
  float frecuencia=ObtenerFrecuencia(); //obtenemos la Frecuencia de los pulsos en Hz
  float caudal_L_m=frecuencia/factor_conversion; //calculamos el caudal en L/m
  float caudal_L_h=caudal_L_m*60; //calculamos el caudal en L/h
  float caudal_L_d=caudal_L_h*24; //calculamos el caudal en L/d
```

```

dt=millis()-t0; //calculamos la variación de tiempo
t0=millis();
volumen=volumen+(caudal_L_m/60)*(dt/1000); // volumen(L)=caudal(L/s)*tiempo(s)

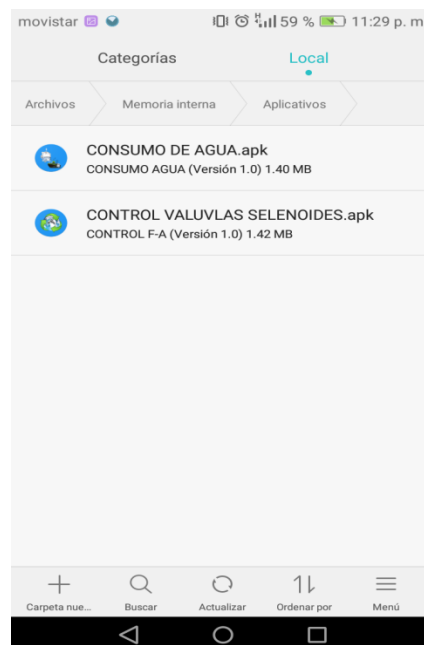
//Enviamos por el puerto serie
Serial.println (" Caudal: ");
Serial.print (caudal_L_m);
Serial.println (" L/m ");
Serial.print (caudal_L_h);
Serial.println (" L/h ");
Serial.print (caudal_L_d);
Serial.println (" L/d ");
Serial.println (" Volumen(consumo): ");
Serial.print (volumen);
Serial.println ("L");
Serial.println('~');
Serial.println ();
}

```

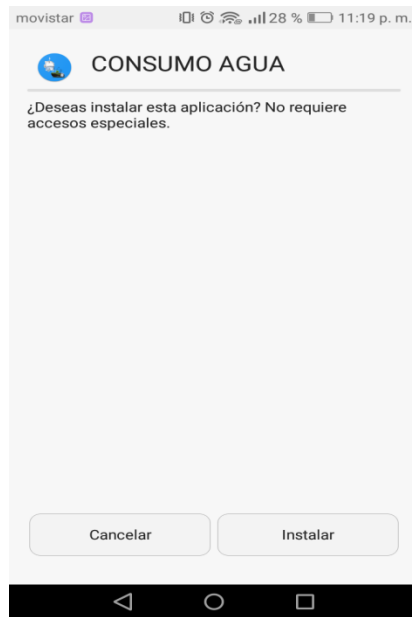
## ANEXO K: MANUAL DE USUARIO APLICACION MOVIL

A continuación se detalla el funcionamiento del aplicativo móvil del control de flujo de agua y consumo de agua potable.

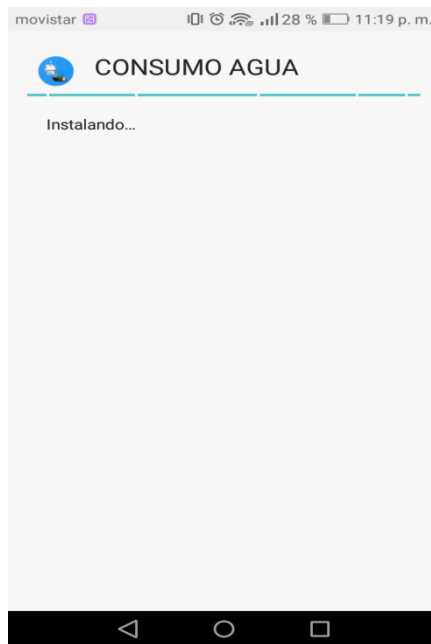
1. Primeramente, procedemos a instalar los aplicativos tanto del consumo de agua como en control de válvulas.



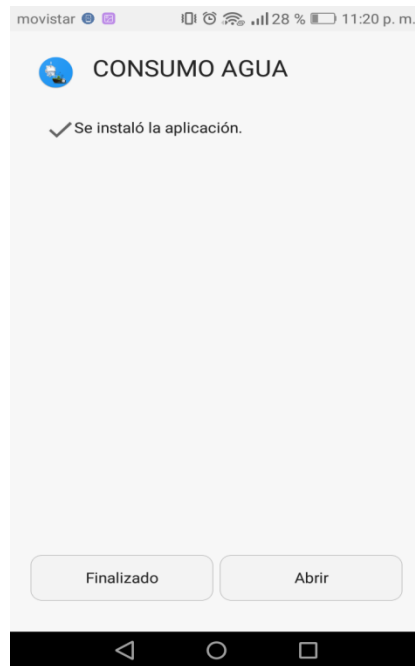
2. Seleccionamos el primer APK el cual es el instalador de consumo de agua, para instalar pulsamos dos veces y este empezara a ejecutaren seguida.



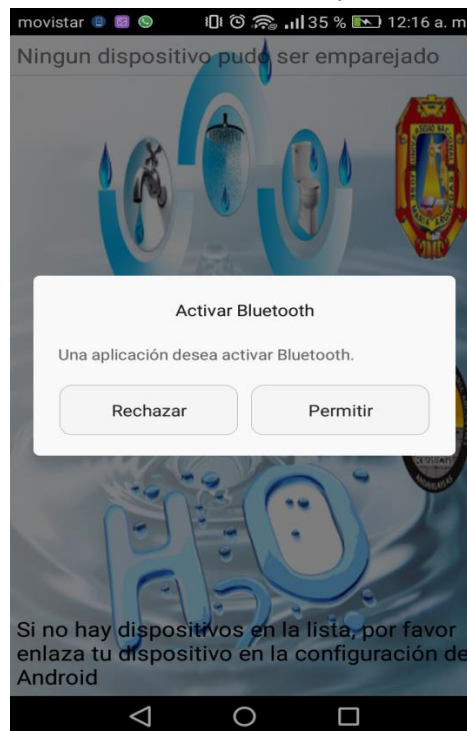
3. Seguidamente pulsamos en el botón instalar para que pueda ejecutarse y forme parte de todo nuestros aplicativo.



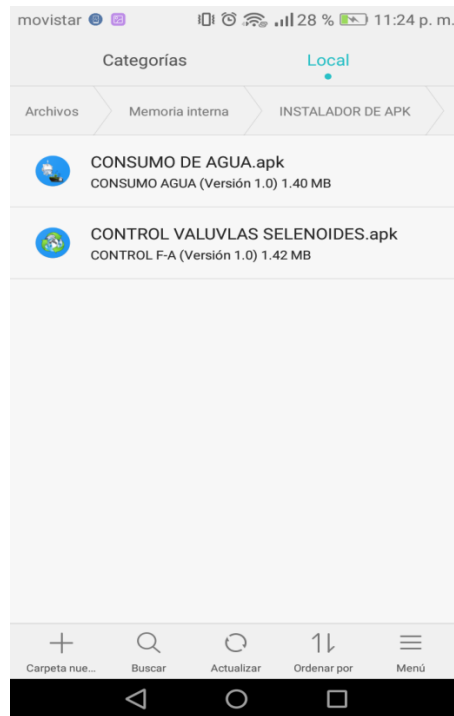
4. Esperamos a que este proceso termine puede tardar unos cuantos segundo y seguidamente nos emergerá una nueva ventana en nuestro dispositivo móvil.



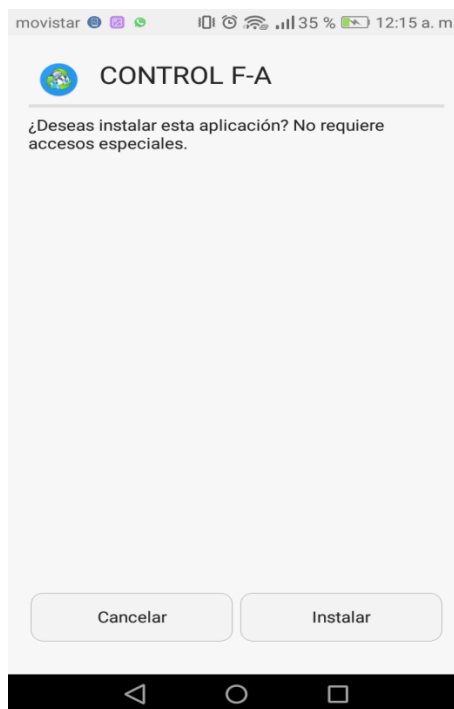
5. Para terminar con este proceso le damos en abrir, el archivo se ejecutará y nos mostrara una nueva interfaz con otras opciones en el cual le damos al botón permitir, para que pueda activar el modulo Bluetooth del dispositivo.



