



**UNIVERSIDAD CATÓLICA ANDRÉS BELLO**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES**

**DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE  
CONSUMO, RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN  
UNA ESTRUCTURA DE ALMACENAMIENTO**

**TRABAJO ESPECIAL DE GRADO**

Presentado ante la

**UNIVERSIDAD CATÓLICA ANDRÉS BELLO**

Como parte de los requisitos para optar al título de

**INGENIERO EN TELECOMUNICACIONES**

REALIZADO POR

Br. Johan Agustín Palacios ~~Carrero~~

Br. Ybrahim José García Ríos

PROFESOR GUIA

Ing. José Gregorio Castillo Pacheco

FECHA Septiembre 2015

**UNIVERSIDAD CATÓLICA ANDRÉS BELLO**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES**

**DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE  
CONSUMO, RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN  
UNA ESTRUCTURA DE ALMACENAMIENTO**

**REALIZADO POR**

**Br. Johan Agustín Palacios Carrero**

**Br. Ybrahim José García Ríos**

**PROFESOR GUIA**

**Ing. José Gregorio Castillo Pacheco**

**FECHA Septiembre 2015**



**DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE  
CONSUMO, RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN  
UNA ESTRUCTURA DE ALMACENAMIENTO**

Este Jurado; una vez realizado el examen del presente trabajo ha evaluado su contenido con el resultado:

DIEZ Y NUEVE (19)

JURADO EXAMINADOR

Firma:

Firma:

Firma:

Nombre: Nicole Buonanno

Nombre: José Riviere

Nombre: José Castillo

REALIZADO POR

Br. Johan Agustín Palacios Carrero

Br. Ybrahim José García Ríos

PROFESOR GUIA

Ing. José Gregorio Castillo Pacheco

FECHA Septiembre 2015

## **RESUMEN**

### **DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO, RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE ALMACENAMIENTO**

Palacios, Johan Agustín  
Johanpalacios.ing@gmail.com

García Ríos, Ybrahim José  
Ybra91@gmail.com

El objetivo del Trabajo Especial de Grado es lograr una manera de alertar a los usuarios que hagan uso del sistema de la entrada o no de agua externa al tanque, en conjunto, con sensores de nivel ubicados dentro del tanque de almacenamiento para poder detallar cuáles son los niveles de aguas disponibles aproximados en ciertos rangos de la estructura en un estimado de tiempo real, con el que el usuario puede contar para realizar las distintas actividades. A su vez en el caso de encontrarse en un estado crítico de agua poder notificar al usuario de las circunstancias ocurrentes y que el mismo tome las medidas adecuadas, ocurrirá lo mismo en caso contrario, que el tanque de almacenamiento se encuentre en un estado lleno de agua, y así el mismo usuario pueda tomar las medidas pertinentes. Dicho objetivo se pretende llevar a cabo creando un sistema de automatización y gestión que funcione de manera remota donde la información sea recopilada por sensores de nivel de agua y de flujo que envíen dicha información a un dispositivo encargado de analizar la misma y pueda ser monitoreada mediante una aplicación Android.

DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO,  
RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE  
ALMACENAMIENTO

---

Este sistema funciona gracias al trabajo en conjunto de un sensor de nivel y sensores de entrada y salida para indicar los niveles de agua disponible y si existe agua entrando o saliendo en el tanque de almacenamiento. La información es recopilada a través de un dispositivo Raspberry Pi y este mismo será el encargado de analizarlo y enviarlo a un servidor local para guardar esta información en una base de datos. La aplicación Android leerá las prestaciones de la base de datos y mostrará la respectiva información a los usuarios del sistema de Automatización.

**Palabras claves:** Tanque de almacenamiento, Raspberry PI, Android, Sensores de agua, Servidores.

## **AGRADECIMIENTOS**

Ante todo queremos dar gracias a nuestros padres, por todos los sacrificios realizados, y su apoyo omnipresente en cada una de nuestras etapas dentro de la universidad. A ellos, gracias infinitas.

A nuestro tutor José Gregorio Castillo por todo su apoyo en los momentos de dificultad a lo largo del trabajo especial de grado.

A nuestros compañeros Carlos Sardi, Eudys Level, Maurizzio Risso, Carlos Lozano, Andrés Torres, Daniel Dacorte, Omar Pedraza y Elías Magallanes por acompañarnos mutuamente en todo momento y prestarnos el apoyo y la ayuda siempre que lo necesitamos.

A Carolina Rueda por todos aquellos momentos y experiencias vividas juntos durante tanto tiempo, aprendiendo y madurando juntos. Te amo.

A Stefanie Parejo por apoyarme en cada uno de los momento vividos y estar conmigo sin importar las circunstancias o dificultades, gracias hoy y siempre. Te amo.

DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO,  
RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE  
ALMACENAMIENTO

---

## ÍNDICE GENERAL

RESUMEN.....	I
AGRADECIMIENTOS .....	III
ÍNDICE GENERAL.....	V
ÍNDICE DE FIGURA.....	IX
ÍNDICE DE TABLA.....	XI
INTRODUCCIÓN .....	XIII
CAPITULO I.....	1
Planteamiento del proyecto .....	1
I.1 Planteamiento del Problema .....	1
I.2 Objetivos de la Investigación.....	4
I.2.1 Objetivo General.....	4
I.2.2 Objetivos Específicos .....	4
I.3 Alcances y Limitaciones.....	5
I.3.1 Alcances.....	5
I.3.2 Limitaciones .....	7
I.4 Justificación .....	7
CAPITULO II .....	11
Marco Teórico .....	11
II.1 Estructura de reserva de agua .....	11
II.2 Sistemas de automatización.....	12
II.3 Medios para la Comunicación.....	13
II.3.1 Protocolos de Comunicación.....	13
II.3.2 Protocolos Inalámbricos .....	14
II.3.3 Comparación entre Wifi y Bluetooth .....	14



DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO,  
RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE  
ALMACENAMIENTO

---

II.4 Raspberry PI .....	17
II.4.1 Arduino.....	21
II.4.2 Comparación entre Raspberry PI y Arduino .....	21
II.5 Base de Datos .....	22
II.5.1 Sistema Gestor de Base de Datos .....	23
II.6 Sistema operativo .....	23
II.6.1 Linux.....	24
II.6.2 Raspbian .....	24
II.7 Android.....	25
II.7.1 Arquitectura de Android.....	26
II.7.2 Bloque básico de una aplicación.....	32
II.7.3 Comparativa de Android con otras plataformas .....	33
II.7.4 Ambiente de desarrollo.....	34
II.7.5 Android Studio .....	34
II.7.6 Principales características que incluye Android Studio.....	35
II.7.7 Comparativa de Android Studio con Eclipse.....	37
II.7.8 Ventajas uso Android Studio .....	39
II.8 XamppServer .....	40
II.9 SQL.....	41
II.10 MySQL .....	41
II.11 Python.....	42
II.12 Samba .....	42
II.13 Smartphone.....	43
II.14 Sensor de Agua.....	44
II.15 Sensores de nivel .....	45
II.15.1 Reed Switch.....	45
CAPITULO III .....	47
Marco Metodológico .....	47
III.1 Investigación y documentación.....	47
III.2 Diseño del sistema.....	49

DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO,  
RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE  
ALMACENAMIENTO

---

III.3 Requerimientos del Diseño de Sistema.....	51
III.4 Desarrollo del sub-sistema detector de los niveles de agua .....	52
III.5 Desarrollo de la base de datos de los estados.....	52
III.6 Desarrollo de las aplicaciones necesarias en el sistema .....	53
III.7 Desarrollo de la aplicación Android .....	53
III.8 Prueba del sistema en entorno real.....	54
III.9 Recomendaciones y Conclusiones .....	54
III.10 Elaboración del Tomo.....	54
CAPITULO IV .....	55
Desarrollo.....	55
IV.1 Diseño del sub-sistema detector de los niveles de agua.....	55
IV.1.2 Análisis y Selección del sensor de nivel .....	57
IV.1.3 Desarrollo y pruebas del sub-sistema detector de los niveles de agua.....	63
IV.1.4 Comunicación utilizada con el Raspberry PI.....	65
IV.2 Diseño del programa de procesamiento y transmisión de los datos recopilados. ..	66
IV.2.1 Análisis del Raspberry PI como dispositivo procesador.....	68
IV.2.2 Análisis y selección del Lenguaje de programación.....	69
IV.2.3 Desarrollo y pruebas del programa de procesamiento y transmisión de los datos recopilados .....	71
IV.2.4 Comunicación del Raspberry PI a Servidor Local.....	73
IV.3 Base de datos de la estructura de almacenamiento .....	74
IV.3.1 XamppServer como servidor local para base de datos .....	77
IV.4 Aplicación LlenarBaseDatos .....	77
IV.4.1 Comunicación de la Aplicación LlenarBaseDatos con MySQL.....	82
IV.5 Aplicación Android Mi Tanque .....	82
IV.5.1 Clases creadas en la aplicación Mi tanque App.....	85
IV.5.1.1 loginSync.java.....	85
IV.5.1.2 login.java .....	85
IV.5.1.3 PantallaUsuario.java .....	86
IV.5.1.4 ConsultaUsuarioSync.java .....	86

DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO,  
RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE  
ALMACENAMIENTO

---

IV.5.1.5 ConsultaUsuario.java .....	86
IV.5.1.6 ModificarUsuarioSync.java .....	87
IV.5.1.7 ModificarUsuario.java .....	87
IV.5.1.8 Refresh.java.....	88
IV.5.2 Layouts utilizados en el desarrollo de la aplicación.....	88
IV.5.2.1 Activity_login.xml .....	88
IV.5.2.2 Activity_Usuario_Pantalla.xml .....	89
IV.5.2.3 Activity_Usuario_Consulta.xml.....	89
IV.5.2.4 Activity_Usuario_Modificar.xml.....	90
IV.5.3 Servicios PHP .....	90
IV.5.3.1 Config.php.....	91
IV.5.3.2 login.php .....	91
IV.5.3.3 consulta.php .....	92
IV.5.3.4 consultaIO.php .....	92
IV.5.3.5 ModificarUsuario.php .....	92
IV.5.4 Permisos de la aplicación.....	92
IV.6.1 Pruebas en entornos simulados .....	93
IV.6.2 Pruebas en entorno real.....	95
CAPITULO V .....	97
Resultados.....	97
V.1 Sub-sistema detector de los niveles de agua.....	97
V.2 Programa de procesamiento y transmisión de los datos recopilados en el Raspberry Pi .....	99
V.3 Estructura de la base de datos.....	103
V.3.1 Tabla Sensores_Nivel.....	103
V.3.2 Tabla Entrada_Salida .....	105
V.3.3 Tabla Usuario_Tanque .....	107
V.4 Aplicación LlenarBaseDatos .....	108
V.5 Aplicación Android “Mi tanque APP” .....	110
V.5.1 Ventanas desarrolladas en la aplicación.....	110

DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO,  
RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE  
ALMACENAMIENTO

---

V.5.1.1 Iniciar Sesión.....	110
V.5.1.2 Menú de inicio.....	111
V.5.1.3 Consulta actualizada del tanque .....	112
V.5.1.4 Modificación de datos .....	114
V.5.2 Servicios PHP .....	115
V.6 Transmisión de datos del Sistema .....	115
V.7 Pruebas realizadas en entornos simulados y real .....	116
CAPITULO VI.....	119
Conclusiones y Recomendaciones .....	119
VI.1 Conclusiones.....	119
VI.2 Recomendaciones .....	121
BIBLIOGRAFIA .....	122

## ÍNDICE DE FIGURA

Figura 1 Estructura de almacenamiento.....	12
Figura 2: Raspberry PI .....	17
Figura 3: Circuito integrado de la Raspberry PI .....	18
Figura 4: GPIO de la Raspberry PI .....	20
Figura 5: Pines del Raspberry PI.....	20
Figura 6: Kernel de Linux .....	27
Figura 7: Librería y Ejecución de Android .....	29
Figura 8: Estructura de la Aplicación Android .....	31
Figura 9: Arquitectura de Android.....	32
Figura 10: bloque básico de una aplicación Android.....	32
Figura 11: Porcentajes de dispositivos adquiridos hasta el 3° trimestre del 2014 .....	34
Figura 12: Funcionamiento Samba .....	43
Figura 13: Reed Switch.....	46

DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO,  
RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE  
ALMACENAMIENTO

---

Figura 14: Funcionamiento del Sistema .....	50
Figura 15: Planteamiento del diseño. ....	54
Figura 16: Circuito alarma de nivel de agua.....	58
Figura 17: Tipos de transductores de presión .....	59
Figura 18: Sensor de flujo de agua .....	60
Figura 19: Sensor flotante tipo eléctrico.....	61
Figura 20: Tanque de almacenamiento.....	65
Figura 21: Tipo de cable utilizado para transmitir los datos de los sensores. ....	66
Figura 22: Diagrama de Flujo del programa de recopilación de datos.....	71
Figura 23: Circuito para las pruebas realizadas.....	72
Figura 24: Diagrama EER de la herramienta Workbench para la creación de la base de datos. ....	77
Figura 25: Diagrama de flujo de la aplicación LlenarBaseDatos. ....	81
Figura 26: Diseño de la aplicación Android. ....	83
Figura 27: Porcentaje de personas usando esta versión de Android.....	84
Figura 28: Diagrama de flujo del funcionamiento de la aplicación Android. ....	94
Figura 29: Sensor de nivel tipo flotante eléctrico.....	98
Figura 30: Diagrama Circuital de los Sensores con el Raspberry PI.....	99
Figura 31: Archivo de texto de sensores de nivel.....	100
Figura 32: Archivo de texto referente a los sensores de entrada y salida.....	101
Figura 33: Captura de pantalla de las distintas configuraciones del GPIO. ....	102
Figura 34: Captura de pantalla de la vista de la aplicación desde el punto de vista de un usuario .....	102
Figura 35: Tabla Sensores de nivel con distintos valores que arrojaron las pruebas respectivas .....	105
Figura 36: Captura de pantalla de la tabla de entrada y salida de agua y algunos resultados obtenidos de las pruebas.....	106
Figura 37: Tabla usuario de la base de datos del sistema.....	108

DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO,  
RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE  
ALMACENAMIENTO

---

Figura 38: Interfaz vista desde la consola de esta aplicación corriendo en el respectivo sistema.....	109
Figura 39: Login de la Aplicación Android. ....	111
Figura 40: Menú principal de la aplicación.....	112
Figura 41: Caso en el que el uno de los sensores se encuentre dañado .....	113
Figura 42: Ventana de la aplicación referente a modificaciones de un usuario.....	114
Figura 43: Diagrama de Telecomunicaciones.....	116
Figura 44: Prueba en un entorno real de todos los parámetros del sistema .....	117
Figura 45: Resultados de pruebas cuando están activos los sensores 1 y 2 .....	118

## ÍNDICE DE TABLA

Tabla 1: Comparación Bluetooth vs Wifi. ....	16
Tabla 2 Comparación de modelos de Raspberry PI.....	19
Tabla 3 Comparación entre Raspberry PI y Arduino.....	22
Tabla 4 Comparativa de Android con otras plataformas .....	33
Tabla 5: Comparación de Android Studio y Eclipse.....	38
Tabla 6: Cuadro comparativo de los posibles sensores a utilizar.....	62

DISEÑO DE UN SISTEMA DE AUTOMATIZACION EN EL CONTROL DE CONSUMO,  
RACIONAMIENTO Y ANALISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE  
ALMACENAMIENTO

---

## INTRODUCCIÓN

La importancia del agua en el mundo es fundamental para todo ser vivo. Los humanos consumen agua potable, que es agua dulce pasada por un proceso de purificación, para que no represente un riesgo para la salud.

El agua dulce no solo es primordial para el consumo humano sino que también abarca varias funciones esenciales para la calidad de vida, como por ejemplo las industrias encargadas del regadío, limpieza, alimentación, hospitales entre otras. Debido a que estas industrias o áreas locales que consumen el agua no pueden subsistir sin este recurso se llevó al almacenamiento de la misma como medida de prevención para evitar la escasez.

La mayoría de las zonas residenciales, sitios públicos o industrias que dependen del recurso del agua para poder llevar a cabo sus funciones por lo general cuentan con una reserva en una estructura o un tanque especial para prevenir la escasez, sin embargo, en momentos de racionamiento o contingencia, frecuentemente por la falta de planificación y de gestión, las personas no son alertadas con la anticipación necesaria de los niveles de agua disponibles dentro del tanque, para así poder tomar las medidas que el momento amerita. Durante estas situaciones, muchas veces, consumos de agua que podríamos catalogar como no fundamentales, tales como fuentes, piscinas, regadíos de áreas verdes entre otros permanecen encendidos, por descuidos del personal encargado, durante rangos de tiempo considerados fatales para su uso, consumiendo así los niveles de agua que restan y de esta manera reduciendo la cantidad de recursos disponibles para las necesidades esenciales e incluso en casos extremos ocasionar daños a la bombas hidráulicas, lo que puede convertirse en pérdidas monetarias y atrasos, específicamente en las zona residenciales se puede durar horas, días y meses sin este recurso tan primordial como es para la vida.



El propósito del proyecto entonces es poder mejorar la manera de alertar a las personas de la entrada o no de agua externa al tanque de almacenamiento y conociendo las cantidades de agua que entran y las reservadas, poder determinar cuáles son los niveles de aguas disponibles en un estimado de tiempo real. A su vez en el caso de encontrarse en un estado crítico de agua poder notificar al usuario y que el mismo tome las medidas preventivas necesarias, y en el caso contrario avisar si el tanque se encuentra en un estado lleno de agua y así el usuario poder tomar las decisiones pertinentes referente a su tanque de almacenamiento. Dicho objetivo se pretende llevar a cabo creando un sistema de automatización que funcione de manera remota donde la información sea recopilada por sensores de nivel de agua y de flujo que envíen dicha información a un dispositivo puente y luego pueda ser monitoreada mediante una aplicación Android.

En el siguiente Trabajo Especial de Grado se presentara la forma en que se realizó dicho sistema y como fue dividido el mismo para así obtener las prestaciones anteriormente mencionadas. A continuación se presenta una breve descripción de los capítulos.

Capítulo I: donde se describe el problema a tratar y se observa la necesidad de contar con un sistema que mejore el alerta a los usuarios de los niveles de agua disponibles, también se describen los objetivos del proyecto, los alcances, las limitaciones y la justificación del mismo.

Capítulo II: contiene las bases teóricas que son necesarias para el desarrollo y total comprensión del tema a desarrollar.

Capítulo III: este capítulo corresponde a la metodología utilizada para la elaboración del sistema, se explican las actividades realizadas para llevar a cabo el mismo.

Capítulo IV: hace referencia al desarrollo del sistema, donde se describen todos los pasos que se realizaron durante el proceso de elaboración del proyecto y se explican las herramientas utilizadas tanto en entorno físico como a nivel de programación.

Capítulo V: es este capítulo donde se presentan todos los resultados del trabajo especial de grado, donde se describen todas las etapas de las diferentes aplicaciones y se verifica su total funcionamiento.

Capítulo VI: corresponde al último capítulo del proyecto y en él se exponen las conclusiones y recomendaciones para futuras investigaciones



## **CAPITULO I**

### **Planteamiento del proyecto**

En este capítulo se le muestra al lector la razón por la cual se decidió la realización de este Trabajo Especial de Grado, se hace la descripción del problema y necesidades que hicieron que se planteara la idea, así como el objetivo al cual se pretende llegar con la finalización de este proyecto mostrando también sus alcances y limitaciones.

#### **I.1 Planteamiento del Problema**

La importancia del agua en el mundo es fundamental para todo ser vivo. Los humanos consumen agua potable, que es agua dulce pasada por un proceso de purificación, para que no represente un riesgo para la salud.

A lo largo de los años el número de habitantes en Venezuela y el mundo ha crecido rápidamente y la utilización de agua crece aún en más grande medida, por lo que se ha vuelto mucho más difícil el almacenamiento de agua, sobre todo en sectores o países con menos recursos.

El agua dulce no solo es primordial para el consumo humano sino que también abarca varias funciones esenciales para la calidad de vida, como por ejemplo las industrias encargadas del regadío, limpieza, alimentación, hospitales entre otras. Debido a que estas industrias o áreas locales que consumen el agua no pueden subsistir sin este recurso se llevó al almacenamiento de la misma como medida de prevención para evitar la escasez.

Se entiende por almacenamiento de agua el contar con una estructura, ya sea de pequeñas dimensiones para áreas locales o de grandes dimensiones para industrias y zonas residenciales con una población grande, con el propósito fundamental de proveer una cantidad adecuada de agua. El correcto almacenamiento de agua requiere unas condiciones y normas, tanto en la forma de almacenamiento como al tipo de depósito que se utilice, esto es de mucha importancia, ya que de esto dependerá la calidad en que el agua llegue a los usuarios o a la maquinaria respectiva que usa este elemento para funcionar, además de esto, el correcto almacenamiento influirá en la vida útil de las maquinaria.

El almacenamiento y /o reutilización del agua dulce se ha convertido en un requisito imprescindible, el cual abarca desde el ámbito industrial hasta el ámbito doméstico, y una de las causas principales del almacenamiento del agua es cuando existe el racionamiento.

Generalmente, cuando existe un racionamiento del agua, la empresa encargada del suministro tiene la labor de avisar con antelación en la respectiva zona de los cambios en el envío de agua, pero muchas veces, el agua deja de llegar sin previo aviso, lo que se convierte en un evento inesperado, que puede ocasionar pérdidas millonarias en una empresa o afectar la vida diaria de los usuarios. Este tipo de hechos realza la importancia de contar con cantidades de agua de reserva almacenadas que permitan contar con este recurso de forma continua.

A pesar de que Venezuela está dentro de los 15 primeros países de mayor producción de agua dulce según el Sistema de Información Global en Agua y Agricultura de la FAO (Parra, 2012), el país enfrenta intensos períodos de racionamiento que compromete el abastecimiento del vital líquido, lo cual se podría atribuir, más allá de los fenómenos naturales de sequía, a la falta de planificación y

gestión en forma apropiada de la red de distribución de agua a nivel nacional, incluyendo también lo cambiante de las inversiones en el sector que dependen directamente de las fluctuaciones en los precios del petróleo. A los problemas anteriormente mencionados se suma la mala utilización del agua, las fugas en los sistemas de distribución y la contaminación de ríos y embalses, así como el bajo porcentaje de tratamiento que reciben las aguas servidas en el país lo cual debería de ser prioritario en la gestión pública.

La mayoría de las zonas residenciales, sitios públicos o industrias que dependen del agua para poder llevar a cabo sus funciones por lo general cuentan con una reserva de esta en una estructura o un tanque especial para prevenir la escasez, sin embargo, en momentos de racionamiento o contingencia, frecuentemente por la falta de planificación y de gestión, las personas no son alertadas con la anticipación necesaria de los niveles de agua disponibles dentro del tanque, para así poder tomar las medidas que el momento amerita. Durante estas situaciones, muchas veces, consumos de agua que podríamos catalogar como no fundamentales, tales como fuentes, piscinas, regadíos de áreas verdes entre otros permanecen encendidos, por descuidos del personal encargado, durante rangos de tiempo considerados fatales para su uso, consumiendo así los niveles de agua que restan y de esta manera reduciendo la cantidad de recursos disponibles para las necesidades esenciales e incluso en casos extremos ocasionar daños a las bombas hidráulicas, lo que puede convertirse en pérdidas monetarias y atrasos, específicamente en las zonas residenciales se puede durar horas, días y meses sin este recurso tan primordial como es para la vida.

El objetivo del proyecto entonces es poder mejorar la manera de alertar a los usuarios de la entrada o no de agua externa al tanque y conociendo las cantidades de agua que entran y un aproximado de las cantidades reservadas, poder detallar cuáles son los niveles de aguas disponibles en tiempo real, con el que se puede contar, para

realizar las distintas actividades. A su vez en el caso de encontrarse en un estado crítico de agua, que el usuario pueda recibir una notificación, para así saber en qué momento el mismo pueda deshabilitar la respectiva bomba hidráulica, encargada de enviar el agua a los distintos servicios y evitar pérdidas o daños a la misma. Dicho objetivo se pretende llevar a cabo creando un sistema de automatización que funcione de manera remota donde la información sea recopilada por sensores de nivel de agua verticales y sensores de entrada y salida que envíen dicha información a un Raspberry Pi y pueda ser monitoreada mediante una aplicación Android.

## **I.2 Objetivos de la Investigación**

### **I.2.1 Objetivo General**

Diseñar y construir el prototipo de un sistema para la automatización en el control de consumo y racionamiento de los niveles de agua disponibles en una estructura de almacenamiento de uso local para un ambiente no comercial.

### **I.2.2 Objetivos Específicos**

1. Investigación de los elementos, a nivel de red y transmisión, que van a conformar el sistema para la automatización en el control de consumo, racionamiento y análisis del nivel del agua disponible.
2. Diseñar desde el punto de vista conceptual el sistema de automatización.

3. Realizar la escogencia y adaptación del subsistema de sensores de entrada y salida de agua y de nivel.
4. Realizar el desarrollo de la adaptación del software para la transmisión de datos del subsistema de sensores a la aplicación Android.
5. Implementar una base de datos de la información adquirida en el sistema de automatización.
6. Diseñar la aplicación Android del sistema de entrega de datos al usuario.
7. Construir un prototipo funcional mediante el cual se muestren las prestaciones del sistema.

### **I.3 Alcances y Limitaciones**

#### **I.3.1 Alcances**

El presente proyecto utiliza como referencia los elementos que se desprenden del caso de análisis descrito en el planteamiento del problema, referente a la necesidad de mejora en el control de consumo, racionamiento y análisis de los niveles de agua disponibles en una estructura de almacenamiento de uso local.

