UNIVERSIDAD DEL AZUAY

FACULTAD DE CIENCIA Y TECNOLOGÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA

“Sistema electrónico de medición de consumo de agua potable.”

Trabajo de graduación previo a la obtención del título de:

INGENIERO ELECTRÓNICO.

Autor:

MAURO ISRAEL MOSQUERA MONTERO.

Director:

LEOPOLDO CARLOS VÁSQUEZ RODRÍGUEZ.

CUENCA-ECUADOR

2015

DEDICATORIA

El presente trabajo está dedicado a mis padres Mauro y Graciela por haberme apoyado incondicionalmente siempre y haber creído en mí en todo el transcurso de mi carrera estudiantil y vida. A mis hermanas Dámaris y Sara por haberme dado siempre un gran ejemplo de lucha, sacrificio y superación.

Israel Mosquera.

AGRADECIMIENTOS

Agradezco en primer lugar a Dios por todas y cada una de las bendiciones que inmerecidamente me ha otorgado, a mi familia por todo el gran e incondicional apoyo a través del tiempo, y por su puesto a todos mis amigos; sería imposible nombrar a todos pero injusto no nombrar a algunos, principalmente a mi compañero y querido amigo Ing. Xavier Gordillo sin el cual hubiera sido imposible el presente trabajo, al Ing. Andrés Cabrera cuya disposición y ayuda han sido absolutamente invaluables, a Julio Andrés Muñoz cuya lealtad y apoyo no pueden expresarse correctamente en palabras; y a todos y cada uno de los que directa o indirectamente, ya sea por un buen gesto o lo contrario ayudaron a que nunca claudicara en el camino profesional que escogí.

SISTEMA ELECTRÓNICO DE MEDICIÓN DE CONSUMO DE AGUA POTABLE.

RESUMEN

El propósito general del presente proyecto de tesis es generar un sistema que pueda unificar, la tecnología de servicios web y software libre, basados en gestión de base de datos junto con dispositivos de medición y tratamiento de señales de caudal y dispositivos móviles basados en sistema operativo Android.

Los elementos de control, medición, gestión y comunicación que intervienen en el presente tema de tesis son microcontroladores, sensor de caudal, pantalla LCD y herramientas de software tales como NetBeans y AppInventor; cuya interacción conforman, en un todo, un sistema electrónico capaz de realizar mediciones de caudal de gran exactitud y ponerlas al servicio de los usuarios a través de servicios web y aplicaciones moviles.

Palabras clave: Sensor, PWM, Arduino, Ethernet, Servidor, Java, Android.

UNIVERSIDAD DEL AZUAY
ESCUELA Ingeniería Electrónica

Lcdo. Leopoldo Carlos Vásquez Rodríguez
Director de Tesis.

Ing. Francisco Eugenio Vásquez Calero
Director de Escuela.

Mauro Israel Mosquera Montero
Autor.

WATER CONSUMPTION ELECTRONIC MEASURING SYSTEM

ABSTRACT

The general purpose of this research project aims to create a system that can unify web service technology and free software based on database management along with devices for flow signal measuring and treatment, and mobile devices based on Android operating system.

The control, measurement, management and communication elements that are part of this research are microcontrollers, flow sensor, LCD screen and software tools such as NetBeans and App Inventor; whose interaction form as a whole, an electronic system capable of high accuracy flow measurements, which will be available to users through web services and mobile applications.

Keywords: Sensor, PWM, Arduino, Ethernet, Server, Java, Android

Lcdo. Leopoldo Carlos Vásquez Rodríguez
Thesis Director

Ing. Francisco Eugenio Vásquez Calero
School Director

UNIVERSIDAD DEL AZUAY
ESCUELA Ingeniería Electrónica

Mauro Israel Mosquera Montero
Author

UNIVERSIDAD DEL AZUAY
DPTO. IDIOMAS

Translated by,
Lic. Lourdes Crespo

ÍNDICE DE CONTENIDOS.

ÍNDICE DE FIGURAS.

Mosquera Montero Mauro Israel.
Trabajo de Graduación.
Lcdo. Leopoldo Carlos Vásquez Rodríguez.
Febrero 2015.

SISTEMA ELECTRÓNICO DE MEDICIÓN DE CONSUMO DE AGUA POTABLE

INTRODUCCIÓN

Medidor de Caudal y Agua Potable

Un medidor de caudal, como lo indica su nombre, mide la cantidad de líquido, (en este caso agua) utilizado desde algún tipo de fuente. Estos instrumentos se utilizan como herramientas para que las empresas que proveen servicios de agua potable, puedan medir el consumo del mismo de cada uno de sus clientes. Partiendo de esa premisa, un medidor de consumo de agua potable es un instrumento que registra, con relativa exactitud, la cantidad de agua que ingresa, desde la red de distribución de la empresa proveedora del servicio, hasta el interior del domicilio del usuario (abonado) (SANITARIOS, 2005).

Debido a la importancia de dicho dispositivo, es necesario proteger al mismo de maltrato y manipulación de personas ajenas, ya que solo debe ser manipulado por personal autorizado perteneciente a la empresa que provee el servicio.

Medidores Mecánicos de Consumo de Agua Potable

Entre los muchos tipos de medidores, los que generalmente se encuentran en los domicilios son los medidores mecánicos de desplazamiento positivo (Figura 1), cuyo funcionamiento consiste en medir el caudal de agua que pasa por una recamara medida previamente que se encuentra en el interior del equipo, la medición del

caudal requerido se calcula de acuerdo a la cantidad de veces que se llena y vacía dicha recamara. Mediante un mecanismo de disco oscilante o pisto al que está conectado el dispositivo, se visualizan los datos de caudal en la parte superior del equipo (BURKE, 2000).

Figura 1. Medidor típico de consumo de Agua.

Fuente: (GUTIERREZ, 2011)

Sin embargo a pesar de la innegable utilidad que posee un medidor convencional, podemos ver las siguientes desventajas:

- Solo pueden utilizarse en sistemas de aguas altamente filtrada, ya que cualquier partícula, arena, sales o material en general obturarían el paso del agua.
- Si no están debidamente instalados, se producen vibraciones en la línea de salida con las consecuentes molestias para el consumidor.
- Las lecturas que se aprecian en el medidor no son del todo claras, pues en muchos casos hay que transformar unidades para tener una idea clara del consumo, cosa que no es tan simple para gran parte de consumidores.

- Se requiere que personal de la empresa constantemente revise los datos del medidor, lo que se traduce en pérdida innecesaria de tiempo y recursos.
- Los medidores convencionales no permiten adicionar un servicio complementario que minimice los tiempos de ejecución, permita ahorrar recursos e implemente servicios adicionales complementarios.
- El porcentaje de errores en medición se encuentra entre un ± 4% al ± 10% (GUTIERREZ, 2011).
- En caso de fallos se requiere necesariamente un cambio total del medidor.

El Medidor Electrónico de consumo de Agua potable como solución.

A partir de estos hechos, el presente proyecto de tesis pretende crear un sistema electrónico que permita entre otras cosas:

- Un medidor electrónico de consumo de agua potable con mínimo error
- Un registro electrónico de consumo de agua potable.
- Envío de los registros de consumo hacia una base de datos.
- Generación de facturas.
- Monitoreo de consumo y facturación en tiempo real.
- Sistema de estado del equipo y respaldos.
- Servicios adicionales para usuarios (páginas web de consulta).

A su vez, para el desarrollo de un sistema que cumpla con estos requerimientos, es necesario tener claros algunos conceptos indispensables en materia de electrónica y programación que, junto con los detalles del desarrollo de los diferentes aspectos del proyecto, es lo que se efectuará a medida que se avance en el tema a través de los capítulos que lo componen.

CAPITULO 1

MEDIDOR ELECTRONICO DE CONSUMO DE AGUA POTABLE

1.1 Conceptos Preliminares.

A continuación se mostrarán conceptos importantes a tener en cuenta para el desarrollo del presente tema de tesis, cabe mencionar que se presentarán los conceptos relacionados únicamente con el hardware del equipo, es decir, el medidor físico incluyendo el firmware contenido en el microcontrolador, los aspectos relacionados con software, es decir, programación, páginas web y bases de datos, se analizarán en capítulos posteriores

1.1.1 PWM (Pulse-Width Modulation).

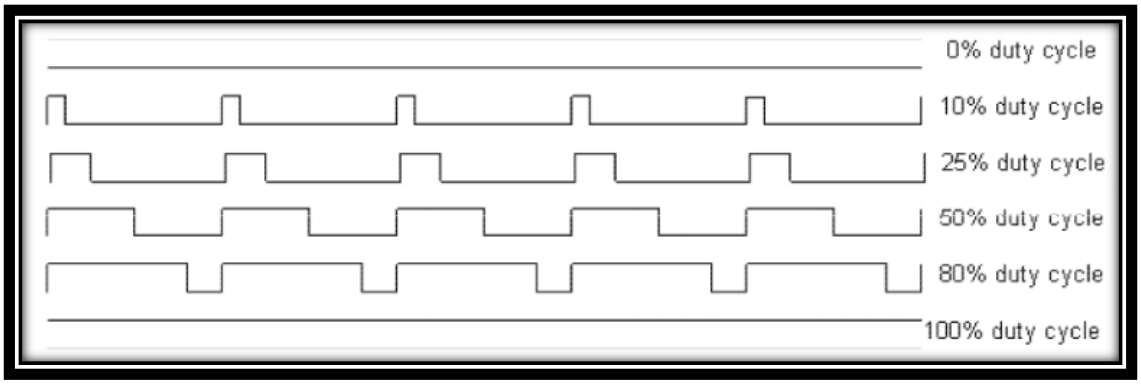
PWM cuyas siglas en ingles se traducen al español como modulación por ancho de pulso, se trata de una técnica que permite generar ondas cuadradas con su respectiva frecuencia y ciclo de trabajo, como en toda onda, la frecuencia no es más que el número de ciclos por unidad de tiempo, y consecuentemente el periodo es el tiempo que demora la onda en dar un ciclo completo (ARCOS, 2011).
Podríamos expresar el ciclo de trabajo (duty cycle), como la relación entre la parte positiva de la onda vs el período (Figura 2). Si lo expresamos matemáticamente tendríamos:

$$D = \frac{\tau}{T}$$

En donde:

- D es el ciclo de trabajo.
- τ es el tiempo en que la función es positiva (ancho del pulso).
- T es el período de la función.

Figura 2. Ejemplos de diferentes ciclos de trabajo de una onda PWM.



Fuente: (ARCOS, 2011)

Como se analizará a continuación, el sensor utilizado como elemento captador de señal, genera una onda PWM que posteriormente será procesada por el microcontrolador partiendo de una relación PWM/caudal proporcionada por el fabricante del sensor, que nos permite obtener instantáneamente el caudal y consecuentemente el consumo de agua potable dependiendo de la variación de frecuencia del PWM entregado.

1.1.2 Efecto Hall y Sensor de Caudal.

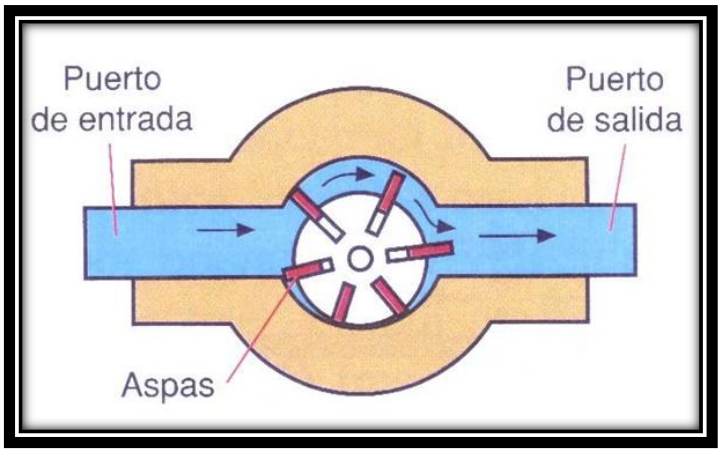
1.1.2.1 Efecto Hall.

Descubierto en 1879, el fenómeno Hall, llamado así en honor al físico estadounidense Edwin Herbert Hall, este efecto consiste básicamente en la aparición de un campo eléctrico en un material conductor o semiconductor por el cual circula una corriente, cuando dicho material es atravesado por un campo magnético genera una diferencia de potencial cuantificable junto con su correspondiente campo eléctrico denominado campo Hall. Cabe destacar que el campo magnético que atraviesa el material es perpendicular y proporcional al campo eléctrico y la diferencia de potencial que circulan por el mismo, de ahí que cualquiera de las variables pueden ser obtenidas a partir de uno de los campos (WIKISPACES, 2012).

1.1.2.2 Sensor de Caudal.

El sensor de caudal utilizado se compone de una válvula de plástico, un rotor (parte giratoria), y un sensor de efecto Hall. Cuando el agua fluye a través de los rodillos del rotor, este comienza a girar con mayor o menor velocidad dependiendo del caudal de líquido que fluye a través de él (Figura 3) (ROMAN, 2010).

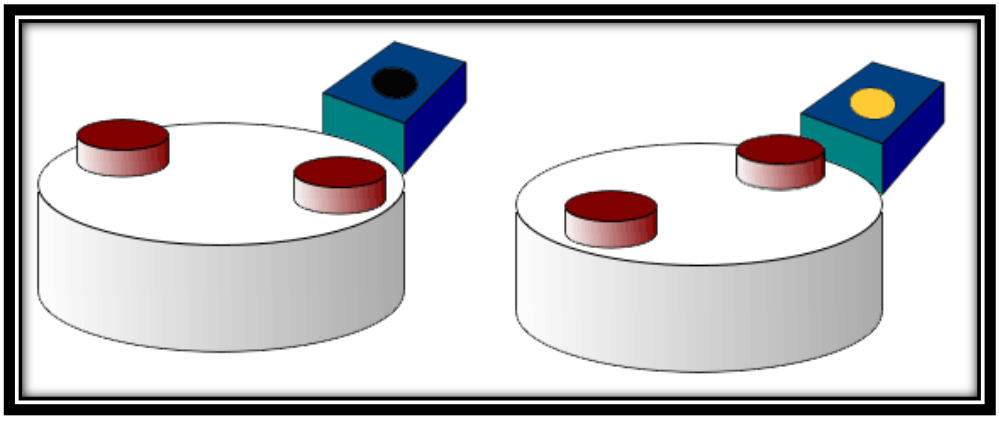
Figura 3. Esquema sencillo del funcionamiento del sensor de caudal.



Fuente: (ROMAN, 2010)

Cuando los rodillos del rotor se ponen en contacto con el campo magnético del sensor, el sensor de efecto Hall emite la señal eléctrica correspondiente en forma de pulsos eléctricos (efecto Hall) (Figura 4), dichos pulsos equivalen a una señal en forma de onda PWM que se entrega al microcontrolador para su posterior conversión en caudal (ROMAN, 2010).

Figura 4. Ejemplo de la generación de diferencia de potencial (PWM) a partir de un campo magnético a través de un conductor, si el conductor está lejos del campo del sensor, no genera potencial (led apagado) y viceversa (led encendido).



Fuente: (GORDILLO, 2014).

Concretamente, el sensor utilizado es el sensor de flujo de agua G ½ pulgada (Figura 5 y Figura 6), que se adapta de mejor forma a los requerimientos necesarios por tener las siguientes características:

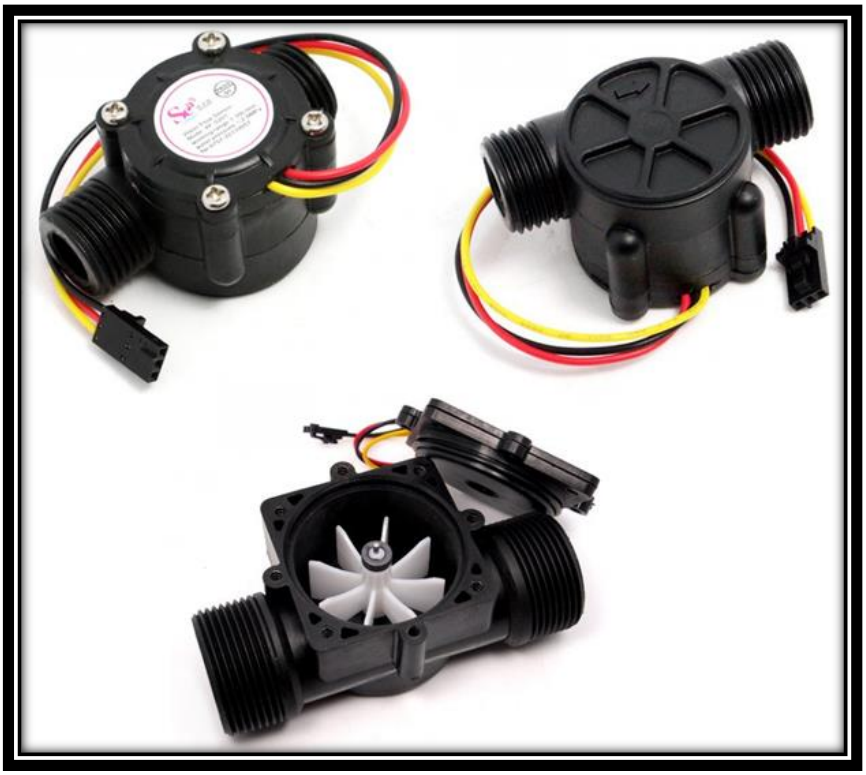
- Compacto, de fácil instalación.
- Enroscado hermético
- Sensor de efecto Hall de alta calidad
- Cumple con la norma RoHS (SISWANTORO, 2013).

A continuación se pasa a describir en resumen sus especificaciones más importantes:

- Voltaje de funcionamiento 5V-24V.
- Corriente máxima 15 mA (DC 5V).
- Peso 43 g.
- Rango de caudal 1 ~ 30 L / min.
- Temperatura de funcionamiento 0C ~ 80 °C.
- Temperatura del líquido <120 °C.
- Humedad de funcionamiento 35% ~ 90% RH.
- Presión de trabajo 1.75 Mpa.
- Temperatura -25C ~ +80 C.
- Humedad 25% ~ 90% RH.

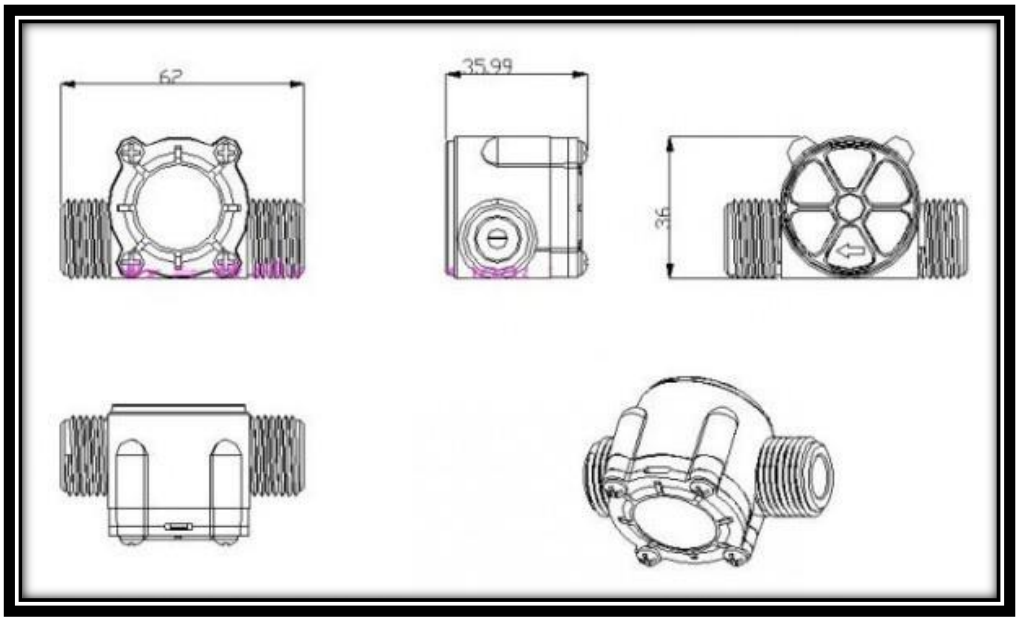
- Dimensiones: 2.44 pulgadas x 1.42 pulgadas x 1.34 pulgadas (6,2 cm x 3,6 cm x 3,4 cm).
- Peso: 1,76 oz (50 g).
- Color: Negro.
- Material: Lista PVCPacking: 1 x sensor (longitud del cable: 15 cm) (SISWANTORO, 2013).

Figura 5. Sensor de flujo de agua G 1/2" utilizado.



Fuente: (SISWANTORO, 2013).

Figura 6. Dimensiones Mecánicas.

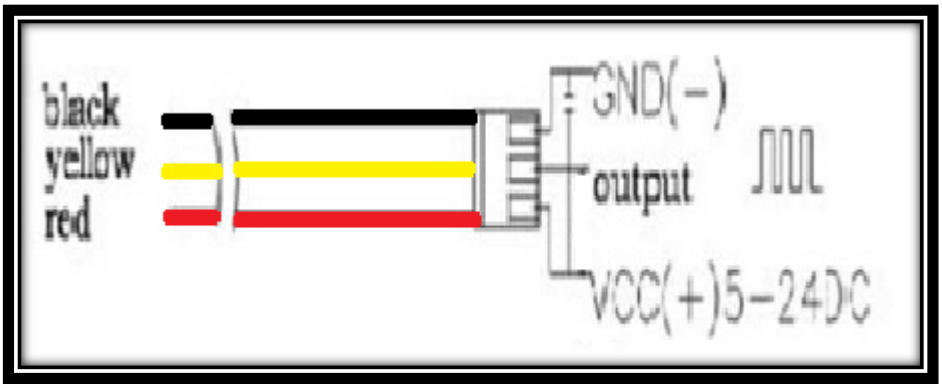


Fuente: (SISWANTORO, 2013).