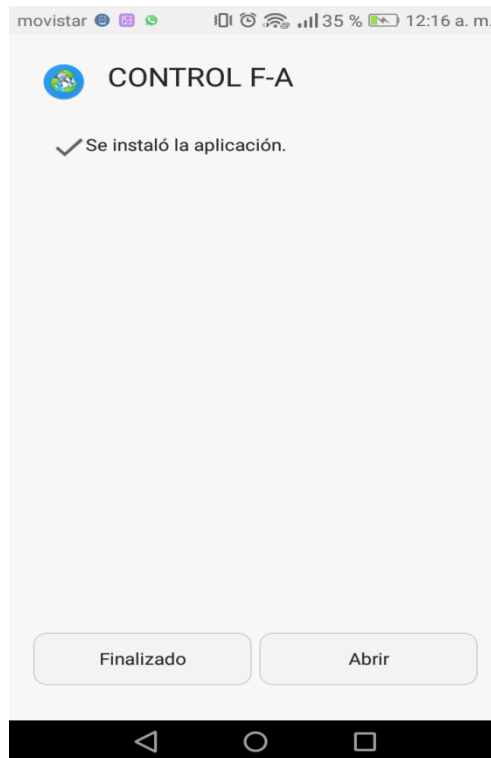
6. Del mismo modo aremos en la instalación del aplicativo, para el control de las válvulas solenoides.



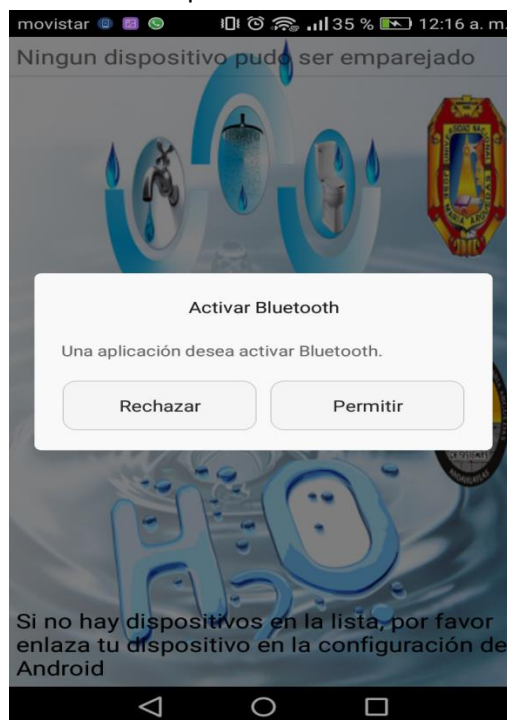
7. Le damos doble clic y este nos ejecuta para que nos emerja una nueva interfaz en este caso le damos a instalar.



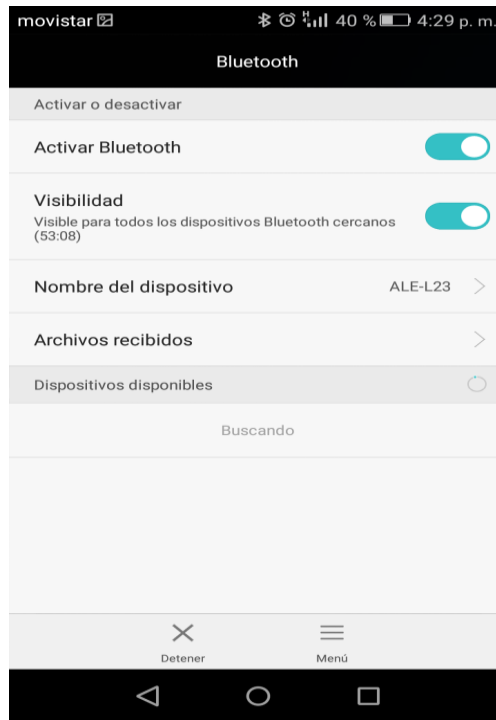
8. Esperamos a que el proceso de instalación finalicé satisfactoriamente y ponemos abrir el aplicativo.



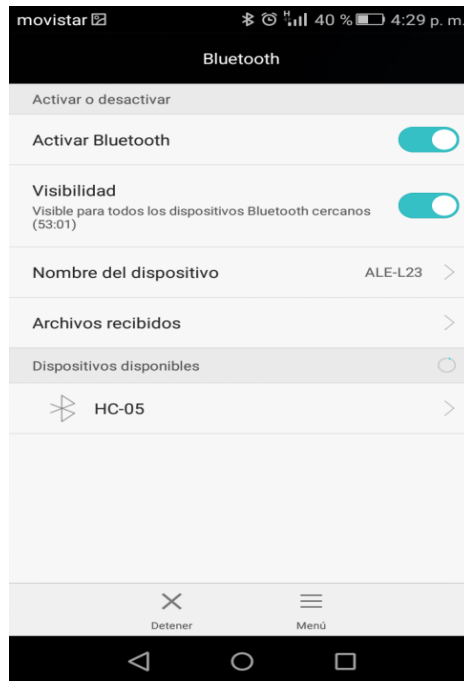
9. A continuación, nos aparece la opción de poder activar el módulo Bluetooth en caso de no tenerlo activo le damos clic en permitir.



10. Seguidamente procedemos a activar el módulo Bluetooth en caso de que no este activo, para poder hacer la conexión correspondiente.



11. A continuación, nos aparece el nombre del módulo Bluetooth para seguir con el paso seleccionamos en HC-05.

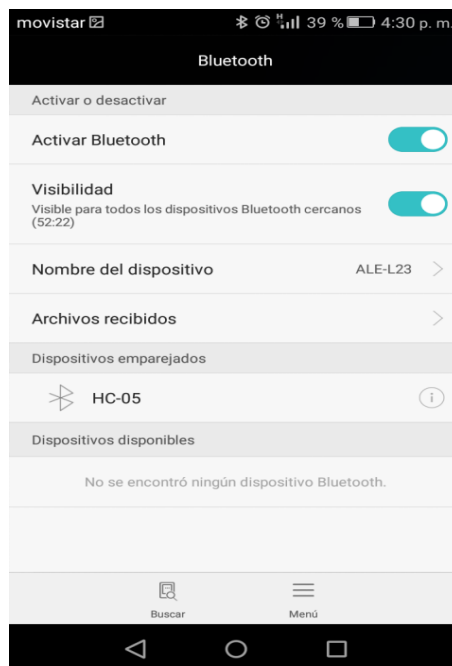


12. A continuación, nos aparece una nueva ventana donde nos pide la contraseña en el recuadro, para ello ponemos una contraseña de **1234** y pulsamos en el botón aceptar.





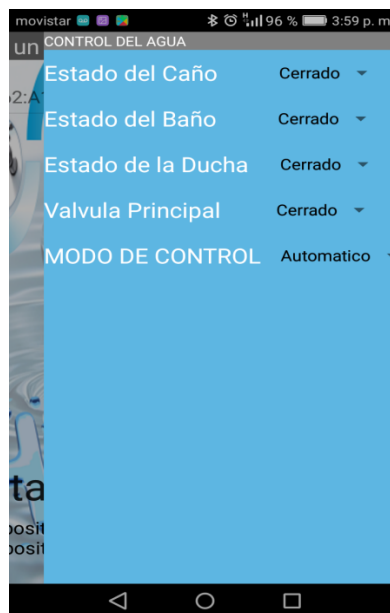
13. Luego de ingresar la contraseña el dispositivo estará emparejado con el modulo Bluetooth.



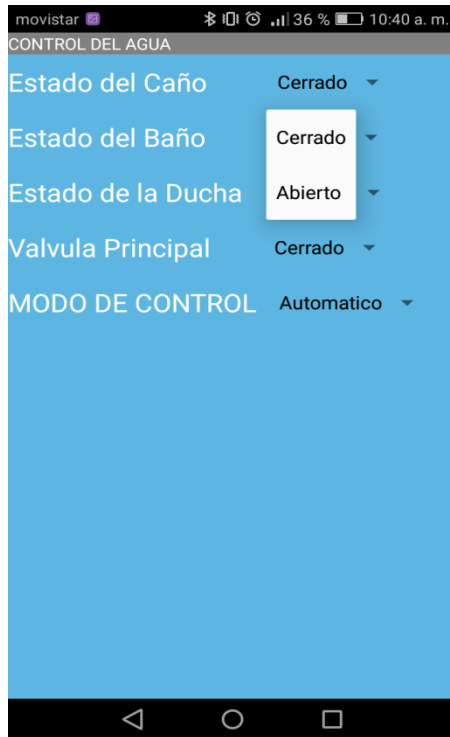
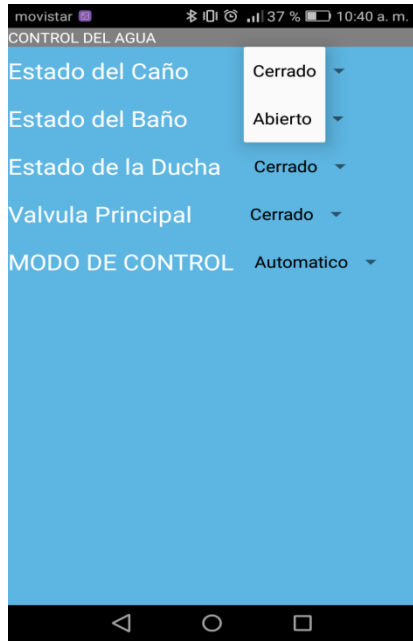
14. Luego de hacer el emparejamiento de Bluetooth procedemos a abrir el aplicativo móvil de control de flujo de agua que esto lo encontramos en la pantalla del dispositivo móvil.

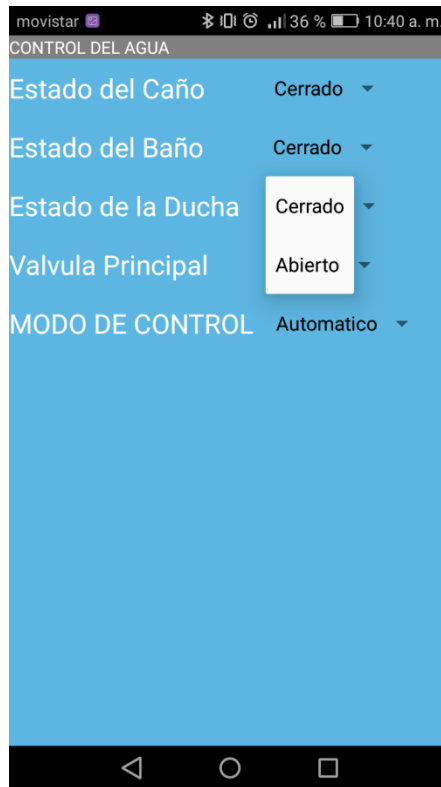


15. A continuación, tendremos la interfaz de control de las válvulas (Inodoro, Ducha y Lava mano).

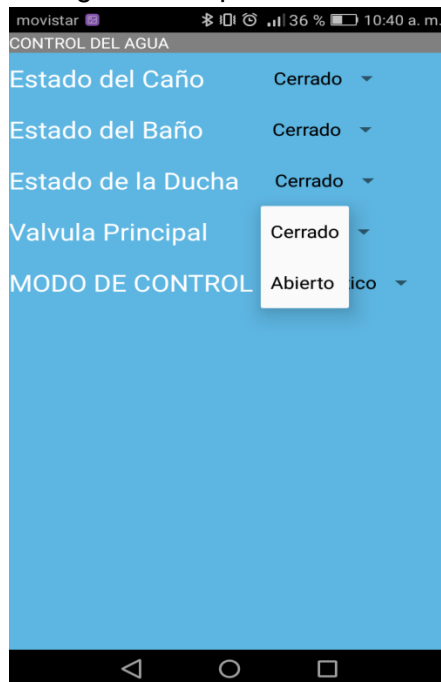


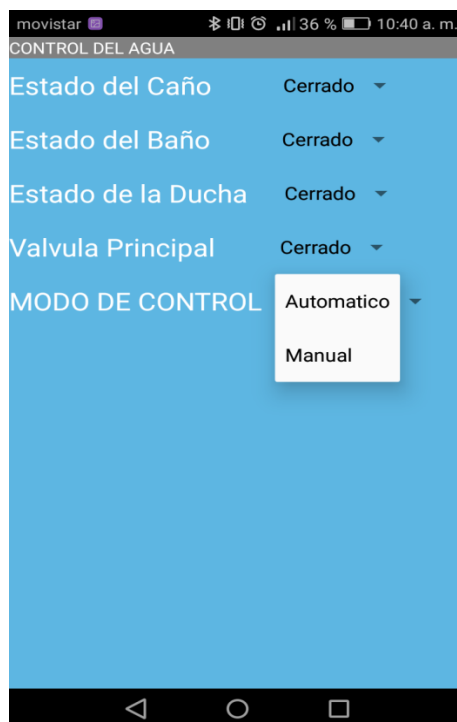
16. Seguidamente podemos pulsar en la barra deslizable para poder encender como apagar ya sea la Ducha, Lava mano y Inodoro.