Dado que se asume únicamente como una referencia de diseño, se entenderá que:



1. No todos los requerimientos de diseño obedecerán a este caso en particular, por lo que podrán existir nuevas premisas que se deriven de otros criterios de decisión.
2. El Trabajo de Grado, ni sus lineamientos, guardan ningún tipo de relación con cualquier ente público o privado sino que es pensado con las iniciativas de mejora que se pretenden desarrollar, existiendo la posibilidad de vincular estos desarrollos en un futuro al área comercial.
3. El Trabajo de Grado y sus premisas se realizan con suficiente amplitud para cumplir con los objetivos propuestos, por lo que la justificación de determinados elementos del proyecto pudieran encontrarse fuera del marco de referencia, apuntando a una solución más general que abarque a otras áreas, en su versión más elaborada.
4. Se tiene por objeto diseñar un sistema de automatización que permita el control y racionamiento de los niveles de aguas disponibles dentro de un tanque de almacenamiento de uso local presentado al usuario a través de una aplicación Android.
5. El proyecto incluirá la identificación y caracterización de los parámetros de lectura de parte de los sensores a usar, la creación de un software para la transmisión de datos de los sensores y una aplicación Android que permita la presentación de la información recopilada.
6. Se plantea la posibilidad de demostrar el funcionamiento del sistema de control mediante simulaciones de un prototipo no comercial elaborado por los autores del Trabajo Especial de Grado.

7. No se contemplará un acabado ni uso profesional, ya que la implementación será con propósitos experimentales y académicos pero si se deja abierta la posibilidad de poder modificar el sistema con el propósito de mejorarlo hasta tal punto que pueda ser utilizado en zonas residenciales o lugares con un almacenamiento de agua considerable que se necesite monitorear.

### **I.3.2 Limitaciones**

La adquisición de los materiales estará sujeta a la disponibilidad inmediata de los mismos, a la existencia en el mercado de soluciones accesibles desde el punto de vista económico y de importación, y al requerimiento final que se derive del diseño, en términos de capacidad de procesamiento. El caso más adverso lo constituiría aquel en el que ciertos sensores de los varios a utilizar no puedan ser adquiridos, en cuyo caso, el alcance del prototipo se degradaría a la demostración del funcionamiento sin el uso de los mismos en el sistema.

### **I.4 Justificación**

Como se mencionó anteriormente en el planteamiento del problema, el Trabajo Especial de Grado parte del almacenamiento de agua requerido por distintas situaciones que pueden existir en diferentes zonas del país, ya sea una zona urbana o una zona industrial, con el fin entonces de mejorar la calidad de vida y evitar pérdidas económicas a los usuarios. Entonces el presente trabajo tiene por justificación buscar una mejoría en la forma en que los usuarios tienen de saber la cantidad de agua que existe reservada en sus respectivas estructuras de almacenamiento, además de incluir en el mismo sistema los conocimientos de entrada o no de agua al tanque de almacenamiento.

Para conseguir las prestaciones del sistema se utilizó como dispositivo el llamado Raspberry Pi encargado de gestionar los respectivos sensores que tendrá la estructura y además de este se utilizó una aplicación Android para que exista una comodidad plena para el usuario al momento de saber las prestaciones del sistema planteado.

El Raspberry Pi representa una ventaja bastante considerable para el proyecto ya que a pesar de ser un ordenador de placa reducido, fundamentalmente creado para estimular la enseñanza de la computación en las escuelas permite poder llevar a cabo procesos más complejos que incluyan web con manejo de bases de datos, que requieran aplicaciones con reloj en tiempo real entre otras tantas, sabiendo esto, el proyecto incluirá el desarrollo del software que, en conjunto con la información obtenida de dichos sensores, se pueda obtener una estimación de los niveles de agua disponibles, como también mensajes de alerta que se activen, según la actividad que se presente, como por ejemplo, el momento en que los niveles de agua sean bajos.

El uso de una aplicación Android para poder mostrar los resultados obtenidos es debido más que todo a que se ha convertido en el sistema operativo más utilizado en Smartphones y Tabletas permitiéndonos poder adaptar el proyecto a lo actualmente más comercial en el mercado gracias a la versatilidad y condición de código abierto, que tiene la plataforma de Google, lo hace el producto perfecto para ser adaptado. Además el Sistema Operativo proporciona todas las interfaces necesarias para desarrollar la aplicación y que se acceda en todas las funciones del teléfono.

En tal sentido, el proyecto diseñado propone la integración tecnológica de diversos elementos en materia de redes, transmisión de datos y sistemas telemáticos en general, conjuntamente con desarrollos particulares e integración tecnológica en

las áreas de automatización; a fin de demostrar su funcionamiento, se pretende construir un prototipo, el cual será para un ambiente no comercial, con el fin de poder mostrar las prestaciones del sistemas y detallar las ventajas que permiten facilitar el control de datos, referentes a los niveles de agua registrados, buscando la mayor eficiencia y simplicidad en la integración e instalación, y con miras a contribuir en la estandarización de procesos en el área. En tal sentido, se buscará desarrollar el trabajo con suficiente amplitud, tal que los resultados de los diversos desarrollos sean replicables en otros campos de la automatización, partiendo de un único diseño modular, escalable y configurable.



## **CAPITULO II**

### **Marco Teórico**

En el presente capítulo, Marco Teórico del Trabajo Especial de Grado, se describen los elementos necesarios a utilizar en la realización del proyecto. Esta descripción es hecha a los fines que un lector no conocedor de la materia pueda entender por qué se usan estos elementos y que importancia tienen dentro del proyecto. Entre tales elementos se nombran: Estructura de Reserva de Agua, Sistema de Automatización, Lenguaje de programación Python, Sensores de Nivel, Dispositivos móviles, Raspberry PI, Bases de datos, Servidores y Sistema operativo Android entre otros elementos a nombrar.

#### **II.1 Estructura de reserva de agua**

Se puede definir como la estructura apta para contener un cierto volumen de agua, con las instalaciones complementarias precisas para cumplir funciones de regulación de caudales, de carga o ambas y de seguridad del servicio.

En consecuencia, el depósito debe garantizar la inalterabilidad de la calidad de las aguas, evitando variaciones de temperatura, desarrollo de algas, contaminación exterior, etc. En este sentido, presenta grandes ventajas la adopción de depósitos cerrados. (Cruz, 20011)



**Figura 1 Estructura de almacenamiento.**

**Fuente: Almazan, 2011**

## **II.2 Sistemas de automatización**

Los sistemas de automatización son diseñados con el fin de usar la capacidad de las máquinas para llevar a cabo determinadas tareas anteriormente efectuadas por seres humanos, y para controlar la secuencia de las operaciones sin intervención humana. El término automatización también se ha utilizado para describir sistemas en los que dispositivos programados o automáticos pueden funcionar de forma independiente o semi-independiente del control humano. (Almazan, 2008)

Los sistemas de automatización en la actualidad resultan de mucha importancia, ya que no siempre es posible para el hombre, estar al tanto de todo lo que ocurre a su alrededor y es una forma de mejorar la calidad de vida.

## **II.3 Medios para la Comunicación**

Para ser transmitido un mensaje, se requiere de un sistema de comunicación que permita que la información sea transferida, a través del espacio y el tiempo, desde un punto llamado fuente hasta otro punto de destino. (Gonzalez, 2001)

Pese a las diferentes vías en las que es posible la transmisión de la información, subyace la siguiente teoría: -ya sea que necesiten un soporte material, como un cable, para que la corriente eléctrica que conduce dicha información llegue al destino en el que será decodificada-, -o que no sea requerido un elemento material conductor-, es la electricidad la encargada de la tarea de trasladar un mensaje, en forma de señal. Las ondas electromagnéticas permiten la transmisión de mensajes o señales. (Gonzalez, 2001)

### **II.3.1 Protocolos de Comunicación**

Llamamos protocolo de comunicaciones a una serie de normas que usan los equipos informáticos para gestionar sus diálogos en los intercambios de información. Dos equipos diferentes de marcas diferentes se pueden comunicar sin problemas en el caso en que usen el mismo protocolo de comunicaciones. (Gonzalez, 2001)

A lo largo del tiempo ha ido mejorando la tecnología de las comunicaciones, y se han podido ir usando protocolos más útiles para las nuevas máquinas. Por ello han ido apareciendo nuevos protocolos a los que se han ido adaptando los productos de cada fabricante para asegurarse la compatibilidad con el resto de las marcas. (Gonzalez, 2001)



### **II.3.2 Protocolos Inalámbricos**

Entre los distintos protocolos existentes en el mercado que pueden utilizarse como medios de transmisión inalámbrica para los sistemas de comunicaciones, se encuentran como más destacables:

- **Wifi:** Es un protocolo de conexión inalámbrica a internet registrado bajo el estándar IEEE 802.11, su principal uso consiste en la interconexión de distintos artefactos electrónicos entre sí; sin la necesidad de la presencia de un cableado entre ellos. Gracias a esta tecnología es posible que las computadoras, televisores, celulares dotados con la tecnología; puedan acceder a internet a través de un punto de acceso inalámbrico. (Gonzalez, 2001)
- **Bluetooth:** La tecnología inalámbrica denominada Bluetooth, proporciona una vía de interconexión inalámbrica mediante una conexión segura de radio de corto alcance, soporta la transferencia de voz y datos entre los diversos aparatos que incorporen esta tecnología. Está basada en el estándar IEEE 802.15, y es el principal recurso para el intercambio de información a corta distancia entre dispositivos móviles, computadoras, cámaras fotográficas entre otros. (Gonzalez, 2001)

### **II.3.3 Comparación entre Wifi y Bluetooth**

Bluetooth, es adecuado para distancias menores a 10 metros y principalmente para conectar dispositivos a alta velocidad, mientras que Wifi es para mayores distancias, para usar como medio para acceder a Internet principalmente y para la creación de redes locales, LAN, inalámbrica (Cortez, Noticias Dot, 2014).

El uso más generalizado de Bluetooth es para el intercambio de información entre dispositivos electrónicos que se encuentren a pequeña distancia. Así se puede conectar dispositivos tales como PDA, teléfono móvil o cámara digital a un ordenador o impresora sin necesidad de cables para intercambiar información, descargarla e incluso imprimir las fotos recibidas o realizadas (Cortez, Noticias Dot, 2014).

El intercambio, por supuesto, es bidireccional entre todos los dispositivos que cuenten con esta tecnología integrada o con un adaptador que se puede adquirir en cualquier tienda especializada. (Cortez, Noticias Dot, 2014)

Bluetooth es para periféricos y para una mejor transferencia de datos entre ordenador y dispositivos, no es como una red de transmisión. Es un desarrollo muy sofisticado de la tecnología infrarrojo. (Cortez, Noticias Dot, 2014)

Otra de las diferencias entre Bluetooth y Wifi radica en su velocidad, porque mientras que la comunicación en el caso de Bluetooth se realiza a una velocidad de un megabyte por segundo, la velocidad de Wifi es 10 veces mayor que la de Bluetooth. (Cortez, Noticias Dot, 2014)

Pero eso no significa que cuando se accede desde una cafetería, restaurante, universidad o un domicilio con cobertura Wifi la conexión a Internet se realice a esa velocidad. (Cortez, Noticias Dot, 2014)

Los 10 megabytes por segundo del Wifi corresponden a la velocidad con la que el ordenador se conecta con la antena y el dispositivo que gestiona el tráfico de la red inalámbrica, mientras que la velocidad a la que se navegue por Internet dependerá del servicio (vía teléfono, DSL o cable, por ejemplo). Lo más normal, es que la velocidad de intercambio de información a través de la red interna sea muy superior a la conseguida cuando accedemos a Internet. (Cortez, Noticias Dot, 2014)

Es esta diferencia de velocidad, lo que hace que el uso o el provecho que podamos sacar de cada tecnología. Al poder vincularnos con una red "LAN" a alta velocidad hace que Wifi se preste más para el envío de paquetes de datos mientras que Bluetooth sea utilizado como una tecnología para la comunicación directa con dispositivos determinados. (Cortez, NoticiasDot, 20014)

**Tabla 1: Comparación Bluetooth vs Wifi. Fuente: [www.noticiasdot.com](http://www.noticiasdot.com)**

Bluetooth	Wifi
Tecnología para comunicar entre dispositivos. Es la evolución de la comunicación infrarroja.	Tecnología de comunicación para acceder a internet de manera inalámbrica, a redes locales (LAN)
Permite comunicación en distancias de hasta 10 metros aproximadamente.	Su alcance es de alrededor de 30 metros
Su velocidad de comunicación es de un megabyte por segundo.	Su velocidad de conexión es de 10 megabytes por segundo.
No se utiliza para conectarse a internet salvo como “modem” para ordenadores portátiles utilizando redes GPRS.	Se utiliza para comunicación entre dispositivos sin cableado.

**Fuente: Realización Propia**

## II.4 Raspberry PI

El Raspberry PI es un dispositivo electrónico que cuenta con un sistema operativo el cual se encuentra cargado en una tarjeta SD. Este dispositivo dispone de una salida de audio HDMI, una salida de vídeo RCA y un puerto Ethernet. Esto permite que se pueda emplear como un ordenador, llegando a realizar funciones tales como la navegación en internet.

Se puede decir que los proyectos que cuentan con la herramienta Raspberry PI son más ligados al software que al hardware. Como se trata simplemente de un ordenador con Linux, la mayoría de los proyectos se basan en software, centros de medios, gráficos, sonido y multimedia. Sin embargo, se puede hacer un poco de control de hardware de manera sencilla con ayuda de los pines GPIO. (Fundación Raspberry PI, S.F)

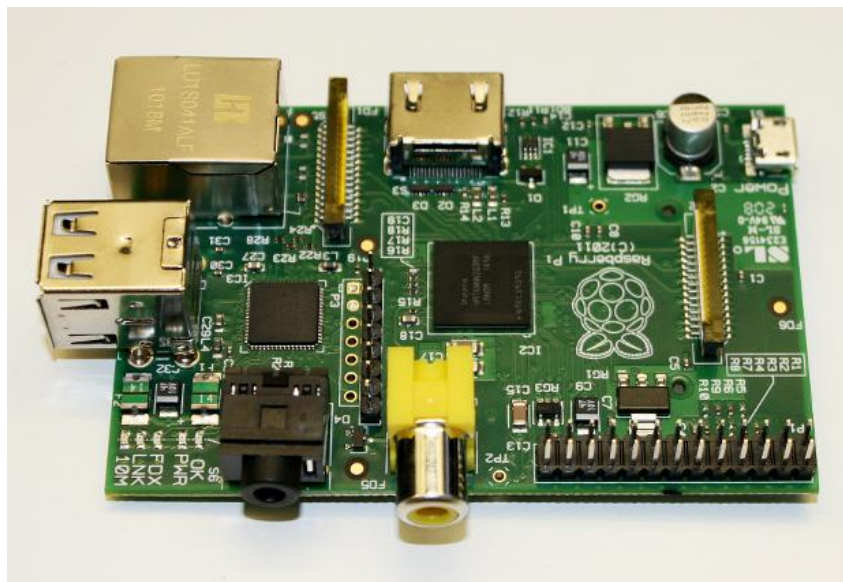


Figura 2: Raspberry PI

Fuente: página web: [www.zdnet.de](http://www.zdnet.de)

Este dispositivo, como se muestra en la Figura 2, está constituido por un circuito integrado, del tamaño de una tarjeta de crédito, que se alimenta a través de un puerto Micro USB. Adicionalmente, posee un conector *ETHERNET* y unos pines GPIO (*General Purpose Input/Output*) para que se pueda interactuar con la placa utilizando sensores o botones, cuatro puertos USB (*Universal Serial Bus*), una ranura para conectar una tarjeta de memoria SD y una salida de video HDMI (*High Definition MultiMedia Interface*). Su precio oscila entre los US\$ 25 y los US\$ 35 y corre el sistema operativo LINUX. Su procesador es un ARM1176JZF-S a 700 MHz y cuenta con 256 Mb de memoria RAM. Para usarlo basta con conectarle un teclado, alimentación a través del puerto Micro USB, y enchufarlo a un monitor o una televisión.

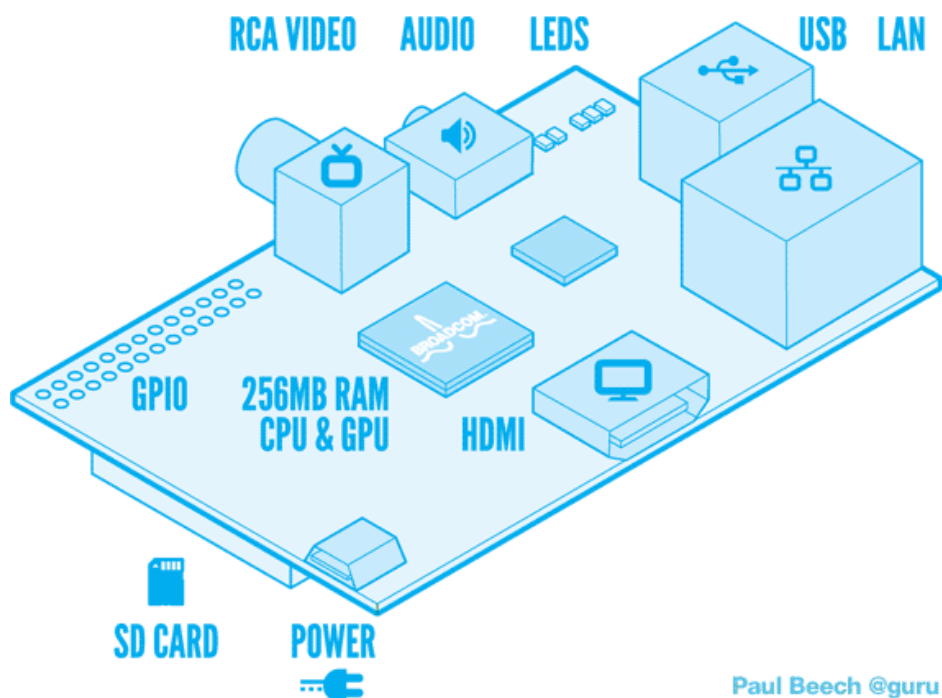


Figura 3: Circuito integrado de la Raspberry PI

Fuente: <http://www.raspberrypi.es/hardware-raspberry-pi.php>

Existen dos modelos: el modelo A y el modelo B. El modelo A solo tiene un puerto USB, carece de controlador *ETHERNET*, y cuesta menos que el modelo B, el cual tiene cuatro puertos USB y controlador *ETHERNET* 10/100. A pesar que el Modelo A no tiene un puerto *ETHERNET* se puede conectar a una red usando un adaptador USB-*ETHERNET* suministrado por el usuario. Por otro lado, a ambos modelos se puede conectar un adaptador Wifi por USB, para tener acceso a redes inalámbricas, o internet. En la Tabla 2 se muestra una comparación entre ambos modelos.

**Tabla 2 Comparación de modelos de Raspberry PI**

	Modelo A	Modelo B
Precio:	\$ 25	\$ 35
CPU:	ARM1176JZF-S a 700 MHz (familia ARM11)	
Memoria (SDRAM):	256 MB	512 MB
Puertos USB	1	4
Salidas de vídeo:	Conector RCA, HDMI	
Almacenamiento integrado:	SD / MMC / ranura para SDIO	
Conectividad de red:	Ninguna	10/100 ETHERNET (RJ45) via hub USB
Periféricos de bajo nivel:	GPIO	
Consumo energético:	500 mA, (2.5 W)	700 mA, (3.5 W)
Fuente de alimentación:	5 V vía Micro USB o GPIO header	
Sistemas Operativos	GNU/LINUX: Debian (RASPBIAN),	

**Fuente: Realización Propia**

## GPIO Raspberry Pi

Raspberry Pi, puede comunicarse con dispositivos externos mediante el conector GPIO incorporado. En dicho conector, se integran pines de alimentación (+5 y +3.3 V), tierra, y entradas/salidas capaces de implementar diferentes protocolos. Se observa en la Figura 4, un esquema del GPIO y el uso que se le da a cada uno de los pines.

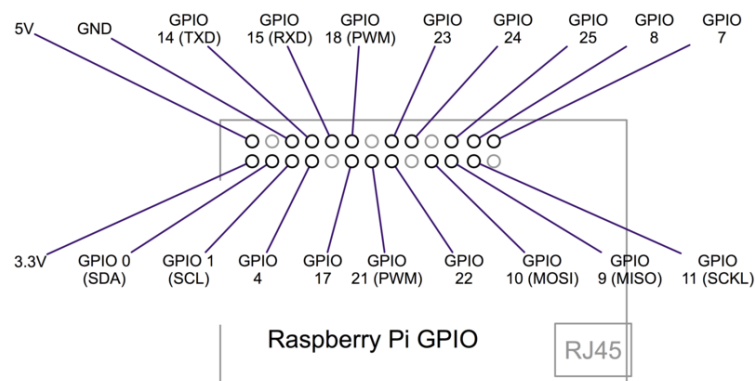


Figura 4: GPIO de la Raspberry PI

Fuente: (Adafruit). <http://learn.adafruit.com/assets/3059>

Es importante tener en cuenta que, a nivel software, se debe saber con qué pin se quiere comunicar, ya que además de los pines correspondientes a +5 voltios, +3.3 voltios y tierra, se tienen varios pines de uso genérico donde se pueden conectar dispositivos *hardware*, como se muestra en la siguiente Figura 5



Figura 5: Pines del Raspberry PI

Fuente: (Adafruit). <http://learn.adafruit.com/assets/3059>

### **II.4.1 Arduino**

Arduino es una plataforma de hardware y software de código abierto, basada en una sencilla placa con entradas y salidas, analógicas y digitales, en un entorno de desarrollo que está basado en el lenguaje de programación *Processing*, es decir, una plataforma de código abierto para prototipos electrónicos. Al ser *open source*, tanto su diseño como su distribución, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin necesidad de licencia. (García, 2011)

### **II.4.2 Comparación entre Raspberry PI y Arduino**

Si se desea hacer un proyecto de hardware se dice que el Arduino es, con mucho, la mejor opción ya que las entradas y salidas analógicas PWM añaden todo un espectro de compatibilidad que por su parte el Raspberry PI no puede hacer de forma predeterminada. Además, si se habla acerca de pines de E / S, se sabe que se le permiten conectar múltiples sensores y componentes de retroalimentación al Arduino, sin embargo, este último no es tan poderoso como el Raspberry PI, debido a que no cuenta con audio adecuado, vídeo o Internet fuera de la caja (sin embargo, puede agregar la funcionalidad básica de esta). Por último se puede decir que el Arduino puede enviar datos a su PC o Pi, en serie y, a continuación, puede crear un programa para leer estos datos y hacer algo. (Benitez, 2008)

Si se desea hacer un proyecto de software, entonces el Pi es el camino a seguir.



A continuación se muestra una tabla en la cual se puede comparar ambos dispositivos:

**Tabla 3 Comparación entre Raspberry PI y Arduino**

	<b>Arduino</b>	<b>Raspberry Pi modelo B</b>
Sistema operativo	Custom	Linux
Adecuado para	Hardware	Software
Numero de terminales de E/S	14 digital (6PWM), 6 analógica	8 digital
Audio / Video	Funcionabilidad programable básica	1 puerto HDMI, 1 puerto RCA
Internet	Vía escudo	1 puerto Ethernet

**Fuente: Realización Propia**

## **II.5 Base de Datos**

Un sistema de base de datos es, básicamente, un sistema computarizado para llevar registros, manipulados por los usuarios, a través de una interfaz gráfica. Cada registro constituye una colección de archivos que recogen diferentes campos o tipos de datos, dependiendo de la estructura que se desee utilizar. Los usuarios pueden realizar una variedad de operaciones sobre dichos archivos, por ejemplo, agregar o eliminar archivos, así como también, insertar, recuperar, modificar o eliminar datos de los archivos.

La finalidad general de la base de datos es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base a peticiones. La información en cuestión puede ser cualquier cosa que sea de importancia para el individuo u organización; en otras palabras, todo lo que sea necesario para auxiliarle en el proceso general de su administración.

### **II.5.1 Sistema Gestor de Base de Datos**

Un sistema gestor de bases de datos (SGBD) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. (Leon, 2009)

### **II.6 Sistema operativo**

Un sistema operativo es un software el cual permite la operatividad o funcionamiento ya sea una pc, un móvil, entre otros.

Las funciones más básicas de un sistema operativo son administrar los recursos, coordinar el hardware, etc.

## **II.6.1 Linux**

LINUX es un sistema operativo, compatible UNIX. Dos características muy peculiares lo diferencian del resto de los sistemas que se encuentran en el mercado, la primera, es que es libre, esto significa que no se tiene que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo, la segunda, es que el sistema viene acompañado del código fuente.

El sistema lo forman el núcleo del sistema (KERNEL) más un gran número de programas y librerías que hacen posible su utilización. El núcleo del sistema sigue en continuo desarrollo bajo la coordinación de Linus Torvalds, la persona de la que partió la idea de este proyecto, en 1991.

El sistema operativo LINUX se distribuye bajo la GNU (*General Public License*), ha sido diseñado y programado por multitud de programadores alrededor del mundo, y presenta numerosas características, entre las más importantes se encuentran: multitarea, multiusuario, multiplataforma, multiprocesador, protección de la memoria entre procesos, librerías compartidas, consolas virtuales múltiples y, además, incluye diversos protocolos de red.