1.1.2.3 Conexiones del Sensor.

Como se puede apreciar en la Figura 7, el sensor de caudal utilizado presenta 3 salidas claramente diferenciables por su color, negro para la conexión a tierra, rojo para el voltaje de alimentación y amarillo como el elemento que entrega la señal. Basados en la tabla de pruebas de señales de salida proporcionada por el fabricante, a continuación de la Figura 7 se puede ver que el nivel de error máximo es del 3%, cuyo valor es significativamente menor que el de los medidores convencionales que presentan un error desde el 4 al 10% (GUTIERREZ, 2011).

Figura 7. Conexión de las salidas del sensor de caudal.



Fuente: (SISWANTORO, 2013).

Tabla de pruebas de Señales de Salida (SISWANTORO, 2013):

Alto nivel de impulsos de salida	Tensión de señal> 4,5 V (entrada de 5 V DC)
Bajo nivel de impulsos de salida	Tensión de señal <0,5 V (entrada DC 5V DC)
Precisión	3% (tasa de flujo de 1L/min a 15l/min)
Ciclo de trabajo de la señal de salida	38%～65%

1.1.3 Microcontrolador.

Basados en las necesidades ya mencionadas de gestionar señales de PWM, junto con otras adicionales que se verán en capítulos posteriores, tales como memoria para respaldos, protocolos de comunicación Ethernet, bluetooth, etc; a más de tener en cuenta otro tipo de factores como costos y espacio, se optó por el microcontrolador y placa ARDUINO UNO, cuyas características generales más importantes se pasan a describir a continuación.

1.1.3.1 Placa Electrónica Arduino UNO.

Arduino UNO (Figura 8) es nombre con el que se conoce a la tarjeta electrónica basada en el microcontrolador ATmega328. Resumiendo sus características principales, cuenta con 14 pines que pueden ser dedicados a entradas/salidas

digitales, de los cuales 6 pueden ser utilizadas como salidas PWM y servir como entradas al mismo tipo de señal, lo cual es altamente conveniente para nuestro propósito; cuenta también con 6 entradas análogas de 10 bits, por lo que entregan valores entre 0 y 1023, un cristal oscilador de 16Mhz, conexión y/o alimentación vía USB, conector adicional de alimentación, conector ICSP (In circuit serial Programing) ideal para grabar el firmware en el microcontrolador sin necesidad de desmontarlo de la placa, y botón de reset. Adicional a estas características, su microcontrolador permite mayores velocidades de transmisión por su puerto USB y no requiere drivers adicionales para Linux o MAC, cuenta también con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc (ARDUINO, 2010).

Arduino UNO está equipado para gestionar el microcontrolador ATmega328, basta con conectar la placa a un ordenador por medio del cable USB, mediante una batería o un adaptador AC-DC. Si se alimenta a través de USB, la alimentación externa no es necesaria (ARDUINO, 2010).

Para su programación solo es necesaria el entorno de programación (IDE) de Arduino, dicho entorno se puede descargar gratuitamente desde la sección de descargas en el sitio web de *www.arduino.cc.* (ARDUINO, 2010).

A continuación se muestra las características generales y puntuales más destacadas tomadas de la hoja de datos del fabricante que corroboran la predilección por dicho instrumento (ARDUINO, 2010).

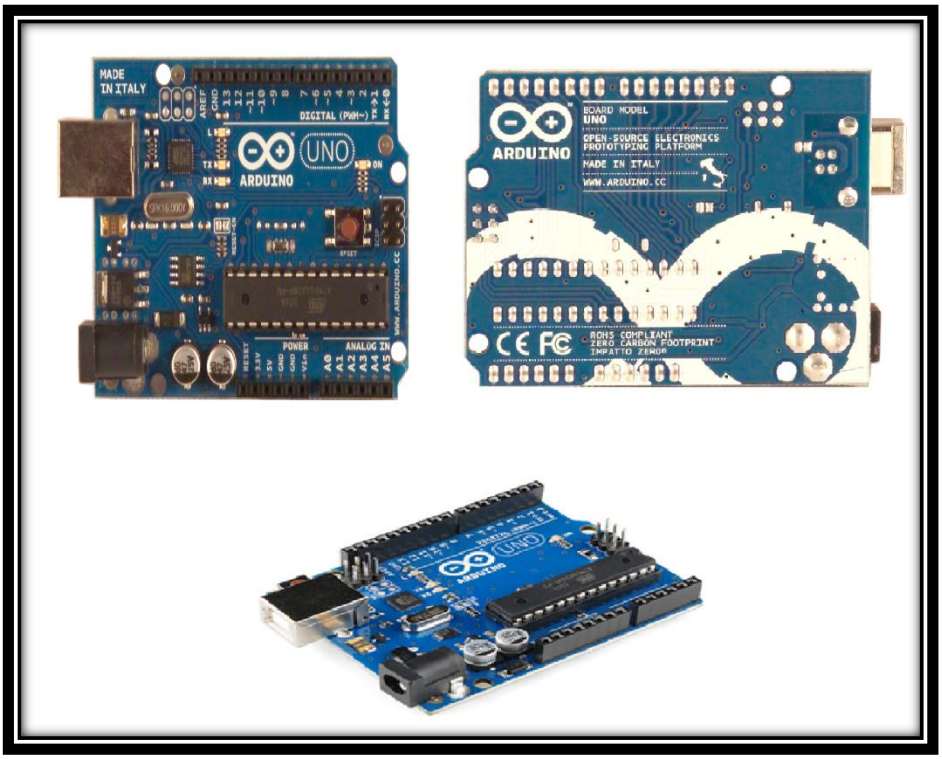
1.1.3.2 Características Generales.

- 14 entradas/salidas digitales, 6 de las cuales pueden ser usadas como entradas/salidas PWM.
- Posee 6 entradas analógicas.
- Los pines 0 y 1 pueden funcionar como transmisión/recepción RX y TX serial.
- Un oscilador de cristal de 16 MHz.
- Conector USB.
- Un jack de poder para alimentación.
- Un conector ICSP.
- Botón de Reset (ARDUINO, 2010).

1.1.3.3 Características Adicionales Importantes.

- Microcontrolador ATmega328.
- Voltaje de Alimentación. 5V.
- Voltaje de entrada recomendado 7 – 12 V.
- Límite de Voiltaje de Entrada 6 – 20 V.
- Pines I/O Digitales 14 (de los cuales 6 pueden proveer salidas PWM).
- Corriente DC por pin I/O 40mA.
- Corriente DC por pin 3.3V 50mA.
- Flash Memory 32 KB (de los cuales 0.5 KB son utilizados para arranque (bootloader)).
- SRAM 2KB.
- EPROM 1KB.
- Velocidad del reloj. 16Mhz (ARDUINO, 2010).

Figura 8. Placa electrónica Arduino UNO.



Fuente: (ARDUINO, 2010).

Como se ha visto, la placa electrónica utilizada cumple con la mayoría de los requerimientos para el desarrollo del proyecto, sin embargo, por sí sola no puede efectuar una comunicación Ethernet para enviar datos de consumo hacia una base de datos, por lo que los usuarios no podrían acceder a ellos a través una página web gestionada por un servidor, por ende, se requiere la utilización de un instrumento adicional conocido como Ethernet Shield, el cual se pasa a describir a continuación.

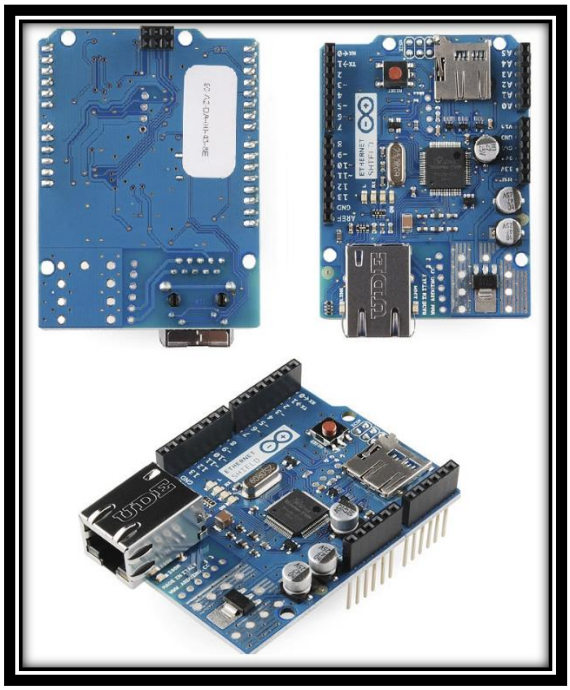
1.1.3.3.1 Ethernet Shield.

Como referencia, al hablar de una placa electrónica en general, un SHIELD es un aditamento que permite a la placa principal efectuar diversas funciones adicionales a las que originalmente desempeña, como es de suponerse, el Ethernet Shield permite a la placa electrónica Arduino UNO, realizar funciones de comunicación mediante protocolo Ethernet para de esta manera, poder enviar datos a una base de datos, y mediante un servidor poderlos exhibir a través de una página web (SHIELD, 2010).
El Ethernet Shield (Figura 9) está diseñado para encajar perfectamente sobre una placa Arduino UNO formando un conjunto sólido, junto con la ayuda de la librería proporcionada, descargable desde la misma página web de Arduino, se puede realizar tanto un pequeño servidor web, como un cliente. La configuración de red se realiza mediante software, a través de un cable RJ45 (Figura 10), por lo que se podrá adaptar con facilidad la placa a nuestra red local (SHIELD, 2010).

Entre sus características generales a tomar en cuenta se puede enumerar:

- Requiere una placa Arduino UNO.
- 5V Tensión de funcionamiento (provenientes de la alimentación de la placa).
- Controlador Ethernet: W5100 con buffer interno de 16 K.
- Velocidad de conexión: 10/100 Mb.
- Conexión con Arduino en el puerto SPI (SHIELD, 2010).

Figura 9. Modelo del Ethernet Shield para Arduino UNO.



Fuente: (SHIELD, 2010).

Figura 10. Cable RJ45 para conexión Ethernet.

Fuente: (SINCABLES.COM, 2011).

1.1.3.3.2 Descripción del Shield.

En pocas palabras, Arduino Ethernet Shield permite a una placa Arduino UNO conectarse a internet. Dicho escudo está basado en el chip de Ethernet Wiznet W5100 (Figura 11) cuyas características se enumera a continuación:

- Protocolos TCP / IP cableadas.
- TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE.
- Motor de hardware de red Un-atacable para la prevención de ataques de red tales como las inundaciones, la suplantación, la inyección.
- Host Interface: directa / indirecta BUS & Serial Peripheral Interface (SPI MODO 0,3).
- 4 socket hardware independiente.
- 16Kbytes memoria interna para el procesamiento de paquetes TCP / IP.
- 10BaseT / 100Base TX Ethernet PHY Embedded.
- Soporte Auto-Negociación (Full y Half Duplex, 10 y 100 Based).
- Soporte Auto-MDIX
- 3.3V Operación con la tolerancia de la señal de E / S 5V I.
- Salidas LED (TX, RX, full / half duplex, colisión, de enlace, velocidad).
- Paquete sin plomo 80LQFP (10x10mm)

A partir de sus características (WIZNET, 2013), el Wiznet W5100 ofrece una red (IP) capilar, capaz de efectuar comunicación TCP y UDP, y permite hasta cuatro conexiones de socket simultáneas.

Figura 11. Chip de Ethernet Wiznet W5100. En el que está basado el escudo.

Fuente: (WIZNET, 2013).

El escudo de Ethernet se conecta a la placa Arduino mediante largos encabezados por arrollamiento de hilo que se extienden a través del escudo, lo que permite que la disposición de pines permanezca intacta y a su vez permite, de ser necesario, ensamblar otro escudo en la parte superior (SHIELD, 2010).

El escudo Ethernet tiene una conexión estándar RJ-45, con un transformador de línea integrada y alimentación a través de Ethernet activado (SHIELD, 2010).

Posee también, una ranura para tarjetas micro-SD, que se puede utilizar para almacenar archivos para servir a través de la red compatible con el Arduino Uno, muy útil para respaldos como se analizará en capítulos posteriores. El lector de tarjetas microSD es funcional mediante la Biblioteca SD, también descargable gratuitamente desde la página oficial. Cuando se trabaja con esta biblioteca, SS es el Pin 4 (SHIELD, 2010).

El escudo también incluye un controlador de reajuste, para asegurarse de que el módulo Ethernet W5100 se restablece correctamente en el encendido

El escudo actual tiene un módulo de alimentación a través de Ethernet (PoE), diseñado para extraer energía de un cable Ethernet de par trenzado Categoría 5 convencional. El escudo no viene con el módulo PoE integrado, que es un componente separado que, de ser necesario, debe ser añadido a él (SHIELD, 2010).

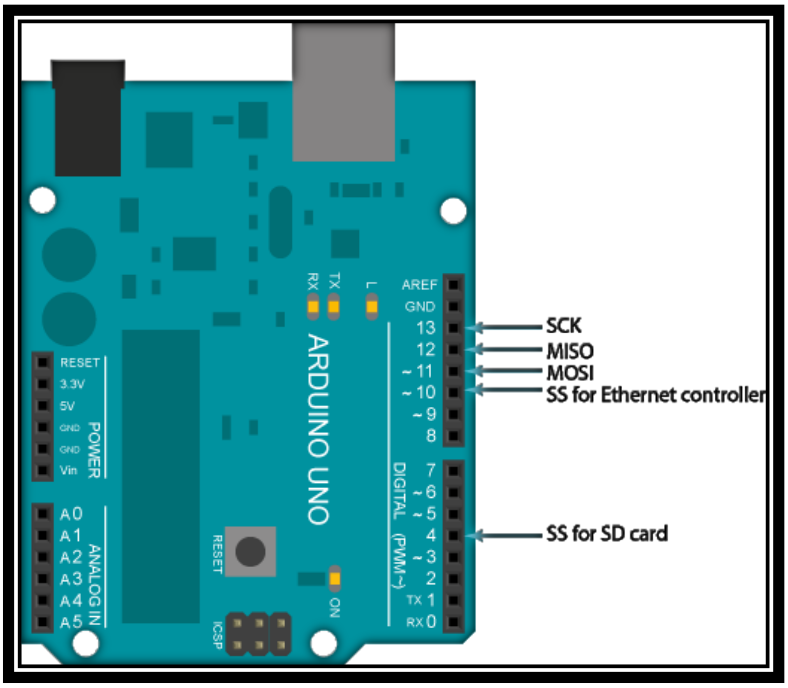
Arduino se comunica tanto con el W5100 y la tarjeta SD utilizando el bus SPI (a través de la cabecera ICSP). Esto es en los pines digitales 10, 11, 12, y 13 en el caso de Arduino UNO. El pin 10 se utiliza para seleccionar el W5100 y el pin 4 para la tarjeta SD. Estos pines son exclusivos, no se pueden utilizar para propósitos generales de entrada o salida (Figura 12).

Cabe mencionar que entre el chip W5100 y la tarjeta SD, sólo uno puede estar activo a la vez. Si está utilizando ambos periféricos, este debe ser atendido por las bibliotecas correspondientes para excluir a uno u otro o volverlos a utilizar dependiendo de la aplicación. Para hacer esto con la tarjeta SD, se debe configurar el pin 4 como salida y escribir un alto a la misma. Para el W5100, establezca pin digital 10 como una salida de alta. El botón de reinicio en el escudo restablece tanto el W5100 y la placa Arduino (SHIELD, 2010).

El escudo contiene un número de LEDs informativos que se enumera a continuación:

- PWR: indica que la placa y el escudo están energizados
- ENLACE: indica la presencia de un enlace de red y parpadea cuando el escudo transmite o recibe datos
- FULLD: indica que la conexión de red es full dúplex.
- 100M: indica la presencia de un Mb / s 100 conexión de red (en contraposición a 10 Mb / s)
- RX: parpadea cuando el escudo recibe datos
- TX: parpadea cuando el escudo envía datos.
- COLL: parpadea cuando se detectan colisiones de red.

Figura 12. Esquema de pines dedicado a la comunicación Ethernet en Arduino UNO.



Fuente: (SHIELD, 2010)

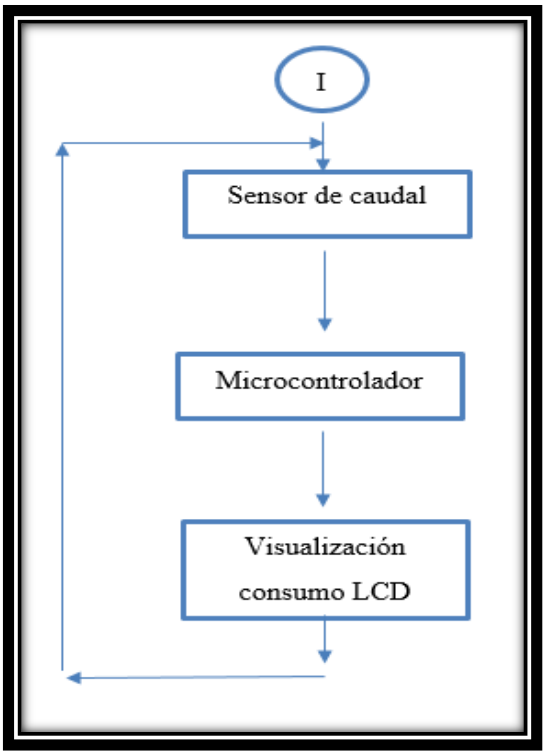
1.2 Medidor Electrónico de Consumo de Agua Potable.

A partir de los conceptos ya analizados, el proceso a seguir para la realización del medidor electrónico de consumo de agua potable es relativamente simple (Figura 13 y Figura 14).

El sensor de caudal es el elemento que captará la señal, es instalado directamente en la tubería de donde proviene el servicio de agua, cuando el líquido atraviese el sensor con mayor o menor intensidad de caudal, el sensor entregará una señal PWM con mayor o menor frecuencia al microcontrolador (Placa Arduino UNO), obviamente si no existe consumo, no existirá caudal y el sensor no entregará ninguna señal. Dicha señal irá a una entrada (pin 2) del microcontrolador, mediante firmware se efectuará la relación existente entre la señal entregada y la cantidad de líquido que se está consumiendo (relación señal/caudal), posteriormente los resultados se exhibirán en un indicador (pantalla lcd). Por otro lado, estos datos, también irán a una base de datos donde, por medio de software (Java Net Beans) se los exhibirá en una página web gestionada por un servidor. En este capítulo, solo se analizará la parte exclusiva del medidor físico, posteriormente en capítulos siguientes se verán las partes del

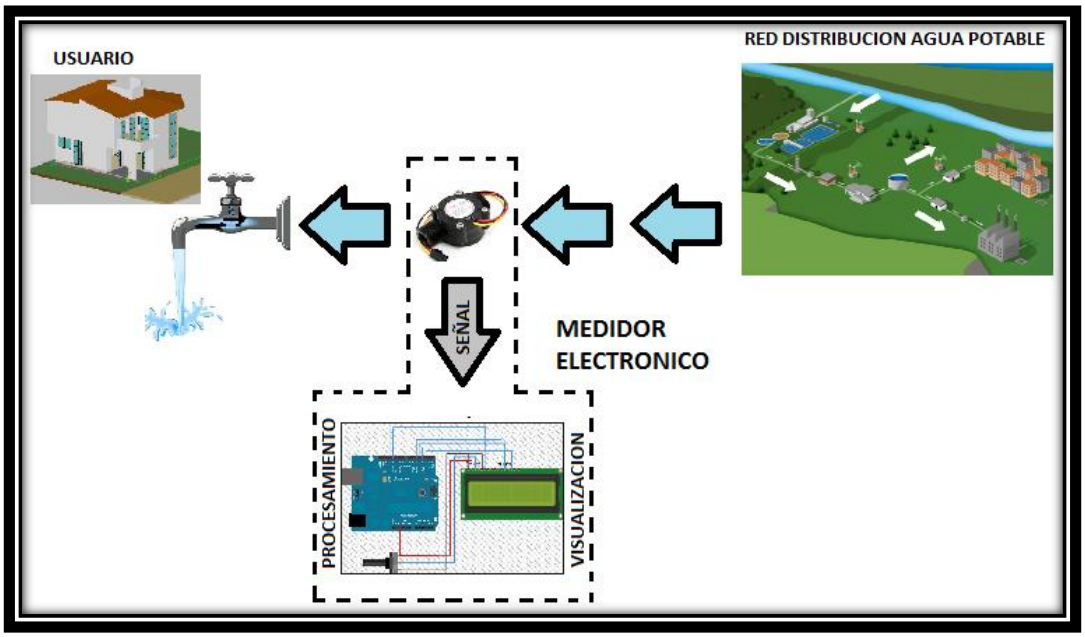
proyecto que tienen que ver con la comunicación, bases de datos, aplicaciones Android, páginas web y respaldos.

Figura 13. Diagrama de bloques del medidor de consumo electrónico de agua potable.



Fuente: Autor.

Figura 14. Esquema sencillo del modo de operación y montaje del medidor electrónico.



Fuente: Autor

1.2.1 Firmware del Microcontrolador.

Dentro de la programación en Arduino lo primero que se debe hacer es declarar las librerías que se van a utilizar, en este caso la *librería "SPI.h"* que es un protocolo de datos en serie síncrono utilizado por los microcontroladores para comunicarse con uno o más dispositivos periféricos rápidamente en distancias cortas, *"Ethernet.h"* que permite a la placa Arduino conectarse a internet mediante protocolo Ethernet., *"Arduino.h"* que permite trabajar con los elementos del entorno de programación (IDE) de Arduino, y *"LiquidCrystal.h"* para el funcionamiento del LCD (Figura 15).