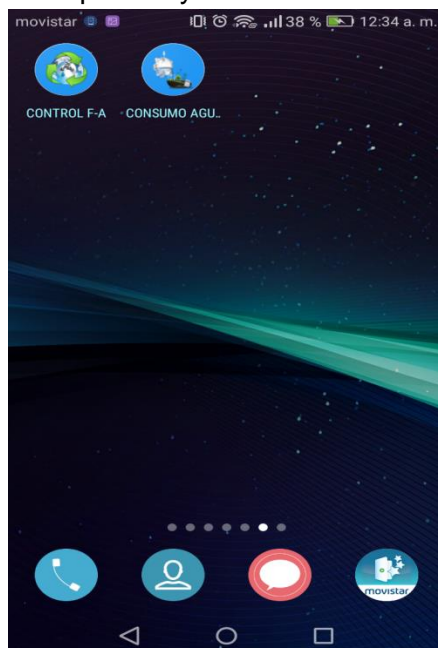


17. A continuación, tendemos la activación de la válvula principal como poner el control en modo manual o modo automático para lo cual seleccionamos en la barra y desplazamos y escogemos la opción.





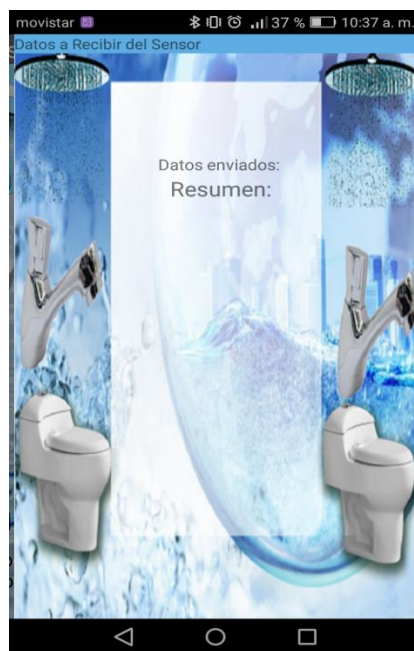
18. De la misma manera podemos acceder a aplicativo de consumo de agua, para poder visualizar el consumo por día y mes en el domicilio.



19. A continuación, también nos conectamos mediante el Bluetooth a la interfaz e nuestro dispositivo para lo cual seleccionamos el dispositivo Bluetooth.

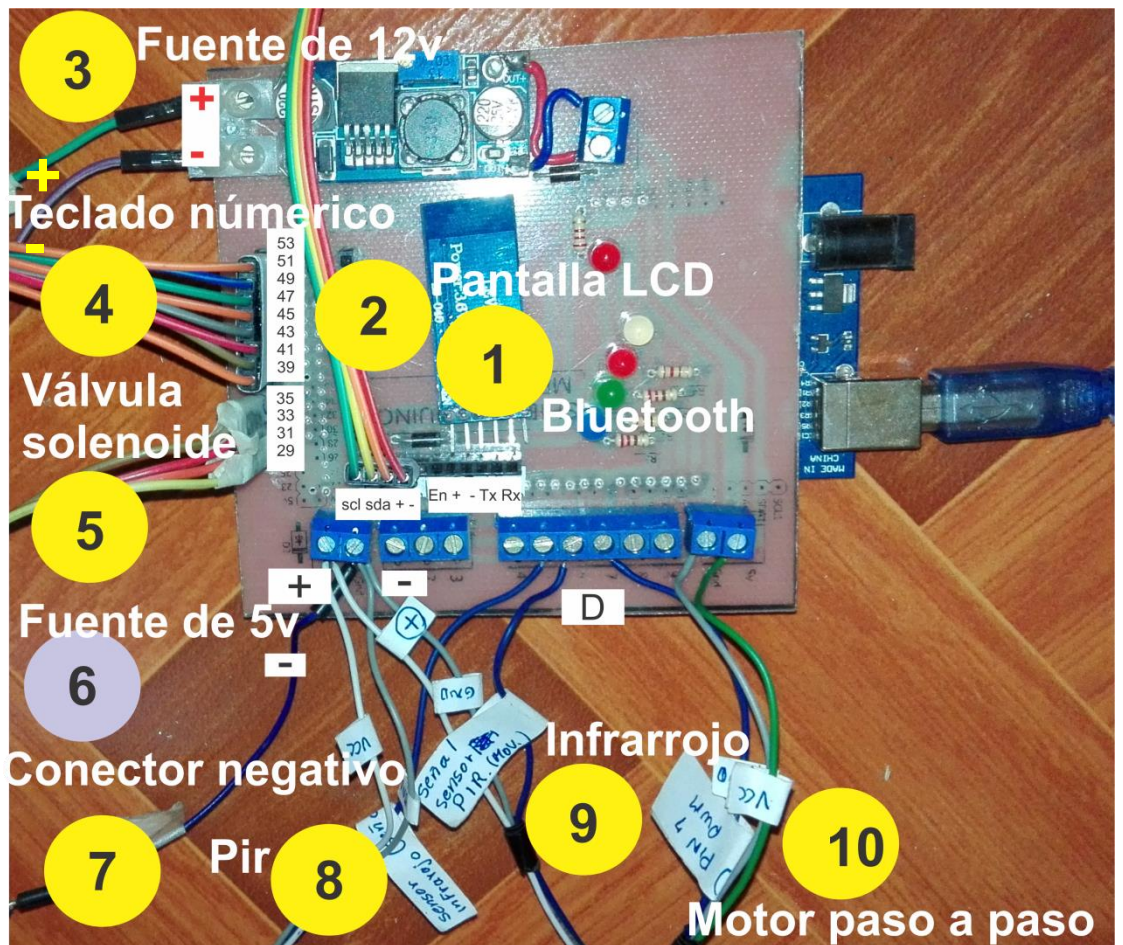


20. Seguidamente tendremos la interfaz donde se visualiza el consumo de agua potable en el domicilio.

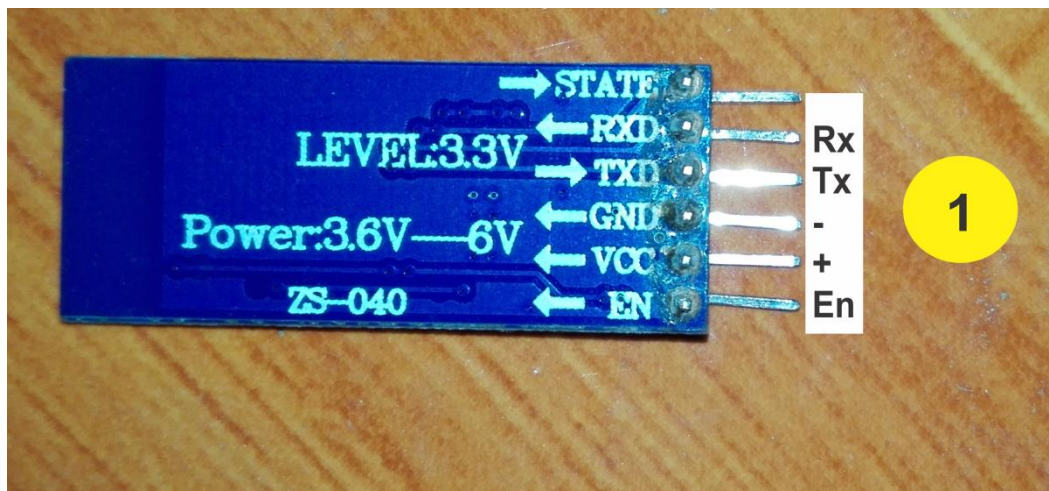


#### ANEXO L: MANUAL DE USUARIO CIRCUITO

1. Circuito del sistema con las salidas y entradas se detalla cada uno de los componentes del circuito enumerados en orden.



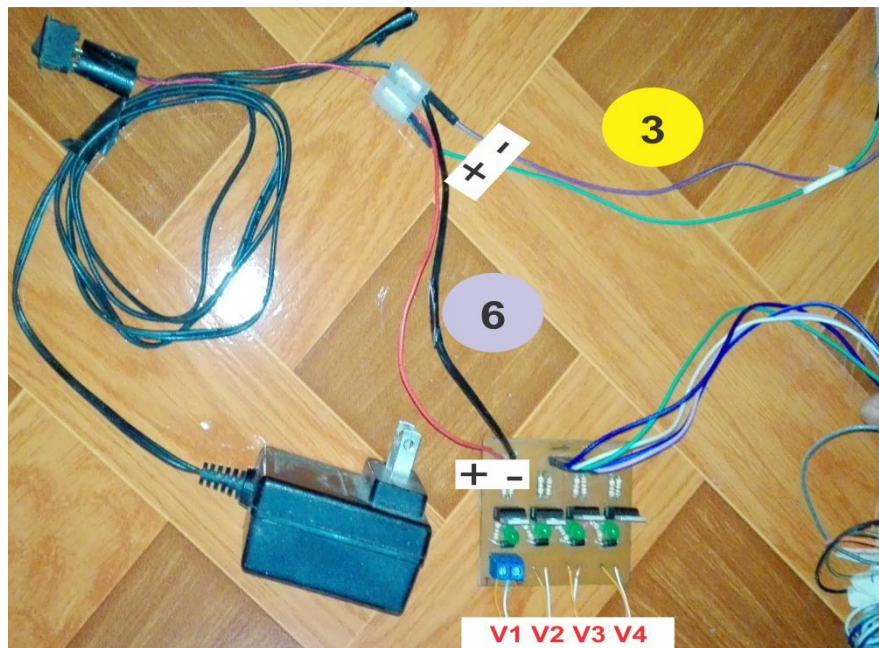
- Bluetooth N°1 conectamos en la posición que muestra la imagen, ingresar los pines con mucho cuidado debido a que puedan sufrir ropturar y no coger el Bluetttoh de un costado sin manchar con el circuito.



- Pantalla LCD N°2 Conectar los cables en la posición que corresponda según a los colores que se muestra, para evitar cruse entre algún componente.



4. Fuente de poder **N°3** este se alimenta de negativo y positivo, en este caso el color verde cumple la función de positivo + el azul de negativo – colocar en esa misma posición al momento de la instalación y suministrar con un cargador de 12 v el regulador de voltaje se calibra con ese suministro de energía.