## **II.6.2 Raspbian**

El sistema operativo RASPBIAN es una distribución de GNU/LINUX y por lo tanto libre. Técnicamente, el sistema operativo es una versión no oficial de Debian Wheezy armhf para el procesador del RASPBERRY PI, con soporte optimizado para cálculos en coma flotante por hardware, lo que permite dar más rendimiento según

sea el caso. RASPBIAN ofrece más que un conjunto de programas básicos y utilidades; viene con 35000 paquetes programas pre compilados, incluido en un formato que hace más fácil la instalación en la RASPBERRY PI. La distribución usa LXDE (Lightweight X11 Desktop Environment) como escritorio y MIDORI como navegador web. Además, contiene herramientas de desarrollo como IDLE (Integrated Development Environment), para el lenguaje de programación PYTHON o SCRATCH, y diferentes ejemplos de juegos usando los módulos PYGAME.

Posteriormente se añadió a los repositorios de RASPBIAN una versión oficial de ORACLE JAVA JDK ARM con soporte para coma flotante por hardware, que ofrece más rendimiento que la versión OpenJDK ARM ya existente hasta ese momento y con más compatibilidad con las aplicaciones. También se anunció que esta versión de ORACLE JAVA JDK se incluiría dentro de la distribución en futuras versiones de RASPBIAN.

## **II.7 Android**

Android es considerado como:

Sistema operativo basado en Linux, orientado a Smartphones y tabletas desarrollado por Android Inc. de Google en el año 2007. El sistema operativo posee una arquitectura basada en Apps y el uso de librerías basadas en C y C++ para acceder a los componentes de los diferentes dispositivos donde el sistema operativo se esté ejecutando. (W. Frank Ableson, 2011)

La programación para sistema operativo Android está basada en Java ya que esta es de código abierto, pero al ser creada por Google se vieron en la necesidad de crear un entorno de desarrollo para este lenguaje, por lo que crearon Android Studio.

### **II.7.1 Arquitectura de Android**

La arquitectura del sistema operativo Android está basada en cuatro capas principales que se nombran a continuación:

- El núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos.

Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

Android utiliza el Kernel 2.6 de Linux por su robustez demostrada y por la implementación de funciones básicas para cualquier sistema operativo, por ejemplo: seguridad, administración de memoria y procesos, implementación de conectividad de red (Network stack) y varios intérpretes (drivers) para comunicación con los dispositivos físicos (hardware). (Catalan, 2010)

Android utiliza como base el Kernel de Linux pero los dos sistemas no son lo mismo, Android no cuenta con un sistema nativo de ventanas de Linux ni tiene soporte para Glibc (librería estándar de C) ni tampoco es posible utilizar la mayoría de aplicaciones de GNU de Linux. (Catalan, 2010)



**Figura 6: Kernel de Linux**

Fuente: Catalán A. (2011)

- Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones. (W. Frank Ableson, 2011)

Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: que ejecuta ficheros Dalvik ejecutables (.dex) –formato optimizado para ahorrar memoria. Además, está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al Kernel de Linux algunas funciones como Threading y el manejo de la memoria a bajo nivel. (W. Frank Ableson, 2011)

A partir de Android 5.0 se reemplaza Dalvik por ART. Esta nueva máquina virtual consigue reducir el tiempo de ejecución del código Java hasta en un 33%.

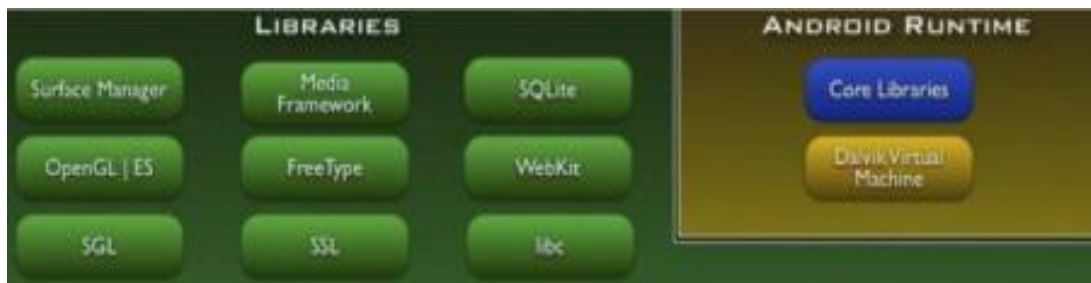
También se incluye en el Runtime de Android el “core libraries” con la mayoría de las librerías disponibles en el lenguaje Java. (W. Frank Ableson, 2011)

- Librerías Nativas

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

- System C library: una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- Media Framework: librería basada en PacketVideo's OpenCORE; soporta codecs de reproducción y grabación de multitud de formatos de audio vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- Surface Manager: maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- WebKit: soporta un moderno navegador Web utilizado en el navegador Android y en la vista Webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- SGL: motor de gráficos 2D.

- Librerías 3D: implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador Hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- FreeType: fuentes en bitmap y renderizado vectorial.
- SQLite: potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- SSL: proporciona servicios de encriptación Secure Socket Layer (capa de conexión segura). (W. Frank Ableson, 2011)



**Figura 7: Librería y Ejecución de Android**

**Fuente: Catalán A. (2011)**

- Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones, etc.).



Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad). Este mismo mecanismo permite a los usuarios reemplazar componentes. (W. Frank Ableson, 2011)

Los servicios más importantes que incluye son:

- Views: extenso conjunto de vistas, (parte visual de los componentes).
- Resource Manager: proporciona acceso a recursos que no son en código.
- Activity Manager: maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- Notification Manager: permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- Content Providers: mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos). (Tomás, J. (2011))

Sobre las librerías encontramos una estructura que nos brinda un contexto para desarrollar, este Framework permite a los desarrolladores aprovechar un sistema de vistas ya construido, administrar notificaciones y Accesar datos a través de proveedores de contenido entre otras cosas. (Catalan, 2010)



**Figura 8: Estructura de la Aplicación Android**

**Fuente: catalán A. (2011)**

- Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema. (W. Frank Ableson, 2011)

Normalmente las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción podemos utilizar el Android NDK (Native Development Kit). (W. Frank Ableson, 2011)

Las aplicaciones centrales que incluye el sistema por defecto son: teléfono, navegador, manejo de contactos, etc. En esta capa de la arquitectura es donde trabajaremos desarrollando aplicaciones.

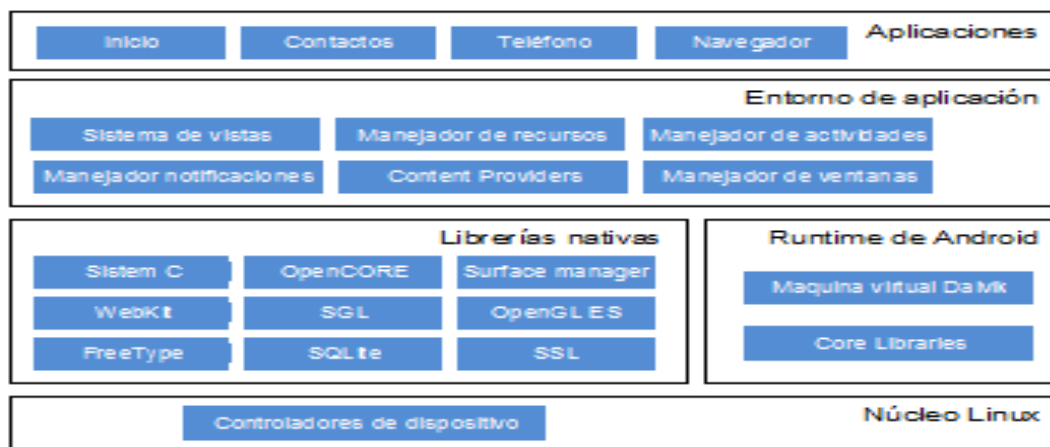


Figura 9: Arquitectura de Android

Fuente: Tomas, J. (2011).

## II.7.2 Bloque básico de una aplicación

Los componentes básicos de una aplicación Android son: Activities, Intents, Views, Services, Content Providers y Broadcast receivers.








Figura 10: bloque básico de una aplicación Android

Fuente: catalán A. (2011)

### II.7.3 Comparativa de Android con otras plataformas

Haciendo una comparación del S.O Android con otros S.O obtenemos la siguiente tabla:

Tabla 4 Comparativa de Android con otras plataformas

					
	Apple iOS 8	Android 5.0	Windows Phone 8	BlackBerry 10	Firefox OS 2.2
Compañía	Apple	Open Handset Alliance	Microsoft	BlackBerry	Mozilla Foundation
Núcleo del SO	Mac OS X	Linux	Windows NT	QNX	Linux
Licencia de software	Propietaria	Libre y abierto	Propietaria	Propietaria	Libre y abierto
Año de lanzamiento	2007	2008	2010	1999	2013
Fabricante único	Sí	No	No	Sí	No
Variedad de dispositivos	Modelo único	Muy alta	Media	Baja	Muy baja
Soporte memoria externa	No	Sí	Sí	Sí	Sí
Motor del navegador web	WebKit	WebKit/Chromium (Blink)	Trident	WebKit	WebKit
Tienda de aplicaciones	App Store	Google Play	Windows Marketplace	BlackBerry World	Firefox Marketplace
Número de aplicaciones	800.000 (marzo 2013)	800.000 (marzo 2013)	130.000 (enero 2013)	100.000 (enero 2013)	¿?
Coste publicar	\$99 / año	\$25 una vez	\$99 / año	Sin coste	Sin coste
Otras tiendas sin supervisión	No	Sí	No	Sí	Sí
Familia CPU soportada	ARM	ARM, MIPS, x86	ARM	ARM	ARM, x86
Soporte 64 bits	Sí	Sí	No	No	No
Máquina virtual	No	Dalvik / ART	.net	No	Navegador Web
Lenguaje de programación	Objective-C, C++	Java, C++	C#, Visual Basic, C++	C, C++, Java	HTML5, CSS, JavaScript
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows	Windows, Mac	Windows, Mac, Linux
Multiusuario	No	Sí	No	No	No
Modo invitado	Sí	Sí	No	No	No

Fuente: Tomás, J. (2011)

Otro aspecto fundamental a la hora de comparar las plataformas móviles es su cuota de mercado. En la siguiente gráfica podemos ver un estudio realizado por la empresa Gartner Group, donde se muestra la evolución del mercado de los sistemas operativos para móviles según el número de terminales vendidos.

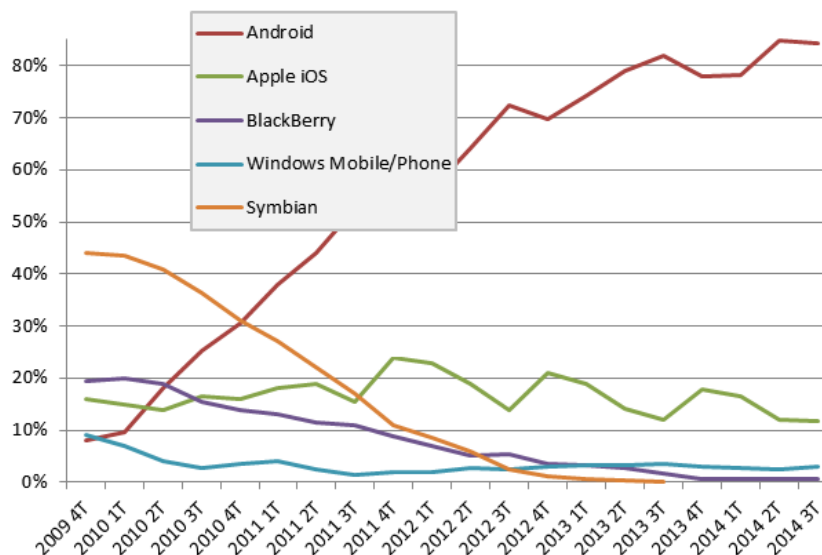


Figura 11: Porcentajes de dispositivos adquiridos hasta el 3° trimestre del 2014

Fuente: Tomás, J. (2011).

## II.7.4 Ambiente de desarrollo

Un ambiente o entorno de desarrollo es un programa de software que le facilita al programador la iteración con la computadora, traduce el lenguaje de maquina a un lenguaje que el programador entienda con facilidad.

Hoy en día existen diversos entornos de desarrollo con diferentes características que satisface las necesidades del programador.

## II.7.5 Android Studio

Android Studio es un entorno de desarrollo integrado (IDE), basado en IntelliJ IDEA de la compañía JetBrains, que proporciona varias mejoras con respecto al

Plugin ADT (Android Developer Tools) para Eclipse. Android Studio utiliza una licencia de software libre Apache 2.0, está programado en Java y es multiplataforma. Fue presentado por Google el 16 de mayo del 2013 en el congreso de desarrolladores Google I/O, con el objetivo de crear un entorno dedicado en exclusiva a la programación de aplicaciones para dispositivos Android, proporcionando a Google un mayor control sobre el proceso de producción. Se trata pues de una alternativa real a Eclipse, el IDE recomendado por Google hasta la fecha, pero que presentaba problemas debido a su lentitud en el desarrollo de versiones que solucionaran las carencias actuales (es indispensable recordar que Eclipse es una plataforma de desarrollo, diseñada para ser extendida a través de Plugins). (Android, 2014)

Android Studio se ha mantenido durante todo este tiempo en versión beta, pero desde el 8 de diciembre de 2014, en que se liberó la versión estable de Android Studio 1.0, Google ha pasado a recomendarlo como el IDE para desarrollar aplicaciones para su sistema operativo, dejando el Plugin ADT para Eclipse de estar en desarrollo activo. (Android, 2014)

### **II.7.6 Principales características que incluye Android Studio**

- Soporte para programar aplicaciones para Android Wear (sistema operativo para dispositivos corporales como por ejemplo un reloj).
- Herramientas Lint (detecta código no compatible entre arquitecturas diferentes o código confuso que no es capaz de controlar el compilador) para detectar problemas de rendimiento, usabilidad y compatibilidad de versiones.

- Utiliza ProGuard para optimizar y reducir el código del proyecto al exportar a APK (muy útil para dispositivos de gama baja con limitaciones de memoria interna).
- Integración de la herramienta Gradle encargada de gestionar y automatizar la construcción de proyectos, como pueden ser las tareas de testing, compilación o empaquetado.
- Nuevo diseño del editor con soporte para la edición de temas.
- Nueva interfaz específica para el desarrollo en Android.
- Permite la importación de proyectos realizados en el entorno Eclipse, que a diferencia de Android Studio (Gradle) utiliza ANT.
- Posibilita el control de versiones accediendo a un repositorio desde el que poder descargar Mercurial, Git, Github o Subversion.
- Alertas en tiempo real de errores sintácticos, compatibilidad o rendimiento antes de compilar la aplicación.
- Vista previa en diferentes dispositivos y resoluciones.

### **II.7.7 Comparativa de Android Studio con Eclipse**

Para una mayor comprensión de las diferencias y novedades que presenta Android Studio con respecto al IDE Eclipse, y más concretamente con el ADT para Android, se propone la siguiente tabla comparativa entre ambas opciones:

En la tabla siguiente se pueden apreciar una serie de características que permiten comparar ambos entornos, aunque sería necesario tener en cuenta otros aspectos que puedan facilitarnos la tarea de decidir si utilizar uno u otro IDE.



DISEÑO DE UN SISTEMA DE AUTOMATIZACIÓN EN EL CONTROL DE CONSUMO, RACIONAMIENTO Y ANÁLISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE ALMACENAMIENTO

---

**Tabla 5: Comparación de Android Studio y Eclipse. Fuente: Academia Android, 2011**

Características	Android Studio	ADT
Sistema de construcción	Gradle	ANT
Construcción y gestión de proyectos basado en Maven (herramienta de software para la gestión y construcción de proyectos Java, similar a Apache ANT, pero su modelo es más simple ya que está basado en XML)	Si	No (es necesario instalar un plugin auxiliar)
Construir variantes y generación de múltiples APK (muy útil para Android Wear)	Si	No
Refactorización y completado avanzado de código Android	Si	No
Diseño del editor gráfico	Si	Si
Firma APK y gestión de almacén de claves	Si	Si
Soporte para NDK (Native Development Kit: herramientas para implementar código nativo escrito en C y C++)	Próximas versiones	Si
Vista en tiempo real de renderizado de layouts	Si	No
Nuevos módulos en proyecto	Si	No
Editor de navegación	Si	No
Generador de assets	Si	No

### **II.7.8 Ventajas uso Android Studio**

- Android Studio ha pasado a ser el entorno recomendado para el desarrollo de aplicaciones en Android, al tratarse de un IDE oficial de Google en colaboración con JetBrains (compañía de desarrollo software especializada en diseño de IDEs).
- Android Studio permite la creación de nuevos módulos dentro de un mismo proyecto, sin necesidad de estar cambiando de espacio de trabajo para el manejo de proyectos, algo habitual en Eclipse.
- Con la simple descarga de Android Studio se disponen de todas las herramientas necesarias para el desarrollo de aplicaciones para la plataforma Android.
- Su nueva forma de construir los paquetes .apk, mediante el uso de Gradle, proporciona una serie de ventajas más acorde a un proyecto Java.
- Facilita la distribución de código, y por lo tanto el trabajo en equipo.
- Reutilización de código y recursos.
- Permite compilar desde línea de comandos, para aquellas situaciones en las que no esté disponible un entorno de desarrollo.
- Mayor facilidad para la creación de diferentes versiones de la misma aplicación, que proporciona numerosas ventajas como puede ser la creación

de una versión de pago y otra gratuita, o por ejemplo diferentes dispositivos o almacén de datos.

- Integración con Google Cloud Platform, para el acceso a los diferentes servicios que proporciona Google en la nube.
- Editor de diseño que muestra una vista previa de los cambios realizados directamente en el archivo xml.

## **II.8 XamppServer**

Xampp es un acrónimo utilizado para describir un sistema de infraestructura de internet y utiliza las siguientes herramientas:

- X: Cualquier sistema operativo.
- A: Apache, como servidor web.
- M: MySQL, como gestor de base de datos.
- P: PHP (generalmente).
- P: Perl o Python, como lenguaje de programación

XamppServer es un entorno de desarrollo web para cualquier sistema operativo que permite crear aplicaciones web y base de datos. También incluye PHPMyAdmin y SQLiteManager para el manejo fácil de las bases de datos.

XamppServer provee a los desarrolladores de un manejador de base de datos (MySQL), un software para servidor web (Apache) y un software de programación de script Web (generalmente PHP).

## II.9 SQL

SQL (*Structured Query Language* ó Lenguaje Estructurado de Consulta), es un lenguaje bastante sencillo, principalmente orientado a bases de datos y, sobre todo, al manejo de consultas. El objetivo principal de SQL es la realización de consultas y cálculos con los datos de una o varias tablas.

El lenguaje SQL es utilizado para trabajar con base de datos relacionales. SQL es una herramienta que controla todas las funciones que proporciona un sistema de gestión de base de datos, como por ejemplo consultar, definir, obtener o manipular los datos, y al mismo tiempo controlar el acceso y definir las restricciones de integridad en la base de datos.

## II.10 MySQL

Es un sistema de administración de bases de datos de código fuente abierto que se ejecuta en muchas plataformas, incluyendo Windows, Solaris, Linux y Macintosh. (Meneses, 2007)

## **II.11 Python**

Python es un lenguaje muy expresivo, es decir, los programas Python son muy compactos: un programa Python suele ser bastante más corto que su equivalente en lenguajes como C. (Python llega a ser considerado por muchos un lenguaje de programación de muy alto nivel.) (Marzal, 2008)

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

## **II.12 Samba**

Samba son un conjunto de aplicaciones para Linux, que implementan el protocolo de comunicación SMB (Server Message Block), utilizado por los sistemas operativos Microsoft Windows, cuya finalidad es compartir carpetas e impresoras.

Básicamente Samba permite a ordenadores que emplean Linux, conectarse a carpetas compartidas en ordenadores con Windows y compartir carpetas como si de un sistema Windows se tratara. Gracias a Samba, en una red se puede tener ordenadores con Windows y ordenadores con Linux de forma que estos puedan intercambiar información a través de carpetas compartidas, de la misma forma que se haría si todos los ordenadores fueran Windows.

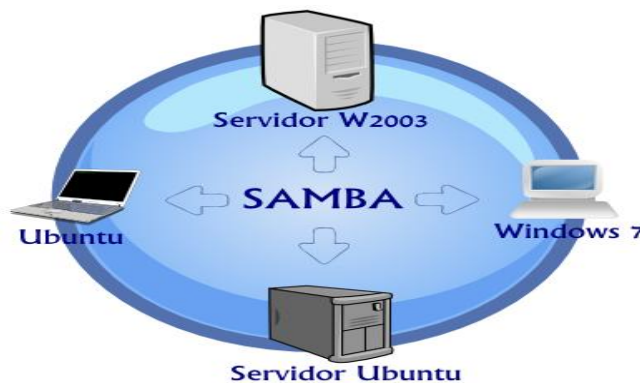


Figura 12: Funcionamiento Samba

## II.13 Smartphone

El término Smartphone pertenece a la lengua inglesa y hace referencia a aquello que, en nuestro idioma, conocemos como teléfono inteligente. Se trata de un teléfono celular (móvil) que ofrece prestaciones similares a las que brinda una computadora (ordenador) y que se destaca por su conectividad. (Rosales, 2010)

Es habitual que se ubique al Smartphone a mitad de camino entre un teléfono celular convencional y una computadora portátil. El Smartphone cuenta con todas las funciones básicas del celular (permite realizar llamadas telefónicas, enviar mensajes de texto, etc.) y le agrega características avanzadas (conexión a Internet, capacidad multimedia, pantalla táctil). (Rosales, 2010)

En la actualidad existen diversos tipos de Smartphone, algunos con mayor cantidad de funciones que otros. Con un Smartphone, es posible que una persona pueda conectarse a la Web a través de una red 3G o Wifi; consultar una ubicación mediante un GPS entre otros. (Rosales, 2010)

Entre los sistemas operativos que se emplean en los Smartphone, podemos mencionar a iOS, Android, BlackBerry OS, Symbian OS y Windows Phone.

Se entra a fondo en este concepto ya que todas las prestaciones del servicio serán enviadas al usuario a través de un teléfono móvil pero no puede ser un teléfono móvil sencillo sino uno que abarque con las especificaciones específicas planteadas en el concepto de Smartphone. (Rosales, 2010)

## **II.14 Sensor de Agua**

Las moléculas de agua cambian el volumen de materiales orgánicos, la conductividad y peso de materiales higroscópicos y absorbentes químicos, y en general la impedancia de casi cualquier material. Estos cambios son utilizados por los métodos o principios de medición de los instrumentos para la medición de humedad. Así, el contenido de agua puede ser medido removiendo el agua (vapor) de una muestra y medir el cambio de peso. Otros principios fundamentales son la evaporación de una superficie de agua con una corriente de gas (psicrometría) y el enfriamiento de una muestra de gas hasta que la condensaciones detectada. (Bentley, 2008)

## **II.15 Sensores de nivel**

Al igual que en la medición de muchas otras variables físicas, la medición de nivel en el interior de depósitos puede obtenerse en formas directas e indirecta. En el primer caso se aprovecha directamente la variación del nivel para hacer la medición, este es el caso de los flotadores, de los sensores basados en electrodos, etc. En el segundo caso, se detecta la diferencia de altura, en forma indirecta, como lo son por ejemplo los sensores sónicos. Los niveles se pueden medir y mantener mediante dispositivos mecánicos de caída de presión, eléctricos y electrónicos entre otros (Machado, 2012)

En el caso de este proyecto el sensor de nivel funciona gracias a un dispositivo llamado Reed Switch que es el encargado de indicar los cambios de nivel del agua y el mismo se encuentra en forma vertical alrededor de un tubo Pavco que se encuentra dentro del tanque de almacenamiento.

### **II.15.1 Reed Switch**

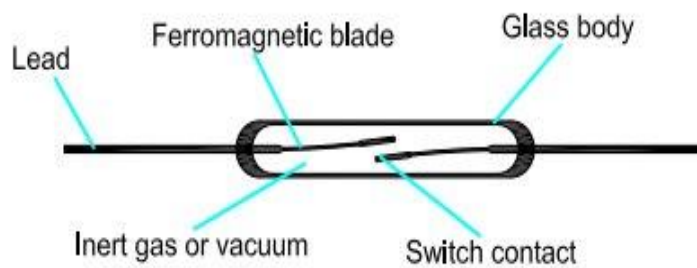
Como se puede deducir por el título en primer lugar estamos hablando de un interruptor (switch), la particularidad que distingue al reed switch respecto a los otros interruptores es que se ve activado por la presencia de un campo magnético. Cuando es normal abierto los contactos se cierran en la presencia del campo; cuando es normal cerrado se abren en presencia de un campo magnético. Fue inventado por W. B. Elwood en 1936 cuando trabajaba para Laboratorios Bell. (General, 2010)



Un reed switch (fig.13) se compone de dos laminas ferromagnéticas, generalmente compuestas de Ni y Fe, herméticamente selladas en una cápsula de vidrio. Las hojas se superponen internamente en la cápsula de vidrio dejando solo un pequeño espacio entre ellas, y se ponen en contacto ante la presencia de un campo magnético adecuado. El área de contacto de ambas hojas se recubre con metales duros (rodio o Rutenio) que elevan la vida útil de los contactos. (General, 2010)

El gas de la cápsula por lo general consiste de nitrógeno o algún gas inerte equivalente. Algunos se encuentran al vacío para trabajar con tensiones elevadas. Las láminas actúan como conductores del flujo magnético cuando se encuentran expuestas a un campo magnético externo, ya sea un imán permanente o una bobina electromagnética. (General, 2010)

Se generan polos de polaridad opuesta y la contactos se cierran cuando la fuerza magnética excédela fuerza elástica de las láminas. A medida que se reduce el campo magnético externo se reduce la fuerza magnética hasta que se hace menor a la fuerza elástica de las láminas momento en el cual se abre el contacto. (General, 2010)



**Figura 13: Reed Switch.**

**Fuente: [www.ayudaelectronica.com](http://www.ayudaelectronica.com)**

## **CAPITULO III**

### **Marco Metodológico**

En este capítulo se pretende dar una clara descripción de las diferentes actividades empleadas para dar una resolución al problema propuesto en el capítulo I y como fueron implementadas a lo largo de la elaboración de este proyecto, las etapas fueron:

1. Investigación y documentación
2. Diseño del Sistema
3. Requerimientos del diseño del sistema
4. Desarrollo de los sensores de nivel e indicadores de entrada y salida de agua
5. Desarrollo de la base de datos de los estados del tanque de almacenamiento
6. Desarrollo de las aplicaciones necesarias en el sistema
7. Desarrollo de la aplicación Android
8. Pruebas del sistema en entornos reales
9. Recomendaciones y Conclusiones
10. Elaboración del Tomo

#### **III.1 Investigación y documentación**

Se realizó una detallada investigación acerca de los diferentes dispositivos a utilizar en la elaboración del proyecto y en el diseño del sistema de medición, como los son: Raspberry, base de datos, servidores web, sensores de nivel, sensores de flujo y dispositivos Android así como también acerca de cada lenguaje de programación que podían ser utilizados en cada uno de los dispositivos que lo requiriera, de los

cuales se tuvo que investigar acerca de: MySQL, Python y Java bajo ambiente Android.