Figura 15. Segmento de código correspondiente a la declaración de librerías.

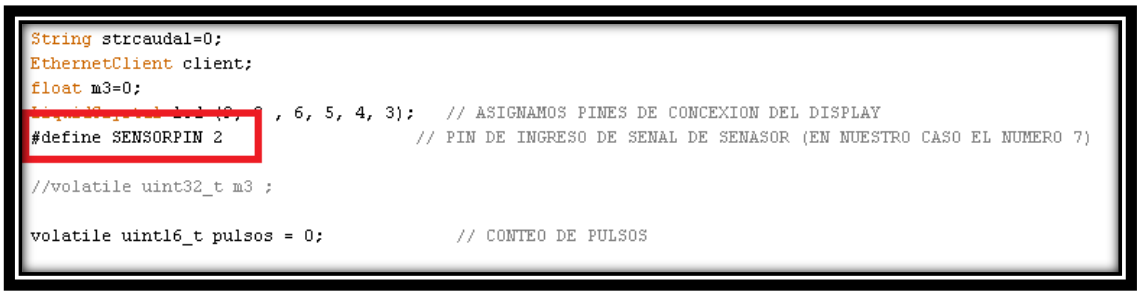
```
*************************************************************/
// INICIALIZAR LIBRERIAS

#include <SPI.h>                    // LIBRERIA SPI PARA TRABAJAR CON PERIFERICOS
#include <Ethernet.h>               // LIBRERIA ETHERNET PARA COMUNICACION Y CONEXION A INTERNET
#include <Arduino.h>                // LIBRERIA PARA TRABAJAR CON IDE ARDUINO
#include "LiquidCrystal.h"          // LIBRERIA DE DISPLAY
```

Fuente: Autor.

Después de esto, y muy importante, se debe definir el pin que se va a utilizar para recibir la señal del sensor, en este caso utilizaremos el pin número dos, definido dentro del microcontrolador con la sentencia *"#define SENSORPIN 2"* como se resalta en la Figura 16.

Figura 16. Segmento de código correspondiente a la declaración del pin para ingreso de la señal.

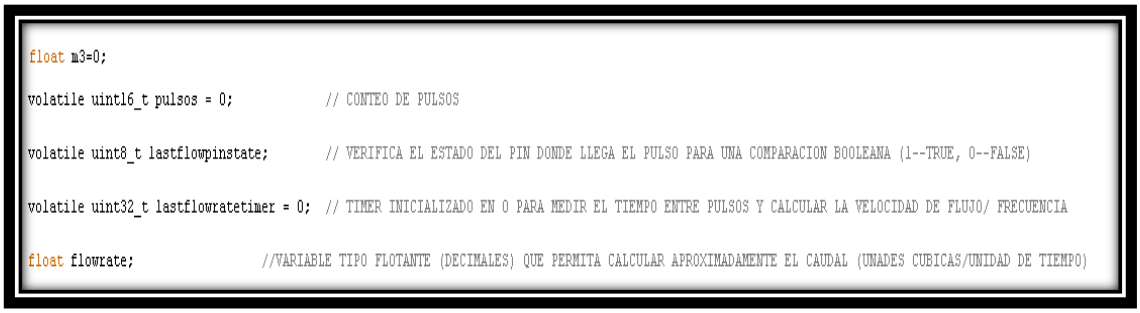


Fuente: Autor.

Después de esto, el procedimiento típico es declarar las variables a utilizar, a algunas de ellas se les añade el prefijo "volatile" ya que, al estar dentro de rutinas de interrupción, su valor tiende a cambiar. A su vez, dependiendo de la función de cada variable, están definidas ya sea como valores enteros solo positivos con capacidad de memoria para albergar un número que ocupe 8, 16 o 32 bits dependiendo del requerimiento (uint8_t, uint16_t, uint32_t) (VARIABLES, 2010) Las variables que se utilizan (Figura 17) puestas entre comillas se enumeran a continuación:

- *"pulsos"* Inicializado en cero y dedicado al conteo de pulsos.
- *"Lastflowpinstate*" verifica el ultimo estado (1 o 0) del pin donde llega los pulsos del sensor (pin 2) para una posterior comparación booleana (1--TRUE, 0--FALSE).
- *"Lastflowratetimer"* Nombre del timer, inicializado en 0, utilizado para, a través de interrupciones, obtener valores de pulsos y posteriormente calcular valores de flujo y frecuencia.
- *"Flowrate"* Variable que dedicada al cálculo del caudal
- *"m3"* Variable utilizada para la conversión de litros a metros cúbicos.

Figura 17. Segmento de código de la declaración de variables.

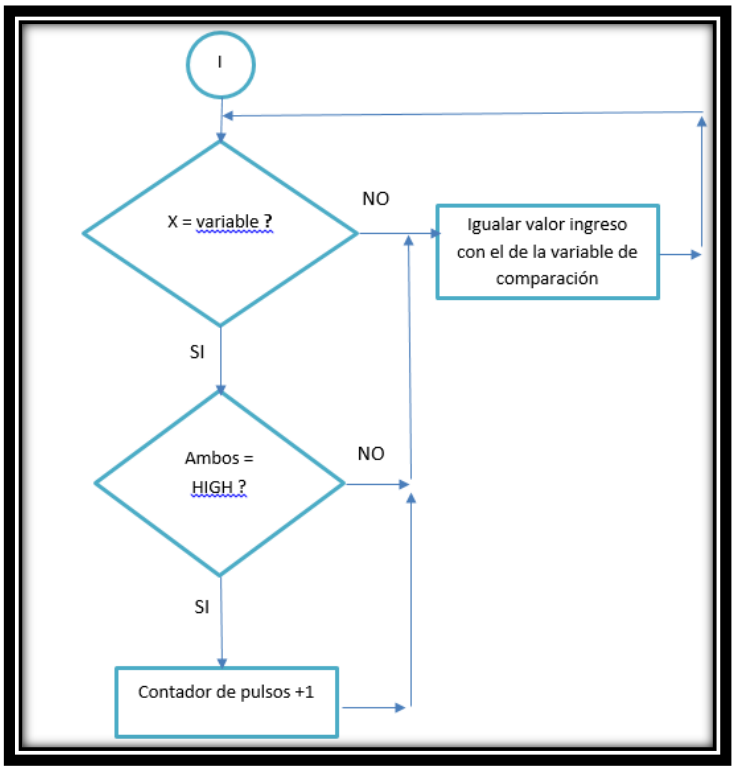


```
float m3=0;
volatile uint16_t pulsos = 0;            // CONTEO DE PULSOS
volatile uint8_t lastflowpinstate;       // VERIFICA EL ESTADO DEL PIN DONDE LLEGA EL PULSO PARA UNA COMPARACION BOOLEANA (1--TRUE, 0--FALSE)
volatile uint32_t lastflowratetimer = 0; // TIMER INICIALIZADO EN 0 PARA MEDIR EL TIEMPO ENTRE PULSOS Y CALCULAR LA VELOCIDAD DE FLUJO/ FRECUENCIA
float flowrate;                 //VARIABLE TIPO FLOTANTE (DECIMALES) QUE PERMITA CALCULAR APROXIMADAMENTE EL CAUDAL (UNADES CUBICAS/UNIDAD DE TIEMPO)
```

Fuente: Autor.

Una vez declarado el pin para la entrada de la señal, establecido variables, y exportado librerías, lo que viene es la generación de la rutina de interrupción (SIGNAL(TIMER0_COMPA_vect)), llamada una vez cada milisegundo; a continuación se efectúa una comparación continua del valor que ingresa al pin de la señal y el valor de la variable *Lastflowpinstate,* cabe resaltar que ambos son valores booleanos HIGH o LOW o 1 y 0 pues, al ser una señal cuadrada de PWM solo puede haber los dos valores ya mencionados, igual es el caso de la variable *Lastflowpinstate.* Si los valores no son iguales, se iguala el valor de la variable al del sensor y se vuelve a comparar; por otro lado, si ambos valores son iguales pueden existir dos casos, que ambos sean HIGH o que ambos sean LOW, si ambos valores son de nivel lógico LOW, solamente se iguala el valor de la variable al del sensor y se vuelve a comparar, por el contrario, si ambos valores son altos (HIGH), quiere decir que existe un pulso, por lo tanto se incrementará el valor de la variable pulsos en 1, y después se volverá a igualar el valor de la variable y a efectuar de nuevo una comparación, de esta manera se evita contar doblemente un mismo pulso si ambos coinciden en un valor sea alto o bajo. Puesto que en cada comparación el timer se incrementa su valor en uno las comparaciones se efectúan cada milisegundo, se divide el valor del timer entre mil para obtener los valores de frecuencia de señal en Hertz (Figura 18 y Figura 19).

Figura 18. Diagrama de bloques del método de obtención de los pulsos de la señal de ingreso.



Fuente: Autor.

Figura 19. Segmento de código de la comparación entre variable e ingreso de señal, incremento de contadores y cálculo de frecuencia.

```
SIGNAL(TIMER0_COMPA_vect) {
  uint8_t x = digitalRead(SENSORPIN);      //LEEMOS EL VALOR EN EL PIN DEL CAUDAL AL QUE ESTA C

  if (x == lastflowpinstate) {               // COMPARAMOS EL VALOR DEL PIN DEL SENSOR CON EL
    lastflowratetimer++;                     // ASI QUE INCREMENTAMOS EL TIMER EN 1 (para lueg
    return; // nada cambia
  }

  if (x == HIGH) {                            // SI EL VALOR DEL PIN DEL SENSOR ES ALTO (HIGH)
    pulsos++;                                 // NOTA: SOLO SE TOMA EN CUENTA EL ESTADO LOGICO
  }
  lastflowpinstate = x;
  flowrate = 1000.0;                  // para calcular el caudal en unidades (hertz)
  flowrate /= lastflowratetimer; // en hertz
  lastflowratetimer = 0;         // se vuelve a cero el timer para iniciar de nuevo el ciclo
}
```

Fuente: Autor.

Una vez obtenido el número de pulsos, solo queda establecer la relación entre el número de pulsos vs el consumo en litros, dicha relación está dada por el fabricante del sensor y es:

Litros= Numero de pulsos / (7.5 * 60)

Si se requiere el valor en metros cúbicos, se debe dividir el valor de litros entre mil, los datos de consumo se irán almacenando en las respectivas variables "litros" y "m3" descritas anteriormente (Figura 20).

Figura 20. Segmento de código de la conversión de pulsos en su equivalente en litros y metros cúbicos.

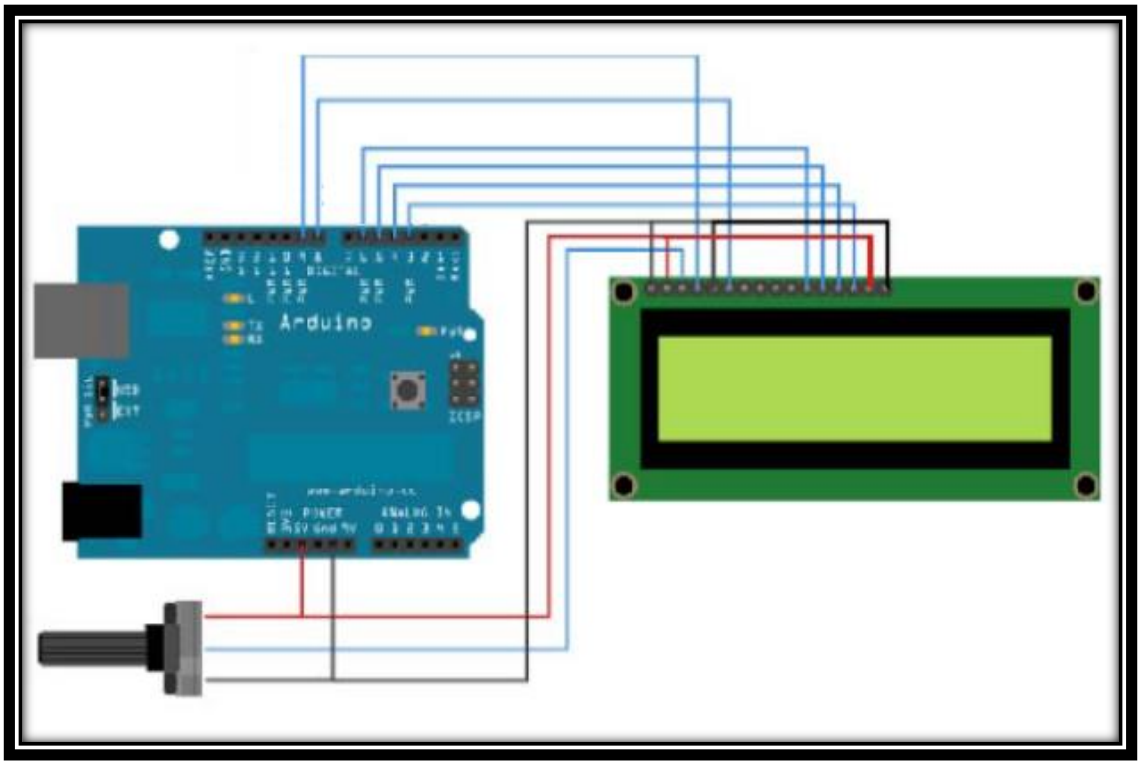
```
// POER CATALOGO, SI ES UN SENSOR DE CAUDAL PLASTICO UTILIZAMOS LAS SIGUIENTES RELACIONES:
// FRECUENCIA SENSOR (HZ) = 7.5 * Q (LITROS/MIN)---------- POR CONSIGUIENTE Q = FRECUENCIA SE
// LITROS = Q * TIEMPO TRANSCURRIDO (SEGUNDOS)/ 60 (SEGUNDOS -- MINUTO)
// LITROS = (FRECUENCIA(PULSOS/SEGUNDO)/ 7.5)* (TIEMPO TRANSCURRIDO (SEGUNDOS)/60 )
// LITROS = PULSOS / (7.5 * 60)
float litros = pulsos;
litros /= 7.5;
litros /= 60.0;
m3=litros/1000;
```

Fuente: Autor

1.2.2 Conexión y Exhibición de datos en Display.

Para una comunicación satisfactoria con el display LCD de cristal líquido, a más de exportar su librería se debe declarar al inicio del programa los pines destinados a la conexión del mismo con la Arduino, puesto que el lcd que se utiliza es de 2 x 16, se destinan los pines 3,4,5,6,8 y 9. Y su declaración y conexión correspondiente con la placa se aprecia en la Figura 21 y Figura 22 (LCD, 2010).

Figura 21. Esquema de la conexión del LCD con la placa Arduino.



Fuente: (LCD, 2010)

Figura 22. Segmento de código para asignar los pines correspondientes al LCD.

```
LiquidCrystal lcd (9, 8 , 6, 5, 4, 3);   // PINES DE CONCEXION DEL DISPLAY
```

Fuente: Autor.

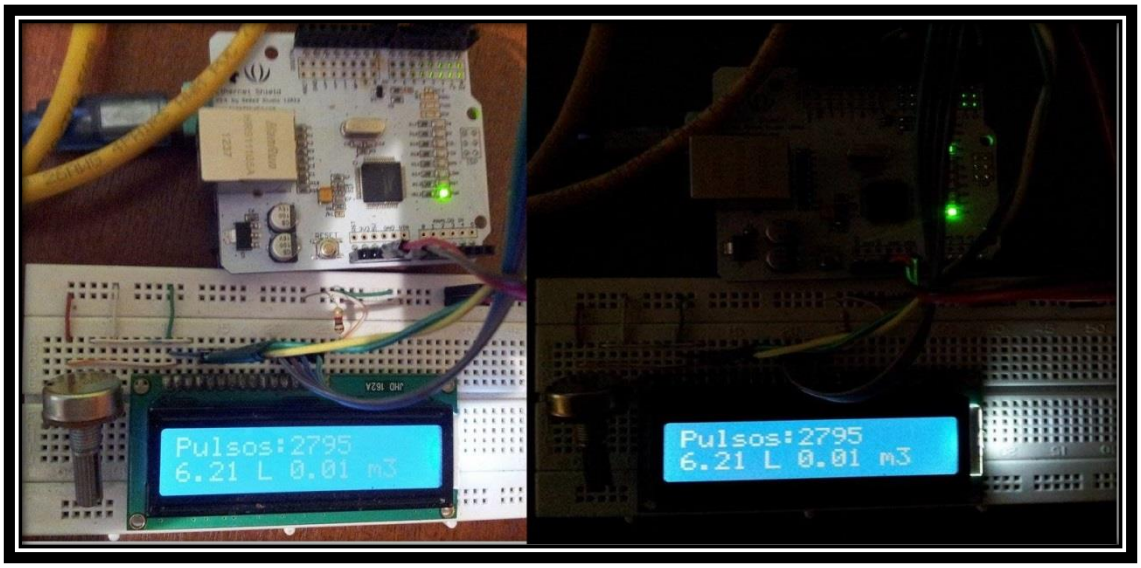
A su vez para enviar datos y poder visualizarlos en el lcd. Es necesario primero direccionar el puntero hacia el sitio específico de la fila y columna donde se quiere comenzar a escribir, esto se logra mediante la sentencia lcd.setCursor(c,f); en lugar de “f”se pone el número de fila (0,1) y en lugar de “c” se ubica el número de columna (de 0 a 15). A continuación en las Figuras 23 y 24 se muestra el segmento de código para escribir en el lcd tanto la variable litros como la variable m3 que son los que se requieren apreciar por el usuario junto con su visualización real.

Figura 23. Segmento de código para escribir en el lcd tanto la variable litros como la variable m3.

```
lcd.setCursor(0, 1);
lcd.print(litros); lcd.print(" L ");
lcd.print(m3); lcd.print(" m3 ");
```

Fuente: Autor

Figura 24. Vistas previas del medidor electrónico funcional.



Fuente: Autor.

CAPITULO 2

CREACION Y ESTRUCTURA DE BASE DE DATOS

2.1 Conceptos Preliminares.

2.1.1 Base de Datos.

Se puede definir una base de datos como una "bodega" que permite almacenar gran cantidad de información relacionada entre sí, de manera ordenada y organizada, de tal forma que pueda ser encontrada y utilizada posteriormente con relativa facilidad, mediante un programa o programas que manejen dicha cantidad de información.
Generalmente una base de datos está compuesta de una o más tablas que guarda un conjunto de datos. Cada tabla está conformada por una o más columnas **y** filas. Las columnas guardan una parte de la información sobre cada elemento que se quiera guardar en la tabla, cada fila de la tabla conforma un registro de cada una de las columnas (VALDES, 2007).

2.1.2 Características.

Entre los principales atributos que poseen las bases de datos se pueden enumerar:

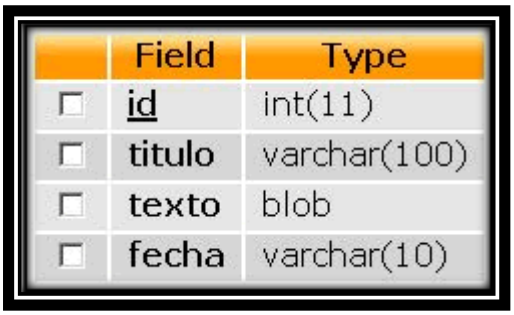
- Independencia lógica y física de la información.
- Mínima redundancia (repetición de datos).
- Acceso periódico por parte de varios usuarios.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación en caso de pérdida.
- Acceso a través de lenguajes de programación estándar (programas).

2.1.3 Estructura de una Base de Datos

Con el propósito obvio de almacenar información de la manera más organizada y lógica posible una base de datos obedece a un orden que debe ser cumplido para tener acceso a la información de manera coherente. Como ya se mencionó, cada base de datos se conforma por tablas, que cumplen la función de contener los campos a partir de los cuales, se alberga ordenadamente la información (VALDES, 2007).

En el siguiente ejemplo a través de la Figura 25 se muestra una tabla que contiene 4 campos.

Figura 25. Campos de información.



	Field	Type
☐	id	int(11)
☐	titulo	varchar(100)
☐	texto	blob
☐	fecha	varchar(10)

Fuente: (VALDES, 2007).

Los datos quedarían organizados como se muestra a continuación en la Figura 26:

Figura 26. Tabla generada a partir de los campos declarados.

←T→	id	titulo	texto	fecha
☐ ✎ ✕	1	saludos	[BLOB - 0 B]	22-10-2007
☐ ✎ ✕	2	como estas ???	[BLOB - 0 B]	23-10-2007

Fuente: (VALDES, 2007)

2.1.4 Tipos de Campos

Cada Sistema de Base de Datos posee tipos de campos que si bien es cierto son propios de cada uno con sus respectivas diferencias, no dejan de tener ciertas similitudes. (VALDES, 2007) Entre los más comunes se pueden nombrar:

- **Numérico:** Se refieren en concreto a enteros "sin decimales" y reales "decimales".
- **Booleanos:** Conocido como variables de estados: Verdadero "Si" y Falso "No".
- **Memos:** son campos alfanuméricos (letras y números) de longitud prácticamente ilimitada pero con el problema de no poder ser indexados.
- **Fechas:** Como es de suponerse, almacena fechas lo que posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.
- **Alfanuméricos:** contienen tanto números como letras. Presentan una longitud limitada (255 caracteres).
- **Autoincrementables:** son campos de números enteros que incrementan su valor en uno para cada registro que se va incorporando al proceso, utilizado generalmente como identificador de un registro.