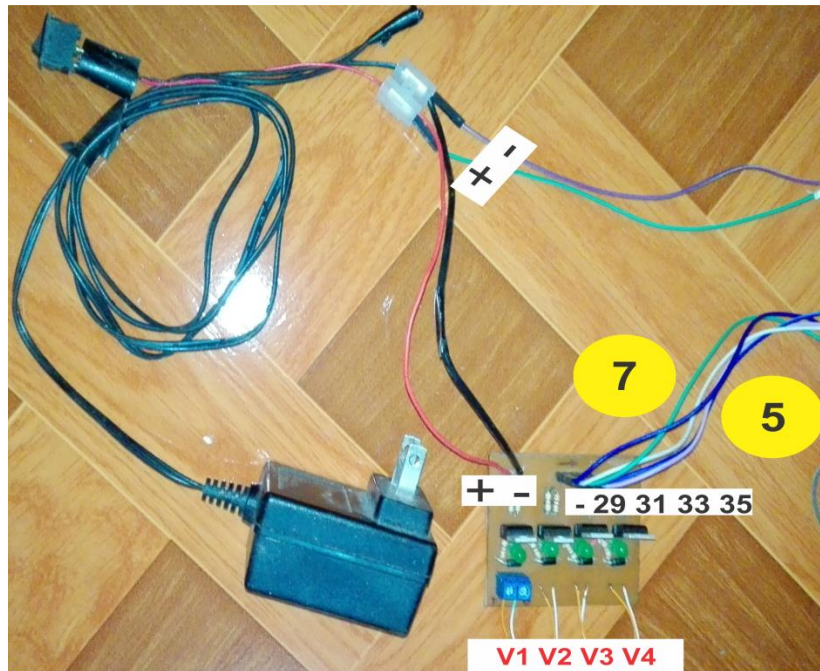


5. Teclado numérico **N°4** poner los pines según al color que se muestra en la imagen en caso contrario no responderá las teclas correspondientes.

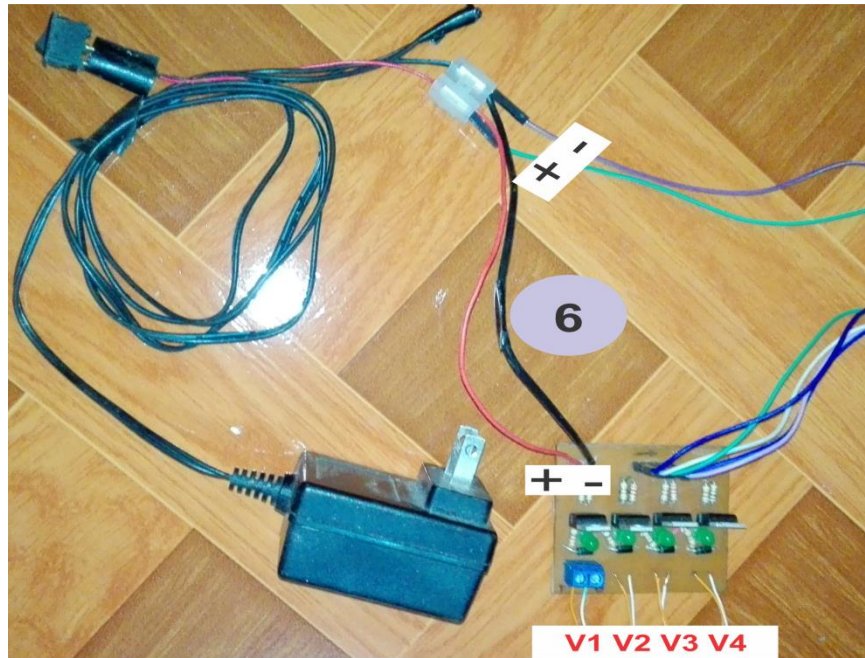




6. Válvula solenoide y conexión a tierra N° 5 y N°7 la válvula solenoide conecte en el orden de la imagen para poder ser mas precisos para detectar los puertos en la placa de las valvulas, la conexión del cable azul que es negativo - conectar donde se encuentran las conexiones negativas.



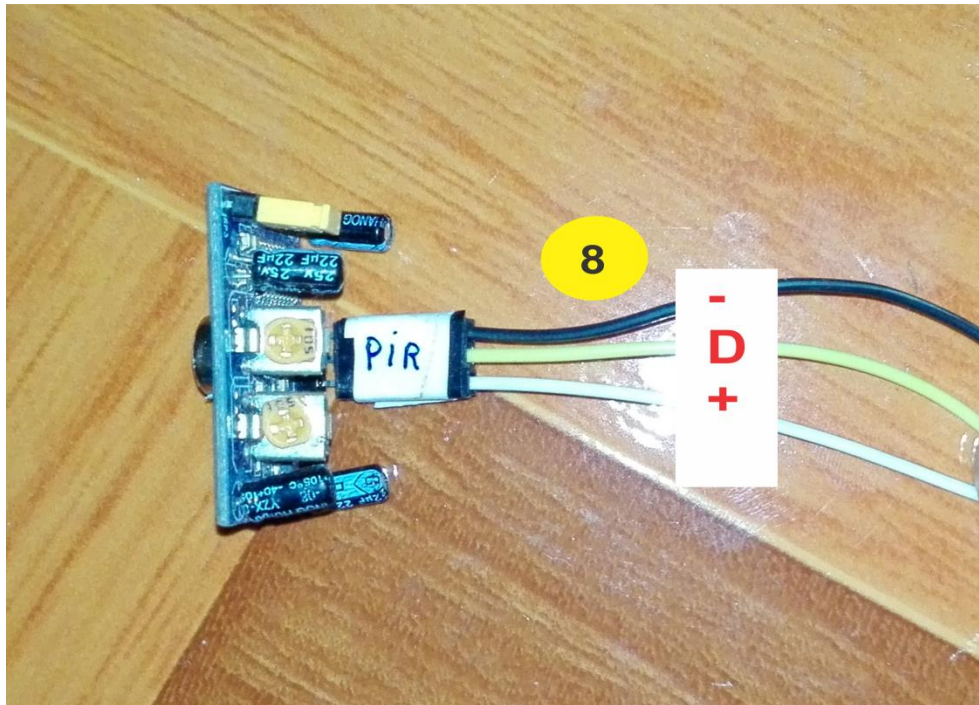
7. Alimentación a la placa de la válvula 12v N°6 y salidas de V1, V2, V3 y V4 el cable rojo en positivo + y el negro es negativo - .



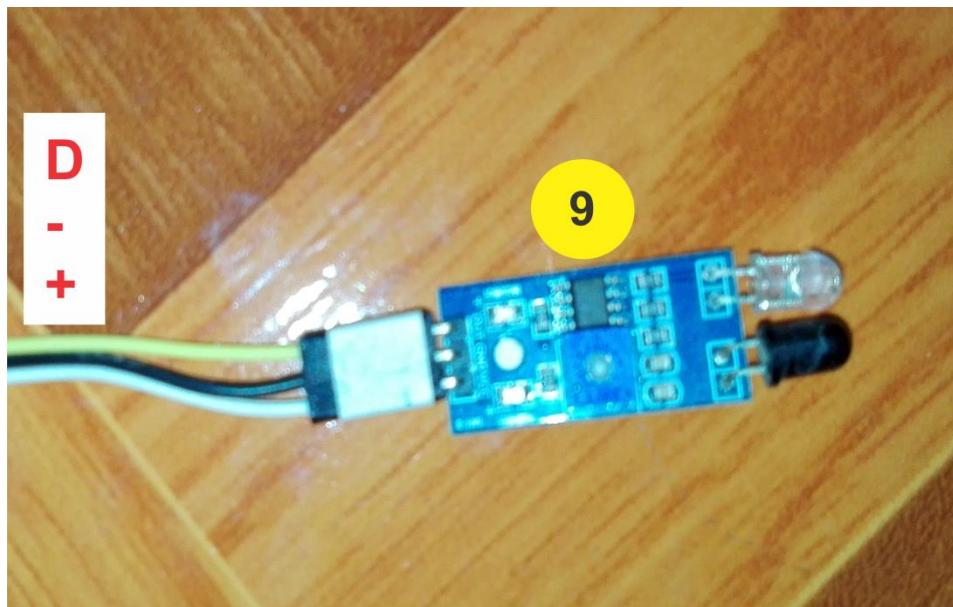
8. Seguidamente se conecta las tres válvulas no tiene polaridad.



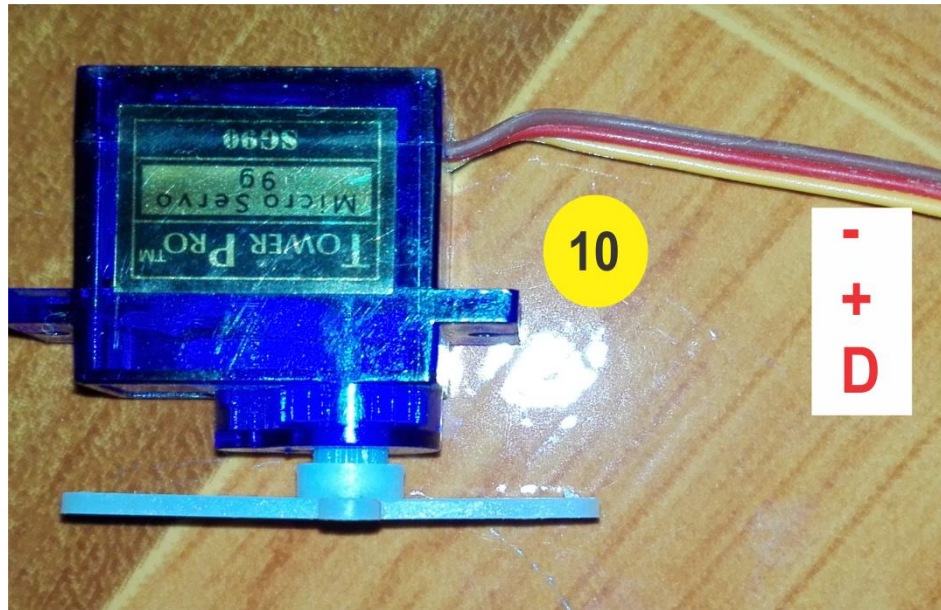
9. Sensor PIR N° 8 Conectar según a los colores de los cables blanco voltaje + plomo negativo - y el azul que datos.



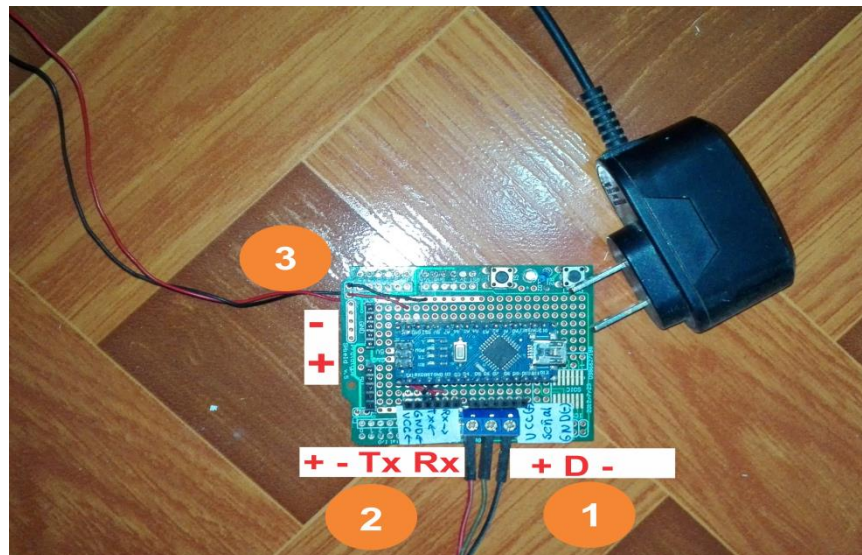
10. Infrarrojo N°9 Conectar según al orden los colores de los cables blanco voltaje + plomo negativo - y el azul que datos.