Debido a la extensa cantidad de información recopilada perteneciente a los distintos tópicos anteriormente nombrados se procedió a realizar una selección y análisis del contenido que permitiera conformar parte del marco teórico del trabajo de grado.

También se investigó acerca de las diferentes maneras existentes que permitieran la transmisión de datos que se podían utilizar para poder comunicar los dispositivos antes mencionados; entre ellas están el protocolo TCP/IP, el efecto hall visto como protocolo de transmisión de una señal, redes de acceso local inalámbrico bajo el estándar IEEE 802.11 y señales eléctricas digitales mediante el diferencial de potencial.

También se realizó una investigación bibliográfica extensa que permita desarrollar los siguientes objetivos:

- Mostrar de una manera dinámica al usuario la cantidad de agua aproximada disponible en el depósito de agua.
- Almacenar la información relevante del sistema.
- Comunicar el dispositivo Raspberry PI con el servidor web en el cual se encuentra la base de datos.

- Elaboración de un circuito detector de agua que permita su uso como un sensor de nivel.

### **III.2 Diseño del sistema**

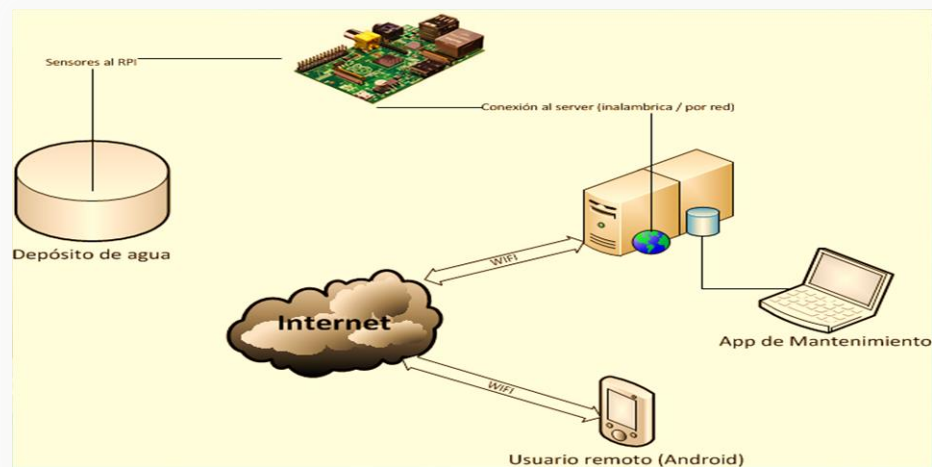
Luego del análisis de los objetivos planteados para este trabajo de grado y la información recopilada de la investigación teórica acerca de los dispositivos a utilizar, se procedió a definir el desarrollo de un diseño preliminar del sistema, tanto a escala de diseño como a escala real como se expresa a continuación.

Se planteó la construcción de un sistema de alerta para distintos niveles de agua que permita predecir un estimado de la cantidad del líquido disponible en las reservas de un tanque de almacenamiento, donde la información es captada por algún tipo de sensor que posibilite censar el nivel de agua. Dicha información debe ser transmitida al Raspberry PI mediante algún medio compatible, el cual se encargara de tener un software capacitado de detectar el nivel del fluido dentro de la estructura de almacenamiento en tiempo real y además poder ser capaz de notificar la existencia de fallas en algunos de los sensores en el caso de que existiera la eventualidad.

El sistema debe contar con un indicador que proporcione la información referente a la entrada y salida de agua al tanque de almacenamiento; dicha información es enviada al Raspberry PI mediante algún medio compatible, el cual es el encargado de manejar esa información en conjunto a los datos expresados por los sensores de nivel.

La información recopilada en el Raspberry PI debe pasar a ser parte de un servidor dedicado a una base de datos encargado de almacenar la información proveniente de los sensores, la cual debe ser catalogada en diferentes tablas según sus características que permitan establecer un orden al momento del intercambio de datos.

Por último el diseño del sistema debe contar con una aplicación Android que permita la consulta por parte del usuario final de los niveles de agua que se encuentran disponibles en tiempo real, alertas que permitan dar a conocer si hay entrada de agua al tanque de almacenamiento y advertencias que informen bajos niveles de agua dentro de la estructura o el posible daño de algún sensor; todas estas acciones estarán disponibles y se hace usos de ellas según los datos que sean consultados en la base de datos del sistema.



**Figura 14: Funcionamiento del Sistema**

**Fuente: Realización Propia**

### **III.3 Requerimientos del Diseño de Sistema**

Luego de realizar toda la investigación teórica acerca de los posibles dispositivos a utilizar en el proyecto y como funciona cada uno, además del uso de los conocimientos adquiridos a lo largo de la carrera para hacer un análisis acerca de las especificaciones se procedió a documentar los requerimientos de hardware y software implicados en el desarrollo del diseño del sistema, así como también se analizó el desempeño que debía de ofrecer el mismo.

Teniendo claros los requerimientos de hardware y software considerados adecuados se realizó la evaluación de los distintos dispositivos y elementos que podían optar como posible parte del sistema eligiendo solo aquellos que bajo sus especificaciones técnicas comprobaran que podían ofrecer el mayor rendimiento al sistema.

También se analizaron las diferentes configuraciones de los dispositivos y tecnologías utilizadas para la transmisión de datos y la comunicación entre ellos; al igual que diferentes lenguajes de programación con los cuales se podría haber realizado las distintas aplicaciones necesarias en el sistema.

Con esto se logró obtener cuales son las soluciones de los distintos tópicos que puedan cubrir los requerimientos que permita al sistema tener un buen funcionamiento tanto físico como lógico.

### **III.4 Desarrollo del sub-sistema detector de los niveles de agua**

Durante esta etapa se inició el desarrollo de lo que serían los sensores encargados de poder captar la información referente a la cantidad de agua disponible en el interior del depósito y los indicadores que permitieran conocer la entrada o salida de agua; datos que luego son enviados al Raspberry PI.

Para dicha tarea se elaboró un sistema autónomo basado en el funcionamiento de *reed switches* que facilitaba la lectura de las cantidades de agua existente en sus diferentes puntos sin que este afectara la medición por posibles daños debido a su estado líquido; permitiendo también una lectura individual de cada nivel y el envío al Raspberry PI de la información mediante pulsos digitales que consistían en el diferencial de potencial existente.

En el caso de los indicadores de entrada y salida se hizo uso de un pequeño circuito electrónico que permitía conocer la información en aquellos puntos que los que pudiese existir un flujo del agua.

### **III.5 Desarrollo de la base de datos de los estados**

En esta fase se procedió a llevar a cabo el desarrollo de la base de datos donde se encontrara toda la información relevante del sistema tales como: estados de los diferentes niveles del agua, donde se encuentra el agua en momentos específicos e información de los indicadores de entrada y salida.

La herramienta Workbench de MySQL facilitó el proceso de la elaboración de la base de datos, ya que cuenta con un entorno gráfico en el cual se pueden modelar las bases de datos y una vez realizada de manera gráfica la misma herramienta nos

arroja el código SQL que fue el lenguaje escogido para la elaboración de la base de datos, y se utilizó el gestor MySQL para el manejo de la misma.

Con esto, se procedió a crear una base de datos llamada “Nivel\_db” la cual se alojó en un servidor local, , utilizando el servidor MySQL incluido en XamppServer.

### **III.6 Desarrollo de las aplicaciones necesarias en el sistema**

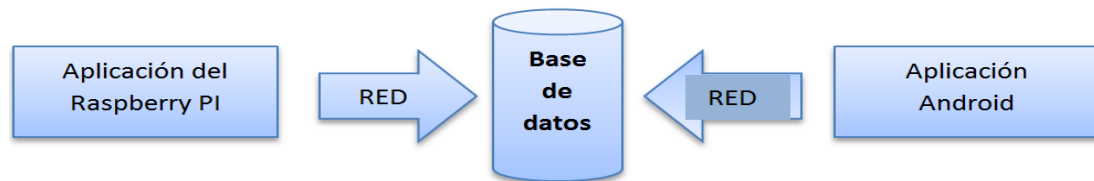
Para desarrollar el sistema, era necesario que el Raspberry PI interpretara la información obtenida desde los sensores al igual que en los indicadores de entrada y salida para que de esta manera se pudiera conocer la cantidad de agua existente, las posibles fallas y la entrada o salida de agua del tanque de almacenamiento; por lo que en el Raspberry PI se programó la aplicación “LlenarBaseDatos” en lenguaje Python que se encarga de ver los diferentes estados y enviará dicha información a la base de datos.

### **III.7 Desarrollo de la aplicación Android**

Unos de los objetivos del proyecto es que el usuario pueda desde su teléfono móvil ver la información referente al estado de los niveles de agua.

La aplicación “TuTanqueApp” está desarrollada para dispositivos con sistema operativo Android y permite al usuario conectarse por medio de su dispositivo móvil con la base de datos del sistema y observar los niveles de agua del mismo, así como también los indicadores de entrada y salida de agua.





**Figura 15: Planteamiento del diseño.**

**Fuente: Realización Propia**

### **III.8 Prueba del sistema en entorno real**

Luego de culminar con la elaboración de todo el diseño del sistema se dio paso a las pruebas en un entorno real en el cual se pudieran corroborar el buen funcionamiento de cada una de las etapas que lo conforman.

### **III.9 Recomendaciones y Conclusiones**

Después de haber realizado tanto pruebas simuladas como en entorno real, se identificaron las recomendaciones a tomar en cuenta para futuras investigaciones. Así como también se observaron los resultados del proyecto.

### **III.10 Elaboración del Tomo**

En esta fase se procedió a documentar todo lo relacionado al trabajo de grado así como también los resultados y conclusiones obtenidas de todas las pruebas.

## **CAPITULO IV**

### **Desarrollo**

En el presente capítulo se pretende dar una explicación de forma detallada de cada una de las etapas realizadas para lograr llevar a cabo el trabajo especial de grado de manera exitosa.

El desarrollo de este proyecto se lleva a cabo siguiendo la metodología expuesta con todos los detalles a través del capítulo III, en el cual se definieron cada una de las etapas de dicha metodología.

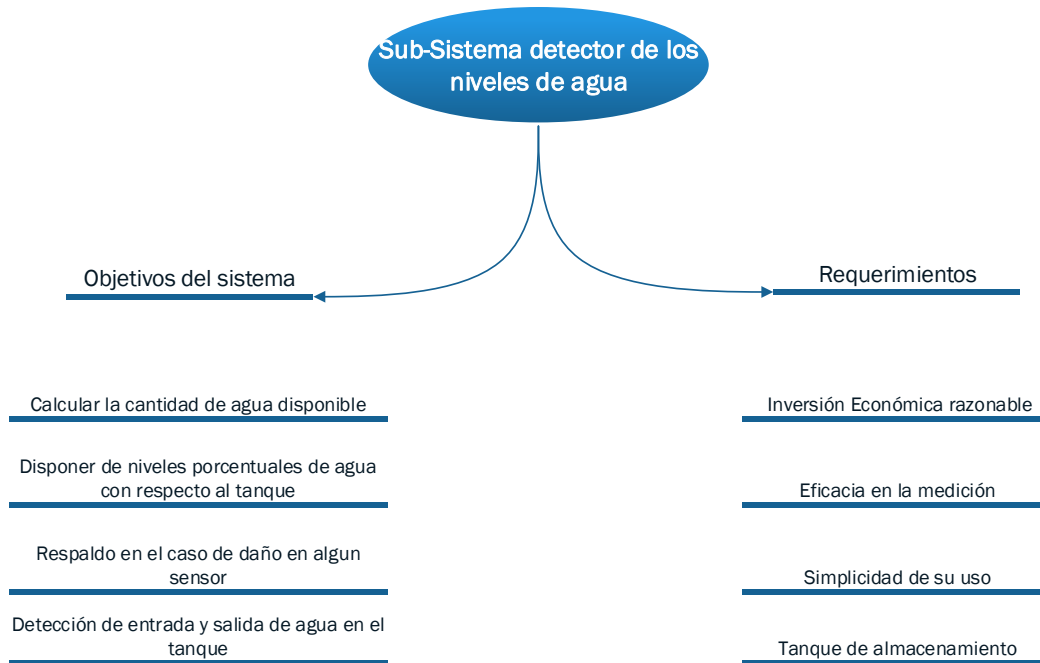
A continuación se explica con detalle las acciones realizadas para llevar a cabo el funcionamiento del sistema que permite el censo de los niveles de agua o entrada y salida de agua en la estructura de almacenamiento del respectivo usuario.

De la etapa 1, se puede decir que fue la base fundamental del diseño en construcción, debido a que en la misma se analizó el correcto funcionamiento del Raspberry PI, (donde se procedió con la instalación del sistema operativo Raspbian), así como el funcionamiento de los sensores de nivel y sensores de entrada y salida de agua, en conjunto con la programación respectiva para cada uno de los casos, los cuales representan dos grandes pilares para la construcción del sistema.

#### **IV.1 Diseño del sub-sistema detector de los niveles de agua**

Para la construcción del prototipo del sistema se planteó la posibilidad de poder crear un sub-sistema detector de nivel de agua, necesario para conocer la cantidad de agua almacenada. Dicha propuesta fue evaluada por los autores del

trabajo llegando a la conclusión de definir principalmente los objetivos que tendría tal estructura y los requerimientos que debía cumplir para el desarrollo y buen funcionamiento del subsistema; para esto se hizo uso de un diagrama de ideas como herramienta que permite representar ambos tópicos, el cual se presenta a continuación



Dentro de los requerimientos encontramos los siguientes puntos:

- La inversión económica: Se refiere a la cantidad de capital monetario que se debe invertir en la compra o construcción del sensor a utilizar.
- Eficacia en la medición: Se refiere al porcentaje de error existente en cada uno de los sensores y de las limitaciones que estos presenten.
- La simplicidad: Se refiere al grado de complejidad que representa su instalación y uso.

- Tanque de almacenamiento: El tanque necesario para el sub-sistema será un tanque de medidas 0,4x1x0,4 metros (ancho x alto x profundo), de una entrada y salida de agua con un diámetro de ½ pulgada y una salida independiente que simule una fuga dentro del tanque.

Mediante una investigación realizada se encontró que existen diferentes tipos de sensores que según el uso que les sea asignado pueden cumplir con los diferentes objetivos que se plantean en el sub-sistema; entonces, partiendo de eso se hizo un análisis y selección de los tipos de sensores de nivel existentes para luego dar paso a la creación del sistema.

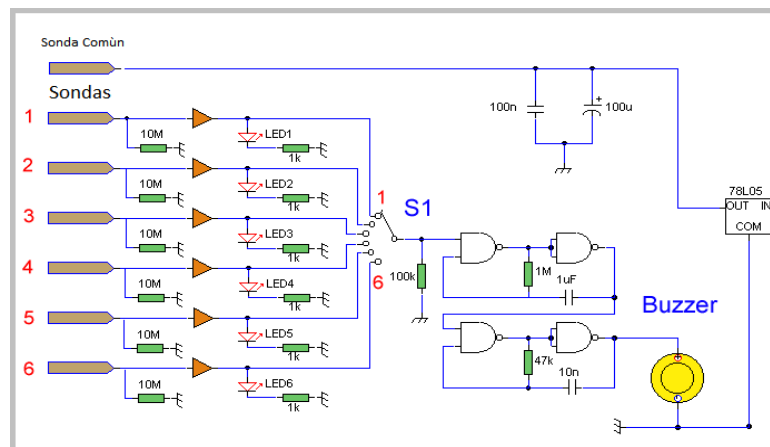
#### **IV.1.2 Análisis y Selección del sensor de nivel**

Para poder medir los distintos objetivos del sub-sistema detector de los niveles de agua disponibles en el tanque de almacenamiento se procedió a analizar los diferentes tipos de sensores existentes y disponibles actualmente en el mercado que realizara dicha actividad y que diera por respuesta a cada cambio de los niveles del líquido señales eléctricas que pudieran ser interpretadas en el Raspberry Pi. Como resultado de referida tarea, se pudo obtener un abanico de opciones que se postulaban como posible resolución a la medición de los niveles del líquido disponible; cada una de ellas presentadas a continuación a través de una breve explicación de sus funciones.

**Opción 1:** En la primera opción tenemos un circuito tipo alarma del nivel de agua que ofrece una indicación visual del nivel de líquido en un tanque, con una alarma

sonora conmutable; el cual funciona mediante la conductancia y resistencia de los fluidos a medir según la temperatura y el volumen que posea dicho líquido.

El circuito utiliza un buzzer trabajando bajo una alimentación de 5 voltios, una sonda por cada nivel de agua que tenga y una sonda "común". La sonda "común" está conectada a la alimentación de 5 voltios. Si la sonda del primer nivel y la sonda "común" detecta agua entre ellas entonces se activa la compuerta número 1 y se envía la señal a la entrada correspondiente al sensor que indica el primer porcentaje de agua detectado y así sucesivamente para el resto de los niveles.

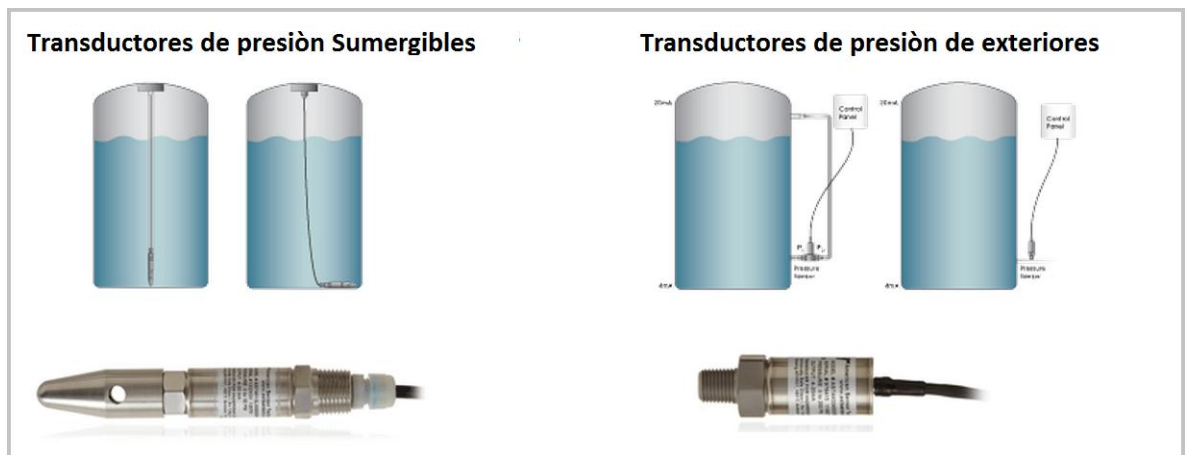


**Figura 16: Circuito alarma de nivel de agua**  
Fuente: [http://www.zen22142.zen.co.uk/Circuits/Alarm/water\\_lev\\_alm.htm](http://www.zen22142.zen.co.uk/Circuits/Alarm/water_lev_alm.htm)

**Opción 2:** Como segunda opción tenemos los sensores transductores de presión, los cuales ofrecen una medición de los niveles del líquido de una manera rentable y precisa mediante los valores de la densidad del líquido y su altura los cuales crean presión sobre el diafragma del transductor de presión.

Los transductores de presión o bien se pueden sumergir en el líquido o montados externamente. Los transductores de presión sumergibles pueden instalarse libremente o con un conducto rígido para controlar el líquido por encima de ellos. A medida que el nivel del líquido disminuye, la intensidad de la señal de salida disminuye.

Los transductores de presión también se pueden montar al lado o en la parte exterior del depósito a través de una línea independiente, línea de alimentación o drenaje.



**Figura 17: Tipos de transductores de presión**  
Fuente: <http://www.astensors.com/liquid-level-sensors.php>

**Opción 3:** Como tercera opción con un sistema de sensores de flujo, el cual está compuesto por un sensor de flujo que mide la cantidad de agua que pasa por una entrada o salida mediante la proporción de pulsos eléctricos derivado del paso del líquido por la tubería y el efecto Hall producido por un aspa interna; entonces, conociendo la cantidad de pulsos obtenidos se puede calcular la cantidad de agua que ha pasado por el sensor.

El sistema está compuesto por un sensor de flujo colocado en la tubería de entrada de agua y el mismo sensor ubicado en cada una de las salidas del tanque de almacenamiento; de esta manera sabiendo la cantidad de agua que sale y que entra podemos obtener un estimado de la cantidad del líquido disponible en el tanque.

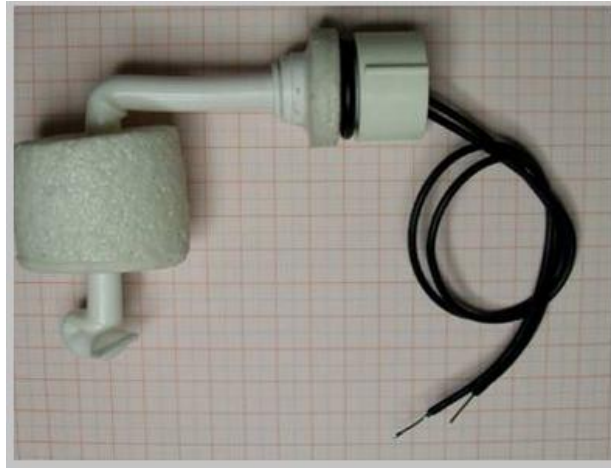


**Figura 18: Sensor de flujo de agua**

**Fuente:** <http://www.hobbytronics.co.uk/yf-s201-water-flow-meter>

**Opción 4:** Como última opción tenemos el sensor flotante tipo eléctrico el cual se presenta como un dispositivo simple, compuesto por un reed switch; componente que permite el cierre de un circuito abierto al estar presente un campo magnético, como por ejemplo, el de un imán contenido en el material flotante.

La manera de trabajo del sensor de nivel flotante consiste en que una vez que los niveles de agua empiezan a ascender, el material flotante se ve obligado a subir de manera recíproca a los niveles de agua, entonces; al pasar el imán por el reed switch interno cierra un circuito y deja pasar una señal eléctrica hacia el puerto de recepción de datos; para que así de esta manera, dependiendo de la cantidad de señales que existan en el puerto de recepción se obtenga un estimado de los niveles de alguna en el tanque de almacenamiento.



**Figura 19: Sensor flotante tipo eléctrico**  
Fuente: <http://goo.gl/mGuDRi>

Ya teniendo claro cuáles son los tipos de sensores que existen como opción y conociendo su método de trabajo se dio paso a ciertas evaluaciones referentes a distintos tópicos relacionados a los requerimientos que debía tener el sub-sistema detector de nivel haciendo uso de algunas de las opciones. Para esto se realizó un cuadro comparativo entre las distintas opciones donde se presenta la condición de cada dispositivo en relación a los puntos ya mencionados; dicho cuadro es presentado a continuación.



DISEÑO DE UN SISTEMA DE AUTOMATIZACIÓN EN EL CONTROL DE CONSUMO, RACIONAMIENTO Y ANÁLISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE ALMACENAMIENTO

---

Tabla 6: Cuadro comparativo de los posibles sensores a utilizar

	<b>Opción 1:</b>  <b>Circuito Alarma nivel de agua</b>	<b>Opción 2:</b>  <b>Transductores de presión</b>	<b>Opción 3:</b>  <b>Sensor de flujo de agua</b>	<b>Opción 4:</b>  <b>Sensor flotante tipo eléctrico</b>
<b>Inversión económica</b>	Implica la compra de todos los componentes que lo conforman dando un estimado de 1500 Bs el sistema.	Implica compra en el exterior del sensor y el cable sumergible dando un estimado de 700\$ por sensor.	Implica compra en el exterior del sensor dando un estimado de 7\$ por sensor.	Implica la compra de todos los componentes que lo conforman dando un estimado de 300Bs por sensor.
<b>Eficacia</b>	Su eficacia está sujeta a una temperatura y conductancia del agua exacta y de los componentes que conforman el circuito.	Su eficacia erradica principalmente en tanques de gran tamaño y es de 0,25% de error.	Su eficacia varía en un porcentaje del 10% según la cantidad de litros que se transmitan.	Su eficacia se ve limitada por no poseer adaptabilidad para distintos tanques, es decir; solo es funcional para el tanque que son empleados
<b>Simplicidad</b>	Sus niveles de simplicidad son bajos ya que al ser un circuito eléctrico cuenta con muchos detalles que hay que tener presente.	Presenta altos niveles de simplicidad debido a la sencillez de su aplicación y el tamaño del sensor.	Presenta altos niveles de simplicidad debido a que solo consiste en su conexión a la tubería de entrada de agua.	Presenta altos niveles de simplicidad gracias a las bondades ofrecidas por el reed switch como componente electromagnético.