Por consiguiente una base de datos posee el siguiente orden jerárquico:

- Tablas
- Campos
- Registros
- Lenguaje SQL

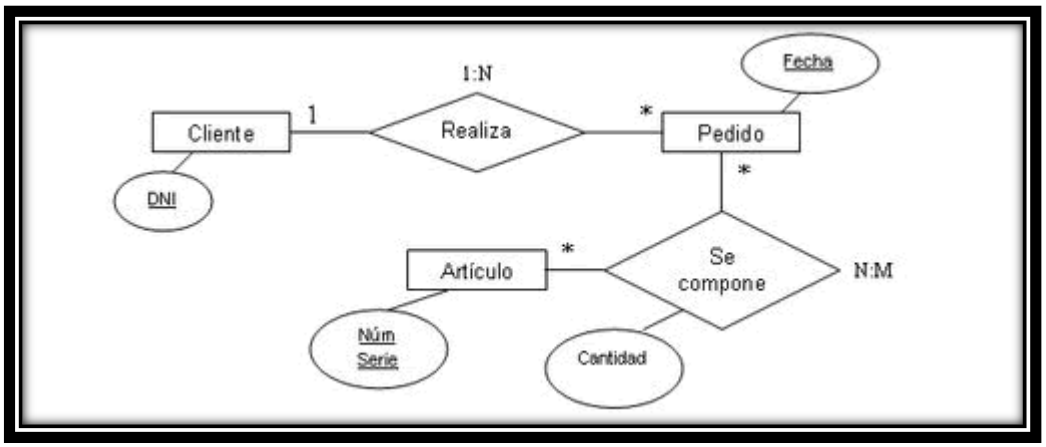
El lenguaje SQL es el lenguaje por excelencia utilizado en los sistemas de base de datos, permite consultar, mostrar, insertar, actualizar y borrar, etc. a nuestras bases de

datos, en resumen manipular la información de una forma muy completa (VALDES, 2007).

2.1.5 Modelo entidad-relación.

Los diagramas relacionales o modelos entidad-relación, (siglas en inglés: ERD "Diagram Entity relationship") son una herramienta que permite modelar los distintos tipos de datos para su organización, tanto su lugar como el orden que ocupan junto con la función que cumplen; para conformar, en un todo, un sistema completo de información. Estos modelos expresan entidades importantes para un sistema de información, sus inter-relaciones y propiedades individuales (Figura 27) (VALDES, 2007) .

Figura 27. Ejemplo de un Diagrama Entidad Relación.



Fuente: (VALDES, 2007)

2.1.6 Cardinalidad de las Relaciones.

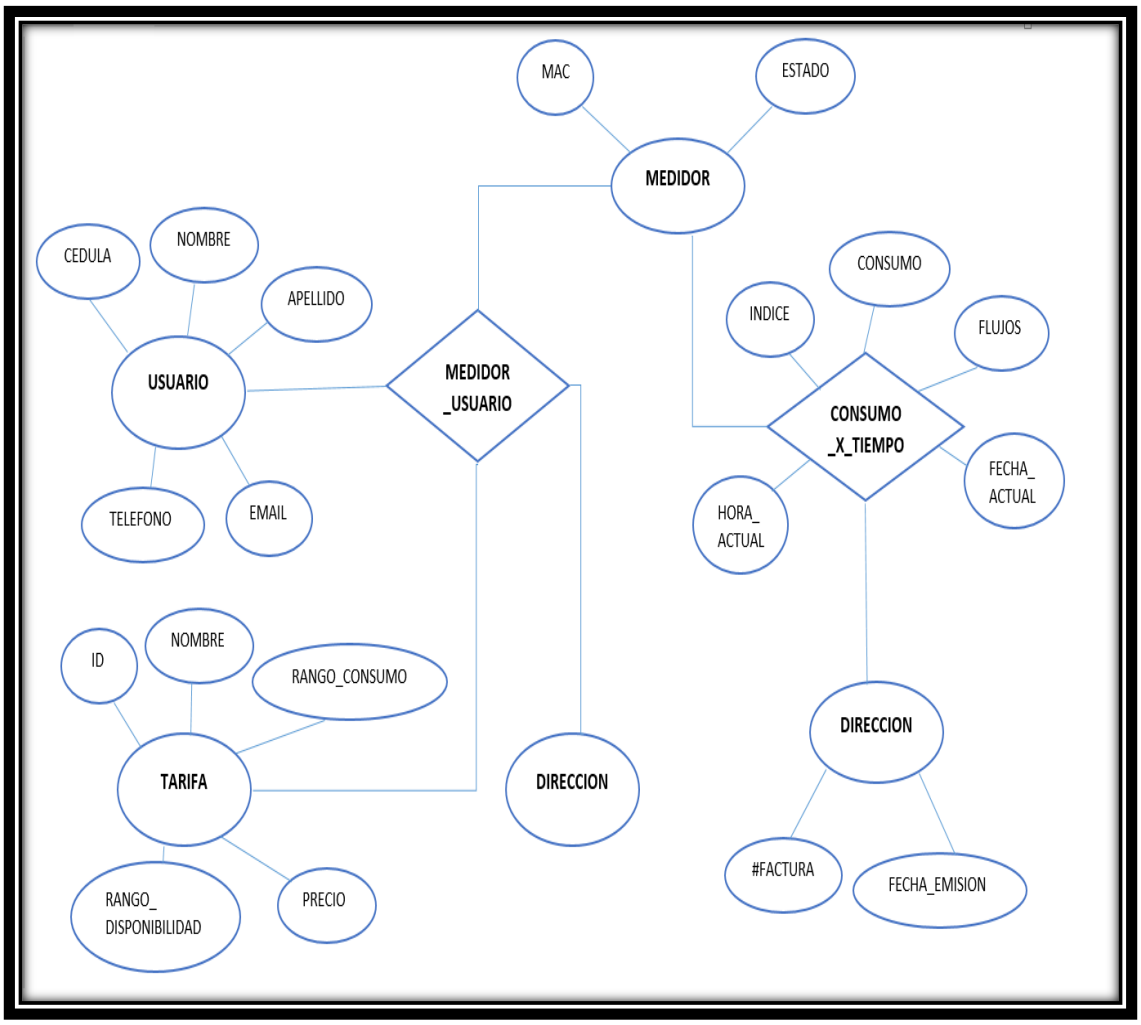
El diseño de relaciones entre las tablas de una base de datos puede ser la siguiente (VALDES, 2007):

- **Relaciones de uno a uno:** una instancia de la entidad A se relaciona con una y solamente una de la entidad B.
- **Relaciones de uno a muchos:** cada instancia de la entidad A se relaciona con varias instancias de la entidad B.

- **Relaciones de muchos a muchos:** cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B.

En cuanto a la organización de los datos en el desarrollo del presente tema de tesis, en la Figura 28, mostrada a continuación se puede apreciar el diagrama entidad relación.

Figura 28. Diagrama Entidad Relación del Proyecto.



Fuente: Autor

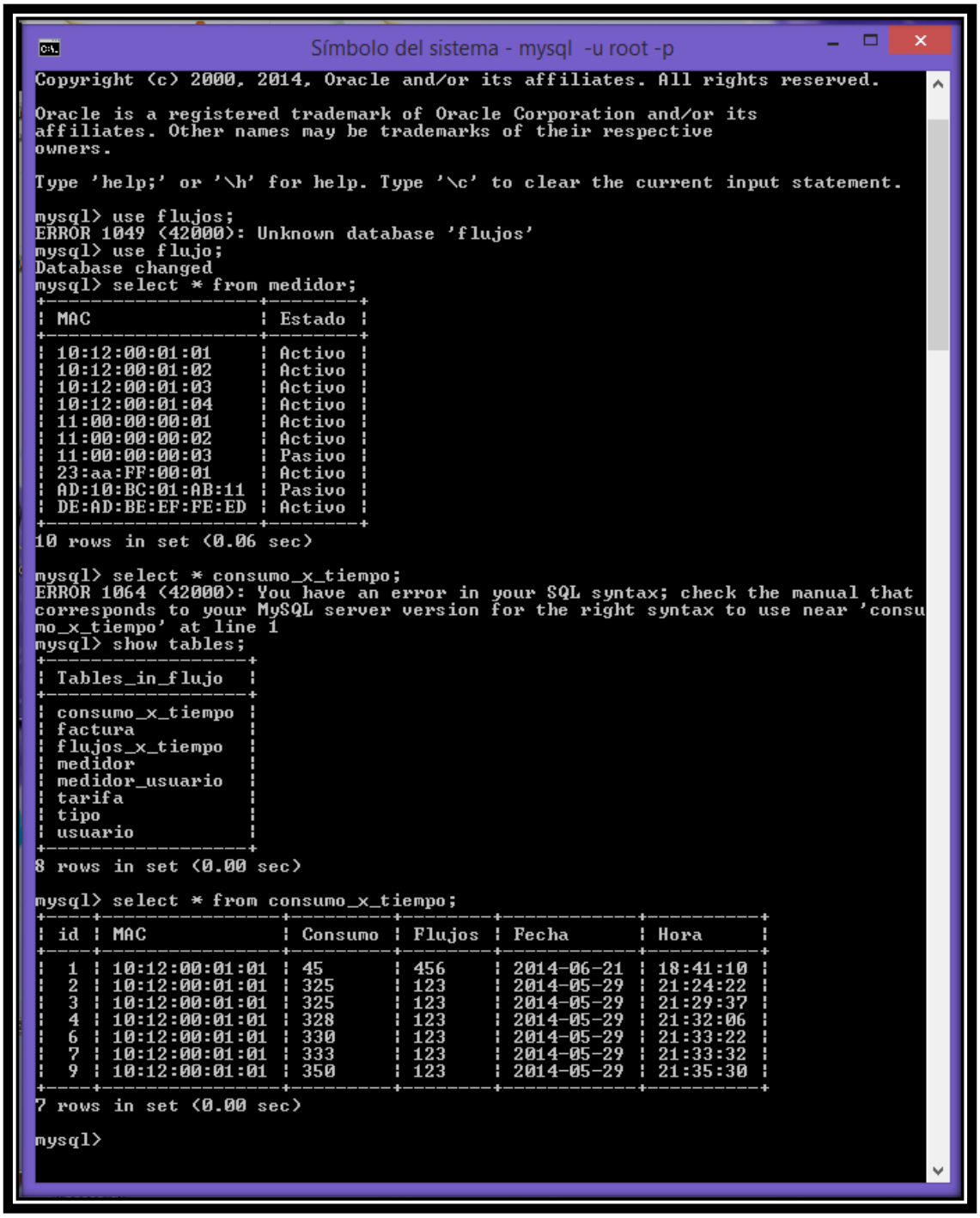
A pesar de contener un considerable número tablas relacionadas entre sí, una vez entendiendo la mecánica del cómo funciona el proyecto, viene siendo relativamente muy sencillo entender el diagrama antes expuesto.

Partiendo del medidor electrónico, se generan los datos de consumo, número de mac, y estado del equipo, que se registra y genera en dicho medidor, los datos de estado y

número de mac, son almacenados en una tabla llamada **MEDIDOR,** por otro lado, los datos de consumo se registran en una tabla denominada **CONSUMO,** que junto con las tablas INDICE, FLUJOS, FECHA_ACTUAL y HORA_ACTUAL, que, generadas por las aplicaciones a partir del consumo, conforman la tabla **CONSUMO_X_TIEMPO,** utilizada para emisión de facturas mediante una aplicación que se verá más adelante ; simultáneamente, utilizando cada número de mac, se asigna un medidor a un usuario mediante otra aplicación, también analizada posteriormente; de ahí que se requiera una tabla de nombre **USUARIO** que, lógicamente, contenga todos los datos pertinentes a dicho usuario, esto es: nombre, apellido, número de cédula, teléfono, e-mail. A su vez, basándose en los datos de consumo y tiempo de la tabla ya mencionada, se utilizan los valores de tarifas de consumo de agua potable (tomados de la página oficial de ETAPA) para generar las facturas, de ahí que sea necesario dentro del grupo **MEDIDOR_USUARIO**, generar una tabla de datos llamada **TARIFA**, lógicamente compuesta de datos correspondientes a: RANGO_ DE_CONSUMO, PRECIO, y NOMBRE, correspondientes, a su vez, al tipo de usuario, ya sea, comercial, residencial, especial etc; pues según la normativa, existen diferentes tarifas para cada uno de los usuarios del servicio.

Como se puede ver, y en resumen, partiendo de los datos de consumo generados por el medidor, se van almacenando ordenadamente los diferentes tipos de datos que se necesitan para asignar usuarios, facturas y valores de consumo en base a las tarifas al momento de emitir facturas, utilizando de forma eficiente la información por medio de múltiples tablas (Figura 29) a las que las aplicaciones irán accediendo o manipulando dependiendo del requerimiento particular de cada aplicación, para convertirlas en otras tablas con diferente información, que, una vez más, son manejadas mediante aplicaciones de software que posteriormente se analizarán con mayor profundidad, que, en un todo, conforman un sistema eficiente de control de servicio de agua potable, disponiendo de los datos requeridos de consumo y tarifas, ya sea por parte del consumidor o de la empresa.

Figura 29. Tablas pertenecientes a la base de datos del proyecto.



Fuente: Autor.

2.2 MySQL y Base de Datos.

MySQL (Figura 30) es un software dedicado a gestión de bases de datos relacional, creada por la empresa sueca MySQL AB, poseedora del código fuente del servidor

SQL como de la marca, bajo la licencia más común en el mundo del software, la GPL de la GNU, MySQL es un software de código abierto, aunque la empresa que la generó oferta también una versión comercial que tiene como ventaja soporte técnico y la posibilidad de integrar el programa a un propietario, lo que brinda seguridades adicionales que en muchos casos son requeridas.

El lenguaje en el que MySQL está basado es el que se le conoce con el nombre de Structured Query Language, por sus siglas en ingles SQL, creado por IBM en 1981; que, por sus características para accesar o gestionar bases de datos relacional, es ampliamente utilizado, no solo por este gestor sino por otros programas también dedicados al manejo de datos (TOLEDO, 2010).

Figura 30. Logo de MySQL y productos asociados.

Fuente: (JONES, 2010)

2.2.1 Características principales.

Una de las razones por las que MySQL atrae la atención de los desarrolladores y programadores, principalmente de páginas web, fue de hecho, su simplicidad, hasta el punto que las bondades faltantes del programa fueron complementados por las aplicaciones desarrolladas donde se utiliza. Paulatinamente, y basándose en experiencias reales de los programadores en sus respectivos campos de acción, y al ser un software de código abierto, los elementos faltantes, van siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre

(TOLEDO, 2010). En las últimas versiones se pueden destacar las siguientes características principales:

- Velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Versatilidad, se puede trabajar en distintas plataformas y sistemas operativos.
- Cada vez que se genera una base de datos, esta, implícitamente se crea o contiene 3 tipos de archivos: De estructura, uno datos y uno de índice.
- Soporta hasta 32 índices por tabla.
- Multihilo, debido a esta característica, soporta sistemas multiproceso.
- Provee un aceptable nivel de seguridad a través de contraseñas (passwords), y gestión de usuarios.
- El servidor puede manejar distintos idiomas (TOLEDO, 2010).

2.2.2 Ventajas.

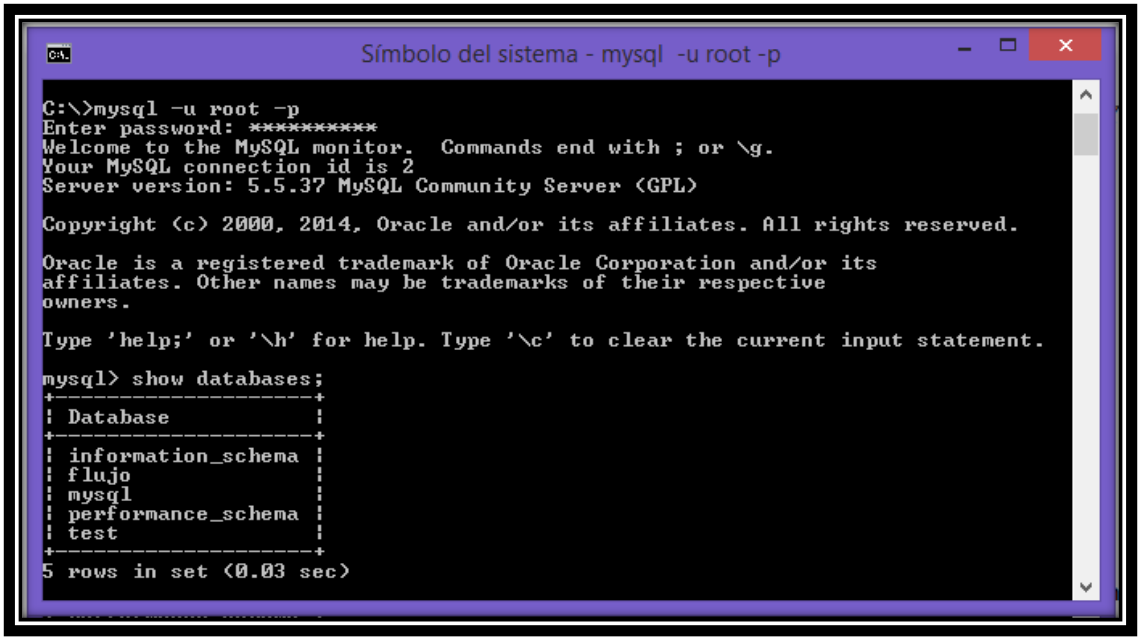
- Velocidad de realización de operaciones, lo que se traduce en un mejor rendimiento.
- Requerimientos sencillos al momento de elaborar bases de datos, gracias a su bajo consumo puede ser ejecutado en un ordenador sin ningún problema a pesar de que dicho ordenador no cuente con características demasiado exquisitas en lo que respecta a la memoria y/o hardware.
- Configuración e instalación relativamente sencillas.
- Puede ejecutarse sobre una gran variedad de Sistemas Operativos
- Baja probabilidad de corrupción de datos. Independientemente del gestor o la red.
- Relativamente buena conectividad y seguridad para un medio gratuito.
- La licencia con la que cuenta este programa (GPL) lo hace, prácticamente, una aplicación universal (TOLEDO, 2010).

2.2.3 Desventajas.

- Una gran cantidad de utilidades de MySQL no están al servicio de todos los usuarios, ya sea la utilidad como tal, o información sobre la misma.
- No es intuitivo.

Una vez analizado las características de este servidor, es posible percibir su idoneidad para la aplicación en el presente tema de tesis, aplicación en la que fueron almacenadas las tablas de datos con las que cuenta el proyecto (Figura 31).

Figura 31. Bases de datos albergados en MySQL del proyecto.



Fuente: Autor.

2.3 Sistema de Gestión de Base de Datos (SGBD).

Estos sistemas, (en inglés DataBase Management System) son un tipo de software específico, utilizado como mediador (interfaz) entre la base de datos, el usuario y las aplicaciones que la utilizan (programas y aplicaciones). Esta elaborado a base de lenguajes de programación, y generalmente se utiliza obedeciendo a sentencias propias del lenguaje de programación en el que está basado el software (lenguaje de definición de datos, lenguaje de manipulación de datos y de un lenguaje de consulta) (AVILA, 2013).

2.3.1 Ventajas y Desventajas de la Gestión de Bases de Datos

Ventajas.

- Control sobre Redundancia: Los sistemas de almacenamiento (ficheros) guardan varias copias de la misma información en diferentes ficheros. Esto hace que se desperdicie memoria, y provoca inconsistencia de datos. En los sistemas de bases de datos, aunque no se puede eliminar del todo la redundancia, puesto que los ficheros están integrados, no se almacenan varias copias de la misma información.
- Consistencia: Gracias al control sobre la redundancia, existe poco riesgo de inconsistencia en los datos, sea que cierto dato se haya almacenado una o de ser necesario más veces, el sistema se actualizará una sola vez, o el mismo sistema garantiza que cada copia se mantenga consistente.
- Compartición: Una base de datos, generalmente perteneciente a una empresa puede ser compartida por todos los usuarios autorizados de dicha empresa, dichos usuarios tendrían acceso a ella incluso con diferentes rangos de privilegios de manipulación sobre la misma.
- Mantenimiento de estándares: Cada empresa que utiliza bases de datos posee sus particulares estándares para manipular información, lo que permite facilidad de intercambio de datos, actualización o acceso, gracias a la estructura de dichas bases, resulta más sencillo regirse a dichos estándares tanto a nivel de empresa como a nivel nacional e internacional.
- Integridad de datos: La integridad tiene que ver con la validez y consistencia de la información almacenada, expresada a través de parámetros y restricciones que no se pueden transigir, aplicadas tanto a los datos, como a sus relaciones, y es su respectivo sistema de gestión (SGBD) quien se debe encargar de mantenerlas.
- Seguridad: Provee alternativas de seguridad para proteger a la información de usuarios no autorizados.
- Accesibilidad: Los sistemas de gestión (SGBD) proporcionan lenguajes que permiten consultar o generan informes e incluso aplicaciones que realizan este tipo de tareas.

- Productividad: Los sistemas de gestión (SGBD) permiten disponer de estas funciones y rutinas que permiten al usuario centrarse mejor en una tarea específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación.
- Mantenimiento: A través de los sistemas de gestión (SGBD) se establece una separación entre los datos y las aplicaciones que los controlan. Conocida como independencia de datos, esta cualidad permite facilitar el mantenimiento de las aplicaciones que acceden a la base de datos.
- Concurrencia: La mayoría de los sistemas de gestión (SGBD) controlan el acceso concurrente a la base de datos y garantizan que no existan problemas si dos usuarios quieren acceder a la misma información relativamente al mismo tiempo.
- Copias de seguridad: Muchos sistemas de ficheros es el usuario quien tiene que realizar copias de seguridad generalmente una vez por día par que, en caso de algún fallo, utilizar estas copias para restaurarlos, lo que provoca pérdida de trabajo realizado desde que se efectuó la última copia. Por otro lado, mediante la utilización de sistemas de gestión (SGBD) se minimiza la cantidad de trabajo perdido cuando se produce un fallo. (AVILA, 2013)

Desventajas.