11. Motor paso a paso N°10 conectar según al orden los colores de los cables verde + plomo negativo - y el azul que datos.



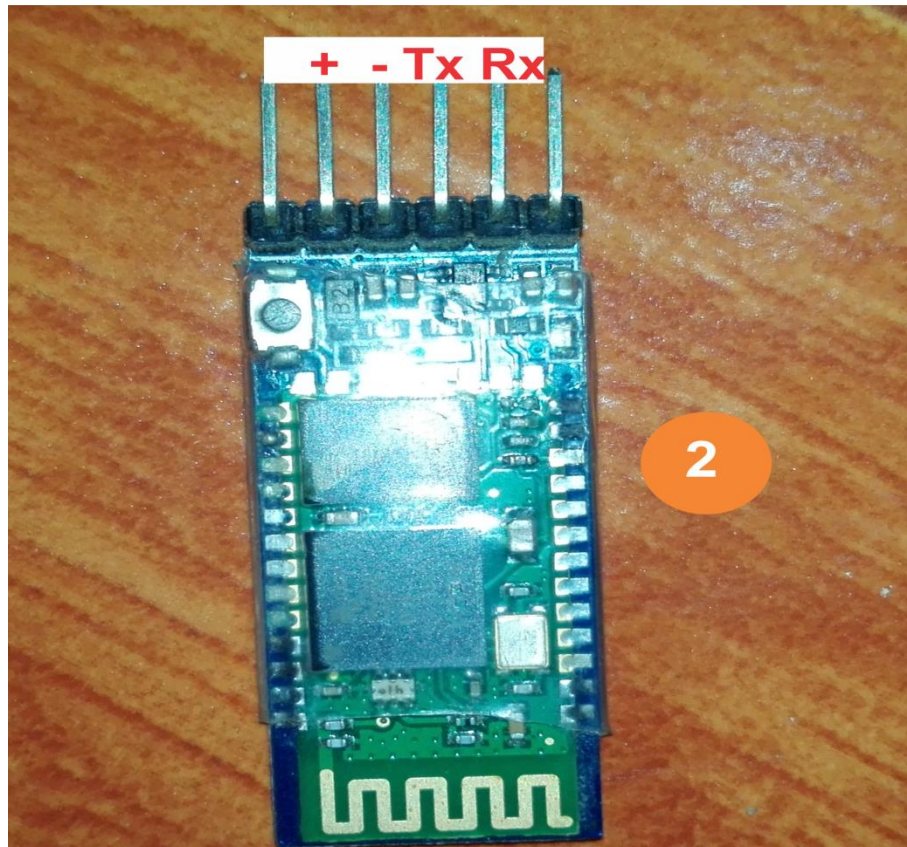
12. Seguidamente se hace la instalación del sensor de flujo de agua con el arduino nano.



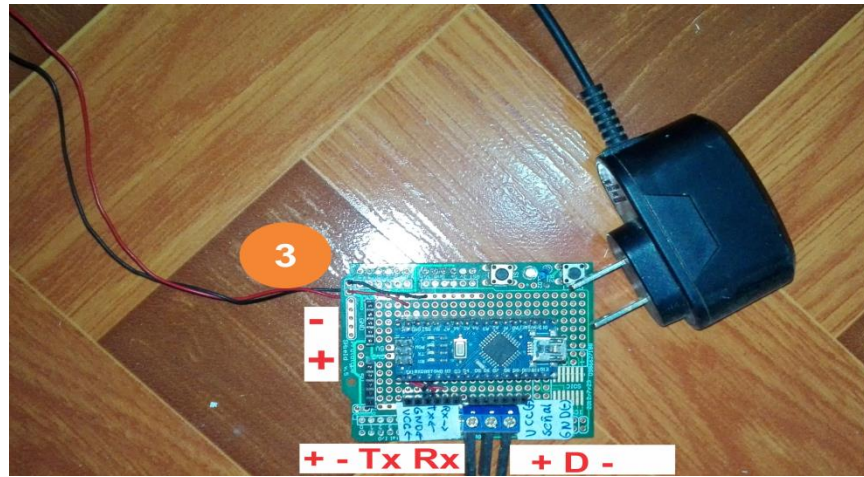
13. Seguidamente N° 1 se conecta en sensor de flujo de agua con el orden especificado, cable rojo + cable marroj **datos** y negro -.



14. Como paso siguiente **N° 2** se hace la conexión del Bluetooth en ese orden que se especifica.



15. Como paso final **N°3** alimentamos al Arduino mega con 5v para lo cual utilizamos un cargado de celular.



ANEXO M: SOLICITUD DE DATOS A ATALSAC TALAVERA

**EMPRESA AGUAS  
DE TALAVERA S.A.C.**

**ESPECIE VALORADA**

Nº 0000777

**FORMULARIO ÚNICO DE TRAMITE**

SOLICITA: duplicado de registros de agua  
del mes de agosto a diciembre de 2016

SEÑOR GERENTE DE LA EMPRESA AGUAS DE TALAVERA S.A.C.

YO: Jose Antonio Rojas

Identificado con D.N.I. Nº 46369753

Con domicilio en J.P. Belgrano

Telf. : 966 444 675

Ante Ud. con el debido respeto me presento y digo:

Que, teniendo la necesidad de

desarrollar un proyecto de investigación tecnológica en  
con el fin de optimizar el uso del agua potable  
en el distrito de Talavera y así contribuir en el  
ahorro en este recurso que es muy importante

**POR LO EXPUESTO:**

Ruego a usted señor  
Gerente que asepte mi petición

Talavera, 6 DE abril DE 2017

[Firma]

FIRMA

D.N.I. Nº 46369753

EMPRESA AGUAS TALAVERA  
RECEPCION DE DOCUMENTOS

06 ABR 2017

Folio: 202 Hora: 8:47 AM



CENTENO ROJAS LINO A  
JR. BELGICA

Para Consultas  
Suministro N°  
**001491**

RECIBO N°

**121885**

Información general				Información de pago	
Titular de la conexión:	CENTENO ROJAS LINO A			Referencia de cobro:	001491
Dirección de suministro:	JR. BELGICA	Distrito:		Mes facturado:	Set 2016
Frecuencia de facturación:	Mensual	Tipo de facturación:		Período de consumo:	
Categoría:		Tarifa:	DOMESTICO II	Fecha de emisión:	10/10/16
	Unidad de uso:	Actividad:		Fecha de vencimiento:	31/10/16

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
585020	5031.00	5049.00	18.00	CONSUMO DEL MES	3.50

Información complementaria

IGV	0.63
EXCESO	0.00
ALCANTARILLADO	0.00
MEDIDOR	0.00
REDONDEO ANTERIOR	-0.03
REDONDEO	0.08

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

**IMPORTE TOTAL A PAGAR S/. \*\*\*4.10**

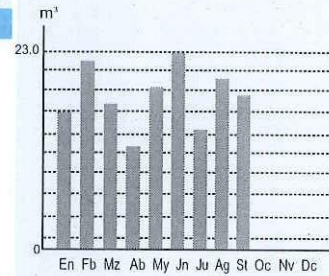


Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SÓLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGÚN CASO AL MENSAJERO

**Números de Consultas**

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

**Evolución de su consumo de agua**



**Reservado para la oficina de cobro**

<b>JR. BELGICA</b>	
Referencia de cobro:	Suministro
121885	001491
Vencimiento:	Importe
31/10/16	S/. ***4.10

**CENTENO ROJAS LINO**



Cod. Suministro: 001491







CENTENO ROJAS LINO A  
JR. BELGICA

Para Consultas  
Suministro N°

**001491**

RECIBO N°

**124513**

Información general				Información de pago	
Titular de la conexión:	CENTENO ROJAS LINO A			Referencia de cobro:	001491
Dirección de suministro:	JR. BELGICA		Distrito:	Período de consumo:	Oct 2016
Frecuencia de facturación:	Mensual	Tipo de facturación:	Tarifa:	Fecha de emisión:	08/11/16
Categoría:		Unidad de uso:	Actividad:	Fecha de vencimiento:	30/11/16

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
585020	5049.00	5070.00	21.00	CONSUMO DEL MES	3.50

Información complementaria	
	IGV
	EXCESO
	ALCANTARILLADO
	MEDIDOR
	REDONDEO ANTERIOR
	REDONDEO

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

**IMPORTE TOTAL A PAGAR** S/. \*\*\*4.34

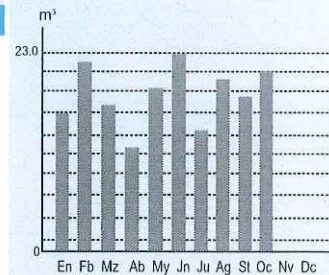


Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SÓLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGÚN CASO AL MENSAJERO

**Números de Consultas**

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

**Evolución de su consumo de agua**



© Sistema desarrollado por MAXINETI SAC email: informes@maxineti.com

**Reservado para la oficina de cobro**

<b>JR. BELGICA</b>	
Referencia de cobro:	Suministro
124513	001491
Vencimiento:	Importe
30/11/16	S/. ***4.34

**CENTENO ROJAS LINO**



Cod. Suministro: 001491





CENTENO ROJAS LINO A  
JR. BELGICA

Para Consultas  
Suministro N°

**001491**

RECIBO N°

**127141**

Información general				Información de pago	
Titular de la conexión:	CENTENO ROJAS LINO A			Referencia de cobro:	Mes facturado:
Dirección de suministro:	JR. BELGICA	Diáritmo:		<b>001491</b>	<b>Nov 2016</b>
Frecuencia de facturación:	Mensual	Tipo de facturación:		Periodo de consumo:	
Categoría:		Tarifa:	DOMESTICO II	Fecha de emisión:	07/12/16
	Unidad de uso:	Actividad:		Fecha de vencimiento:	31/12/16

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
585020	5070.00	5089.00	19.00	CONSUMO DEL MES	3.50

Información complementaria	
	IGV
	EXCESO
	ALCANTARILLADO
	MEDIDOR
	REDONDEO ANTERIOR
	REDONDEO

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

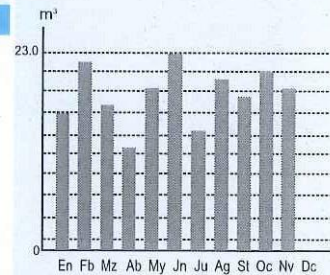
**IMPORTE TOTAL A PAGAR** S/. \*\*\*4.10



Este recibo adquiere valor solamente si posee certificación de cobro. Su pago no cancela deudas anteriores. CANCELAR SOLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGÚN CASO AL MENSAJERO

Números de Consultas

Evolución de su consumo de agua



usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

© Sistema desarrollado por MAXINETI SAC email: informes@maxineti.com

Reservado para la oficina de cobro

<b>JR. BELGICA</b>	
Referencia de cobro:	Suministro
127141	001491
Vencimiento:	Importe
31/12/16	S/. ***4.10

CENTENO ROJAS LINO



Cod. Suministro: 001491





CENTENO ROJAS LINO A  
JR. BELGICA

Para Consultas  
Suministro N°

**001491**

RECIBO N°

**129769**

Información general				Información de pago	
Titular de la conexión:	CENTENO ROJAS LINO A			Referencia de cobro:	001491
Dirección de suministro:	JR. BELGICA	Distrito:		Mes facturado:	Dic 2016
Frecuencia de facturación:	Mensual	Tipo de facturación:	DOMESTICO II	Periodo de consumo:	
Categoría:		Unidad de uso:		Fecha de emisión:	10/01/17
		Actividad:		Fecha de vencimiento:	31/01/17

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
585020	5089.00	5011.00	22.00	CONSUMO DEL MES	3.50

Información complementaria

IGV	0.63
EXCESO	0.00
ALCANTARILLADO	0.00
MEDIDOR	0.00
REDONDEO ANTERIOR	-0.06
REDONDEO	0.03

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

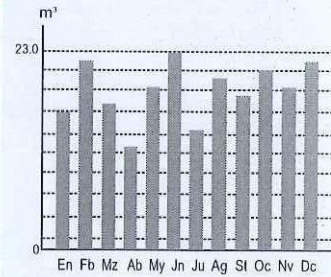