Fuente: Realización Propia

Luego de realizar las comparaciones que presentan cada una de nuestras opciones y de analizar cuál sería la mejor elección para la realización del sub-sistema de censo a los niveles de agua, se decidió utilizar el sensor de flujo y el sensor flotante tipo electrónico como parte del nuevo sistema a utilizar.

#### **IV.1.3 Desarrollo y pruebas del sub-sistema detector de los niveles de agua**

Partiendo de la elección realizada en el punto anterior donde luego de una comparación entre las diferentes opciones se eligió los dos dispositivos que serían parte del sub-sistema detector de niveles de agua se procedió a realizar desarrollo del mismo con sus respectivas pruebas.

Basándose en las características que presentan cada uno de los dos dispositivos que conforman parte del sistema ambos sensores tomaran un rol dentro de la estructura donde se puedan aprovechar de manera considerable las bondades que estos poseen.

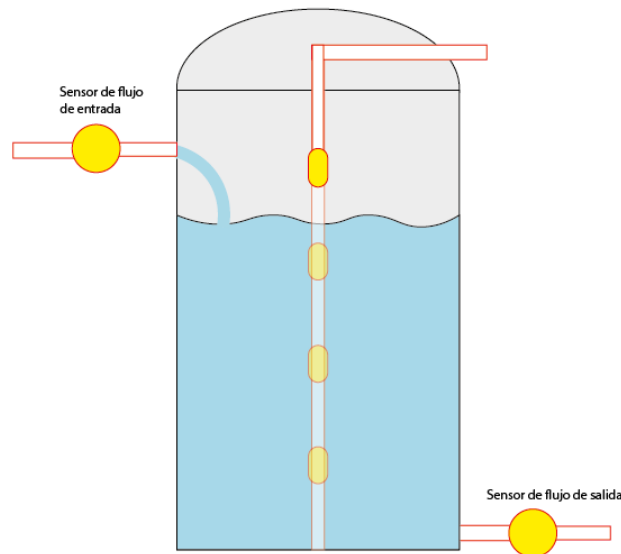
Con el fin de detectar la entrada y salida de agua se asignó el sensor de flujo como el encargado de dicha función, ubicándose en la tubería de entrada y salida de agua al tanque para que de esta manera se pueda censar el ingreso y abandono del recinto. Su instalación consistió en colocar el sensor entre la tubería de ½ pulgada para que de esta manera el agua circule de forma obligatoria a través del rotor del sensor y así obtener los datos referente a la cantidad de litros.

Los sensores de nivel en cambio ofrecen la posibilidad de poder fijar distintos niveles porcentuales de agua con respecto al tanque de almacenamiento debido a su forma de trabajar, enviando cierto porcentaje de agua disponible según la cantidad de niveles que se encuentren activos dentro del tanque.

Para el sensor flotante tipo electrónico existen 4 niveles dentro del tanque. Dado a las dimensiones del tanque los cuatro niveles estén separados a la misma distancia entre ellos y representen el 25% ,50% ,75% y 100% de la capacidad del mismo.

Para llevar a cabo el sensor flotante tipo eléctrico se utilizó una tubería de material PVC de ½ pulgada de diámetro; dentro de esta tubería esta contenido un reed switch para cada nivel (en nuestro caso 4), componente que permite el cierre de un circuito abierto al estar presente a un campo magnético como por ejemplo el de un imán. La tubería se encuentra sellada por una obstrucción intencional que impide el paso del agua hacia los reed switch y los cables en sus dos extremos para así evitar todo tipo de interferencia en la actividad del mismo lo cual fue logrado bajo la realización de pruebas que arrojaran como resultado alguna posible fisura o grieta que permitiera el paso de agua. La parte exterior de la tubería está rodeada por un material cilíndrico de características físicas que le permita flotar en agua y que en su interior tiene contenido un imán. Existe un flotador para cada reed switch y su posición en el exterior de la manguera se verá limitado en 4 secciones diferentes.

En la siguiente imagen se presenta el boceto del acabado final del sub-sistema detector de los niveles de agua.



**Figura 20: Tanque de almacenamiento.**

**Fuente: Realización Propia**

#### **IV.1.4 Comunicación utilizada con el Raspberry PI**

Una vez realizadas las pruebas para comprobar el funcionamiento tanto de los sensores de nivel, como los indicadores de entrada y salida se documentó la forma en que los nombrados anteriormente transmitían los datos recopilados al Raspberry PI. La forma de esta comunicación fue a través de los cables eléctricos de cobre, los mismos estarían conectados a los sensores tipo flotante eléctricos y a su vez a los respectivos pines del GPIO.

Los cables de cobre además también son los encargados de transmitir el voltaje que pone en funcionamiento a los sensores flotantes tipo eléctricos, el cual viene del pin de 3.3 V del GPIO del Raspberry PI.

Los cables de cobre son los encargados entonces de transmitir el voltaje transmitido a través de los sensores dependiendo del nivel del agua que estos indiquen y se diferenciarán ya que cada sensor de nivel (de los 4 sensores) estará conectado a un pin del GPIO distinto. Dicha información será recopilada a través de un programa encargado de analizar la misma y transmitirla.

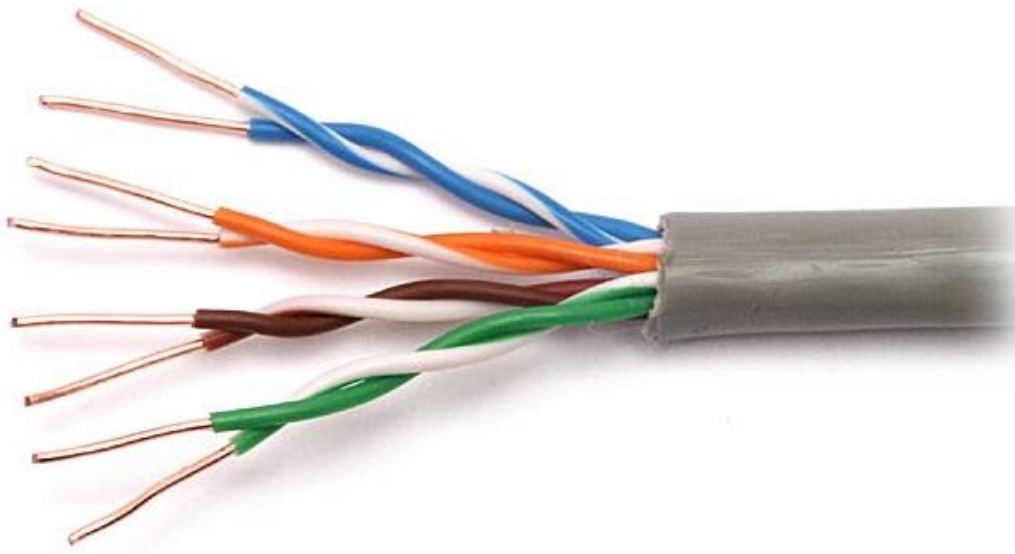


Figura 21: Tipo de cable utilizado para transmitir los datos de los sensores.

Fuente: Electronica.com

## **IV.2 Diseño del programa de procesamiento y transmisión de los datos recopilados.**

En aras de mantener la transmisión de la información de manera eficiente nace la necesidad de crear un programa que permita el procesamiento, análisis y envío de los datos suministrados por nuestro sub-sistema de medición llevándonos a definir principalmente los objetivos que tendría tal programa y los requerimientos que debía

cumplir para el desarrollo y buen funcionamiento del mismo; para esto se hizo uso de un diagrama de ideas como herramienta que permite representar ambos tópicos, el cual se presenta a continuación.



Para la realización de esta aplicación se plantea el manejo de la información que viene de los sensores de la siguiente manera:

Si el sensor detectó un cambio en el nivel del agua significativo este produce un cambio de su estado, es decir que al dispositivo de procesamiento a través de un pin de entrada programable recibe un voltaje indicando que existe una señal que se está generando a través del sensor, creando una respuesta lógica en formato digital a que identifique dicha acción con un "1" o simplemente un "0" si el mismo no detectó ningún cambio o no existe una señal de voltaje que llegue a la placa reducida. Esta

información se analiza de la siguiente forma, si ninguno de los sensores detecta cambio en el nivel de agua, el tanque no tiene suficiente agua o si en cambio todos los sensores marcan el estado “1”, respectivamente indica que el tanque se encuentra lleno.

La recopilación de los estados de los sensores también permitirá saber si existen posibles daños en ellos y la detección de entrada y salida de agua; luego esa información es almacenada en un archivo de texto para que dicha información sea enviada a la base de datos que se encuentra en el servidor local.

#### **IV.2.1 Análisis del Raspberry PI como dispositivo procesador**

Se utilizó un Raspberry PI modelo B, se decidió por este dispositivo, debido a su capacidad de procesador para el manejo de Software y gran capacidad de almacenamiento en proyectos de electrónica que incluyan una programación avanzada, por encima de otros dispositivos en el mercado tal como es el Arduino, en el caso de este proyecto el mismo abarca programación de alto nivel referente a los distintos parámetros que se tienen como objetivo para cumplir el funcionamiento del mismo, entre los distintos objetivos a cumplir se pueden nombrar los niveles de agua que pueda tener la estructura de almacenamiento, la existencia de entrada y salida de agua de este y además poder tener conocimiento referente a si el sistema tiene sensores dañados o con problemas que imposibiliten el funcionamiento correcto. Además de lo nombrado anteriormente para la decisión de usar esta placa reducida se tomó en cuenta que tiene un precio accesible en el mercado, inclusive el Raspberry Pi abarca capacidades más allá del trabajo de grado, que no fueron necesarias, pero en un futuro pudiesen ser funcionales en el caso de llevar a cabo una ampliación del proyecto.

A este dispositivo se le instaló el sistema operativo Raspbian que es un sistema operativo similar a Linux creado especialmente para dispositivos Raspberry PI, luego de familiarizarse con la placa y detallar sus características principales se programó una aplicación en lenguaje Python la cual se encarga de detectar el estado de los sensores y las otras características referentes al sistema, además de generar un archivo de texto con la información de dichos sensores.

#### **IV.2.2 Análisis y selección del Lenguaje de programación**

En esta etapa se decidió cual era el lenguaje más apropiado a utilizar en la placa inteligente, básicamente se tenían dos posibilidades: Java o Python

Java desde su nacimiento fue pensado para ser multidispositivo. Hoy Java está en muchos de los dispositivos, desde un computador, reproductores Blue-Ray, menús de TV digital, chip de celulares, entre otros. Desde el 2014 Raspbian incluye el Java Development Kit (JDK) para la versión SE, versión estándar, es decir la versión que se usa en cualquiera computador. También existe la versión Java ME que es la microedition que también se puede usar en un Raspberry PI.

Sin embargo, Python es el lenguaje de programación que es recomendado por los fundadores del Raspberry PI, ya que es un lenguaje de sintaxis sencilla y clara que puede venir bien en los temas de educación. Es un lenguaje interpretado o de script, fuertemente tipado y dinámico, es multiplataforma y es orientado a objetos. Además, es un lenguaje bastante potente y con muchas librerías que ayudan a realizar distintas actividades de programación.



Adicionalmente a esto, Raspbian (sistema operativo utilizado en la placa) ofrece más que un conjunto de programas básicos y utilidades; viene con 35000 paquetes, programas recompilados, incluido en un formato que hace más fácil la instalación en el Raspberry PI. Además, contiene herramientas de desarrollo como IDLE (*Integrated Development Environment*), para el lenguaje de programación PYTHON o SCRATCH, y diferentes ejemplos de juegos usando los módulos PYGAME.

Es por ello que tomando en cuenta los beneficiosos que trae trabajar con Python en la placa, se ha optado por esta opción, ya que simplifica el trabajo en aspectos como: la instalación (viene en el sistema operativo Raspbian), programación (sintaxis sencilla) y además es un lenguaje dinámico y multiplataforma.

Una vez que se realizó la escogencia del lenguaje de programación a trabajar y sabiendo cuales eran los requerimientos y prestaciones necesarias, para el programa de recopilación y análisis de los datos, que vienen de los sensores tipo flotantes y entrada y salida, se procedió a realizar el diagrama de flujo que indicaría el funcionamiento y proceso lógico necesario para que el mismo fuese óptimo para el sistema.





respectivo sensor está funcionando o en un estado prendido gracias al voltaje generado mencionado anteriormente.

Además de trabajar con los niveles referentes del tanque de almacenamiento en la respectiva aplicación detectora, también se trabajó en tener los conocimientos referentes al estado que se encuentran los sensores, es decir, saber si los mismos están funcionando correctamente y en el caso de que estos estén en un mal estado la información referente a esto quede guardada en un archivo de texto indicando el estado del mismo. Al aumentar los niveles de agua desde cero lógicamente el primer sensor en activarse es el que hace referencia al 25% de la capacidad del tanque, luego el segundo sería el sensor del 50% y así continua hasta llegar a 100% lo que nos hace llegar a la conclusión de que todos los sensores flotantes anteriores al que demarca el nivel actual del agua deben estar activados y de no estarlos presenta fallas.

Sabiendo esto el archivo de texto referente de los sensores de nivel pueden tener tres posibles estados tales como “ENCENDIDO”, “APAGADO” o “MALO” respectivamente para indicar el funcionamiento, después a través de otra aplicación la información es enviada a la base de datos que se encuentra en el servidor local.

#### **IV.2.4 Comunicación del Raspberry PI a Servidor Local**

Para lograr la comunicación del Raspberry Pi con nuestra PC donde se encontraba alojado nuestro servidor local, se utilizó un módulo WIFI mini siguiendo el estándar 802.11.g conectado al puerto USB del Raspberry Pi. Este dispositivo es compatible con el Raspberry Pi sin necesidad de instalar controladores adicionales.

### IV.3 Base de datos de la estructura de almacenamiento

La base de datos usada en este trabajo especial de grado es de tipo dinámica ya que las aplicaciones que se utilizaran en el sistema irán interactuando con ella y cambiando sus valores con el tiempo, fue creada en lenguaje SQL y con la ayuda de la herramienta Workbench, que permite manejar la base de datos de manera gráfica y genera automáticamente el código en el lenguaje seleccionado.

La base de datos del sistema tiene como nombre Nivel\_db y se utilizó una estructura que permite separar la información en tablas, esta base de datos contiene cuatro tablas en las cuales se almacena toda la información necesaria para el sistema, las tablas son:

- **“Sensores\_Nivel”:** En esta tabla se guardan los estados de cada uno de los sensores de nivel, contiene cinco columnas:
  - **1era Columna:** Llamada “Sensor\_Bajo”, en la cual se encuentra la información referente al 25% del nivel de capacidad del tanque.
  - **2da Columna:** Llamada “Sensor\_Medio”, en la cual se almacena la información referente al 50% del nivel de capacidad del tanque.
  - **3era Columna:** Llamada “Sensor\_MedioA”, en la cual se almacena la información referente al 75% del nivel de capacidad del tanque.

- **4ta Columna:** Llamada “Sensor\_Alto”, en la cual se almacena toda la información referente al nivel más alto del tanque o 100%.
- **5ta Columna:** Abarca un estimado muy aproximado del nivel en el que se encuentra el tanque, en esta misma se almacenara toda esa información referente. Dichos estados se guardan de la siguiente manera un “apagado” si el sensor no está marcando ese nivel de agua y un “encendido” si el sensor marca ese nivel de agua.

Esta tabla permite identificar aproximadamente los niveles de aguas disponibles en la estructura de almacenamiento, además de saber si existen sensores dañados o con fallas.

- **“Entrada\_Salida”:** En esta tabla se guarda la información de cada vez que entra agua a la estructura de almacenamiento, la misma contiene dos columnas:
  - **1era Columna:** Llamada “Sensor\_Entrada”, que la misma guarda la información referente a la entrada de agua.
  - **2da Columna:** Llamada “Sensor\_Salida”, en esta columna se guarda la información de cada vez que existe salida de agua de la estructura de almacenamiento.

La funcionalidad de la tabla es la de guardar la información que se almacena con el texto “apagado” si no existe salida o entrada de agua y un “encendido” si hay salida o entrada de agua del tanque.

Esta tabla permite saber la entrada y salida o no de agua al tanque de almacenamiento.

- **“Tanque\_Usuario”:** en esta tabla se guarda la información de cada usuario de la aplicación, contiene tres columnas:
  - **1era Columna:** Llamada “Nombre”, la cual tal y como su nombre lo indica guarda el nombre de cada usuario
  - **2da Columna:** Llamada “ID”, en la cual se almacena el número de cada usuario.
  - **3era Columna:** Llamada “pass”, en la cual se almacena la contraseña de cada usuario.

Esta tabla permite saber el nombre del usuario que esté utilizando la aplicación para conocer el estado de su tanque de almacenamiento.

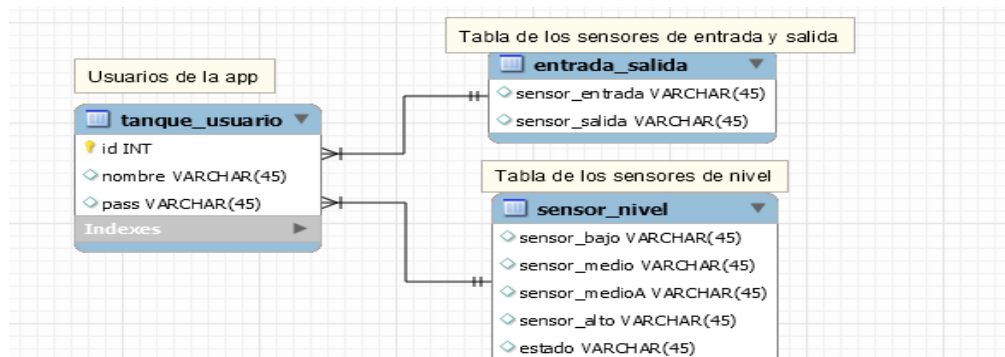


Figura 24: Diagrama EER de la herramienta Workbench para la creación de la base de datos.

Fuente: Realización Propia

### IV.3.1 XamppServer como servidor local para base de datos

Para poder establecer conexión con la base de datos se necesita un servidor web en el cual quede almacenada la misma, para ello utilizó la herramienta XamppServer la cual brinda característica de Apache y funciona como servidor local en una PC y en la que se tiene una base de datos alojada, y también trae características MySQL que permite gestionar la base de datos.

Se decidió utilizar la herramienta XamppServer por las bondades que presenta, las cuales permiten manejar de manera sencilla la base de datos (con MySQL), también funciona como servidor web (Apache) y permite la programación de script PHP.

### IV.4 Aplicación LlenarBaseDatos

LlenarBaseDatos es una aplicación que se corre en una PC donde se encuentra alojado el servidor local, esta fue desarrollada en lenguaje Python la cual se encarga de leer archivos de texto que son generados en el Raspberry PI el cual contiene la información de los estados de los sensores en cada uno de los niveles del tanque y los



respectivos sensores de entrada y salida de agua, la aplicación es la encargada de enviar dicha información a las distintas tablas de la base de datos alojada en XampServer. Esto se logró gracias a la instalación de un servidor llamado Samba en el Raspberry Pi encargado de compartir archivos entre distintos dispositivos pertenecientes a una red local o perteneciente a la misma red de transmisión.

Para el diseño y desarrollo de esta aplicación se utilizó una computadora con sistema operativo Windows 7 Professional en la que se instaló el IDE (*Integrated Development Environment*) Python 2.7, la cual es una herramienta que permite programar aplicaciones en lenguaje Python y fue la herramienta seleccionada para programar todas las aplicaciones utilizado por su fácil sintaxis y también debido a que es el lenguaje utilizado en el Raspberry Pi; además de su parecido con el lengua C++ implementado a lo largo de la carrera, por estas razones se decidió entonces continuar con este lenguaje a lo largo del proyecto.

Para establecer comunicación entre la aplicación y la base de datos, se hizo necesario importar un módulo llamado MySQLdb la que sirve para poder conectar una aplicación Python a un sistema gestor MySQL. Este módulo permite ejecutar comandos SQL, recoger el resultado de consultas SQL en un formato simple.

En el desarrollo de la aplicación se vio en la necesidad de crear diferentes funciones. Una función llamada ConexionBaseDatos() fue la que se estableció para la conexión con la base de datos como se mencionó anteriormente a través del módulo MySQLdb.

Otra de las respectivas funciones fue la llamada LeerArchivo() que es una función encargada de leer el archivo que es enviado del Raspberry PI, esto se hace

posible con la ayuda de un módulo llamado ReadLines() encargado de leer el archivo de manera completa y el mismo almacenarlo en una lista.

El desarrollo y la lógica del funcionamiento de la aplicación y sus funciones nombradas anteriormente pueden ser divididos de la siguiente manera:

1. A través del servidor Samba el archivo de texto generado en el Raspberry Pi con la información de los sensores de nivel y entrada y salida de agua es compartida con la PC que almacena la aplicación para llenar la base de datos local.
2. La aplicación Llenar base de datos es dividida en distintas funciones; la primera función es la de leer el respectivo archivo (LeerArchivo() ) de texto generado por el Raspberry PI y guardar cada línea del archivo en una lista que se almacena en la aplicación, además de esto también se lee un archivo de texto creado en esta aplicación para los valores anteriormente almacenados y se van a guardar en una segunda lista.
3. Se corre la siguiente función que es la comunicación con la Base de datos (ConexionBaseDatos()); si se establece una conexión correcta la misma manda una respuesta de comunicación exitosa y se utiliza un puntero para poder moverse a través de la base de datos; si la base de datos se encuentra vacía de manera directa se guarda la información de la lista referente a la función de leer el archivo de texto y guarda esta lista en un segundo archivo de texto, de lo contrario se va a la siguiente función.
4. Después de establecer la conexión se corre la función principal y se va a hacer una comparación entre las dos listas existentes a través de la

función ReadLines() las cuales son la lista en la que se guarda la información referente al Raspberry Pi y la lista que guarda la información del archivo de texto generado en esta aplicación para guardar los valores anteriormente utilizados o almacenados. De ser iguales estos no existe ningún cambio en la base de datos, de ser distintos automáticamente la información completa de la lista es guardada en la tabla referente, los valores de esta lista son colocados intencionalmente para que cada valor de ella pertenezca a una columna distinta de la tabla perteneciente a la base de datos y después de almacenar la información la misma también es guardado en el archivo de texto referente a los valores anterior.

Esta función también cuenta con un módulo del tipo Timer el cual permite realizar la acción de revisar el archivo de texto que pueda tener nuevos valores del sistema y almacenarlos en la base de datos cada cierto tiempo que en este caso se escogió como tiempo conveniente 10 segundos.

Con esto la base de datos se actualizara automáticamente, solo si tiene nuevos valores el archivo de texto, en un periodo de cada 10 segundos con los nuevos estados de los sensores y se podrá acceder a esos nuevos valores.

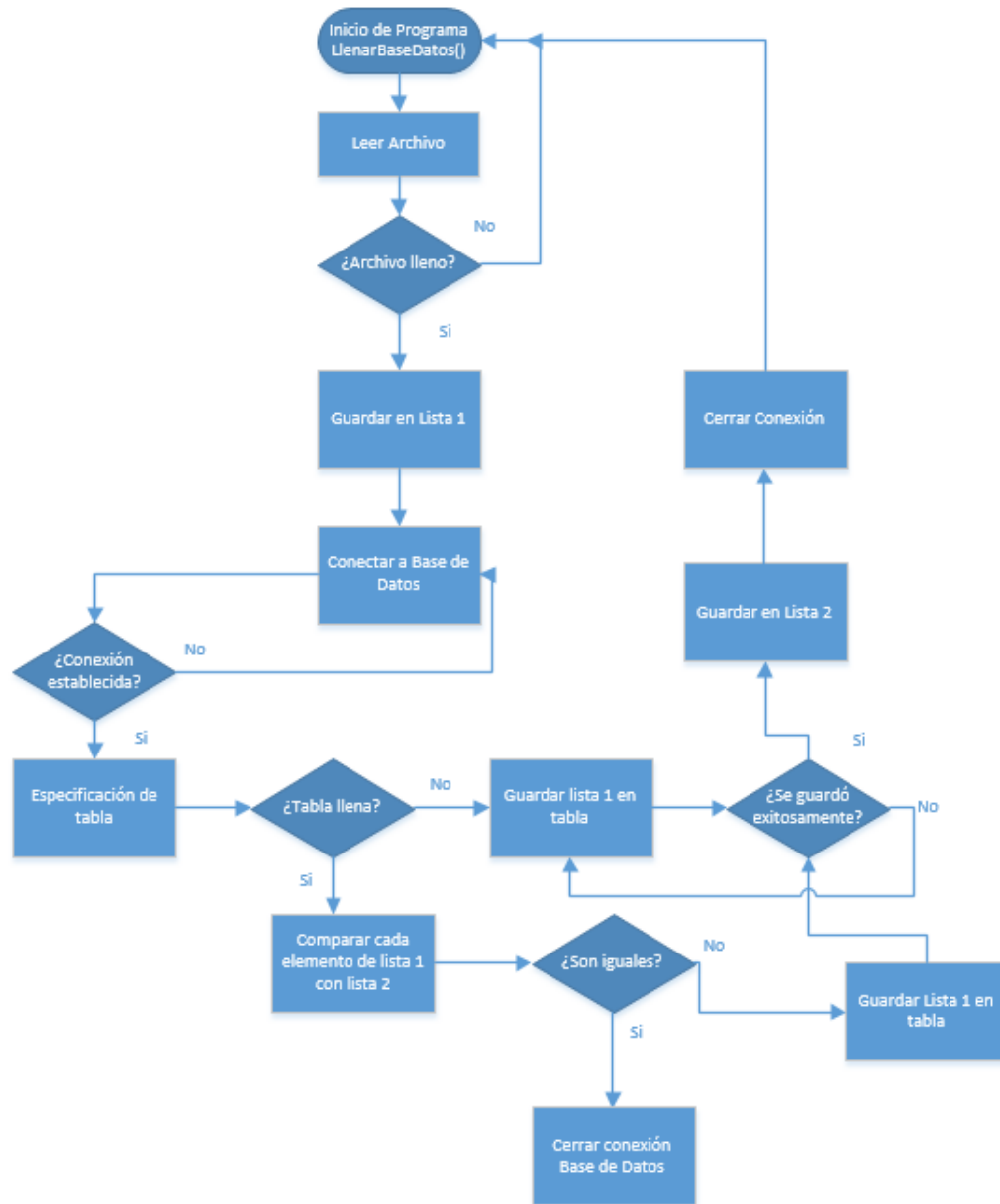


Figura 25: Diagrama de flujo de la aplicación LlenarBaseDatos.