- Relativa Complejidad: Los gestores o sistemas de gestión de bases de datos (SGBD) son programas con una gran funcionalidad pero que, de hecho, pueden llegar a tener un grado relativamente alto de complejidad. Es preciso dominar su estructura para poder realizar un buen uso de estos.
- Coste del equipamiento adicional: Tanto para el SGBD, como para la misma base de datos, pueden ser necesario adquirir más espacio de almacenamiento, e inclusive un ordenador o maquina más grande y que este únicamente dedicado a la gestión de datos. Todo esto elevara el costo de la implantación de sistemas de bases de datos.
- Vulnerable a los fallos: El hecho de que los datos por concepto se manejen con gestores (SGBD) hace que el sistema se vuelva vulnerable y sea imperativo poseer y actualizar con relativa periodicidad copias de seguridad (Backup) (OKAMURA, 2012).

2.3.2 Gestores de Base de Datos

Es una o varias herramientas de software que permiten controlar, crear, modificar, y en términos generales, manipular una base de datos o parte de su contenido. En el presente proyecto, se utilizará el programa "NetBeans", como la herramienta que efectuará la gestión en lo que respecta a la base de datos, misma herramienta que se analizará a continuación (OKAMURA, 2012).

2.4 Java y NetBeans

Java es un entorno de desarrollo (lenguaje de programación) orientado a objetos, desarrollado por la empresa de software Sun MicroSystems, el cual toma como referencia la escritura de código fuente de C y C++ para incorporar un modelo de objetos más simples y donde se eliminen las herramientas de bajo nivel que induce errores (PENARRIETA, 2011).

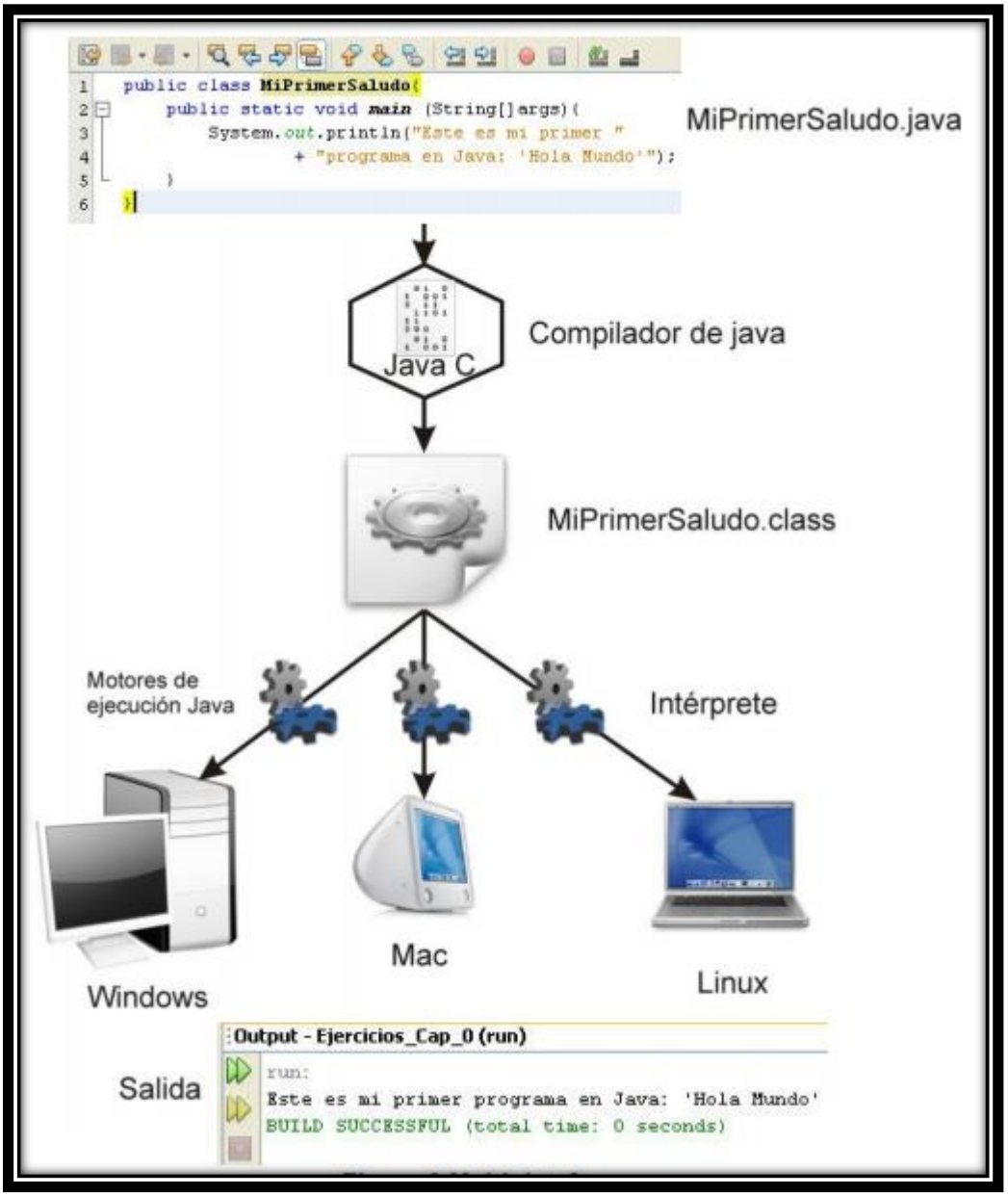
Cuando se habla de Programación Orientada a Objetos (POO) nos referimos a un entorno que permite desarrollar aplicaciones de forma que se simplifique el código dividiéndolo en objetos, permitiendo centrarse en cada objeto, de esa manera se elimina la complejidad de escribir código demasiado extenso (PENARRIETA, 2011).

Desde su creación en 1991, Java ha experimentado una enorme evolución hasta nuestros días, llegando a poseer características importantes como las siguientes:

- **Orientado a objetos.** Planteando el paradigma de "objetos" como piezas reutilizables en proyectos distintos, se hace posible desarrollar aplicaciones de gran complejidad reduciendo drásticamente códigos largos y características de lenguajes de bajo nivel (punteros) reduciendo drásticamente los tiempos de desarrollo.
- **Simple.** Reduciendo hasta en un 50% la cantidad de código.
- **Distribuido.** Java proporciona librerías y herramientas para que los programas sean distribuidos, es decir, para que se ejecuten en varias máquinas e interactuando entre ellas.

- **Multiplataforma.** Programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware (ordenador o maquina) independientemente de la marca o sistema operativo. (Figura 32) (PENARRIETA, 2011).

Figura 32. Estructura del proceso que permite a Java ser Multiplataforma.



Fuente: (PENARRIETA, 2011).

- **Robustez y Recolección de basura**. Java realiza verificaciones periódicas al momento de la compilación y ejecución del programa, lo que ayuda a detectar errores de ciclo de desarrollo a más de verificar referencias innecesarias a objetos y eliminarlos (Recolección de basura).
- **Seguro.** Tanto en el lenguaje, ejecución del programa y código fuente, Java somete a la aplicación a una serie de test que brinden seguridad tanto para el equipo como para el programa en sí.
- **Portable.** Java construye sus interfaces de manera que se puedan utilizar en entornos Unix, Mac o Windows
- **Lenguaje Interpretado.** Su interprete (JDK) puede ejecutarse en cualquier plataforma
- **Multihilo (multithreaded).** Lo que posibilita ejecutar diversas acciones simultáneas en una aplicación, los hilos son procesos independientes dentro de un gran proceso; al ser multihilo, se tiene mejor rendimiento interactivo y mejor desempeño de la aplicación en tiempo real.
- **Lenguaje Dinámico.** Java no intenta conectar simultáneamente todos los módulos de los q consta una aplicación hasta el mismo tiempo de ejecución. También simplifica el uso de protocolos nuevos o actualizados. Java es capaz de "importar" automáticamente cualquier pieza que el sistema necesite para funcionar y, para evitar "importar" cada vez, java implementa las opciones de persistencia para que no se eliminen cuando se limpie el caché de la máquina (PENARRIETA, 2011).

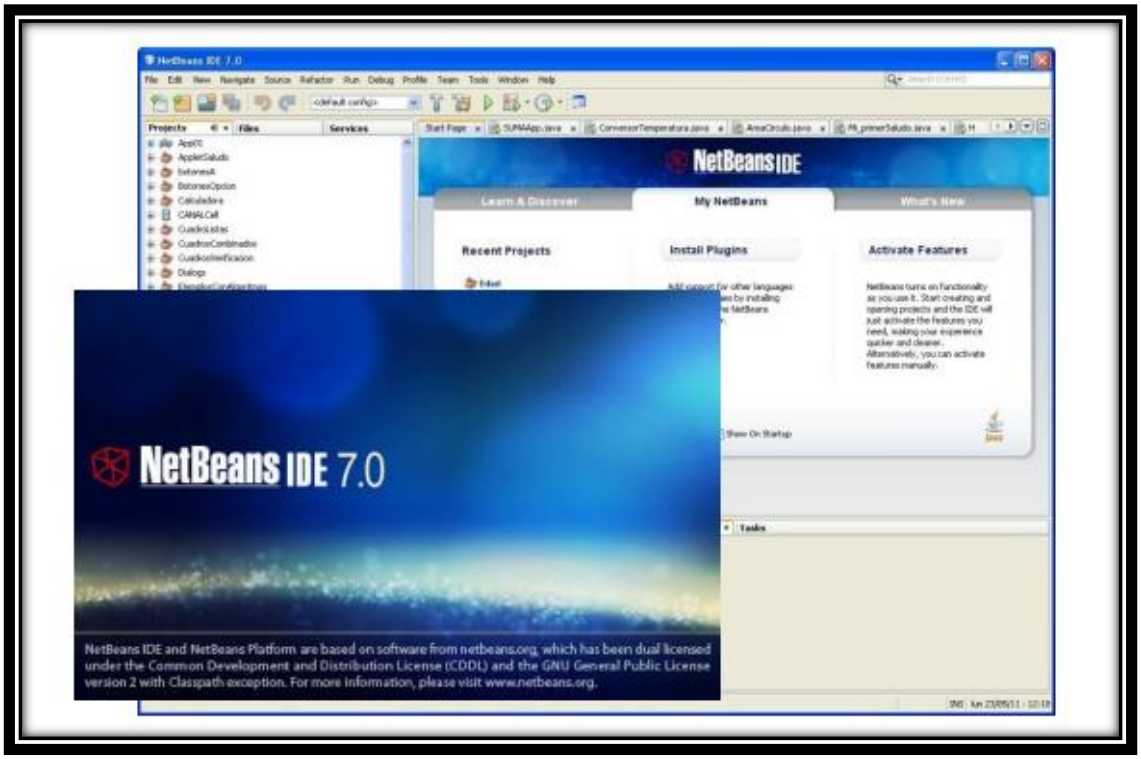
2.4.1 NetBeans

NetBeans (Figura 33) es un poderoso entorno de desarrollo (IDE) por medio del cual se pueden crear todo tipo de aplicaciones desde los programas más simples hasta las aplicaciones más complejas, que pueden llegar a incluir interacción web, UML, base de datos, aplicaciones para telefonía móvil e inclusive Inteligencia Artificial (IA). Debido a su eficiente desempeño y características, se desarrollará la programación de aplicaciones para el presente proyecto en este IDE. Cabe mencionar, que Java no cuenta con un entorno de desarrollo propio, de ahí que se puede utilizar desde un bloc de notas hasta entornos de desarrollo avanzados, al analizar sus características,

no es difícil entender porque se utilizará NetBeans como IDE principal para el desarrollo de aplicaciones en el presente proyecto (PENARRIETA, 2011).

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java, es un producto de código abierto, libre y gratuito sin restricciones de uso. Demás está decir que aunque esta hecho principalmente para el lenguaje de programación Java, puede también ser utilizado para otros lenguajes de programación, junto con esto, existen además un número importante de módulos (archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo) para extender las propiedades y características de NetBeans IDE dependiendo de la naturaleza de la aplicación (PENARRIETA, 2011).

Figura 33. IDE NetBeans 7.0.



Fuente: (PENARRIETA, 2011).

A más de las características ya mencionadas cabe resaltar las siguientes:

- **Asistentes** para la creación y configuración de distintos proyectos, incluido elección de frameworks.

- **Excelente editor de código**, **multilenguaje**, con el habitual coloreado y sugerencias de código, junto con opciones donde se puede usar el pulsar y arrastrar para incluir componentes en nuestro código.
- **Simplifica gestión de grandes proyectos**
- **Herramientas para depurado de errores**: el debugger que viene incluido es extremadamente útil para encontrar fallas en el código.
- **Optimización de código**
- **Acceso a base de datos**: desde el propio Netbeans es posible conectarse a distintos sistemas gestores de bases de datos, como pueden ser Oracle, MySql y demás, y ver las tablas, realizar consultas y modificaciones, y todo ello integrado en el propio IDE.
- **Integración con servidores de aplicaciones**. Es posible gestionar dichos servidores desde el propio IDE. Entre otros, podemos usar: Apache Tomcat, GlassFish, JBoss, WebLogic, Sailfin, Sun Java System Application Server, etc.
- **Es extensible a través de plugins**. Ofrecidos con relativa periodicidad, la página oficial permite obtener, generalmente de forma gratuita, los plugins que se van generando de acuerdo a la demanda de los usuarios.

2.5 Conjunto de Aplicaciones Gestoras de Base de Datos. Tesis Mosquera

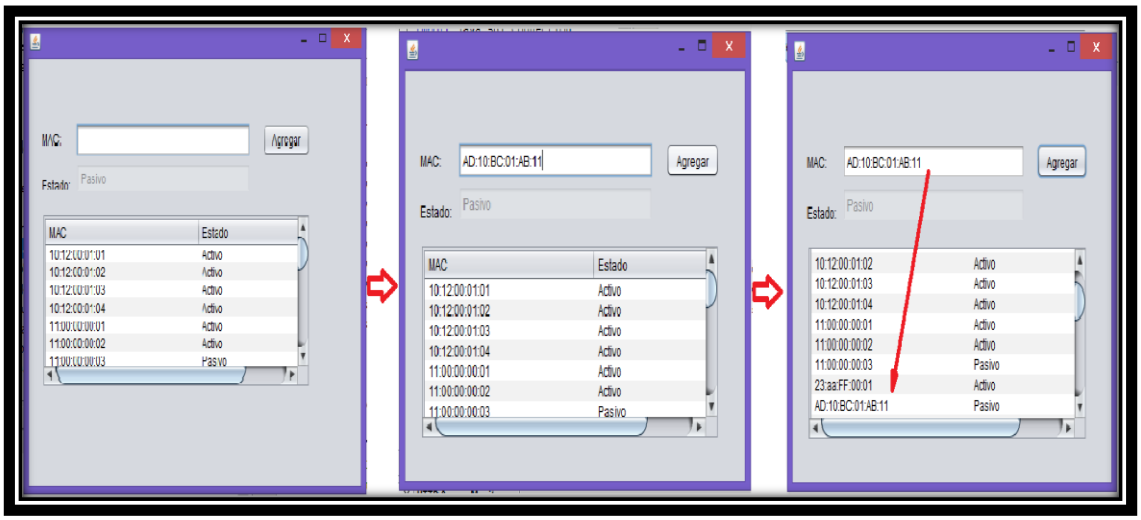
El proyecto "Tesis Mosquera" no es más que un conjunto de programas desarrollados en NetBeans que permiten a la empresa que provee el servicio de agua potable, utilizar, manipular y controlar los datos de consumo de los medidores y datos derivadas de los mismos, ya sea tarifas, fechas, usuarios a los que pertenecen los números de medidores (mac), datos de usuarios, etc; todos estos, albergados en forma de tablas expansibles, dentro de la base de datos. A continuación se ofrecerá una breve revisión de las funciones que ejercen cada una de estas aplicaciones, para dar una idea de la forma en la que se controla el servicio.

2.5.1 Aplicación “Medidor”.

Esta aplicación permite a la empresa, ingresar los números de cada uno de los medidores (mac) que estén actualmente disponibles en la misma, naturalmente, esto permitirá a dicha empresa, asignar un medidor único tomado de la lista generada por esta aplicación, a cada uno de los clientes que se vayan incorporando, o requieran de un medidor, sea por primera vez, o necesiten de un equipo nuevo.

Una vez arrancada la aplicación, se despliega una pantalla donde se aprecian todos los medidores existentes, los que están disponibles están marcados como “pasivo” y los que están actualmente en uso marcados como “activos”, junto con esto, un cuadro de texto que permite ingresar los nuevos números de los medidores, luego de ingresados, al presionar el botón “Agregar” el número ingresa a su tabla correspondiente en la base de datos e inmediatamente se visualiza en la lista en la parte inferior de la ventana, naturalmente dicho medidor ingresado será marcado como “pasivo” pues aún no ha sido asignado a ningún usuario (Figura 34), la tarea de asignación de medidores a usuarios tanto nuevos como para los que ya cuentan con un medidor y consecuentemente están registrados, es desarrollada por la aplicación “RegistroUsuarios” que se describe a continuación.

Figura 34. Proceso de ingreso de medidores a partir de sus mac únicas en base de datos.



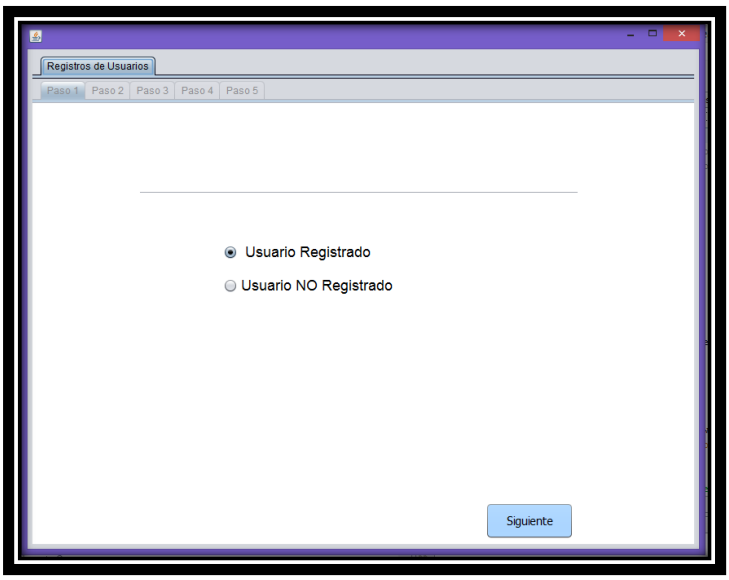
Fuente: Autor.

2.5.2 Aplicación “RegistroUsuarios”

Una vez que un usuario, sea perteneciente al grupo de usuarios registrados o incorporado recientemente, requiere un medidor, nuevo o adicional, sea domiciliario o destinado a uso industrial, la empresa debe ser capaz de registrar eficientemente este tipo de eventos, esta, es básicamente la tarea a la que se dedica la aplicación “RegistroUsuarios”.

Al momento de arrancada la aplicación, “RegistroUsuarios” despliega una ventana que permite escoger entre un usuario registrado, y un usuario no registrado (Figura 35).

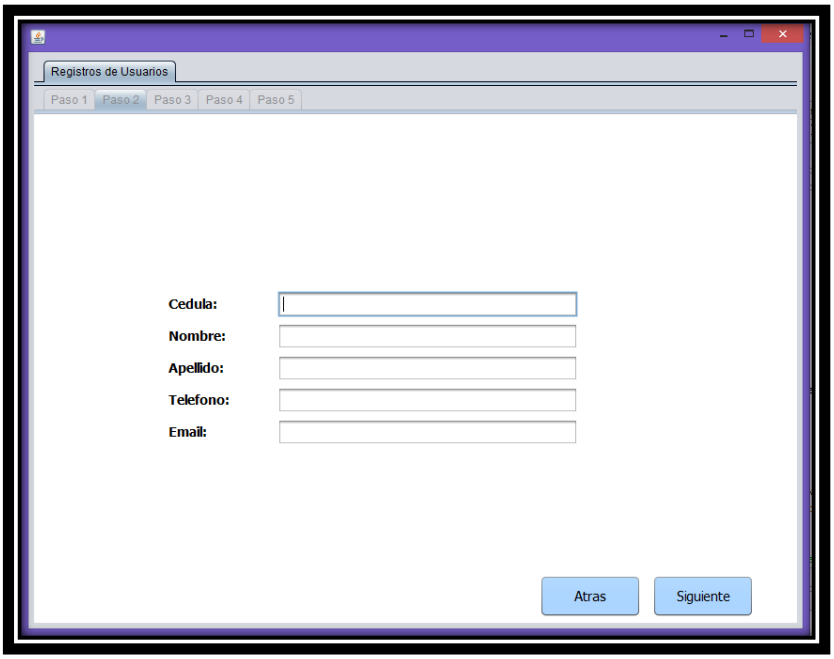
Figura 35. Pantalla de selección para usuarios Registrados y No Registrados.



Fuente: Autor.