**IMPORTE TOTAL A PAGAR S/. \*\*\*4.60**



Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SOLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGUN CASO AL MENSAJERO

Números de Consultas

Evolución de su consumo de agua



usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

© Sistema desarrollado por MAXINETI SAC email: informes@maxineti.com

Reservado para la oficina de cobro

JR. BELGICA

Referencia de cobro:	Suministro:
129769	001491
Vencimiento:	Importe:
31/01/17	S/. ***4.60

CENTENO ROJAS LINO



Cod. Suministro: 001491





CENTENO ROJAS LINO A  
JR. BELGICA

Para Consultas  
Suministro N°

**001491**

RECIBO N°

**135025**

Información general				Información de pago	
Titular de la conexión:	CENTENO ROJAS LINO A			Referencia de cobro:	001491
Dirección de suministro:	JR. BELGICA	Distrito:		Mes facturado:	Feb 2017
Frecuencia de facturación:	Mensual	Tipo de facturación:		Periodo de consumo:	
Categoría:		Tarifa:	DOMESTICO II	Fecha de emisión:	08/03/17
	Unidad de uso:	Actividad:		Fecha de vencimiento:	31/03/17

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
585020	5126.00	5139.00	13.00	CONSUMO DEL MES	3.50

Información complementaria		Detalle de facturación	
		Concepto	Importe
		IGV	0.63
		EXCESO	0.00
		ALCANTARILLADO	0.00
		MEDIDOR	0.00
		REDONDEO ANTERIOR	-0.01
		REDONDEO	0.03

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

**IMPORTE TOTAL A PAGAR S/. \*\*\*4.10**

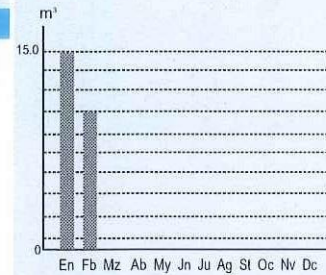


Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SOLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGUN CASO AL MENSAJERO

**Números de Consultas**

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular. Hasta 15 marzo

**Evolución de su consumo de agua**



© Sistema desarrollado por MAXINETI SAC email: Informes@maxineti.com

Reservado para la oficina de cobro

<b>JR. BELGICA</b>	
Referencia de cobro:	Suministro
135025	001491
Vencimiento	Importe
31/03/17	S/. ***4.10

**CENTENO ROJAS LINO**



Cod. Suministro: 001491





CENTENO ROJAS LINO A  
JR. BELGICA

Para Consultas  
Suministro N°

**001491**

RECIBO N°

**132397**

Información general				Información de pago	
Título de la conexión:	CENTENO ROJAS LINO A			Referencia de cobro:	001491
Dirección de suministro:	JR. BELGICA	Distrito:		Mes facturado:	Ene 2017
Frecuencia de facturación:	Mensual	Tipo de facturación:		Periodo de consumo:	
Categoría:		Tarifa:	DOMESTICO II	Fecha de emisión:	10/02/17
	Unidad de uso:	Actividad:		Fecha de vencimiento:	28/02/17

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
585020	5011.00	5026.00	15.00	CONSUMO DEL MES	3.50

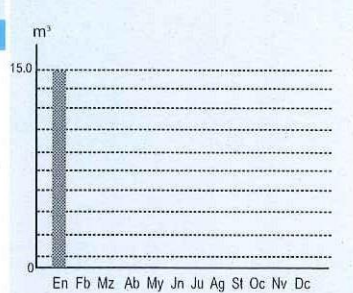
Información complementaria	
	IGV
	EXCESO
	ALCANTARILLADO
	MEDIDOR
	REDONDEO ANTERIOR
	REDONDEO

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

**IMPORTE TOTAL A PAGAR S/. \*\*\*4.10**

Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SÓLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGÚN CASO AL MENSAJERO

Evolución de su consumo de agua



Números de Consultas

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

© Sistema desarrollado por MAXINETI SAC email: Informes@maxineti.com

Reservado para la oficina de cobro

<b>JR. BELGICA</b>	
Referencia de cobro:	Suministro
132397	001491
Vencimiento	Importe
28/02/17	S/. ***4.10

CENTENO ROJAS LINO



Cod. Suministro: 001491





CENTENO ROJAS LINO A  
JR. BELGICA

Para Consultas  
Suministro N°

**001491**

RECIBO N°

**137653**

Información general				Información de pago	
Titular de la conexión:	CENTENO ROJAS LINO A			Referencia de cobro:	Mes facturado:
Dirección de suministro:	JR. BELGICA	Distrito:		<b>001491</b>	<b>Mar 2017</b>
Frecuencia de facturación:	Mensual	Tipo de facturación:		Período de consumo:	
Categoría:		Tarifa:	DOMESTICO II	Fecha de emisión:	11/04/17
	Unidad de uso:	Actividad:		Fecha de vencimiento:	30/04/17

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
585020	5139.00	5154.00	15.00	CONSUMO DEL MES	3.50

Información complementaria

IGV	0.63
EXCESO	0.00
ALCANTARILLADO	0.00
MEDIDOR	0.00
REDONDEO ANTERIOR	-0.03
REDONDEO	0.02

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

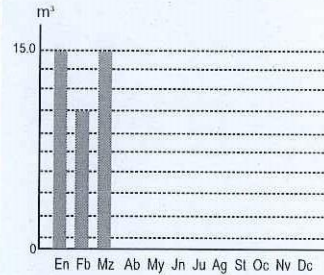
**IMPORTE TOTAL A PAGAR S/. \*\*\*4.10**



Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SOLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGUN CASO AL MENSAJERO

Números de Consultas

Evolución de su consumo de agua



© Sistema desarrollado por MAXINETI SAC email: Informes@maxineti.com

Reservado para la oficina de cobro

JR. BELGICA	
Referencia de cobro:	Suministro:
137653	001491
Vencimiento:	Importe:
30/04/17	S/. ***4.10

CENTENO ROJAS LINO



Cod. Suministro: 001491





ALCCA HUILLCAHUAYA VANESSA ADA  
JR. MASURACCRA 229

Para Consultas  
Suministro N°  
**003450**

RECIBO N°

**136850**

Información general				Información de pago	
Titular de la conexión:	ALCCA HUILLCAHUAYA VANESSA ADA			Referencia de cobro:	Mes facturado:
Dirección de suministro:	JR. MASURACCRA 229	Distrito:		<b>003450</b>	<b>Feb 2017</b>
Frecuencia de facturación:	Mensual	Tipo de facturación:		Período de consumo:	
Categoría:		Tarifa:	DOMESTICO I	Fecha de emisión:	08/03/17
	Unidad de uso:	Actividad:		Fecha de vencimiento:	31/03/17

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
5584	443.00	454.00	11.00	CONSUMO DEL MES	3.50

Información complementaria	
IGV	0.63
EXCESO	0.00
ALCANTARILLADO	0.00
MEDIDOR	0.00
DEUDA ANTERIOR	4.10
REDONDEO ANTERIOR	-0.07
REDONDEO	0.04

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

**IMPORTE TOTAL A PAGAR S/. \*\*\*8.20**



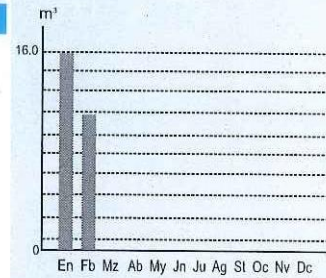
Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SÓLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGÚN CASO AL MENSAJERO

**Números de Consultas**

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

© Sistema desarrollado por MAXINETI SAC email: informes@maxineti.com

**Evolución de su consumo de agua**



**Reservado para la oficina de cobro**

<b>JR. MASURACCRA</b>	
Referencia de cobro:	Suministro
136850	003450
Vencimiento	Importe
31/03/17	<b>S/. ***8.20</b>

ALCCA HUILLCAHUAYA VANESSA ADA



Cod. Suministro: 003450





ALCCA HUILLCAHUAYA VANESSA ADA  
JR. MASURACCRA 229

Para Consultas  
Suministro N°  
**003450**  
RECIBO N°  
**134222**

Información general				Información de pago	
Titular de la conexión:	ALCCA HUILLCAHUAYA VANESSA ADA			Referencia de cobro:	003450
Dirección de suministro:	JR. MASURACCRA 229	Distrito:		Mes facturado:	Ene 2017
Frecuencia de facturación:	Mensual	Tipo de facturación:	DOMESTICO I	Período de consumo:	
Categoría:		Unidad de uso:		Fecha de emisión:	10/02/17