Fuente: Realización Propia

#### **IV.4.1 Comunicación de la Aplicación LlenarBaseDatos con MySQL**

La comunicación entre la aplicación encargada de leer la información que se recibe del Raspberry Pi es a través de módulos de conexión de Python los cuales establecen la conexión a través del puerto 80 que es el utilizado por el Servidor local y es la forma en que se establece la conexión. Además de utilizar librerías Python que permiten establecer dicha comunicación directamente sin necesidad de protocolos externos tal como puede ser el HttpConection, entre dichas librerías se encuentra la librería SQLdb especializada para modificar y gestionar bases de datos MySQL con el lenguaje de programación Python sin necesidad de entornos de desarrollo externos.

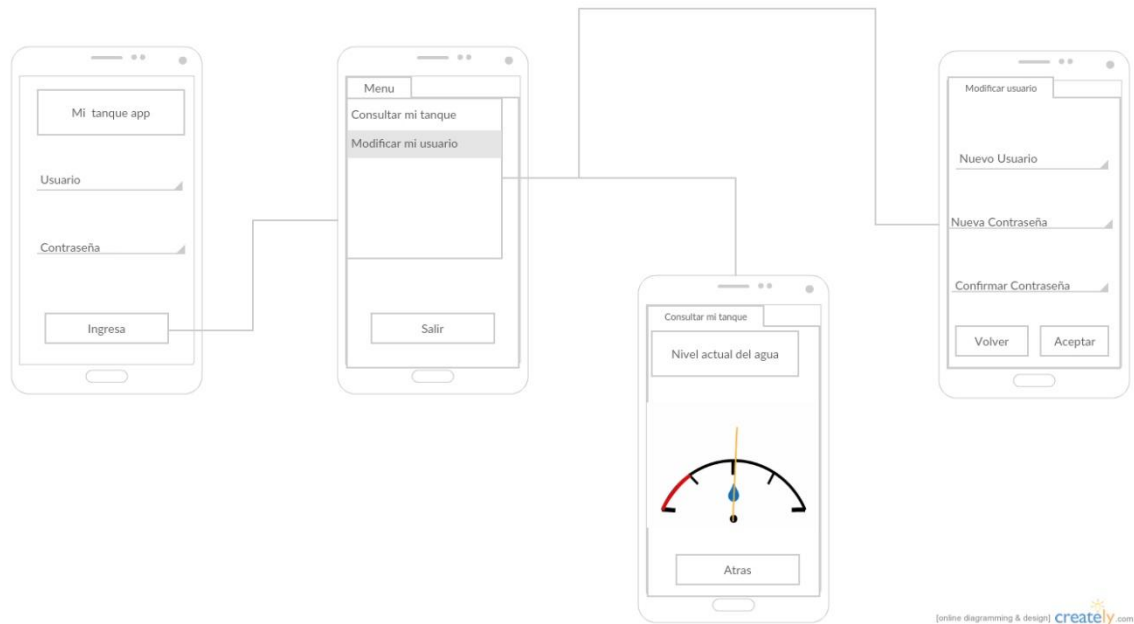
#### **IV.5 Aplicación Android Mi Tanque**

A continuación se explica a detalle todo lo que se realizó para llevar a cabo el funcionamiento de nuestra aplicación móvil.

El objetivo de la aplicación Android en el trabajo de grado es poder facilitar al usuario la información referente al tanque de agua que desea monitorear desde su teléfono celular. Dicha aplicación permitirá al usuario saber el nivel porcentual de agua existente en el tanque como el ingreso o no agua al mismo mediante el inicio de sesión con su nombre de usuario y contraseña particular; además tendrá la oportunidad de poder modificar los valores de usuario y contraseña registrados en la base de datos. Basándonos en el razonamiento explicado anteriormente el siguiente esquema busca plasmar una idea de cómo será la aplicación a primera instancia.

## DISEÑO DE UN SISTEMA DE AUTOMATIZACIÓN EN EL CONTROL DE CONSUMO, RACIONAMIENTO Y ANÁLISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE ALMACENAMIENTO

---



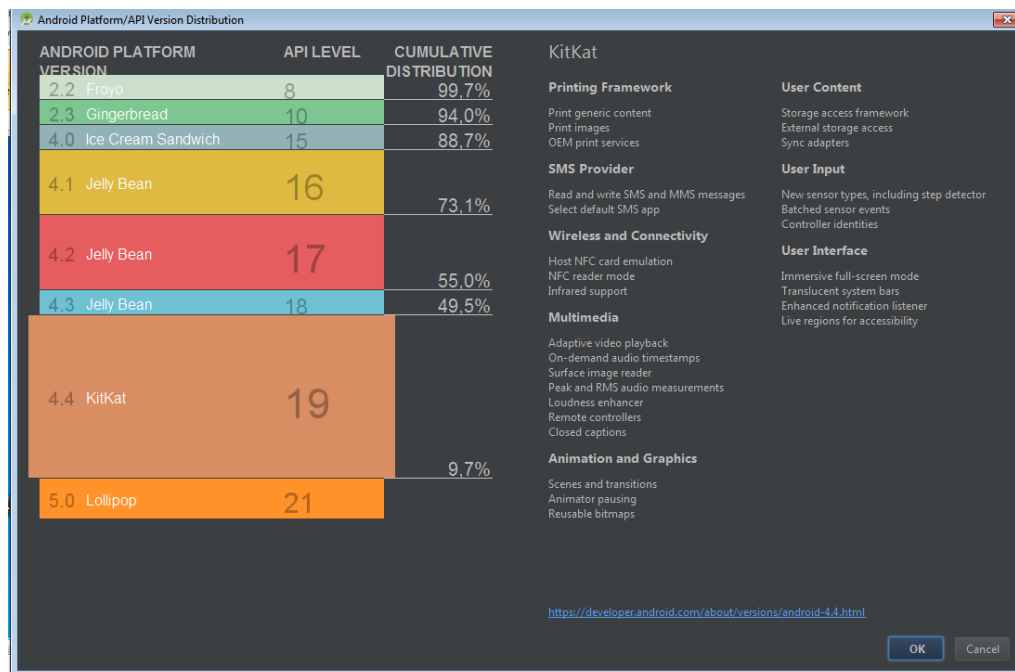
**Figura 26: Diseño de la aplicación Android.**

**Fuente: Realización Propia**

Para el diseño y desarrollo de la aplicación se utilizó un computador con sistema operativo Windows 7 *professional*, en el que se instaló el IDE (*Integrated Development Environment*) Android Studio, obtenido en la página oficial de Android [www.developer.android.com](http://www.developer.android.com) y Java JDK, obtenido en la página web oficial de Oracle [www.oracle.com](http://www.oracle.com). Estas herramientas permite programar aplicaciones en lenguaje Android de manera dinámica y activarle todos los permisos necesarios para la misma, también es el IDE de programación recomendado en cuanto al desarrollo de aplicaciones Android.

La versión Android en la que se llevó a cabo la aplicación fue la 4.4, mejor conocida como KitKat, ya que actualmente está representa el 39,8% de usuarios Android a nivel mundial. En el siguiente cuadro se ve la distribución acumulativa de

usuarios según la versión utilizada y las características generales de la versión Kit Kat.



**Figura 27: Porcentaje de personas usando esta versión de Android.**

**Fuente: Android Studio**

Para lograr que la aplicación tuviera una correcta funcionalidad se vio en la necesidad de crear diferentes ventanas o “activities” las cuales están compuestas por la parte grafica que será lo mostrado al usuario a través de una ventana utilizando en archivos .xml que genera etiquetas similares a las de HTML y la parte lógica que permite manipular e interactuar todos aquellos elemento gráficos y poder colocar el código de la actividad bajo un archivo tipo.java.

### **IV.5.1 Clases creadas en la aplicación Mi tanque App**

A continuación se presentan las clases particulares creadas y las pertenecientes a los “activities” utilizados en el desarrollo de la aplicación.

#### **IV.5.1.1 loginSync.java**

Esta clase es la encargada de llamar a las aplicaciones php por medio del protocolo http y con la ayuda de la clase *HttpEntity* (Entidad http) la cual fue importada en nuestra aplicación.

Dicha clase cuenta con una función de tipo String llamada link y recibe como parámetro la url de los archivos php, con la utilización de objetos de tipos: URL y *HttpConnection* envía una petición de tipo GET al script programado en PHP los cuales se comunican con la base de datos y extraen la información referente al usuario que intenta iniciar sesión.

#### **IV.5.1.2 login.java**

Esta es la clase principal de nuestra aplicación esta entrelazada con la ventana que ve el usuario al momento de abrir o ejecutar la aplicación.

La función de esta clase es permitir la conexión del usuario a la base de datos del sistema mediante la introducción de su nombre de usuario y contraseña; una vez



que el usuario introduce sus datos los mismos son enviados al loginSync.java para ser certificados oprimiendo el botón “ingresar” para dar paso a una nueva ventana.

#### **IV.5.1.3 PantallaUsuario.java**

Esta clase será el resultado de coincidir los datos suministrados por el usuario con la base de datos en el Login.java, dándole al usuario la bienvenida y ofreciéndole la posibilidad de poder cambiar sus datos de ingreso o consultar la información de su tanque.

#### **IV.5.1.4 ConsultaUsuarioSync.java**

Se accederá a esta clase de manera oculta al usuario luego de presionar la opción de consultar la información del tanque en la Activity PantallaUsuario.java. Es la encargada de llamar a las aplicaciones php por medio del protocolo http y con la ayuda de la clase *HttpEntity* (Entidad http) la cual fue importada en nuestra aplicación.

Esta clase recibirá bajo el formato Json los datos vinculados a la última actualización que presenta el tanque y el estado de los sensores para luego hacer uso de estos.

#### **IV.5.1.5ConsultaUsuario.java**

Esta clase recibirá la información recopilada en la clase ConsultaUsuarioSync.java y dependiendo de los resultados obtenidos se procederá a hacer la muestra grafica de los niveles existentes y de la posible falla de sensores.

En conjunto a esta actividad también serán consultados los valores de los sensores como también la entrada y salida de agua para que de una manera gráfica sean mostrados al usuario.

#### **IV.5.1.6 ModificarUsuarioSync.java**

Esta clase al igual que loginSync.java es la encargada de llamar a las aplicaciones php por medio del protocolo http y con la ayuda de la clase *HttpEntity* (Entidad http) la cual fue importada en nuestra aplicación.

Dicha clase cuenta con una función de tipo String llamada link y recibe como parámetro la url de los archivos php, con la utilización de objetos de tipos: URL y HttpURLConnection envía una petición de tipo GET al script programado en PHP los cuales se comunican con la base de datos y modifican la información referente al usuario que intenta iniciar sesión.

#### **IV.5.1.7 ModificarUsuario.java**

Esta clase será visualizada al presionar la opción de cambiar los datos de ingreso del usuario en la Activity PantallaUsuario.java. Se hará la petición del nuevo nombre y la nueva contraseña la cual tendrá que ser escrita en dos campos diferentes para así poder realizar la confirmación de dicho dato. El programa permitirá poder utilizar el mismo nombre de usuario en el caso de solo querer cambiar la contraseña y viceversa.

Ya obtenidos los nuevos datos a actualizar los mismos son enviados al loginSync.java para ser certificados oprimiendo el botón “aceptar” para dar paso a una nueva ventana.

#### **IV.5.1.8 Refresh.java**

Esta clase se mantendrá de manera oculta al usuario, la cual se encargara de actualizar los datos provenientes de la base de datos con el fin de que a través de un Broadcast receiver se pueda generar una alarma en el caso de que existan fallas en el sistema o se necesite enunciar de manera inmediata algo al cliente.

#### **IV.5.2 Layouts utilizados en el desarrollo de la aplicación**

A continuación se presentan los Layouts pertenecientes a los “activities” utilizados en el desarrollo de la aplicación.

##### **IV.5.2.1 Activity \_login.xml**

Es el perteneciente a la Activity Login que contiene la clase login.java. Consiste en la primera pantalla que tendrá como objetivo permitir al usuario iniciar sesión con sus datos.

Para la creación de este diseño se hizo uso de las herramientas graficas suministradas por el IDE Android Studio de las cuales fueron usadas dos EditText; entiéndase como una herramienta que permite el ingreso de datos desde el teclado y un botón que al ser presionado permite el envío de la información al siguiente paso de procesamiento de la misma.

#### **IV.5.2.2 Activity\_Usuario\_Pantalla.xml**

Es el perteneciente a la Activity PantallaUsuario que contiene la clase PantallaUsuario.java. Consiste en la pantalla mostrada al usuario luego de un inicio de sesión exitoso en el cual se presentaran las opciones de modificar sus datos de acceso, consultar el estado del tanque o salir.

Para la creación de este diseño se hizo uso de las herramientas graficas suministradas por el IDE Android Studio de la cuales fue usada un ListView; entiéndase como una herramienta que permite mostrar en una estructura tipo lista las opciones anteriormente mencionadas permitiendo que cada una de ellas al ser presionadas tuvieran su respectiva respuesta al usuario.

#### **IV.5.2.3 Activity\_Usuario\_Consulta.xml**

Es el perteneciente a la Activity ConsultaUsuario que contiene la clase ConsultaUsuario.java. Consiste en la pantalla mostrada al usuario luego de haber elegido la opción de consultar el estado del tanque mostrándole el nivel existente en el tanque registrado en la última actualización como además la posible entrada y salida de agua.

Para la creación de este diseño se hizo uso de las herramientas graficas suministradas por el IDE Android Studio de las cuales fue usado un PlainText; entiéndase como una herramienta que permite mostrar en pantalla un texto simple que en este caso llevaba por contenido la frase “Estado Actual del tanque”, un ListView

que permitía mostrar la información vinculada al estado de los sensores además de la entrada y salida de agua.

Para dar muestra de los niveles se hizo uso de la herramienta Illustrator que permitió la creación de una serie de imágenes que representaban los distintos niveles que podía censar el sistema de tal forma que sea mostrada la indicada según la cantidad de agua existente.

#### **IV5.2.4 Activity\_Usuario\_Modificar.xml**

Es el perteneciente a la Activity ModificarUsuario que contiene la clase ModificarUsuario.java. Consiste en la pantalla mostrada al usuario luego de haber elegido la opción de modificar sus datos de ingreso y tiene por fin permitir el ingreso de los nuevos datos suministrados por el cliente.

Para la creación de este diseño se hizo uso de las herramientas graficas suministradas por el IDE Android Studio de las cuales fue usado tres EditText que permitían el ingreso del nuevo usuario y contraseña desde el teclado y un botón que para el envío de los datos a su procesamiento.

#### **IV.5.3Servicios PHP**

Durante el diseño y desarrollo del sistema existe la necesidad de comunicar la aplicación móvil programada en lenguaje Android con nuestro servidor local en el cual se encuentra alojada nuestra base de datos; para poder lograr que esta comunicación se efectué es necesario utilizar algunos script programados en php los

cuales se encargaran de conectarse con la base de datos y dependiendo la acción que se desee realizar podrán consultar o modificar los datos que en ella se encuentran.

Para la comunicación y todas las funciones que se quiere que la aplicación pueda realizar fue necesario crear varios script php los cuales tendrán diferentes funciones, los script utilizados son los siguientes:

#### **IV.5.3.1 Config.php**

Este script es el encargado de suministrar a los demás scripts los datos pertenecientes al servidor local y la base de datos; los cuales son:

Nombre del usuario del servidor local: “Administrador”

- Nombre del servidor local : “Local Host”
- Contraseña del usuario del servidor local : “prueba123 ”
- Nombre de la Base de Datos: “Nivel\_db”

#### **IV.5.3.2 login.php**

Este script php es el que se encarga de recibir los datos del inicio de sesión del usuario bajo el método GET y compararlo con los que se encuentran en la base de datos en la tabla “Tanque\_Usuario”, devolviendo el id del cliente si los datos suministrados coinciden con los guardados.

#### **IV.5.3.3 consulta.php**

Este script permite conocer el estado actualizado de los niveles porcentuales del tanque mediante la consulta a la tabla “Sensores\_Nivel”, devolviendo el estado del tanque y los sensores en formato Json para su manejo directo en Android Studio.

#### **IV.5.3.4 consultaIO.php**

Este script permite conocer el ingreso o salida del agua en el tanque mediante la consulta a la tabla “Entrada\_Salida”, devolviendo el estado “encendido” o “apagado” según sea el caso en formato Json para su manejo directo en Android Studio.

#### **IV.5.3.5 ModificarUsuario.php**

Este script recibe dos parámetros; uno es el nombre del usuario y otro es la contraseña del mismo; luego utilizando el comando INSERT INTO se envían dichos parámetros a la base de datos donde está la información de los usuarios. Si el nombre de usuario existe lo modifica en la tabla “Tanque\_Usuario” y si está registrado envía un mensaje informándolo.

#### **IV.5.4 Permisos de la aplicación**

Para que nuestra aplicación pudiera tener un correcto funcionamiento y se pueda conectar con nuestra base de datos, utilizar los servicios php y conectarse al servidor es necesario activar el permiso de internet en el AndroidManifest; el permiso

de internet se activa colocando la siguiente línea al principio del AndroidManifest.xml de nuestro proyecto: “<uses-permission android:name=“android.permission.INTERNET”/>”.

#### **IV.6.1 Pruebas en entornos simulados**

Una vez estuvieron realizadas todas las aplicaciones se procedió a realizar algunas pruebas simuladas las cuales describiremos detalladamente a continuación.

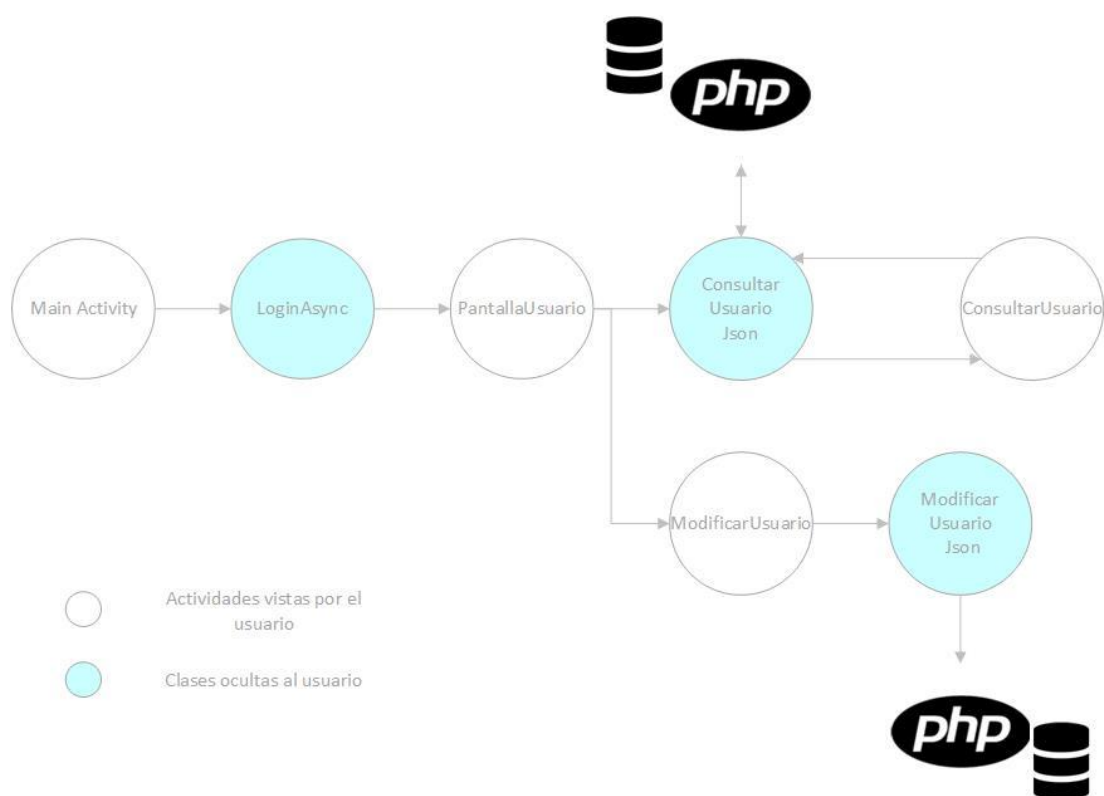
La primera prueba realizada consistió en la conexión del sistema partiendo desde el sub-sistema detector de nivel de agua a los pines correspondientes en el Raspberry PI y de ahí las funciones realizadas en Python que permitían la conexión a la base de datos; todo esto sin hacer uso del tanque donde estarían colocados los sensores en las pruebas del entorno real. Durante estas pruebas de manera manual eran movidos los sensores simulando la paulatina entrada ascendiente de agua con la intención de generar cambios en los datos guardados en la base de datos y de corroborar la actualización de la información cada 10 segundos. Además también se simularon los posibles errores en los sensores y todos los posibles estados de nivel que abarca el sistema.

La segunda prueba realizada consistía en agregar al sistema los scripts realizados en PHP y la aplicación Android conectado mediante un servidor local a través de una conexión inalámbrica WLAN. Con el fin de lograr el óptimo funcionamiento del proyecto se realizaron las mismas pruebas de manera manual dando como resultados ciertos errores que nos llevaron a las modificaciones en los scripts en PHP y en las funciones Python para lograr la transmisión de los datos de manera eficiente .A su vez la aplicación Android presento ciertos inconvenientes



vinculados con el archivo que permite la compilación de la misma llamado Build Gradle, ya que habían ciertas diferencias con respecto al API con el que se estaba trabajando; estos inconvenientes fueron resueltos consultando la página oficial de Android dedicada a las dudas e inconvenientes suscitados a los desarrolladores.

El diagrama de flujo del funcionamiento de la aplicación se encuentra representado en la figura mostrada a continuación.



**Figura 28: Diagrama de flujo del funcionamiento de la aplicación Android.**

**Fuente: Realización Propia**

#### **IV.6.2 Pruebas en entorno real**

Una vez hechas todas las pruebas simuladas y realizados todos los cambios necesarios para su buen funcionamiento se dio a cabo el montaje del sistema en un tanque con medidas específicas para los sensores de nivel y se dio inicio a la prueba de las mismas obteniendo como resultados la respuesta esperada por cada parte del sistema; el cual incluía el Raspberry PI, las funciones en Python, la base de datos, los scripts PHP y la aplicación Android.



## **CAPITULO V**

### **Resultados**

En el siguiente capítulo se pretende describir detalladamente cada uno de los resultados obtenidos luego de haber realizado cada uno de los procedimientos que se describieron en el capítulo anterior (Desarrollo) y se analizaran todos y cada uno de esos resultados.

A grandes rasgos se logró obtener como resultado final un sistema que permite a los usuarios obtener un estado aproximado de la cantidad de agua que tiene su estructura de almacenamiento, además de saber exactamente el estado de los sensores y de cuando existe entrada o salida de agua respectivamente.

#### **V.1 Sub-sistema detector de los niveles de agua**

Al haber realizado varias pruebas con los distintos sensores, se logró tener como resultado que el sensor flotante tipo eléctrico tiene un buen alcance para la medición del agua, sin embargo no se obtiene una medición exactas de los mismos, solo se pueden establecer mediciones aproximadas y por esto mismo se decidió que las distancias entre los 4 sensores sea de un aproximada entre cada uno, en cambio para los sensores de entrada y salida al ser solo indicadores de agua se adaptan perfectamente al sistema sin errores de proximidad.

Para la construcción del sub-sistema detector de los niveles de agua se utilizó un tubo Pavco de  $\frac{1}{2}$  y los respectivos Reed switchs nombrados anteriormente fueron colocados dentro del mismo a una distancia prudencial para evitar errores muy grandes de proximidad. El tubo tiene una altura de 40 cms ideal para el tanque de

almacenamiento de prueba porque este tendría esta altura exacta. Además se le adhirió al tubo de  $\frac{1}{2}$  flotadores de 1 pulgada para que los mismos puedan subir a través del tubo sin dificultades cuando el tanque de almacenamiento se llene, estos flotadores por dentro contienen imán, material que genera el campo magnético para activar el respectivo voltaje que se envía a través del Raspberry Pi.



**Figura 29: Sensor de nivel tipo flotante eléctrico.**

**Fuente: Realización Propia**

Los resultados entonces fueron los esperados para el sub-sistema de sensores ya que a pesar de que no tiene una exactitud en la medida cumple con el requisito de enviar la información referente del estado del tanque de almacenamiento al Raspberry Pi y de las indicaciones de entrada y salida de agua respectivamente.