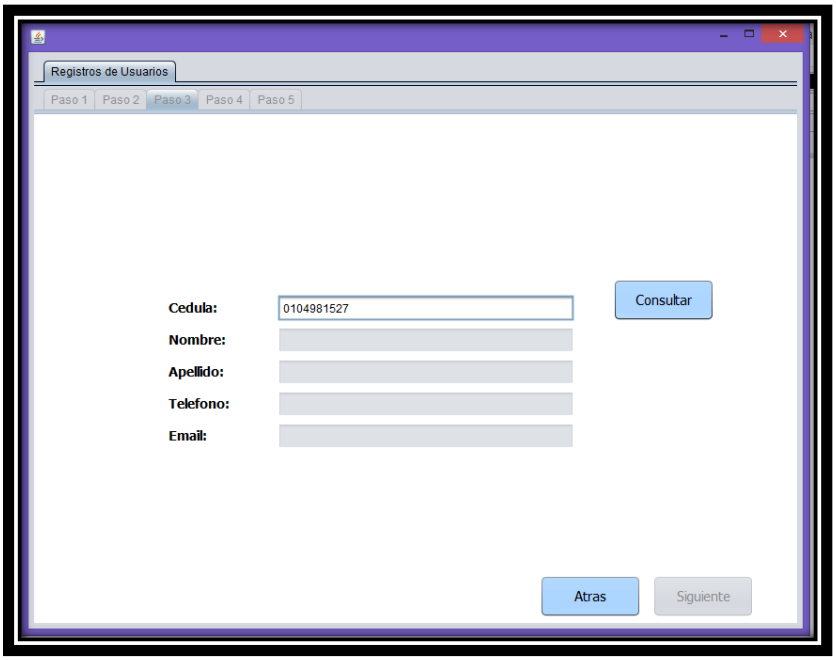
Puesto que la función principal de la aplicación es la misma para ambos, asignará un medidor a un usuario, básicamente los procesos siguientes son muy parecidos indistintamente si es para un usuario nuevo o uno ya registrado, con la diferencia lógica de, que si es un usuario nuevo, primero se registrarán sus datos antes de asignarle un medidor (figura 36),y si es un usuario ya registrado, se utilizará un criterio de búsqueda basado en su número de cédula registrado previamente cuando se ingresaron sus datos por primera vez (figura 37).

Figura 36. Registro de datos de un usuario nuevo.



Fuente: Autor

Figura 37. Cuadro de búsqueda de un usuario ya registrado.

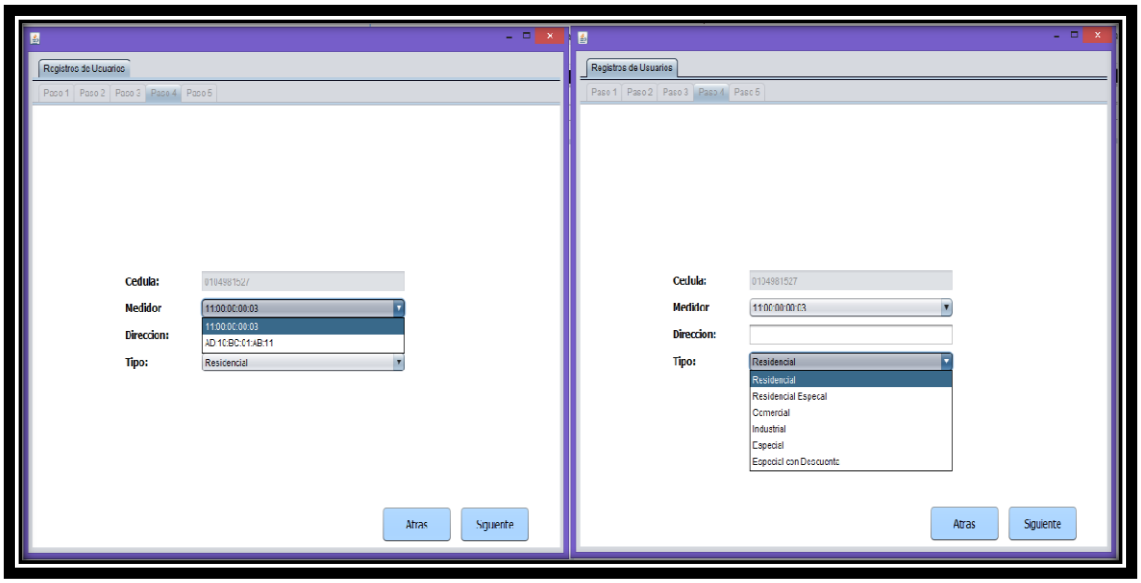


Fuente: Autor.

Luego de esto, para ambos casos, se mostrará una pantalla con el nombre y datos del usuario, y dos pestañas que, al momento de desplegarse, mostrarán una lista de todos

los medidores disponibles ingresados previamente en la aplicación “Medidor” y una lista del tipo de servicio (Residencial, Comercial, Especial, etc) respectivamente, la persona encargada de manejar la aplicación deberá asignar un numero de medidor para el usuario y el tipo de servicio, pues dependiendo del uso al que este dedicado el servicio dependerá la tarifa que se le aplique al momento del cobro (Figura 38).

Figura 38. Pantalla de ingreso de datos de usuario, numero de medidor, y tipo de tarifa.



Fuente: Autor

Posteriormente la acción quedará completada cuando, después de dar click en el botón “Siguiente”, se muestre la pantalla siguiente un cartel con la palabra "DONE" (Figura 39), cuyas siglas en ingles se vierten como “hecho” lo que indica que la acción se completó, y naturalmente se vuelve al inicio.

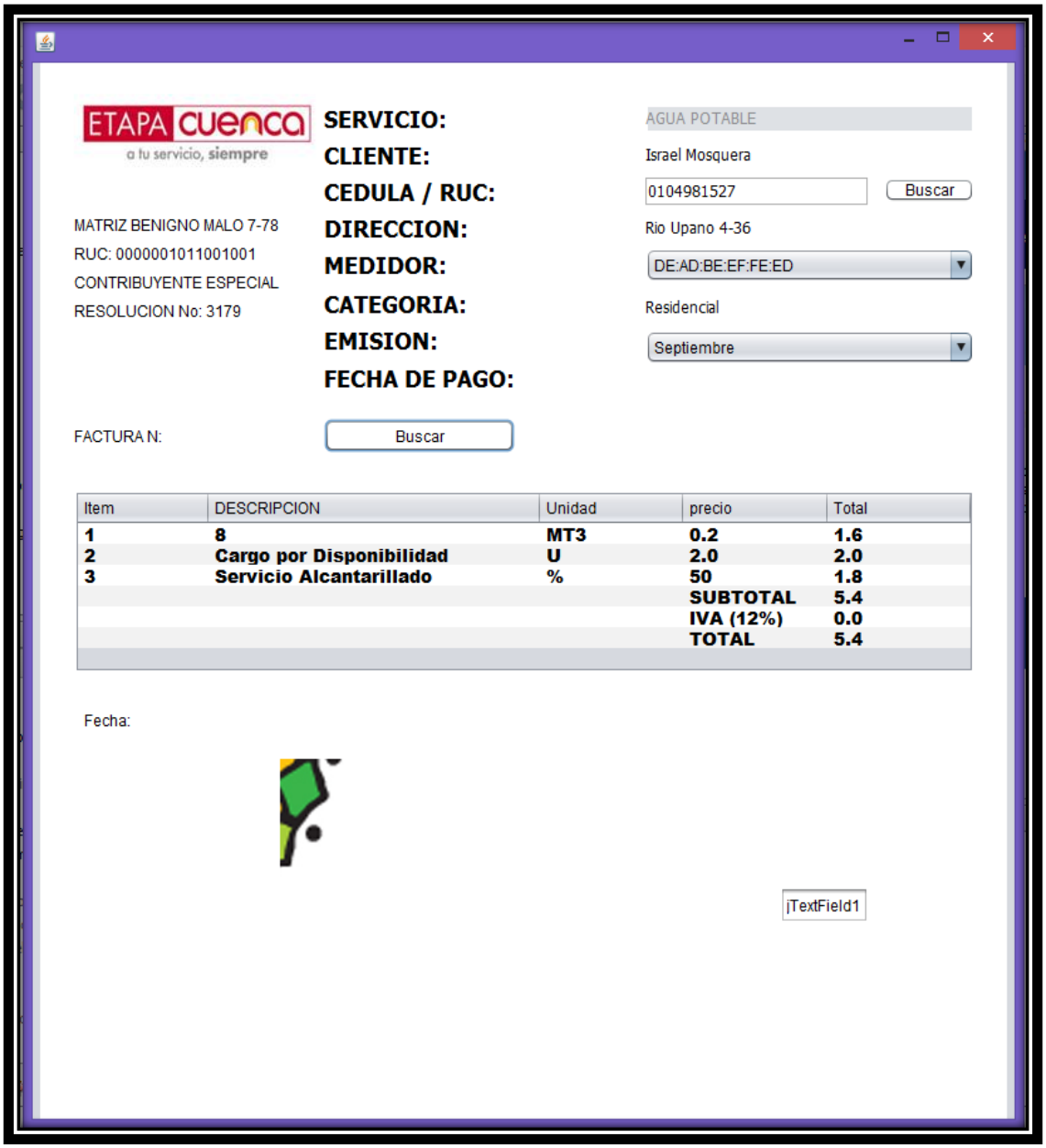
Figura 39. Ventana final que indica que el proceso concluyo correctamente.

Fuente: Autor

2.5.3 Aplicación “Facturar”

Basados en la información generada a partir del consumo, y, tomando la normativa de la página oficial de ETAPA para tarifas de agua potable, es fácil entender el propósito de la aplicación “Facturar” (Figura 40), dicha aplicación no hace más que emitir el precio a pagar por el consumo de agua potable en determinado mes. Su funcionamiento es simple, primero, el usuario ingresa e número de cedula del abonado al cual emitirá la factura, la aplicación busca y enseña los datos de dicho usuario como de los medidores que están bajo su cargo, posteriormente, en la pestaña desplegable de los medidores se escogerá el medidor (mac) del cual se emitirá la factura, en caso de tener varios medidores a su cargo, caso contrario, al tener solo uno, simplemente se va directamente al paso siguiente, que es, escoger el mes que el abonado pretende cancelar, y luego, simplemente se presiona la tecla “Buscar”, inmediatamente aparecerá el consumo y consecuente valor a pagar, dicha acción se repetirá en el caso de que el abonado quiera cancelar el valor de varios meses, en donde lo único que se hace es, lógicamente, escoger el mes adicional las veces requeridas.

Figura 40. Vista de la aplicación “Facturar”.



Fuente: Autor.

CAPITULO 3

PAGINA WEB DE CONSULTA

3.1 Conceptos Preliminares.

3.1.1 Aplicaciones web.

En 1989, un equipo de investigación en Suiza, desarrollaron un programa y protocolo que facilitaría la comunicación y participación de sus investigaciones, dicha aplicación no es más que lo que ahora se conoce como Hypertext Transfer Protocol (HTTP), que, con un nuevo lenguaje conocido actualmente como Hypertext Markup Language (HTML), darían lugar al famoso protocolo por casi todos utilizado: World Wide Web (WWW) (GARCIA, 2007).

A partir de este acontecimiento, se ha potenciado el uso masivo de Internet, y este crecimiento está siendo conducido por aplicaciones que utilizan ideas de desarrollo provenientes de los mismos usuarios, lo que redunda en nuevas utilidades tanto sociales como interactivas que recogen y reutilizan todo tipo de información (GARCIA, 2007).

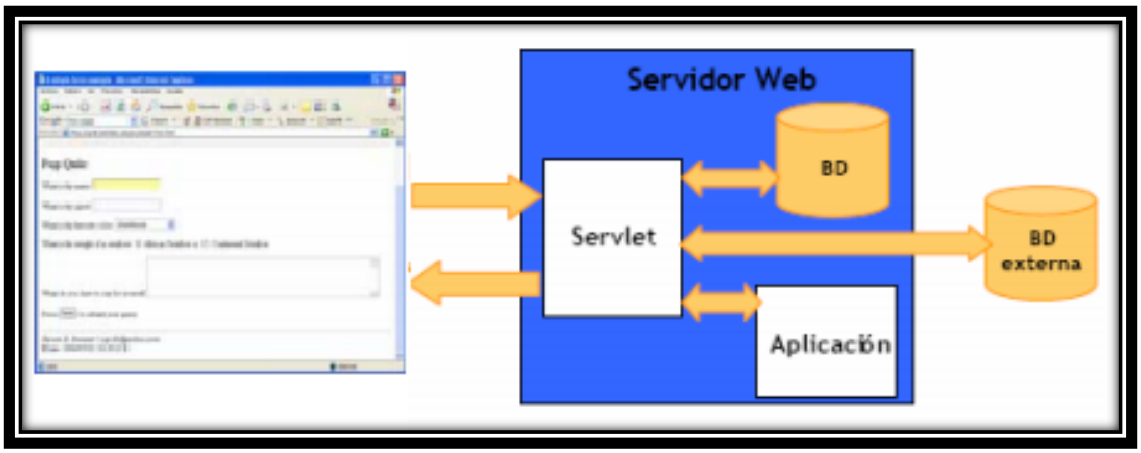
Con el avance de La tecnología, las aplicaciones web se vuelven cada vez más interactivas, ya sea compartiendo datos entre ellas mismas o intercambiando información entre páginas, dichas aplicaciones permiten la interacción con sistemas informáticos de gestión de una empresa, como pueden ser bases de datos, contabilidad o inventario, etc. Todo esto a través de, obviamente, una página web, lo que redunda en alcances tales como generación de contenido, creación de páginas personalizadas o, en general, el desarrollo de comercio electrónico (GARCIA, 2007).

A continuación se describen las tecnologías más empleadas para el desarrollo de aplicaciones web, y por ende, las que en su mayoría se utilizaron el desarrollo del presente tema de tesis.

3.1.2 Servlets.

Los Servlets son los programas con los que están construidas las páginas Web, los servlets se ejecutan a través de un servidor Web que, como se verá más adelante, se encarga de las peticiones de los usuarios, lo que es enormemente útil pues actualmente las página Web en su mayoría no solamente están basada en datos enviados por el usuario, sino que los datos de las paginas cambian frecuentemente, y además, muchas de las páginas Web usan información desde bases de datos corporativas u otras fuentes (OLANDA, 2007)(Figura 41).

Figura 41. Esquema básico del funcionamiento de un servlet en una página web.



Fuente: (OLANDA, 2007)

Se puede decir, por lo tanto, que los Servlets, son programas especiales de carácter web, que actúan como capa intermedia entre una petición proveniente de un cliente HTTP y las Bases de Datos o Aplicaciones en el servidor HTTP

Si se habla de JAVA Servlets, el mismo principio se aplica, con la variación de que la tecnología de software utilizada en cuanto a lenguaje de programación en general, proviene de JAVA, y su funcionalidad está dedicada principalmente a:

- Leer los datos enviados por el cliente.
- Extraer cualquier información útil ya sea en la cabecera HTTP o en el cuerpo de la petición enviado por el cliente.
- Generar resultados dinámicos.
- Formatear los resultados en un documento HTML.

- Establecer los parámetros HTTP adecuados incluidos en la cabecera de la respuesta (por ejemplo: el tipo de documento, cookies, etc.)
- Enviar el documento final al cliente.

Cabe destacar que el concepto de JAVA Servlet está íntimamente ligado al concepto de JSP que se analiza a continuación (OLANDA, 2007).

3.1.3 Páginas Web dinámicas. JSP.

JSP (Java Server Pages) es una alternativa a los servlets provista por java para la generación de contenidos de forma dinámica en el lado del servidor. JSP es de hecho, una tecnología equivalente a los servlets, pues las páginas JSP se traducen en servlets que ejecuta el servidor en cada petición.

Las páginas JSP facilitan la creación de contenido dinámico sin que sea necesario conocer a fondo el lenguaje Java. Una página JSP utiliza tanto código HTML como fragmentos de código Java para generar un contenido Web que convina tanto componentes estáticos como dinámicos. Además de esta combinación una página JSP puede instanciar clases, llamar a otras páginas JSP, Servlets e incluir JavaBeans y applets (Pequeños programas o aplicaciones). Para pasar datos de un JSP a otro, o a un Servlet, se utiliza las solicitudes (*Request)* mediante la utilización de formularios, que es el elemento que interactúa con el usuario (OLANDA, 2007).

Aunque si bien es cierto que un JSP puede imitar la funcionalidad de un Servlet, la principal función de un JSP es ser una especie de capa de exhibición, que separe la presentación del control de la aplicación.

3.1.4 HTTP - Hypertext Transfer Protocol.

Es el protocolo utilizado en cada operación realizada en la Web. Desarrollado por el consorcio W3C (World Wide Web Consortium) y la IETF. HTTP define la forma de escribir (sintaxis) e interpretar (semántica) las sentencias que usan los elementos software dentro de lo que vendrá siendo, en un todo, la arquitectura web (clientes, servidores, etc) para comunicarse. Como en una transacción, este protocolo sigue el modelo petición-respuesta entre un cliente y un servidor. HTTP es un protocolo

carente de estado lo que significa que no guarda ninguna información sobre conexiones anteriores (GARCIA, 2007).

3.1.5 HTML – HyperText Markup Language.

El lenguaje HTML está conformado por una serie etiquetas que definen el contenido y el aspecto de las páginas web. Aunque no se puede considerar un leguaje derivado de él, está basado en el estándar SGML (Standard Generalizad Markup Language), que es un lenguaje que permite ordenar y etiquetar los distintos elementos que componen un documento. Utilizado para manejar documentos relativamente grandes que son sujetos a constantes revisiones y/o se imprimen en distintos formatos o idiomas. Desarrollado y estandarizado por ISO en 1986, el mismo W3C se encarga de su estandarización (GARCIA, 2007).

3.1.6 JavaScript.

Es un lenguaje utilizado generalmente para páginas web, se caracteriza por ser interpretado basado en objetos y guiado por eventos. Todos los navegadores están en la capacidad de interpretar el código JavaScript integrado dentro de las páginas web. Para la interacción con una página o aplicación web, el JavaScript posee una implementación denominada DOM (Document Object Model) que es una interfaz de programación de aplicaciones para documentos HTML y XML. Define la estructura lógica de dichos documentos y el modo en que se accederá e interactuará con ellos (GARCIA, 2007).

3.1.7 Servidores Web.

También llamado servidor HTTP; es el elemento encargado de mantenerse a la espera y procesar las peticiones HTTP en el momento en que son llevadas a cabo por un cliente HTTP, y consecuentemente, enviar la información solicitada a los clientes. Almacena principalmente documentos HTML que, como ya se ha visto, son documentos, con un formato especial para la visualización de páginas web a través de los navegadores utilizados por clientes, dichos documentos son almacenados en forma de archivos y pueden ser imágenes, videos, texto, presentaciones, y en general

todo tipo de información. Todos los sitios web en Internet residen en servidores web. Cuando un usuario se conecta a un sitio, el servidor envía los datos que se muestran en la pantalla (GARCIA, 2007).
En la actualidad los servidores web más utilizados son Apache Server (utilizado en el presente proyecto), y Microsoft Internet Information.

3.1.8 Servidor Apache Server.

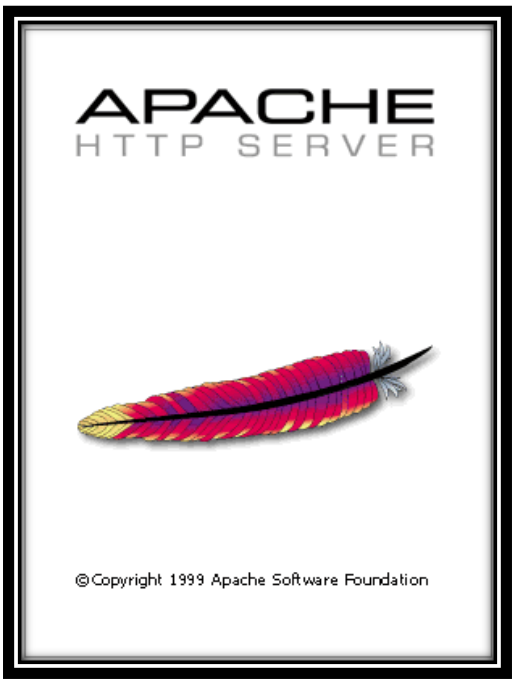
El Servidor Apache Server (Figura 42) es uno de los servidores web más populares en el mundo, es un software libre y de código abierto, es decir, de adquisición gratuita y con la posibilidad de ser mejorado por parte de los usuarios en cuanto a su estructura se refiere. Si bien es cierto que inicialmente este servidor era exclusivo de la plataforma Unix, posteriormente, y gracias al trabajo de múltiples desarrolladores, este software migro a la mayoría de plataformas conocidas y utilizadas hoy en día, como lo son Windows, Mac OSX y UNIX (GNU, BSD, etc) (MALDONADO, 2008). Entre las características que hicieron de este, uno de los servidores HTTP más utilizados en el mundo se destacan las siguientes:

- **Multiplataforma**. Apache es ejecutable sobre una multitud de plataformas y Sistemas Operativos.
- **Software Libre**. La tecnología de Apache es libre y de código abierto, lo que le otorga transparencia y brinda al usuario la posibilidad de conocer que es lo que realmente está instalando.
- **Configurable**. Apache es un servidor maleable, es decir, su diseño y funcionalidad son capaces de ser ampliados, según las necesidades y gusto del usuario, asi como los mensajes de error ocurrentes en el uso del servidor
- **Diversidad de lenguajes de Programación**. Apache trabaja en conjunto con gran cantidad de Lenguajes de Programación tales como PHP (*PHP Hypertext Pre-processor*), Perl, CGI (*Common Gateway Interface*), Java, JSP (*Java Server Pages*) y otros lenguajes de script, lo que complementa los sitios web dinámicos que se utiliza actualmente.
- **Archivos Log**. Apache cuenta con la posibilidad de albergar información global del sistema, errores producidos en un determinado tiempo, esta información se encuentra en los denominados archivos Log, indispensables

para administradores de sistemas en lo que respecta a políticas de seguridad a seguir debido a la gran cantidad de información que contiene.