		Actividad:		Fecha de vencimiento:	28/02/17

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
5584	427.00	443.00	16.00	CONSUMO DEL MES	3.50

Información complementaria		Concepto	Importe
		IGV	0.63
		EXCESO	0.00
		ALCANTARILLADO	0.00
		MEDIDOR	0.00
		DEUDA ANTERIOR	0.00
		REDONDEO ANTERIOR	-0.03
		REDONDEO	0.07

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

**IMPORTE TOTAL A PAGAR** S/. \*\*\*4.10



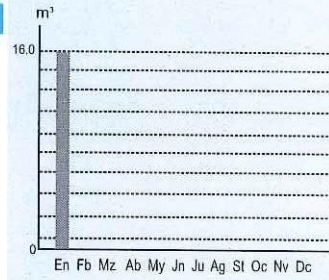
Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SÓLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGÚN CASO AL MENSAJERO

**Números de Consultas**

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

© Sistema desarrollado por MAXINETI SAC email: Informes@maxineti.com

**Evolución de su consumo de agua**



**Reservado para la oficina de cobro**

<b>JR. MASURACCRA</b>	
Referencia de cobro: 134222	Suministro: 003450
Vencimiento: 28/02/17	Importe: S/. ***4.10

ALCCA HUILLCAHUAYA VANESSA ADA



Cod. Suministro: 003450







ALCCA HUILLCAHUAYA VANESSA ADA  
JR. MASURACCRA 229

Para Consultas  
Suministro N°

**003450**

RECIBO N°

**131583**

Información general				Información de pago	
Titular de la conexión:	ALCCA HUILLCAHUAYA VANESSA ADA			Referencia de cobro:	003450
Dirección de suministro:	JR. MASURACCRA 229	Dirección:		Mes facturado:	Dic 2016
Frecuencia de facturación:	Mensual	Tipo de facturación:		Período de consumo:	
Categoría:		Tarifa:	DOMESTICO I	Fecha de emisión:	10/01/17
Unidad de uso:		Actividad:		Fecha de vencimiento:	31/01/17

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m <sup>3</sup> )	Concepto	Importe
5584	412.00	427.00	15.00	CONSUMO DEL MES	3.50

Información complementaria

IGV	0.63
EXCESO	0.00
ALCANTARILLADO	0.00
MEDIDOR	0.00
DEUDA ANTERIOR	4.20
REDONDEO ANTERIOR	-0.03
REDONDEO	0.01

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

**IMPORTE TOTAL A PAGAR S/. \*\*\*8.30**



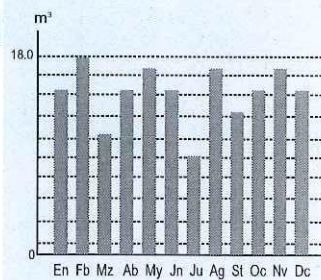
Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SOLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGUN CASO AL MENSAJERO

Números de Consultas

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

© Sistema desarrollado por MAXINETI SAC email: Informes@maxineti.com

Evolución de su consumo de agua



Reservado para la oficina de cobro

JR. MASURACCRA

Referencia de cobro:	131583	Suministro:	003450
Vencimiento:	31/01/17	Importe:	S/. ***8.30

ALCCA HUILLCAHUAYA VANESSA ADA



Cod. Suministro: 003450





ALCCA HUILLCAHUAYA VANESSA ADA  
JR. MASURACCRA

Para Consultas  
Suministro N°

**003450**

RECIBO N°

**123699**

Información general				Información de pago	
Titular de la conexión:	ALCCA HUILLCAHUAYA VANESSA ADA			Referencia de cobro:	003450
Dirección de suministro:	JR. MASURACCRA 229	Distrito:		Mes facturado:	Set 2016
Frecuencia de facturación:	Mensual	Tipo de facturación:		Período de consumo:	
Categoría:		Tarifa:	DOMESTICO I	Fecha de emisión:	10/10/16
	Unidad de uso:	Actividad:		Fecha de vencimiento:	31/10/16

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
5584	367.00	380.00	13.00	CONSUMO DEL MES	3.50

**Información complementaria**

IGV	0.63
EXCESO	0.00
ALCANTARILLADO	0.00
MEDIDOR	0.00
DEUDA ANTERIOR	4.10
REDONDEO ANTERIOR	-0.07
REDONDEO	0.04

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

**IMPORTE TOTAL A PAGAR S/. \*\*\*8.30**

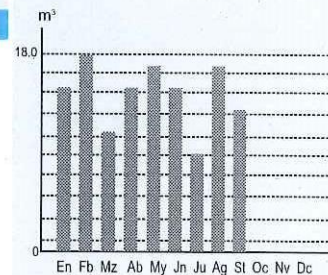


Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores, CANCELAR SÓLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGUN CASO AL MENSAJERO

**Números de Consultas**

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

**Evolución de su consumo de agua**



© Sistema desarrollado por MAXINETI SAC email: informes@maxineti.com

**Reservado para la oficina de cobro**

<b>JR. MASURACCRA</b>		<b>ALCCA HUILLCAHUAYA VANESSA ADA</b>	
Referencia de cobro:	123699	Suministro:	003450
Vencimiento:	31/10/16	Importe:	S/. ***8.30



Cod. Suministro: 003450





ALCCA HUILLCAHUAYA VANESSA ADA  
JR. MASURACCRA 229

Para Consultas  
Suministro N°  
**003450**  
RECIBO N°  
**126327**

Información general				Información de pago	
Titular de la conexión:	ALCCA HUILLCAHUAYA VANESSA ADA			Referencia de cobro:	Mes facturado:
Dirección de suministro:	JR. MASURACCRA 229	Distrito:		<b>003450</b>	<b>Oct 2016</b>
Frecuencia de facturación:	Mensual	Tipo de facturación:	DOMESTICO I	Período de consumo:	
Categoría:		Unidad de uso:		Fecha de emisión:	08/11/16
		Actividad:		Fecha de vencimiento:	30/11/16

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
5584	380.00	395.00	15.00	CONSUMO DEL MES	3.50

Información complementaria	
	IGV
	EXCESO
	ALCANTARILLADO
	MEDIDOR
	REDONDEO ANTERIOR
	REDONDEO

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

**IMPORTE TOTAL A PAGAR** S/. \*\*\*4.10



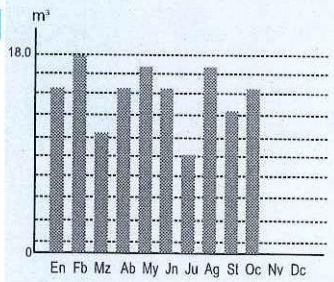
Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SÓLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGÚN CASO AL MENSAJERO

Números de Consultas

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

© Sistema desarrollado por MAXINETI SAC email: informes@maxineti.com

Evolución de su consumo de agua



Reservado para la oficina de cobro

JR. MASURACCRA	
Referencia de cobro: 126327	Suministro: 003450
Vencimiento: 30/11/16	Importe: S/. ***4.10

ALCCA HUILLCAHUAYA VANESSA ADA



Cod. Suministro: 003450





ALCCA HUILLCAHUAYA VANESSA ADA  
JR. MASURACCRA 229

Para Consultas  
Suministro N°

**003450**

RECIBO N°

**128955**

Información general				Información de pago	
Titular de la conexión:	ALCCA HUILLCAHUAYA VANESSA ADA			Referencia de cobro:	003450
Dirección de suministro:	JR. MASURACCRA 229	Distrito:		Mes facturado:	Nov 2016
Frecuencia de facturación:	Mensual	Tipo de facturación:	DOMESTICO-I	Período de consumo:	
Categoría:		Unidad de uso:		Fecha de emisión:	07/12/16
		Actividad:		Fecha de vencimiento:	31/12/16

Registro del medidor				Detalle de facturación	
Medidor	5584	Consumo (m3)	17.00	Concepto	Importe
Lectura anterior	395.00	Lectura actual	412.00	CONSUMO DEL MES	3.50

**Información complementaria**

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

IGV	0.63
EXCESO	0.00
ALCANTARILLADO	0.00
MEDIDOR	0.00
REDONDEO ANTERIOR	-0.07
REDONDEO	0.04

**IMPORTE TOTAL A PAGAR S/. \*\*\*4.20**



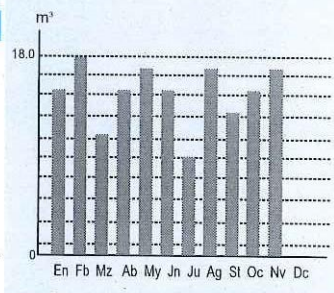
Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SÓLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGÚN CASO AL MENSAJERO

**Números de Consultas**

usuario urgente para empadronarse antiguos y nuevos, requisito: DNI, una fotografía tamaño pasaporte, recibo de agua del titular.