A continuación se presenta el diagrama circuital de cómo es conectado los sensores flotantes tipo electricos con los pines del GPIO del Raspberry PI para el correcto funcionamiento del sistema.

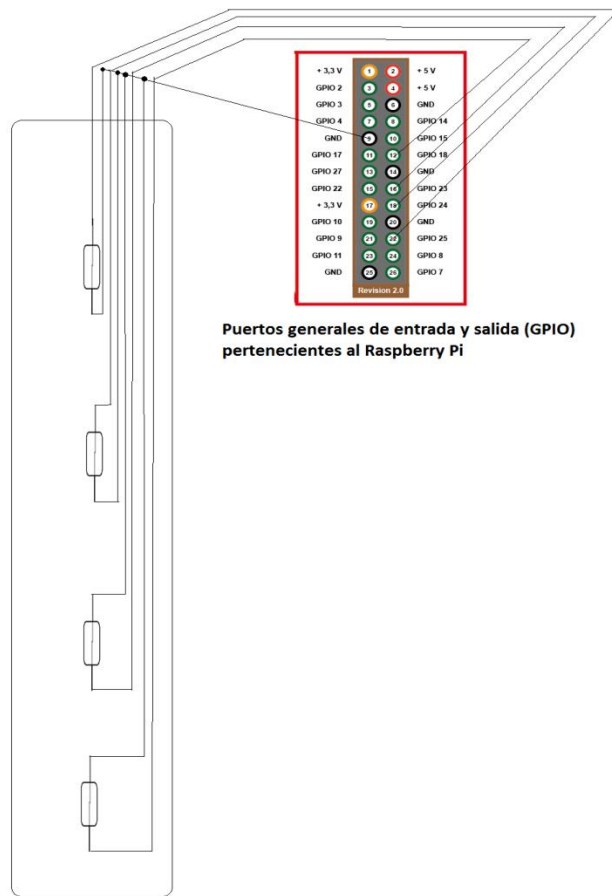


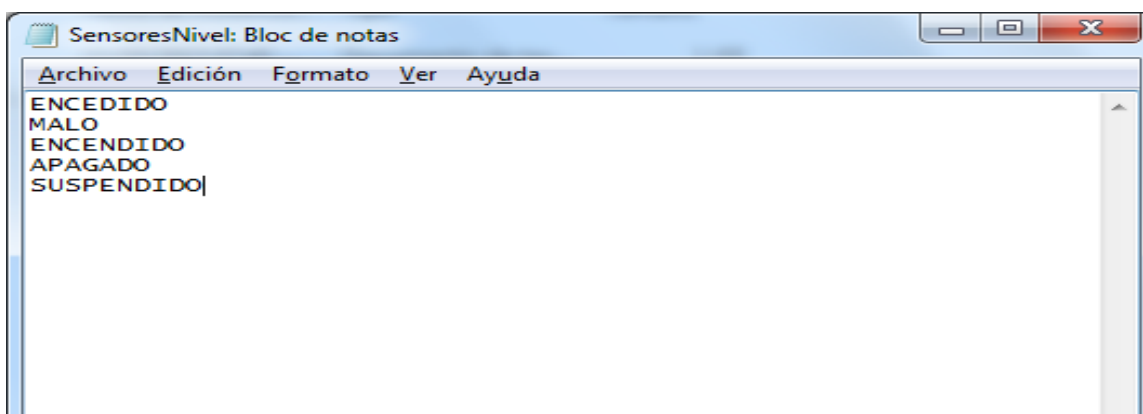
Figura 30: Diagrama Circuital de los Sensores con el Raspberry PI.

Fuente: Realización Propia

## V.2 Programa de procesamiento y transmisión de los datos recopilados en el Raspberry Pi

Con el desarrollo de la aplicación detectora en el Raspberry implementada en lenguaje Python, se logra observar el estado de cada uno de los sensores conectados al dispositivo y de acuerdo a esos estados genera dos archivos de texto con la información de los sensores, respectivamente el archivo de texto llamado

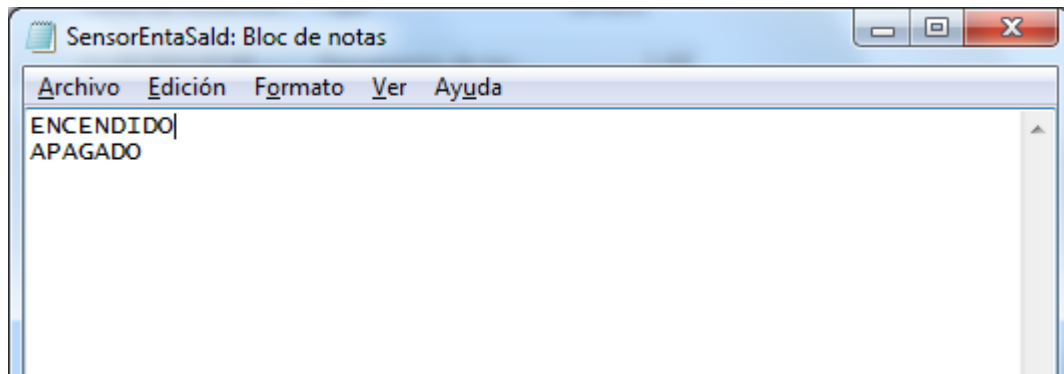
“SensoresNivel.txt” es el archivo que almacena los distintos estados de los sensores de nivel, el cual puede abarcar 5 posibles líneas las cuales cada una representa un sensor de nivel y una quinta línea indicando un estado general del tanque, además pueden aparecer tres formas distintas como puede ser “ENCENDIDO”, “APAGADO”, “MALO”; estas tres palabras representan los posibles estados que puede tener el sensor en ese momento y es lo que leerá la aplicación LlenarBaseDatos para luego insertar los estados en la base de datos.



**Figura 31:** Archivo de texto de sensores de nivel.

**Fuente:** Realización Propia

El segundo archivo de texto es llamado “EntradaSalida.txt” que es el archivo encargado de almacenar los estados de la entrada y salida del agua en el tanque de almacenamiento. En este a diferencial del archivo anterior solo pueden ser llenado dos posibles líneas, cada uno indicando un sensor distinto, la primera línea para el sensor de entrada y la segunda línea para el sensor de salida, en este caso solo pueden ser escritas dos posibles palabras tal como puede ser “ENCENDIDO” y “APAGADO” representando los dos posibles estados que pueden tener ambos sensores. Además esta será la información que leerá la respectiva aplicación encargada de enviar la información a la base de datos.



**Figura 32:** Archivo de texto referente a los sensores de entrada y salida.

**Fuente:** Realización Propia

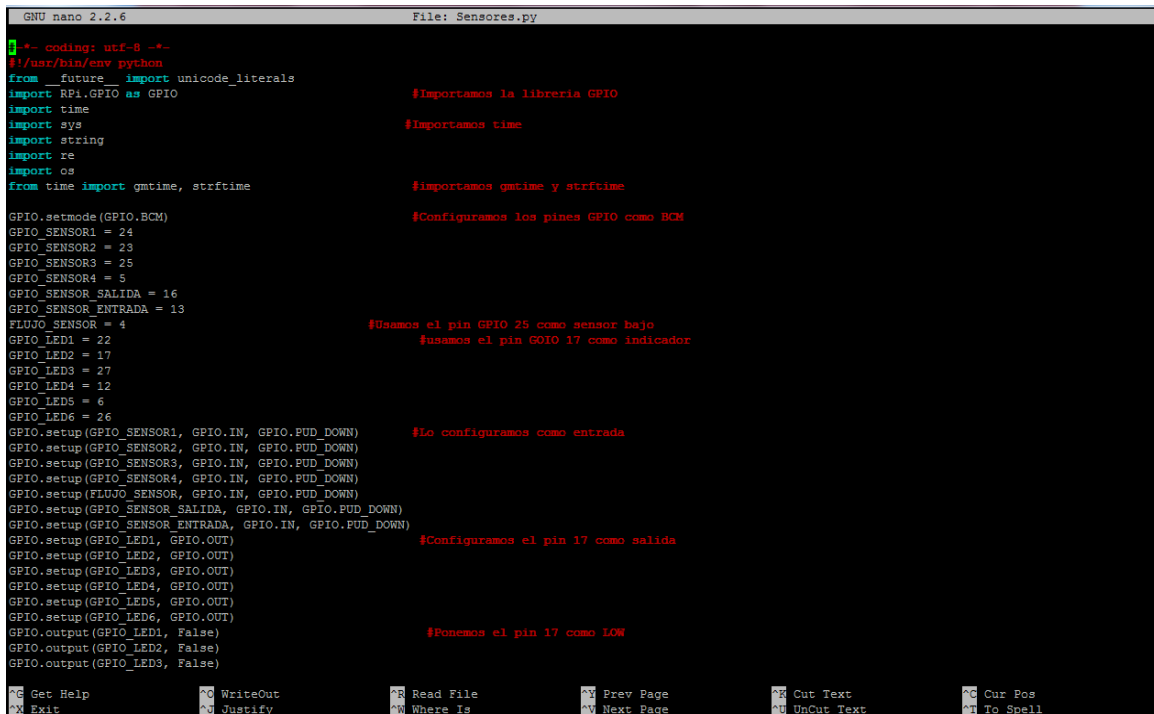
La información de estos archivos de texto es leída por la aplicación “LlenarBaseDatos.py” y esta es la encargada de enviar la información a la base de datos.

En un principio en este programa los resultados de los sensores no fue el esperado ya que los mismos arrojaban resultados inversos, es decir, cuando debía marcar “1” marcaba “0”, esto fue debido a que en el Raspberry PI por defecto las resistencias de la configuración se encuentran en estado Pop-up o en estado alto, por lo que se tuvo que colocar los pines del GPIO usados como entrada en estado Pop-Down. Luego de cambiar esta configuración en cada entrada los resultados fueron los esperados, tal como se observa en la figura 23. Entonces cuando el sensor flotante tipo eléctrico mandaba un voltaje o un estado “uno” entonces se registraba como entrada de dato en el Raspberry PI.



## DISEÑO DE UN SISTEMA DE AUTOMATIZACIÓN EN EL CONTROL DE CONSUMO, RACIONAMIENTO Y ANÁLISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE ALMACENAMIENTO

---

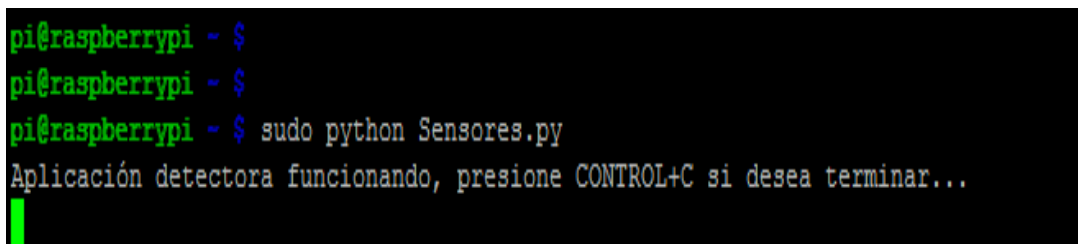


```
GNU nano 2.2.6 File: Sensores.py
#!/usr/bin/env python
from __future__ import unicode_literals
import RPi.GPIO as GPIO
import time
import sys
import string
import re
import os
from time import gmtime, strftime

GPIO.setmode(GPIO.BCM)
GPIO_SENSOR1 = 24
GPIO_SENSOR2 = 23
GPIO_SENSOR3 = 25
GPIO_SENSOR4 = 5
GPIO_SENSOR_SALIDA = 16
GPIO_SENSOR_ENTRADA = 13
FLUJO_SENSOR = 4
GPIO_LED1 = 22
GPIO_LED2 = 17
GPIO_LED3 = 27
GPIO_LED4 = 12
GPIO_LED5 = 6
GPIO_LED6 = 26
GPIO.setup(GPIO_SENSOR1, GPIO.IN, GPIO.PUD_DOWN)
GPIO.setup(GPIO_SENSOR2, GPIO.IN, GPIO.PUD_DOWN)
GPIO.setup(GPIO_SENSOR3, GPIO.IN, GPIO.PUD_DOWN)
GPIO.setup(GPIO_SENSOR4, GPIO.IN, GPIO.PUD_DOWN)
GPIO.setup(FLUJO_SENSOR, GPIO.IN, GPIO.PUD_DOWN)
GPIO.setup(GPIO_SENSOR_SALIDA, GPIO.IN, GPIO.PUD_DOWN)
GPIO.setup(GPIO_SENSOR_ENTRADA, GPIO.IN, GPIO.PUD_DOWN)
GPIO.setup(GPIO_LED1, GPIO.OUT)
GPIO.setup(GPIO_LED2, GPIO.OUT)
GPIO.setup(GPIO_LED3, GPIO.OUT)
GPIO.setup(GPIO_LED4, GPIO.OUT)
GPIO.setup(GPIO_LED5, GPIO.OUT)
GPIO.setup(GPIO_LED6, GPIO.OUT)
GPIO.output(GPIO_LED1, False)
GPIO.output(GPIO_LED2, False)
GPIO.output(GPIO_LED3, False)
```

Figura 33: Captura de pantalla de las distintas configuraciones del GPIO.  
Fuente: Realización Propia

En una vista general los resultados de este programa fueron los esperados ya que transmite y analiza los estados de los sensores de nivel y los sensores de entrada y salida adecuadamente.



```
pi@raspberrypi ~ $
pi@raspberrypi ~ $
pi@raspberrypi ~ $ sudo python Sensores.py
Aplicación detectora funcionando, presione CONTROL+C si desea terminar...
^C
```

Figura 34: Captura de pantalla de la vista de la aplicación desde el punto de vista de un usuario.  
Fuente: Realización Propia

### **V.3 Estructura de la base de datos**

A continuación se describen detalladamente las tablas que conforman la estructura de la base de datos llamada Nivel\_db.

La base de datos (Nivel\_db) cuenta con cuatro tablas en las cuales se almacenan la información necesaria para el correcto funcionamiento del sistema y las mismas son explicadas a continuación.

#### **V.3.1 Tabla Sensores\_Nivel**

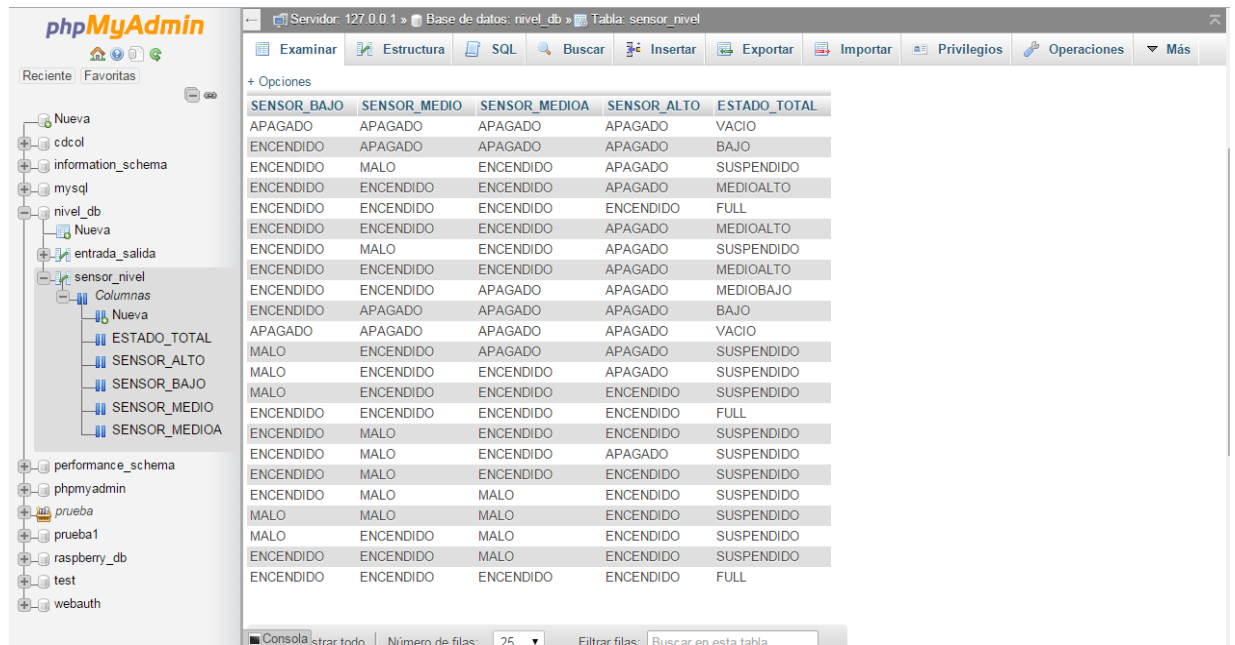
En esta tabla se guarda información referente a los distintos niveles del tanque de almacenamiento de agua, los cuales son representados a través de 4 sensores que se encuentran distribuidos de manera equitativa a lo largo de la estructura. La tabla se encuentra conformada por cinco columnas las cuales son:

- **Sensor\_Bajo:** en esta columna se guarda los dos posibles estados que puede tener el sensor, ubicado en la parte más baja del tanque. Los estados pueden ser respectivamente “apagado” o “encendido” dependiendo si el reed switch está cerrando el respectivo circuito o no.
- **Sensor\_Medio:** en esta columna se guarda los dos posibles estados que puede tener el sensor, ubicado en la parte entre el medio y la parte de abajo del tanque. Los estados pueden ser respectivamente “apagado” o “encendido” dependiendo si el reed switch está cerrando el respectivo circuito o no.

- Sensor\_MedioA: en esta columna se guarda los dos posibles estados que puede tener el sensor, ubicado en la parte entre el medio y la parte alta del tanque. Los estados pueden ser respectivamente “apagado” o “encendido” dependiendo si el reed switch está cerrando el respectivo circuito o no.
- Sensor\_Alto: en esta columna se guarda los dos posibles estados que puede tener el sensor, ubicado en la parte más alta del tanque. Los estados pueden ser respectivamente “apagado” o “encendido” dependiendo si el reed switch está cerrando el respectivo circuito o no.
- Estado: en esta columna se almacena la información referente a un estado aproximado del tanque, la información depende del estado de los sensores de nivel mencionados anteriormente, teniendo en cuenta que esta columna es para un aproximado de agua en el tanque y no una cantidad exacta en litros, es una referencia para saber en qué punto se encuentran los niveles de agua. Esta información se almacena siempre y cuando los sensores estén en un estado activado.

En la siguiente figura se detalla un poco más los resultados de varias pruebas que se realizaron para poder determinar con exactitud si la tabla respectiva estaba siendo llenada de forma adecuada. En líneas generales fueron los resultados esperados ya que la información de cada sensor es guardada de manera correcta y además se observa cada uno de los posibles parámetros que podía tomar el sensor agregándose de forma adecuada a la tabla.

## DISEÑO DE UN SISTEMA DE AUTOMATIZACIÓN EN EL CONTROL DE CONSUMO, RACIONAMIENTO Y ANÁLISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE ALMACENAMIENTO



SENSOR_BAJO	SENSOR_MEDIO	SENSOR_MEDIOA	SENSOR_ALTO	ESTADO_TOTAL
APAGADO	APAGADO	APAGADO	APAGADO	VACIO
ENCENDIDO	APAGADO	APAGADO	APAGADO	BAJO
ENCENDIDO	MALO	ENCENDIDO	APAGADO	SUSPENDIDO
ENCENDIDO	ENCENDIDO	ENCENDIDO	APAGADO	MEDIOALTO
ENCENDIDO	ENCENDIDO	ENCENDIDO	ENCENDIDO	FULL
ENCENDIDO	ENCENDIDO	ENCENDIDO	APAGADO	MEDIOALTO
ENCENDIDO	MALO	ENCENDIDO	APAGADO	SUSPENDIDO
ENCENDIDO	ENCENDIDO	ENCENDIDO	APAGADO	MEDIOALTO
ENCENDIDO	ENCENDIDO	APAGADO	APAGADO	MEDIOBAJO
ENCENDIDO	APAGADO	APAGADO	APAGADO	BAJO
APAGADO	APAGADO	APAGADO	APAGADO	VACIO
MALO	ENCENDIDO	APAGADO	APAGADO	SUSPENDIDO
MALO	ENCENDIDO	ENCENDIDO	APAGADO	SUSPENDIDO
MALO	ENCENDIDO	ENCENDIDO	ENCENDIDO	SUSPENDIDO
ENCENDIDO	ENCENDIDO	ENCENDIDO	ENCENDIDO	FULL
ENCENDIDO	MALO	ENCENDIDO	ENCENDIDO	SUSPENDIDO
ENCENDIDO	MALO	ENCENDIDO	APAGADO	SUSPENDIDO
ENCENDIDO	MALO	ENCENDIDO	ENCENDIDO	SUSPENDIDO
ENCENDIDO	MALO	MALO	ENCENDIDO	SUSPENDIDO
MALO	MALO	MALO	ENCENDIDO	SUSPENDIDO
MALO	ENCENDIDO	MALO	ENCENDIDO	SUSPENDIDO
ENCENDIDO	ENCENDIDO	MALO	ENCENDIDO	SUSPENDIDO
ENCENDIDO	ENCENDIDO	ENCENDIDO	ENCENDIDO	FULL

Figura 35: Tabla Sensores de nivel con distintos valores que arrojaron las pruebas respectivas.

Fuente: Realización Propia

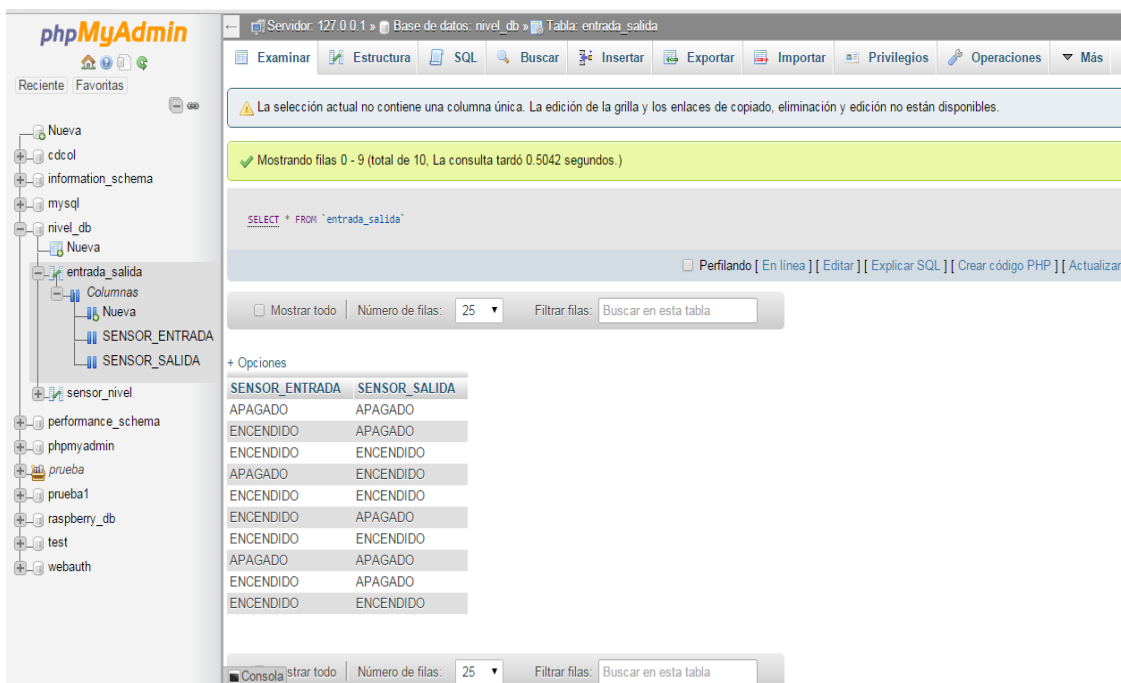
### V.3.2 Tabla Entrada\_Salida

En esta tabla se guarda información referente a la entrada y salida del agua del tanque de almacenamiento. Esta tabla está conformada por dos columnas las cuales se dividieron de la forma a continuación.

- **Sensor\_Entrada:** en esta columna se guarda toda la información perteneciente a la entrada de agua del tanque, cuando existe una entrada de agua en la misma se guarda la información llamada “encendido” referente a que el sensor de la entrada del tanque se encuentra en estado “uno”, mientras que por caso contrario si el mismo se encuentra en estado “cero” se almacena la información “apagado”.

- **Sensor\_Salida:** en esta columna están almacenado los estados referente a la salida del tanque de almacenamiento, los mismos que fueron enviados desde el archivo generado por el Raspberry, los datos están formado de las siguiente forma, un estado “1” generado por el sensor indica que existe salida de agua y en la tabla queda almacenado un “encendido” referente al estado del sensor y en el caso contrario si se genera un estado “0” por parte del sensor, el mismo indica que no hay actividad de agua y en la tabla se almacena un “apagado” referente a que el sensor se encuentra en un estado apagado.

En líneas generales para esta tabla fueron los resultados esperados ya que la misma guardaba la información referente de los sensores de manera correcta y precisa demostración de una buena funcionalidad de la misma.



La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

Mostrando filas 0 - 9 (total de 10, La consulta tardó 0.5042 segundos.)