- **Filosofía Libre.** Apache está muy ligada a su pensamiento y/o filosofía libre, es decir, el enfoque que tienen los usuarios del mismo, el cual, junto con la posibilidad que tienen dichos usuarios de manejar el código del servidor, hace posible encontrar gran cantidad de documentos, ejemplos y ayuda en todos los idiomas.
- **Actualización Constante.** Al ser de código abierto, Apache se actualiza constantemente. Muchos programadores de todo el mundo contribuyen periódicamente con mejoras al software, que están disponibles, en su mayoría de forma gratuita, para cualquier persona que use el servidor web.
- **Costo.** El servidor web Apache es completamente gratuito y accesible para cualquier persona en el mundo. Excepto en los casos en que ha sido una aplicación específicamente diseñada para usos no generales, utilizar el código abierto Apache Web Server representa un ahorro considerable, en especial para las pequeñas empresas o particulares que no cuentan con grandes presupuestos para el servidor (MALDONADO, 2008).

Figura 42. Logo del servidor Apache y productos relacionados.

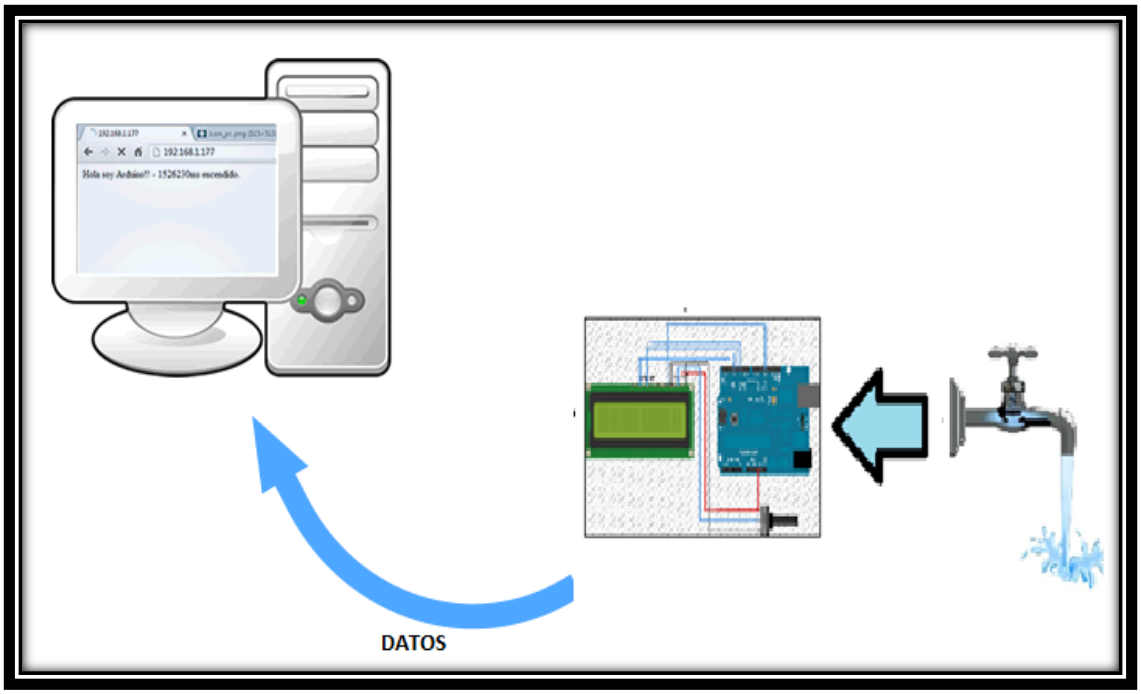


Fuente: (MALDONADO, 2008).

3.2 Comunicación entre el Medidor Electrónico y Software.

Al hablar del medidor electrónico, se habla obviamente del equipo físico, expuesto con mayor profundidad en el primer capítulo, a su vez, al hablar del software, nos referimos a todo elemento no físico desarrollado mediante programación, como lo son las aplicaciones web, servlets, base de datos, y programas en general que gestionan el manejo de la información. Naturalmente de nada servirían estos elementos si no están comunicados entre sí, debido a que, del medidor físico nacen los datos reales con los que se trabaja, y con los elementos de software, se hace el tratamiento de los mismos y la exposición de ellos de la forma requerida, principalmente, en una página web. Una de las partes clave del proyecto, es lograr una comunicación entre el equipo físico y el software que hace el tratamiento de la información (Figura 43)

Figura 43. Comunicación entre el equipo físico y el software.



Fuente: (FJRAMIREZ, 2013)

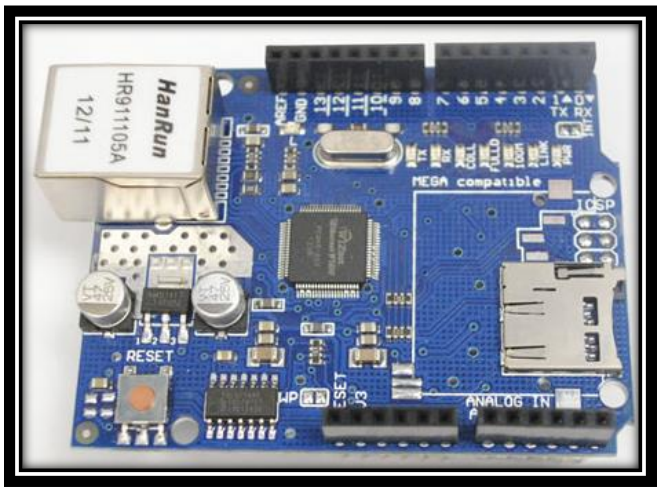
Para empezar, es necesario decir que la placa Arduino UNO por sí sola no es capaz de lograr una comunicación, y peor aún, enviar datos hacia ningún sistema, como ya se mencionó en el capítulo uno, para que dicha placa pueda comunicarse, es

necesario la implementación de los denominados escudos, en este caso un escudo Ethernet, mismo que se pasa a analizar a continuación.

3.2.1 Ethernet Shield

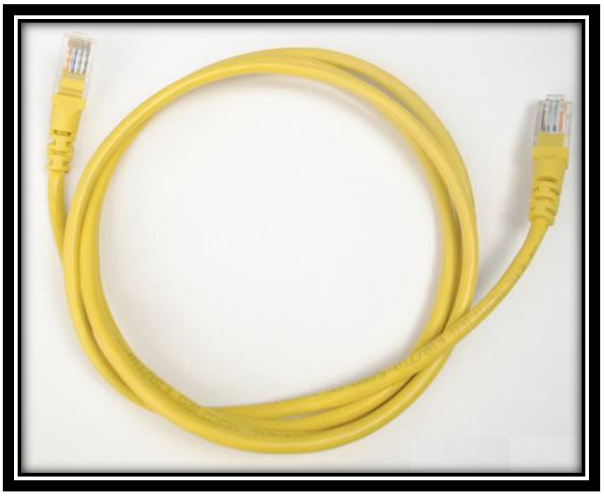
El Arduino Ethernet Shield (Figura 44) se monta sobre la placa principal, en este caso Arduino UNO a Internet para formar un conjunto completo, capaz de conectarse a una red a través de un cable RJ45 (Figura 45).

Figura 44. Ethernet Shield.



Fuente: (FJRAMIREZ, 2013).

Figura 45. Cable RJ45 para la conexión.



Fuente: (FJRAMIREZ, 2013)

El Arduino Ethernet Shield permite a una placa Arduino se conecte a internet, para lo cual al momento de la programación debe exportarse la biblioteca de Ethernet para escribir los fragmentos de código que permiten conectarse a Internet a la placa a través del escudo, utilizando como medio físico para ello una conexión estándar RJ-45, con un transformador de línea integrada y alimentación a través de Ethernet activado (SHIELD, 2010).
El escudo contiene los siguientes LEDs informativos:

- PWR: indica que la placa y el escudo están energizadas.
- ENLACE: indica la presencia de un enlace de red y parpadea cuando el escudo transmite o recibe datos
- FULLD: indica que la conexión de red es full dúplex.
- 100M: indica la presencia de un Mb / s 100 conexión de red (en contraposición a 10 Mb / s)
- RX: parpadea cuando el escudo recibe datos
- TX: parpadea cuando el escudo envía datos
- COLL: parpadea cuando se detectan colisiones de red (SHIELD, 2010).

Todas estas características dan la capacidad a la placa de comunicarse vía Ethernet y enviar datos mediante las sentencias descritas a continuación.

3.2.2 Programación del Medidor Físico.

Para lograr una comunicación Ethernet entre el medidor y un sistema de aplicaciones web, y en general, programar un microcontrolador; lo primero que hay que hacer es declarar las librerías a utilizar que nos permitan, entre otras cosas, programar dicha comunicación, concretamente la librería "*Ethernet*" (Figura 46), descargable desde la página oficial de software de Arduino.

Figura 46. Declaración de librería Ethernet.

```
// INICIALIZAR LIBRERIAS

#include <Ethernet.h>
#include <Arduino.h>
#include "LiquidCrystal.h"
```

Fuente: Autor

Una vez declaradas las librerías, es necesario declarar todas las variables que intervienen en una aplicación que incluya comunicación web (Figura 47), en este caso a través de Ethernet, y son principalmente las siguientes:

- **MAC.** Numero único de cada equipo que consta de 6 pares de dígitos hexadecimales separados en este caso por el símbolo "**:**" cada dos dígitos, mismo que servirá como numero de medidor
- **Dirección IP**. Del ordenador en el que se va a enviar datos.
- **Gateway.** Dirección IP del gateway o router

- **EthernetClient.** Se refiere al nombre de la variable del cliente o usuario en nuestro caso "client".

Figura 47. Declaración de variables que intervienen en la comunicación web.

```
// INICIALIZAR VARIABLES
String macl="DE:AD:BE:EF:FE:ED";
byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED}; //MAC
IPAddress ip(192,168,1,102); //IP
byte gateway[] = { 192, 168, 1, 100 }; // direccion IP del gateway o router
IPAddress server(192,168,1,100);    // direccion para Google (usando DNS)
EthernetClient client;
```

Fuente: Autor.

Teniendo las variables listas y las librerías iniciadas, solo queda por habilitar la comunicación Ethernet mediante la sentencia "*Ethernet.begin(mac, ip, gateway);"* al momento de iniciar el programa para posteriormente enviar los datos de consumo (albergados en la variable caudal), los datos de mac guardados en la variable que lleva su nombre; y por último, los datos de caudal por unidad de tiempo, almacenados en la variable Flowrate. El proceso inicia mediante la realización de una petición de comunicación hacia la página por parte del medidor a través de un método conocido como "get", que permite enviar parámetros y/o formularios desde el cliente hacia el servidor agregando al final del URL un signo **"?"** y pares de parámetros que en en este caso son **"mac","caudal" y "flowrate"** , junto con el valor correspondiente a cada parámetro, cada uno separados por un símbolo**"&" (**Figura 48**);** correspondientemente el servidor devuelve una respuesta a la petición confirmando si los datos se ingresaron correctamente o no, de darse el primer caso, los datos son enviados continuamente hacia el servlet de la página que gestiona la información que llega desde la placa electrónica, para exhibirlos o almacenarlos en la base de datos.

Figura 48. Sentencia de petición para envío de datos hacia la página web.

```
String consulta = "GET http://192.168.1.100:8084/Flujos/Grabar?consumo=" + caudal + "&mac=" + macl + "&caudal=" + flowrate + " HTTP/1.1";
```

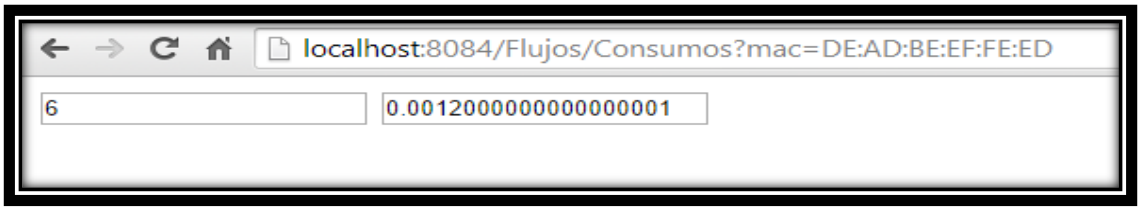
Fuente: Autor.

3.3 Servlets del Medidor Electrónico.

3.3.1 Servlet Consumos.java

El servlet Consumos.java proporciona el valor a pagar del consumo hasta el momento de la consulta. (Figura 49).

Figura 49. Servlet Consumos.java.

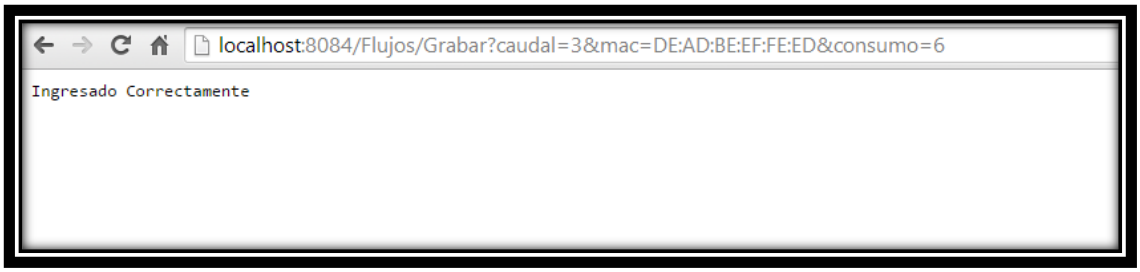


Fuente: Autor.

3.3.2 Servlet Grabar.java

El servlet Grabar.java interactúa con el medidor electrónico directamente para ir grabando los datos recogidos por este en la base de datos. (Figura 50).

Figura 50. Servlet Grabar.java.

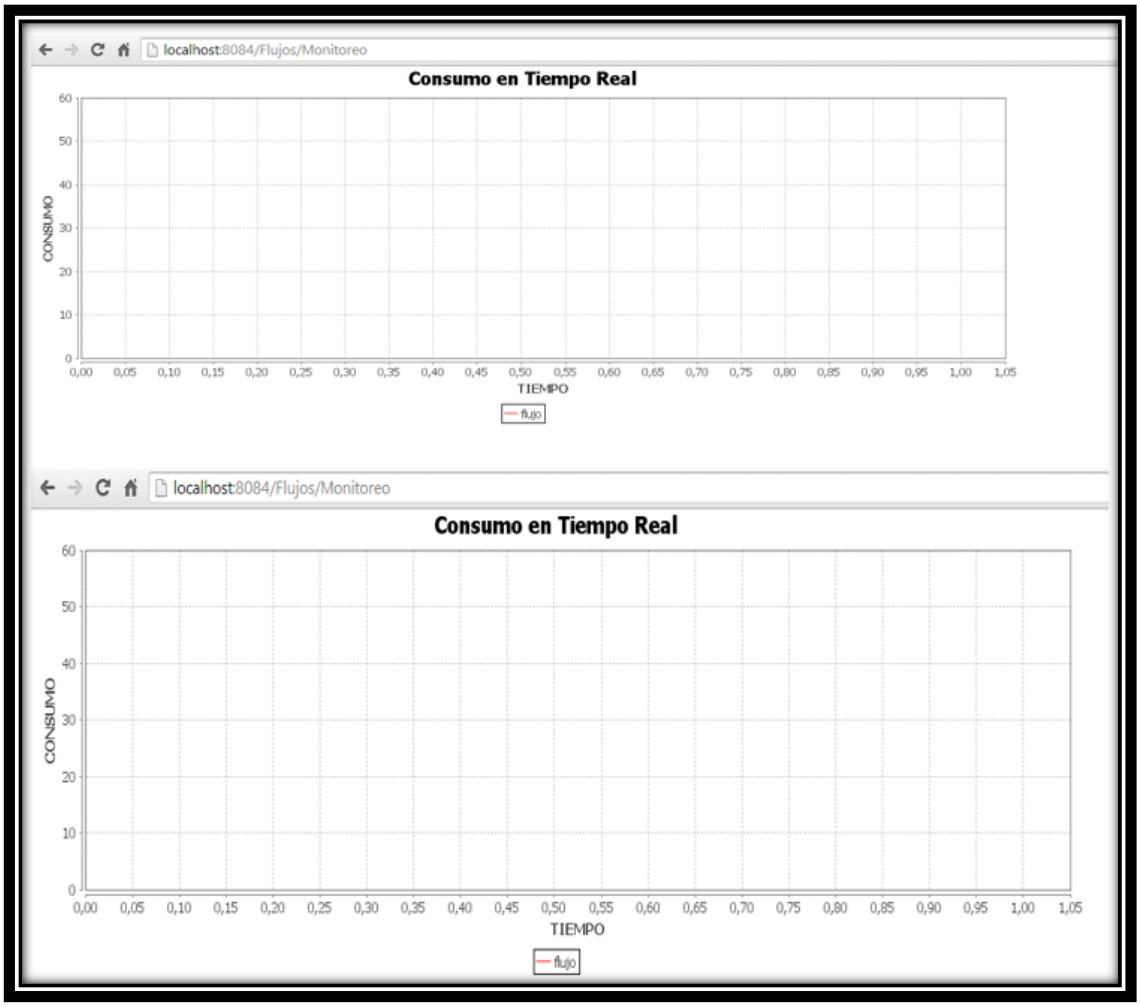


Fuente: Autor.

3.3.3 Servlet Monitoreo.java

El servlet Monitoreo.java **g**rafica el histórico consumo vs tiempo de los últimos 50 datos, para que esto pueda hacerse eficientemente es necesario especificar el número de mac del medidor al que se refiere. (Figura 51).

Figura 51. Servlet Monitoreo.java.

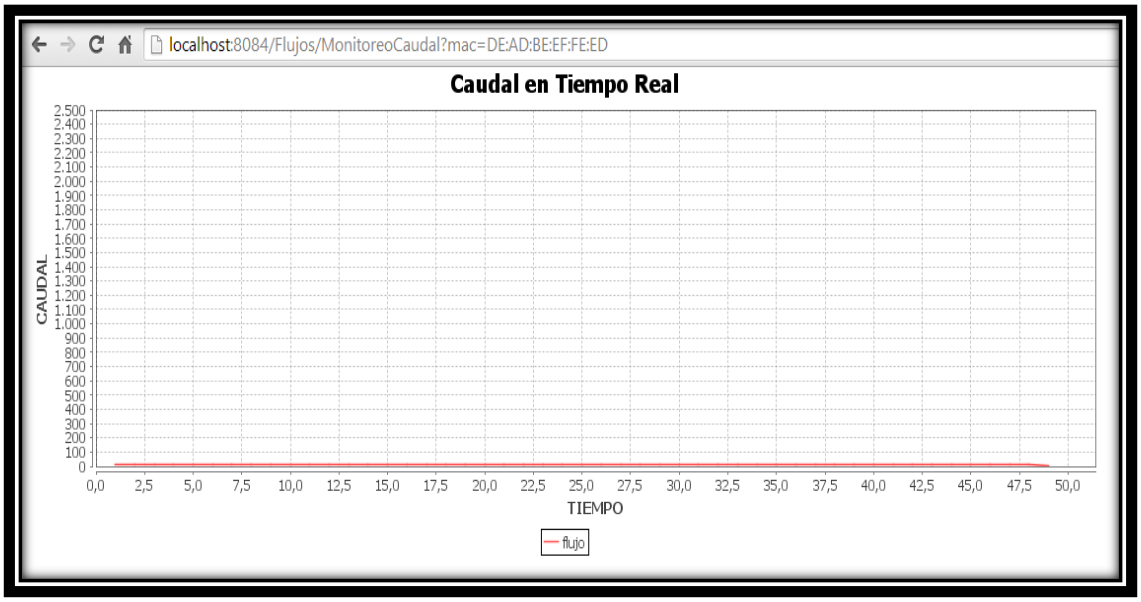


Fuente: Autor.

3.3.4 Servlet MonitoreoCaudal.java

El servlet MonitoreoCaudal.java **g**rafica el caudal vs tiempo de los últimos 50 datos, para esto es necesario especificar la mac del medidor al que se refiere. (Figura 52).

Figura 52. SERVLET MonitoreoCaudal.java.



Fuente: Autor.

Como ya se puede suponer, el funcionamiento simultáneo de cada uno de los servlets antes mencionados da como resultado la página web del proyecto, sin embargo, aunque dicha página sea enteramente funcional, resulta poco agradable o atractivo para el usuario, desde el punto de vista visual, exponer únicamente los servlets funcionando, ya que, una página web, como muchos otros productos, tiene un mayor impacto entre los consumidores no solamente por la calidad de su contenido, sino en gran medida por la forma, apariencia o presentación en general con la que dicho producto se expone al público.

Teniendo en cuenta este hecho, es necesario echar mano de alguna herramienta de software con la cual, la página web ya funcional, tenga un significativo aumento en su atractivo visual. Y dicha herramienta no es más que el programa dedicado al diseño de páginas web **Adobe Dreamweaver**, el cual se pasa a analizar a continuación.

3.4 Entorno de Diseño de páginas web.

Como ya se ha visto, para construir una página web, es necesario valerse de instrumentos de software que permitan generar aplicaciones web dinámicas según las necesidades individuales de cada proyecto, sin embargo, una vez generada dicha

página, es altamente conveniente recurrir a otro tipo de software que permita hacerla altamente agradable, amigable y cómoda para los usuarios a quienes está destinada; en otras palabras, una vez que una página web es funcional, únicamente posee los elementos estrictamente necesarios para su funcionamiento, careciendo completamente de detalles, adornos, y demás elementos de diseño gráfico, que hacen a una página web, un entorno agradable al usuario.

En el presente proyecto de tesis, se utilizó una aplicación semiprofesional para diseño de páginas web que responde al nombre **Adobe Dreamweaver,** cuyas características y aportes en el presente proyecto se pasan a describir a continuación.

3.4.1 Adobe Dreamweaver.

Adobe Dreamweaber (Figura 53), concretamente su versión CS6, es un programa informático que permite el diseño de páginas web. La variedad de funciones que permiten editar la página web agregan con relativa rapidez características que dotan a dicha página de atractivo visual sin necesidad de programar directamente el código HTML (S.L., 2013).