© Sistema desarrollado por MAXINETI SAC email: Informes@maxineti.com

**Evolución de su consumo de agua**



Reservado para la oficina de cobro

<b>JR. MASURACCRA</b>	
Referencia de cobro:	Suministro
128955	003450
Vencimiento	Importe
31/12/16	S/. ***4.20

ALCCA HUILLCAHUAYA VANESSA ADA





ALCCA HUILLCAHUAYA VANESSA ADA  
JR. MASURACCRA 229

Para Consultas  
Suministro N°

**003450**

RECIBO N°

**139478**

Información general			Información de pago	
Titular de la conexión:	ALCCA HUILLCAHUAYA VANESSA ADA		Referencia de cobro:	Mes facturado:
Dirección de suministro:	JR. MASURACCRA 229	Distrito:	<b>003450</b>	<b>Mar 2017</b>
Frecuencia de facturación:	Mensual	Tipo de facturación:	Periodo de consumo:	
Categoría:		Tarifa:	Fecha de emisión:	09/04/17
	Unidad de uso:	Actividad:	DOMESTICO I	Fecha de vencimiento:
				30/04/17

Registro del medidor				Detalle de facturación	
Medidor	Lectura anterior	Lectura actual	Consumo (m3)	Concepto	Importe
5584	454.00	466.00	12.00	CONSUMO DEL MES	3.50
<b>Información complementaria</b>					
IGV					
EXCESO					
ALCANTARILLADO					
MEDIDOR					
REDONDEO ANTERIOR					
REDONDEO					

\* Pagando tu recibo a tiempo contribuyes a que otros tengan agua

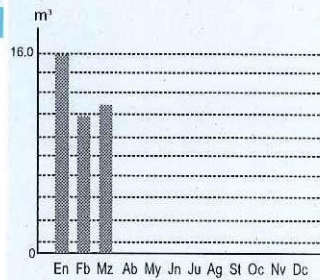
**IMPORTE TOTAL A PAGAR S/. \*\*\*4.10**



Este recibo adquiere valor solamente si posee certificación de cobro, su pago no cancela deudas anteriores. CANCELAR SÓLO EN LUGARES AUTORIZADOS INDICADOS AL REVERSO EN NINGÚN CASO AL MENSAJERO

Números de Consultas

Evolución de su consumo de agua



© Sistema desarrollado por MAXINETI SAC email: informes@maxineti.com

Reservado para la oficina de cobro

<b>JR. MASURACCRA</b>		<b>ALCCA HUILLCAHUAYA VANESSA ADA</b>	
Referencia de cobro:	Suministro	Referencia de cobro:	Suministro
139478	003450	139478	003450
Vencimiento	Importe	Vencimiento	Importe
30/04/17	S/. ***4.10	30/04/17	S/. ***4.10



Cod. Suministro: 003450



## ANEXO N: ENCUESTAS VIVIENDA LINO CENTENO ROJAS

L1

Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1	2	3	4	5

Pregunta N° 1:

- ¿Cómo fue navegar por toda las interfaces del aplicación? 2

Pregunta N° 2:

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema? 4

Pregunta N° 3:

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual? 5

Pregunta N° 4:

- ¿Cómo fue para usted interactuar con cada uno de los sensores? 4

Pregunta N° 5:

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema? 3

Pregunta N° 6:

- ¿Cómo le pareció el sistema en general? 3

Malo	Buena	Muy buena
1	2	3

Pregunta N° 7:

- ¿Cómo respondió la aplicación para usted? 2

Pregunta N° 8:

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria? 3

Pregunta N° 9:

- ¿Qué cree usted de la reducción del agua? 3

Pregunta N° 10:

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable? 3

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI \_\_\_\_\_ NO \_\_\_\_\_

¿Cuál?

---



---

L2

Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1	2	3	4	5

Pregunta N° 1:

- ¿Cómo fue navegar por toda las interfaces del aplicación? 5

Pregunta N° 2:

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema? 4

Pregunta N° 3:

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual? 5

Pregunta N° 4:

- ¿Cómo fue para usted interactuar con cada uno de los sensores? 4

Pregunta N° 5:

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema? 4

Pregunta N° 6:

- ¿Cómo le pareció el sistema en general? 3

Malo	Buena	Muy buena
1	2	3

Pregunta N° 7:

- ¿Cómo respondió la aplicación para usted? 2

Pregunta N° 8:

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria? 3

Pregunta N° 9:

- ¿Qué cree usted de la reducción del agua? 3

Pregunta N° 10:

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable? 3

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI \_\_\_\_\_ NO \_\_\_\_\_

¿Cuál?

---

---

43

### Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1	2	3	4	5

Pregunta N° 1:

- ¿Cómo fue navegar por toda las interfaces de la aplicación ? 4

Pregunta N° 2:

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema? 4

Pregunta N° 3:

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual? 2

Pregunta N° 4:

- ¿Cómo fue para usted interactuar con cada uno de los sensores? 4

Pregunta N° 5:

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema? 3

Pregunta N° 6:

- ¿Cómo le pareció el sistema en general? 3

Malo	Buena	Muy buena
1	2	3

Pregunta N° 7:

- ¿Cómo respondió la aplicación para usted? 3

Pregunta N° 8:

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria? 3

Pregunta N° 9:

- ¿Qué cree usted de la reducción del agua? 3

Pregunta N° 10:

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable? 2

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI \_\_\_\_\_ NO \_\_\_\_\_

¿Cuál?

---

---



Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1	2	3	4	5

Pregunta N° 1:

- ¿Cómo fue navegar por toda las interfaces del aplicación ? 5

Pregunta N° 2:

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema? 4

Pregunta N° 3:

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual? 3

Pregunta N° 4:

- ¿Cómo fue para usted interactuar con cada uno de los sensores? 4

Pregunta N° 5:

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema? 2

Pregunta N° 6:

- ¿Cómo le pareció el sistema en general? 4

Mal	Buena	Muy buena
1	2	3

Pregunta N° 7:

- ¿Cómo respondió la aplicación para usted? 2

Pregunta N° 8:

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria? 3

Pregunta N° 9:

- ¿Qué cree usted de la reducción del agua? 2

Pregunta N° 10:

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable? 3

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI \_\_\_\_\_ NO \_\_\_\_\_

¿Cuál?

---



---

Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1	2	3	4	5

Pregunta N° 1:

- ¿Cómo fue navegar por toda las interfaces del aplicación ? 5

Pregunta N° 2:

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema? 5

Pregunta N° 3:

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual? 5

Pregunta N° 4:

- ¿Cómo fue para usted interactuar con cada uno de los sensores? 4

Pregunta N° 5:

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema? 2

Pregunta N° 6:

- ¿Cómo le pareció el sistema en general? 5

Malo	Buena	Muy buena
1	2	3

Pregunta N° 7:

- ¿Cómo respondió la aplicación para usted? 3

Pregunta N° 8:

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria? 3

Pregunta N° 9:

- ¿Qué cree usted de la reducción del agua? 2

Pregunta N° 10:

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable? 2

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI \_\_\_\_\_ NO \_\_\_\_\_

¿Cuál?

---



---

### Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1	2	3	4	5

Pregunta N° 1:

- ¿Cómo fue navegar por toda las interfaces del aplicación ? 4

Pregunta N° 2:

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema? 4

Pregunta N° 3:

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual? 5

Pregunta N° 4:

- ¿Cómo fue para usted interactuar con cada uno de los sensores? 4

Pregunta N° 5:

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema? 4

Pregunta N° 6:

- ¿Cómo le pareció el sistema en general? 5

Malo	Buena	Muy buena
1	2	3

Pregunta N° 7:

- ¿Cómo respondió la aplicación para usted? 3

Pregunta N° 8:

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria? 3

Pregunta N° 9:

- ¿Qué cree usted de la reducción del agua? 3

Pregunta N° 10:

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable? 3

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI \_\_\_\_\_ NO \_\_\_\_\_

¿Cuál?

---



---

## ANEXO Ñ: ENCUESTAS DE VIVIENDA VANESA ALCCA HUILLCAHUAYA

K<sub>1</sub>

Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1	2	3	4	5

Pregunta N° 1:

- ¿Cómo fue navegar por toda las interfaces del aplicación? 4

Pregunta N° 2:

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema? 4

Pregunta N° 3:

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual? 3

Pregunta N° 4:

- ¿Cómo fue para usted interactuar con cada uno de los sensores? 5

Pregunta N° 5:

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema? 3

Pregunta N° 6:

- ¿Cómo le pareció el sistema en general? 5

Malo	Buena	Muy buena
1	2	3

Pregunta N° 7:

- ¿Cómo respondió la aplicación para usted? 2

Pregunta N° 8:

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria? 3

Pregunta N° 9:

- ¿Qué cree usted de la reducción del agua? 3

Pregunta N° 10:

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable? 3

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI \_\_\_\_\_ NO \_\_\_\_\_

¿Cuál?

---



---

V2

### Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1	2	3	4	5

Pregunta N° 1:

- ¿Cómo fue navegar por toda las interfaces del aplicación? 4

Pregunta N° 2:

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema? 4

Pregunta N° 3:

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual? 4

Pregunta N° 4:

- ¿Cómo fue para usted interactuar con cada uno de los sensores? 5

Pregunta N° 5:

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema? 4

Pregunta N° 6:

- ¿Cómo le pareció el sistema en general? 5

Mal	Buena	Muy buena
1	2	3

Pregunta N° 7:

- ¿Cómo respondió la aplicación para usted? 2

Pregunta N° 8:

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria? 3

Pregunta N° 9:

- ¿Qué cree usted de la reducción del agua? 3

Pregunta N° 10:

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable? 3

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI \_\_\_\_\_ NO \_\_\_\_\_

¿Cuál?

---

---

Diseño de encuesta del sistema ecoeficiente con plataforma Arduino y Android.

La siguiente encuesta se tendrá que responder según a las calificaciones numéricas que se muestra.

Muy difícil	Difícil	Más o menos	Fácil	Muy fácil
1	2	3	4	5

Pregunta N° 1:

- ¿Cómo fue navegar por toda las interfaces del aplicación? 5

Pregunta N° 2:

- ¿Cómo fue para usted manejar automáticamente y manualmente el sistema? 4

Pregunta N° 3:

- ¿Cómo fue para usted acceder a la información del consumo en litros del agua diariamente y mensual? 4

Pregunta N° 4:

- ¿Cómo fue para usted interactuar con cada uno de los sensores? 5

Pregunta N° 5:

- ¿Cómo le resulto conectarse a los dispositivos Bluetooth del sistema? 5

Pregunta N° 6:

- ¿Cómo le pareció el sistema en general? 5

Mal	Buena	Muy buena
1	2	3

Pregunta N° 7:

- ¿Cómo respondió la aplicación para usted? 2

Pregunta N° 8:

- ¿Cómo califica usted el uso de nuevas tecnologías en nuestra vida diaria? 3

Pregunta N° 9:

- ¿Qué cree usted de la reducción del agua? 3

Pregunta N° 10:

- En general ¿Cómo considera el sistema ecoeficiente en el ahorro del agua potable? 3

¿Tiene alguna sugerencia para mejorar en el sistema ecoeficiente?

SI \_\_\_\_\_ NO \_\_\_\_\_

¿Cuál?

---



---