`SELECT * FROM `entrada_salida``

☐ Perfilando [ En línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

+ Opciones

SENSOR_ENTRADA	SENSOR_SALIDA
APAGADO	APAGADO
ENCENDIDO	APAGADO
ENCENDIDO	ENCENDIDO
APAGADO	ENCENDIDO
ENCENDIDO	ENCENDIDO
ENCENDIDO	APAGADO
ENCENDIDO	ENCENDIDO
APAGADO	APAGADO
ENCENDIDO	APAGADO
ENCENDIDO	ENCENDIDO

Consola | Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla

**Figura 36:** Tabla de entrada y salida de agua y algunos resultados obtenidos de las pruebas.  
**Fuente:** Realización Propia

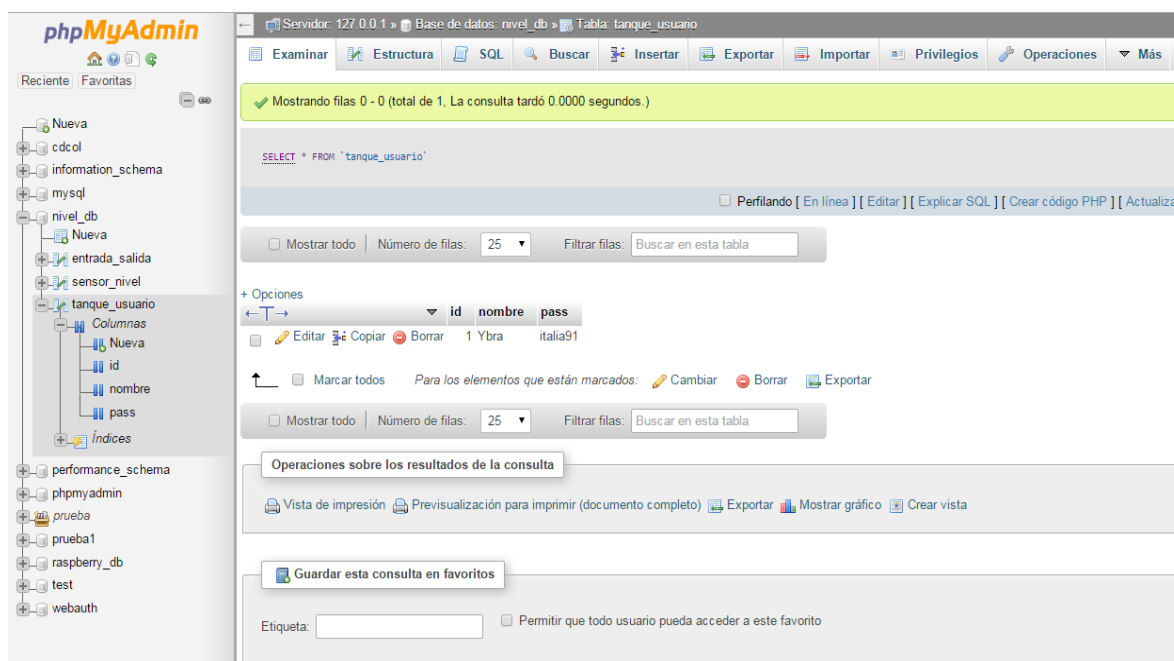
### V.3.3 Tabla Usuario\_Tanque

En esta tabla es en donde se guarda la información de los usuarios que están almacenados para el uso del sistema de automatización, esta tabla se encuentra dividida en tres columnas nombradas a continuación:

- ID: aquí se encuentra almacenado el número con el que se registró el usuario, cada usuario tiene un ID distinto y esto es lo que relaciona cada tanque a cada usuario.
- Nombre: en esta columna se almacena la información del nombre o *username* de cada usuario del sistema, este nombre puede ser modificable a través de la aplicación Android.
- Pass: esta columna contiene la información referente a la clave o *password* del usuario que permite ingresar en su menú principal para saber la información referente de su tanque.

En general para esta tabla fueron resultados esperados ya que a través de la aplicación Android se puede detallar estas columnas funcionando de manera adecuada para ingresar en la sesión correcta y poder ver de manera formal las prestaciones del sistema de automatización.

# DISEÑO DE UN SISTEMA DE AUTOMATIZACIÓN EN EL CONTROL DE CONSUMO, RACIONAMIENTO Y ANÁLISIS DE LOS NIVELES DE AGUA EN UNA ESTRUCTURA DE ALMACENAMIENTO



**Figura 37: Tabla usuario de la base de datos del sistema.**

**Fuente: Realización Propia**

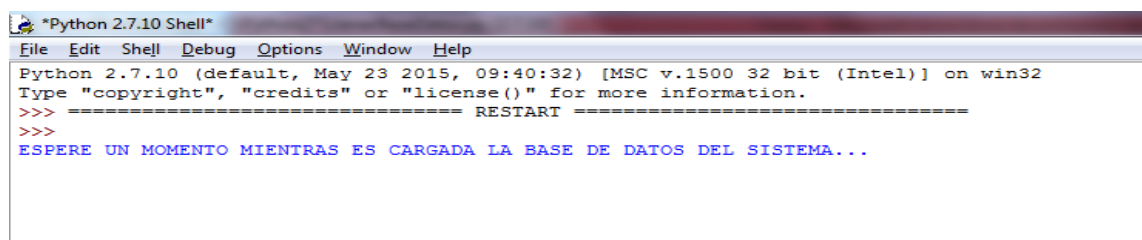
## V.4 Aplicación LlenarBaseDatos

Esta aplicación Python se corrió en un PC donde se encuentra el servidor local y es la que envía la información del archivo de texto a la base de datos, dicha información es enviada cada 10 segundos y se actualiza. La actualización ocurre de la siguiente forma, la aplicación se corre de manera permanente siempre y cuando la PC del servidor local se encuentre encendida y además de esto, que el administrador local corra la aplicación una vez para que comience a funcionar, entonces luego de esto la función LeerArchivo() se va a encargar de leer cada 10 segundos el archivo de texto referente que viene del Raspberry PI, si existe un cambio en este archivo la información será guardada en la tabla correspondiente de la base de datos

“Nivel\_db”, después de establecer conexión con la misma. De caso contrario no se almacena nada en la tabla, y esta rutina se vuelve a cumplir cada 10 segundos hasta que exista un cambio en el archivo de texto correspondiente.

Ocurre lo mismo, no solo para la tabla de los sensores de nivel, sino para la tabla entrada y salida de agua, la información es actualizada cada 10 segundos siempre y cuando existan cambios en el comportamiento de los sensores del sistema de caso contrario la misma permanecerá igual.

Entonces para esta aplicación de llenado los resultados fueron los esperados ya que lee de forma eficiente la información que se encuentra en el archivo de texto que viene del Raspberry Pi. Luego de esto la información antes nombrada se inserta en las diferentes tablas de la base de datos correctamente, además compara de manera adecuada con valores iguales y así poder descartar aquellos valores que no sean referentes para el llenado y cumple con el requisito de actualizar esta rutina cada 10 segundos aproximadamente.



```
*Python 2.7.10 Shell*
File Edit Shell Debug Options Window Help
Python 2.7.10 (default, May 23 2015, 09:40:32) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>> ESPERE UN MOMENTO MIENTRAS ES CARGADA LA BASE DE DATOS DEL SISTEMA...
```

**Figura 38:** Interfaz vista desde la consola de esta aplicación corriendo en el respectivo sistema.

**Fuente:** Realización Propia



## **V.5 Aplicación Android “Mi tanque APP”**

Los resultados obtenidos de aplicación Android “Mi tanque APP” reflejan la sección del sistema que interactúa directamente con el usuario final, reflejando la importancia de su óptimo funcionamiento y fácil manejo para el usuario con el fin de poder cumplir el objetivo de facilitar el control de los niveles porcentuales de agua dentro del tanque y de poder tomar las medidas preventivas que esto implica. Para la muestras de los resultados se utilizó la opción de poder emular la aplicación desde un teléfono Android con permisos de desarrollador, conectado a la PC a través del puerto USB para que de esta manera pudiéramos tener el resultado más parecido a lo que sería para el usuario interactuar con la aplicación.

El teléfono utilizado para emular los resultados fue un LG L35, con sistema operativo Android 4.4.2 (KitKat), API 19.

### **V.5.1 Ventanas desarrolladas en la aplicación**

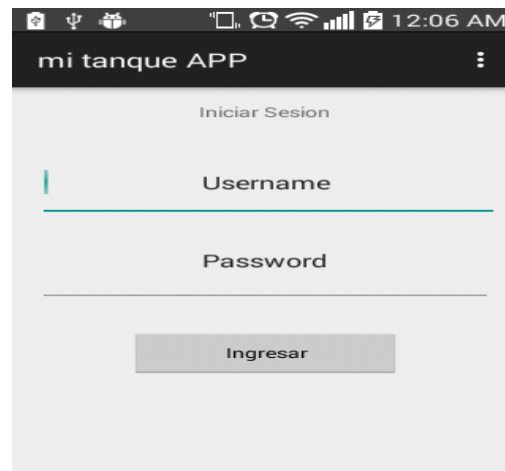
A continuación se muestra las ventanas que serán mostradas al usuario desde el inicio, dando una breve explicación de en qué consiste cada una y cual actividades están vinculadas directamente con ellas.

#### **V.5.1.1 Iniciar Sesión**

Como su nombre lo dice, esta ventana es la primera en aparecer luego de abrir la aplicación, la cual permite al usuario ingresar sus datos para así ingresar al menú de opciones. En el caso de que la información proporcionada por el usuario no sea

correcta, un “Toast” programado aparecerá en la ventana informando que se deben ingresar los datos, ya que son incorrectos; entiéndase por Toast en Android como ventanas emergentes con información corta del tamaño del texto, las cuales desaparecen automáticamente luego de un tiempo de espera.

La ventana está relacionada directamente con la Activity “Login” la cual permite recibir los datos y enviarlos a la Activity “LoginSync” para su posterior verificación.



**Figura 39: Login de la Aplicación Android.**

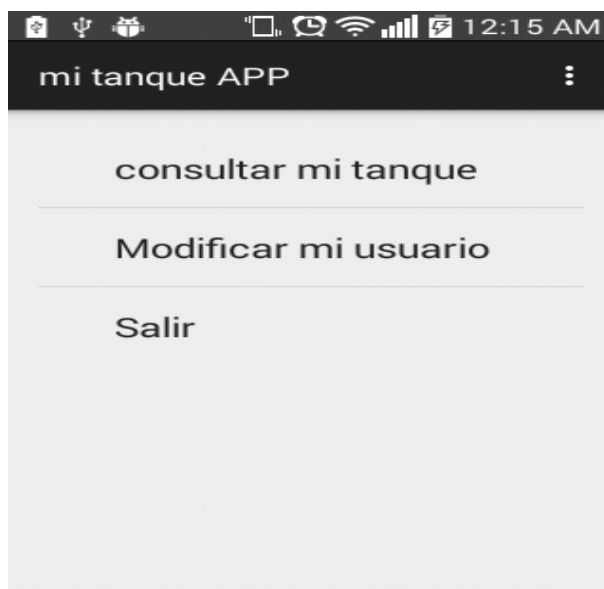
**Fuente: Realización Propia**

### **V.5.1.2 Menú de inicio**

Esta ventana es la siguiente en aparecer en el caso de un exitoso inicio de sesión, la cual permite al usuario poder elegir entre la opción de consultar los niveles actuales de agua en el tanque, modificar sus datos de usuario o salir del menú. El fin

de esta ventana es poder facilitar la manera de poder acceder a la información de una manera ordenada y fácil.

Esta ventana está relacionada directamente con la Activity “PantallaUsuario” la cual permite ser un tipo de puente entre las demás actividades mediante el paso de parámetros.



**Figura 40: Menú principal de la aplicación.**

**Fuente: Realización Propia**

### **V.5.1.3 Consulta actualizada del tanque**

Esta ventana es la siguiente en aparecer en el caso de que sea seleccionada la opción número uno en el menú de inicio, la cual permitirá al usuario poder obtener información actualizada de los niveles porcentuales de agua existentes en el tanque, si

se encuentra o no entrando o saliendo agua del recinto y en el caso del daño de un sensor indica cuál de estos presenta la falla; todo esto a través de una lista que el usuario puede deslizar para poder leer. Para poder dar a conocer parte de la información se hizo uso de ilustraciones realizadas con el fin de poder mostrar de una manera sencilla las distintas posibles situaciones. A continuación se mostrara la imagen correspondiente a un posible caso.



**Figura 41:** Caso en el que el uno de los sensores se encuentre dañado.


**Fuente:** Realización Propia

En la figura se muestra la información referente al daño de alguno de los sensores mediante un símbolo de advertencia en conjunto al sensor que presenta la falla, obligando al usuario tener que revisar el sistema de manera inmediata.

#### V.5.1.4 Modificación de datos

Esta ventana es la siguiente en aparecer en el caso de que sea seleccionada la opción número dos en el menú de inicio, la cual permitirá al usuario poder agregar los nuevos datos que serán modificados en la casilla perteneciente al usuario en los registros. Permitirá la inclusión de un nuevo nombre y contraseña, la cual tendrá que ser escrita dos veces con motivos de confirmación de la misma.

Esta ventana está relacionada directamente con la Activity “ModificarUsuario” la cual permite recibir los datos y enviarlos a la Activity “ModificarUsuarioSync” para su posterior anexo y modificación en la base de datos.



The screenshot shows a mobile application interface for 'mi tanque APP'. The status bar at the top indicates the time is 02:03 PM. The app's header is dark with the text 'mi tanque APP' and a three-dot menu icon. The main content area has a light background and contains the following elements from top to bottom: a text label 'Modificar Usuario', a text input field 'Nuevo Usuario' with a green underline, another text input field 'Nueva Contraseña' with a grey underline, a third text input field 'Confirmar Contraseña' with a grey underline, and a grey rectangular button at the bottom labeled 'ACEPTAR'.

**Figura 42:** Ventana de la aplicación referente a modificaciones de un usuario.

**Fuente:** Realización Propia

### **V.5.2 Servicios PHP**

Los servicio php que se utilizaron fueron 5 script desarrollados en lenguaje PHP, los cuales se encuentran en nuestro servidor, a los que se puede acceder mediante direcciones web y pasándole los parámetros necesarios posibilitando la conexión entre la base de datos y cada una de las actividades que hacen posible el funcionamiento de la aplicación Android mediante la transmisión de datos en un formato ligero.

### **V.6 Transmisión de datos del Sistema**

En el siguiente diagrama se puede detallar una vista general de cómo es la transmisión de datos desde un punto de vista de telecomunicaciones indicando cuales fueron las maneras en que dichos datos iban transmitiéndose hasta que la respectiva información referente del tanque de almacenamiento llegue a la interfaz del usuario pudiendo así cumplir con los objetivos planteados del sistema.

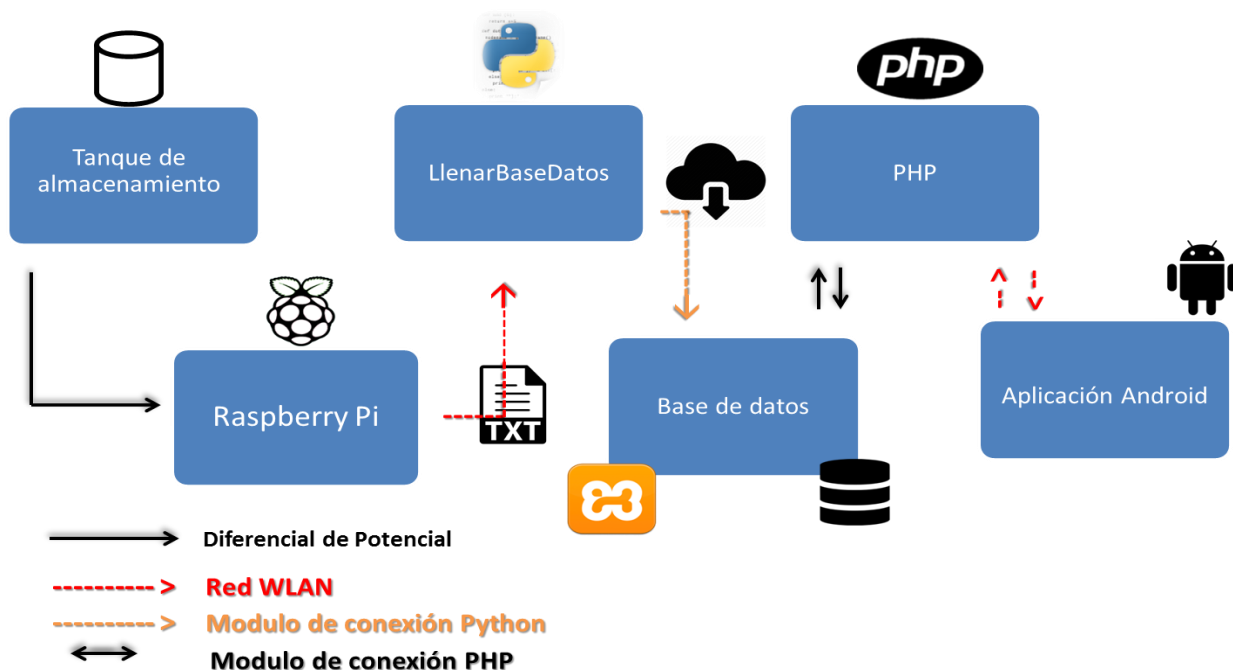


Figura 43: Diagrama de Telecomunicaciones.

Fuente: Realización Propia

## V.7 Pruebas realizadas en entornos simulados y real

Con la realización de las pruebas realizadas tanto en entorno simulado como en entorno real, se lograron resolver algunos errores que nos arrojaba el sistema, también se pudo observar la diferencia de resultados en ambos entornos y cuáles eran las distorsiones que estaban presentes en el entorno real que en el simulado no.

En la siguiente figura se observan los resultados en el caso que el agua no alcanza el primer sensor de nivel, es decir, los niveles de agua eran críticos en el tanque de almacenamiento.



Figura 44: Prueba en un entorno real de todos los parámetros del sistema.

Fuente: Realización Propia

En la siguiente figura se muestran los resultados en el caso que los niveles de agua se encuentran en el nivel 2 es decir se encuentran activos los sensores 1 y 2.



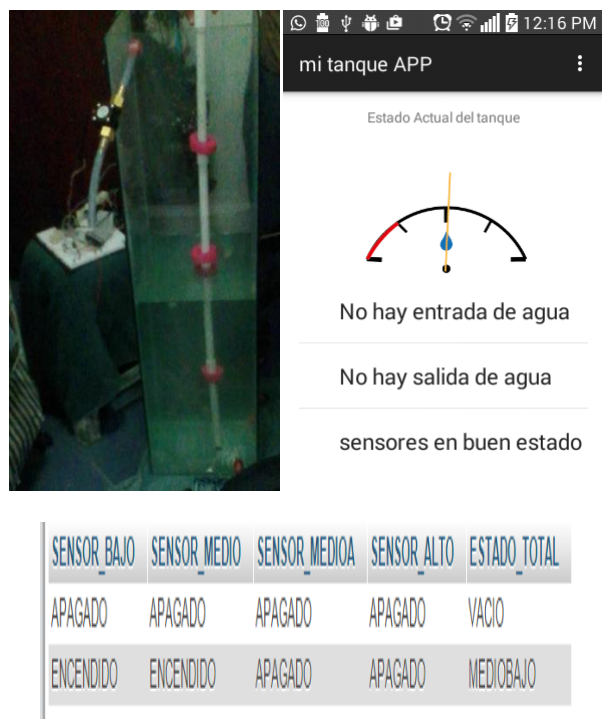


Figura 45: Resultados de pruebas cuando están activos los sensores 1 y 2.

Fuente: Realización Propia

## **CAPITULO VI**

### **Conclusiones y Recomendaciones**

Una vez que fueron elaborados todos y cada uno de los objetivos planteados en el Trabajo Especial de Grado, se procederá a presentar las conclusiones y recomendaciones obtenidas luego de su realización.

#### **VI.1 Conclusiones**

Dado que el contenido expuesto en los capítulos anteriores en el proyecto y tomado en cuenta los resultados obtenido, se puede destacar que a pesar de algunos problemas que se presentaron a lo largo de su desarrollo debido a la dependencia de diversos dispositivos, se logró el diseño de un sistema el cual le permite a los usuarios mejorar la forma de saber los niveles de agua existentes en su tanque de almacenamiento, mejorando así su calidad de vida y evitándole pérdidas monetarias o daños a la bomba hidráulica utilizada para enviar agua a los distintos servicios a utilizar.

Lo antes descrito motivó a los autores de este Trabajo Especial de Grado a la elaboración de un sistema el cual le permitiera a los usuarios tener conocimiento de los estimados de aguas existentes en su estructura de almacenamiento ya sea en un área local, tal como puede ser un apartamento, o ya sea un área comercial y que necesite el conocimiento de los estimados, para evitar pérdidas de dinero innecesaria.

Se cumplieron cada uno de los objetivos específicos nombrados en este Trabajo Especial de grado entre los cuales se pueden destacar:

- Se Investigó con suficiente amplitud elementos, a nivel de red y transmisión, que conforman el sistema para la automatización en el control de consumo, racionamiento y análisis del nivel del agua disponible.
- Diseño completo desde el punto de vista conceptual y en entorno real del sistema de automatización.
- Realización exitosa de la escogencia y adaptación del subsistema de sensores de entrada y salida de agua y de nivel.
- Cumplimiento del desarrollo de la adaptación del software en la transmisión de datos del subsistema de sensores a la aplicación Android.
- Implementación de una base de datos de la información que se adquiere en el sistema de automatización.
- Diseño completo de la aplicación Android del sistema de entrega de datos al usuario.
- Construcción del prototipo funcional mediante el cual se muestran las prestaciones del sistema.

## VI.2 Recomendaciones

Se recomienda algún tipo de fuente externa que pueda alimentar el sistema en el caso de su instalación en lugares remotos o carentes de conexiones eléctricas ya que dicho sistema funciona bajo la utilización de voltaje.

Con el fin de evitar posibles limitaciones y que el sistema se encuentre disponible para una mayor cantidad de usuarios, se recomienda el desarrollo de la aplicación móvil para otros sistemas operativos como IOS, Windows Phone, BlackBerry, etc.

Se recomienda el uso de servidores que no sean locales para aumentar la amplitud del proyecto y que no sea necesaria una red solo local sino el uso del internet.

Se recomienda la implementación de un *plugin* detector de fugas dentro del tanque de almacenamiento haciendo uso del sensor de flujo utilizado y los sensores flotantes tipo eléctrico con el fin de detectar y solventar todas aquellas fugas que existan mayormente en tanque subterráneos.

## BIBLIOGRAFIA

- Almazan, B. (09 de Abril de 2008). *Gestiopolis*. Recuperado el 15 de Febrero de 2015, de Gestiopolis: <http://www.gestiopolis.com/automatizacion-robotica-produccion/>
- Android, A. (11 de Diciembre de 2014). *academiaandroid*. Recuperado el 3 de Agosto de 2015, de academiaandroid: <http://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>
- Benitez, B. (8 de Enero de 2008). *Codeduino*. Recuperado el 6 de Agosto de 2015, de Codeduino: <https://codeduino.com/projects/robotics-automation/>
- Bentley, A. (10 de Mayo de 2008). *Metas*. Recuperado el 4 de Agosto de 2015, de Metas: <http://www.metas.com.mx/guiametas/la-guia-metas-08-05-sensores-de-humedad.pdf>
- Catalan, A. (Julio de 2010). *Catalanes*. Recuperado el 3 de Agosto de 2015, de Catalanes: <https://www.softcatala.org/catalanitzador/android.php>
- Cortez, A. (25 de Mayo de 20014). *NoticiasDot*. Recuperado el 3 de Agosto de 2015, de NoticiasDot: <http://www.noticiasdot.com/publicaciones/2004/0804/0508/noticias050804/noticias050804-11.htm>
- Cortez, A. (25 de Mayo de 2014). *Noticias Dot*. Recuperado el 2015, de Noticias Dot.
- Cruz, F. J. (24 de Febrero de 20011). *Escripd*. Recuperado el 15 de Febrero de 2015, de Escripd: <http://es.scribd.com/doc/123087446/Tema-11-Depositos-Conceptos-Generales#scribd>

- General, E. (17 de Abril de 2010). *ayudaelectronica.com*. Recuperado el 5 de Agosto de 2015, de ayudaelectronica.com: <http://ayudaelectronica.com/ques-un-reed-switch/>
- Gonzalez, J. (8 de Julio de 2001). *Sistemas de Comunicación*. Recuperado el 16 de Febrero de 2015, de Sistemas de Comunicación: <https://sistemascomunic.wordpress.com/sistemas-de-comunicacion/>
- Leon, H. (10 de Octubre de 2009). *Ocitys*. Recuperado el 6 de Agosto de 2015, de Ocitys: [http://www.oocities.org/es/herleon123/ads/SGBD\\_foro.htm](http://www.oocities.org/es/herleon123/ads/SGBD_foro.htm)
- Machado, G. (4 de Enero de 2012). *FacilElectronica*. Recuperado el 10 de Febrero de 2015, de facilElectronica: <http://www.electronicaestudio.com/sensores.htm>
- Marzal, A. (23 de Marzo de 2008). *Aleg*. Recuperado el 4 de Agosto de 2015, de Aleg: <http://www.alegsa.com.ar/Dic/python.php>
- Meneses, R. (24 de Mayo de 2007). *DefinicionAbc*. Recuperado el 5 de Agosto de 2015, de DefinicionAbc: <http://www.definicionabc.com/tecnologia/mysql.php>
- Rosales, R. (20 de Julio de 2010). *Area de tecnologia*. Recuperado el 15 de Febrero de 2015, de Area de tecnologia: <http://www.areatecnologia.com/Que-es-un-smartphone.htm>
- W. Frank Ableson, R. S. (11 de Noviembre de 2011). *Manning.com*. Recuperado el 1 de Agosto de 2015, de Manning.com: <http://www.manning.com/ableson3/>