Figura 53.Logotipo de Adobe Dreamweaber.

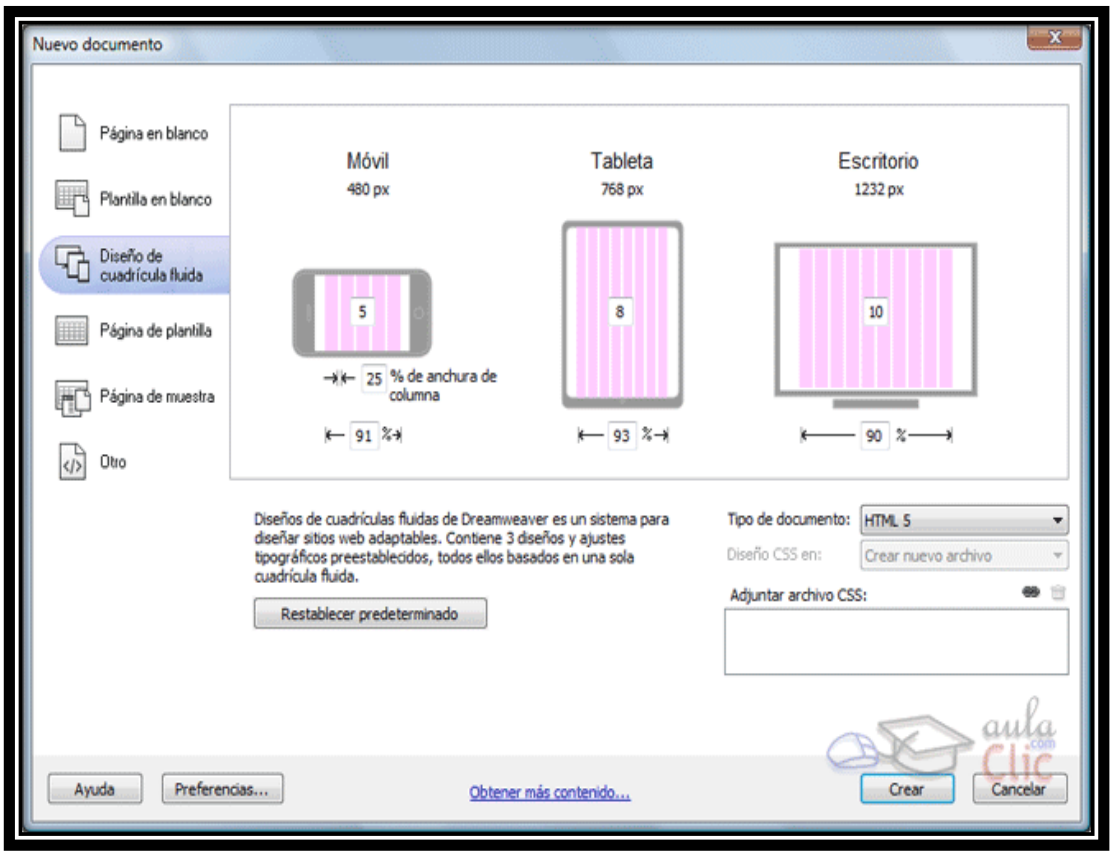
Fuente: (S.L., 2013).

Entre las múltiples funciones de Adobe Dreamweaber lo que se considera más importante es que permite trabajar con mapas visuales del sitio web en cuestión, lo que va actualizando el sitio agregando los cambios y componentes directamente en el servidor sin salir del programa. Lo que permite la continuidad de la página y poder

ver directamente la repercusión de cada cambio en la página en el momento en que se lo efectúa (S.L., 2013).

A más de lo ya mencionado, otra de las cualidades con las que cuenta la aplicación es la de plantillas con diseño fluido, lo que quiere decir, que el software cuenta con la capacidad de generar una página web que pueda acoplarse a las dimensiones del dispositivo desde donde se acceda a dicha página, ya que como se puede suponer, los dispositivos desde donde los usuarios acceden a internet son muy variados (teléfonos, tablets, ordenadores etc), por ende las dimensiones de la pantalla de los mismos también varían, de ahí que la versatilidad de la opción de plantillas con diseño fluido (Figura 54) sea una característica sobresaliente (S.L., 2013).

Figura 54.Plantilla de diseño fluido de Adobe Dreamweaber.

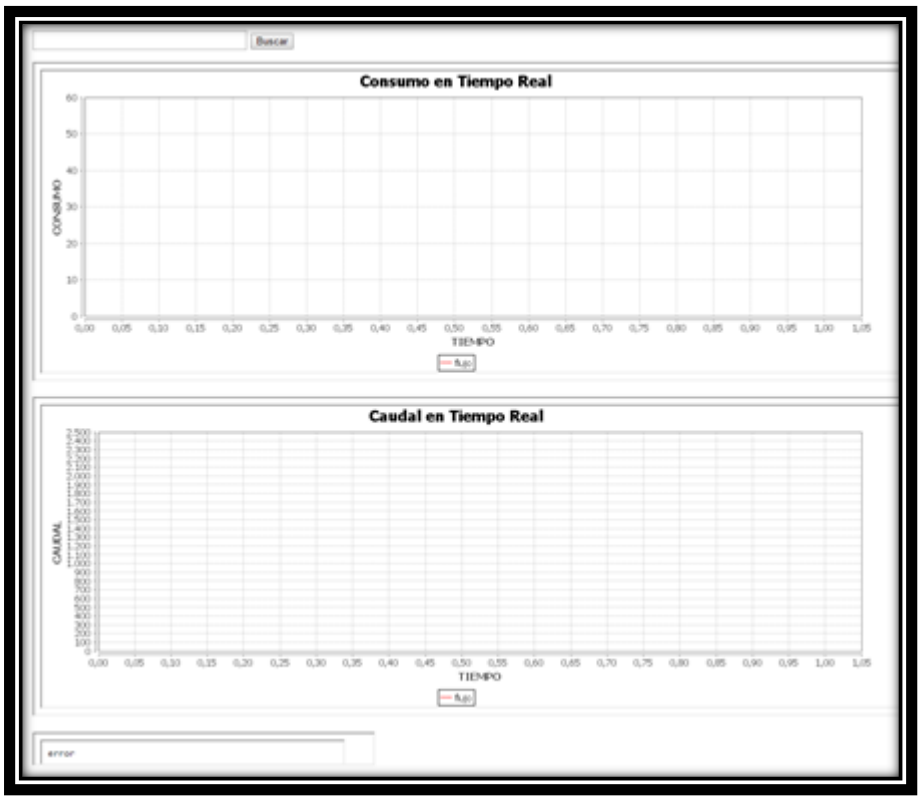


Fuente: (S.L., 2013).

Por lo demás no es de hecho necesario ahondar demasiado en todas las funciones del programa, ni tratar de aplicarlas en la página web del proyecto, pues se estaría cayendo en la categoría del diseño gráfico, lo que se hizo, es ingresar manualmente

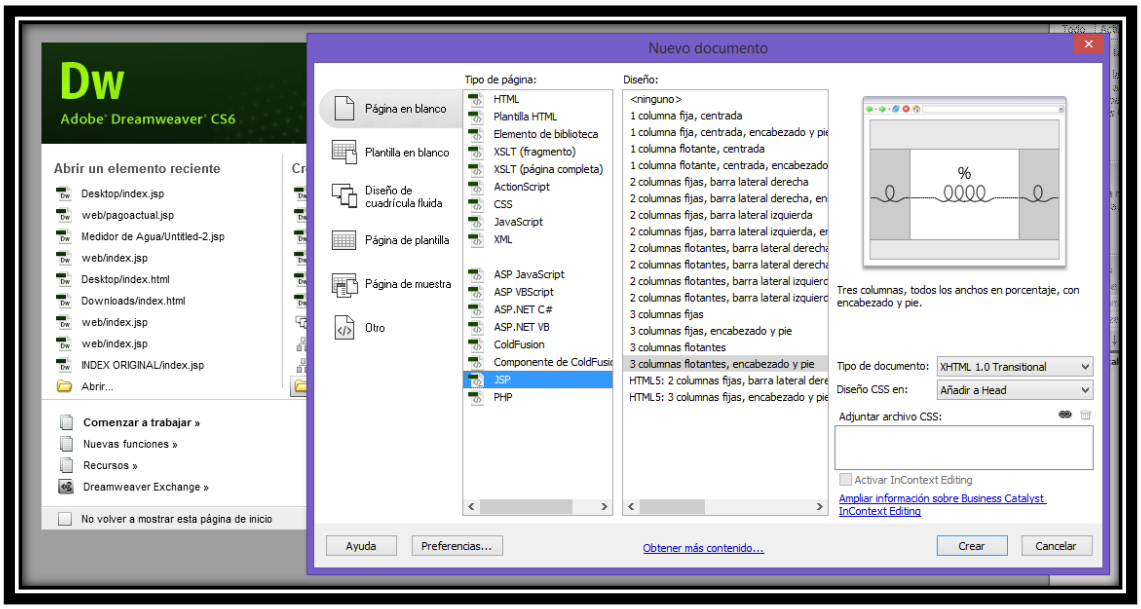
el código (index) de la página web (figura 55)en Adobe Dreamweaber y utilizar la opción que permite, al momento de iniciar un documento nuevo, dotar al código de ciertas características, como cabeceras o cajas de texto en los bordes (Figura 56).

Figura 55. Vista inicial sin diseño de la página web del proyecto.



Fuente: Autor.

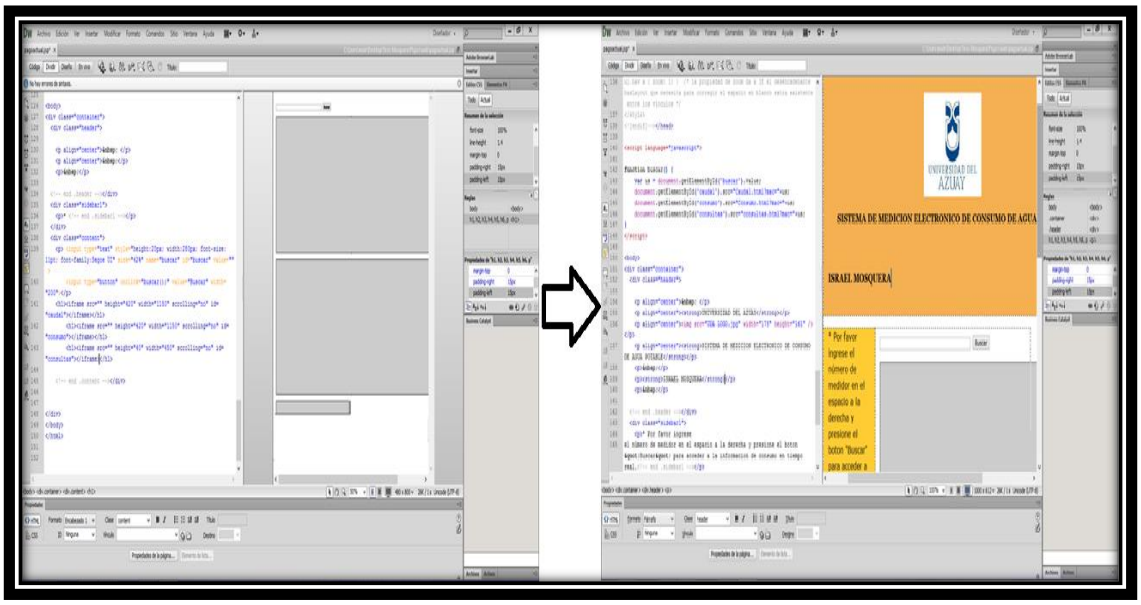
Figura 56. Selección de características del nuevo documento (Columnas, Cabeceras y pies de página).



Fuente: Autor.

Luego de lo ya mencionado, se escoge la opción grafica que permite trabajar en el código (parte izquierda) y simultáneamente en la parte gráfica de la página (derecha), cualquier característica que se le añada a la parte grafica se aplicará correspondientemente en el código y viceversa, naturalmente resulta más sencillo trabajar directamente en la parte gráfica para lograr los efectos deseados como se aprecia en la Figura 57.

Figura 57. Programación del código original a través de Dreamweaver en su entorno gráfico.



Fuente: Autor.

Como ya se mencionó, la aplicación Adobe Dreamweaber permite una alta gama de opciones para hacer de cualquier página web una experiencia visual, altamente agradable, cuando se alcanza los requisitos de diseño el código es automáticamente modificado a través del servidor y los cambios se aplican directamente en la página web, el resultado final de la página web del proyecto se aprecia en la Figura 58 a continuación.

Figura 58. Vista definitiva de la página web del proyecto.

Fuente: Autor.

CAPITULO 4

APLICACION ANDROID DE MONITOREO

4.1 Conceptos Preliminares.

4.1.1 Android.

Android es, entre otras cosas, un sistema operativo desarrollado por la conocida compañía Google, mismo que fue creado a partir del concepto OpenSource o código abierto y dirigido a dispositivos móviles de todo tipo, teléfonos móviles**,** tabletas e incluso mini ordenadores portátiles (MUNDOMANUALES.COM, 2013).

Android, a más de ser un Sistema Operativo, es además una plataforma de Software basada en Linux, lo que es ya decir bastante, pues al hablar de Linux, se está hablando de un sistema operativo de kernel libre o código abierto, muy parecido a UNIX, a su vez, y tomando en cuenta que el núcleo o kernel es el encargado de que tanto el software como el hardware de cualquier dispositivo ya sea móvil u ordenador (figura 59), puedan trabajar juntos correctamente; al hablar de Android se está hablando de un software flexible, que puede ser modificado por cualquier usuario para adaptarlo a las necesidades del mismo, pues de hecho su código fuente está al alcance de cualquier usuario sin mayor tipo de restricción, no solo esto, sino que al ser una plataforma de código abierto, cualquier desarrollador de software puede desarrollar aplicaciones para dicha plataforma independientemente del lenguaje de programación que utilice (Java, C, C++, etc) y compilarlas al código nativo de Android a través de su interfaz de aplicaciones (API)conocida como ARP (MUNDOMANUALES.COM, 2013).

En sus inicios, el sistema operativo Android fue desarrollado por la famosa compañía Google Inc, siendo un proyecto al que poco después se uniría nada más y nada menos que la Open Handset Alliance, que es un consorcio conformado por 48 compañías dedicadas a Hardware, Software y Telecomunicaciones, mismas que acordaron promocionar los estándares de códigos abiertos para dispositivos móviles.

A pesar de este hecho, ha sido Google quien se adjudicado la publicación de la mayoría del código fuente de Android bajo la licencia de Software Apache, que, cabe recalcar, es libre y de código abierto a cualquier desarrollador (MUNDOMANUALES.COM, 2013).

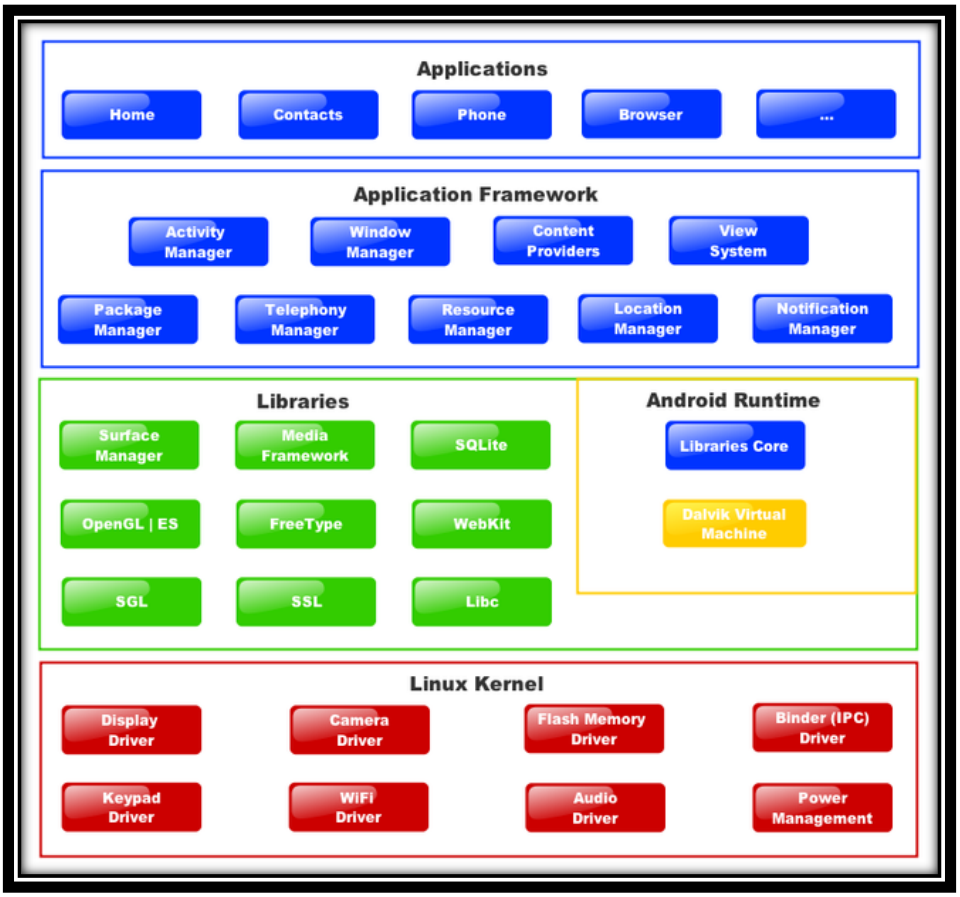
Diseñado principalmente para dispositivos móviles, Android posibilita controlar dichos dispositivos mediante bibliotecas o librerías desarrolladas y/o adaptadas por Google a través del lenguaje de programación conocido con el nombre Java (MUNDOMANUALES.COM, 2013).

4.1.1.1 Arquitectura de Android.

La arquitectura interna de la plataforma Android (Figura 59), está principalmente formada por 4 elementos:

- **Aplicaciones**: Desarrolladas a partir de la plataforma Android, las aplicaciones Android están escritas en lenguaje JAVA, y pueden incluir como base un cliente de email, calendario, SMS, mapas, navegador, contactos, etc.
- **Framework de aplicaciones**: Que es el mismo código fuente de Android utilizado para las aplicaciones base, al cual, todos los usuarios o desarrolladores de aplicaciones tienen acceso total. De esta forma, se evita la generación de gran cantidad de aplicaciones distintas para una misma acción, y así, los programas pueden ser modificados o reemplazados por cualquier usuario sin la necesidad de programar las aplicaciones desde el principio.
- **Librerias**: En la base de datos de Android se incluye un conjunto de librerías basadas en C/C++, que están al alcance de todos desarrolladores a través del framework de aplicaciones Android, que como ya se sabe, es de código abierto.
- **Runtime de Android**: Android consta de un set adicional de librerías que se encargan de la mayoría de las funcionalidades disponibles en las librerías base del lenguaje de programación Java. Su Máquina Virtual funciona a través de registros, y hecha a correr clases compiladas por el compilador de Java, anteriormente transformadas al formato .dex (Dalvik Executable) (MUNDOMANUALES.COM, 2013).

Figura 59. Arquitectura de Android.



Fuente: (GONZALES, 2011)

4.1.1.2 Características:

Muchas de las características con las que cuenta Android van a depender en si del dispositivo móvil sin embargo es posible mencionar algunas para tener una idea del alcance de Android como sistema operativo móvil:

- **Framework de aplicaciones**: Como ya se mencionó, posibilita reemplazo y reutilización de recursos.
- **Navegador integrado**: basado en el motor de código abierto **Webkit**.
- **SQlite**: Es la base de datos ligada directamente con las aplicaciones.
- **Multimedia**: Soporte para formatos de audio, video e imágenes (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF, etc).
- **Máquina virtual Dalvik**: Base de llamadas de instancia parecida a Java.

- **Telefonía GSM**: Según el dispositivo.
- **Bluetooth, EDGE, 3g y Wifi**: Según el dispositivo.
- **Cámara, GPS, brújula y acelerómetro**: Según el dispositivo.
- **Pantalla Táctil**. (MUNDOMANUALES.COM, 2013)

4.1.1.3 Ventajas y desventajas.

Para poder definir las ventajas y desventajas de este sistema operativo, es bueno hacer un contraste con su competidor inmediato, el sistema operativo que se encuentra en los dispositivos móviles de la empresa Apple, conocido con el nombre de iOS (CULTURACION.COM, 2012).

En contraste con iOS, que es un sistema operativo exclusivo de los móviles Apple; Android es completamente libre y potencialmente utilizable que cualquier compañía tecnológica lo pueda implementar en sus dispositivos, lo que se traduce en diversidad para el potencial usuario pues cuenta con una gran cantidad de marcas y modelos de dispositivos móviles que se fabricados basados en Android.

Por otro lado, una de las grandes desventajas de Android con respecto a iOS tiene que ver con las actualizaciones del sistema. En iOS las actualizaciones son automáticas y están al alcance de todos sus dispositivos a la vez, por el contrario en Android no ofrece tal ventaja pues dicho acceso a actualizaciones depende en gran medida de la compañía fabricante, que generalmente, como la mayoría de compañías, deja de lado los modelos más antiguos para centrarse en servicios para los modelos más actuales o de último lanzamiento. Sin embargo, cabe mencionar, que Android, al ser un sistema de código abierto, permite una relativa fácil actualización del móvil en sitios no oficiales de Firmwares, lo que posibilita, de manera extraoficial por así decirlo, darle más tiempo de vida a nuestro dispositivo (CULTURACION.COM, 2012).

4.1.2 App Inventor.

Se conoce con el nombre de App Inventor (Figura 60) a un entorno de desarrollo de aplicaciones dedicado a la creación de las mismas para dispositivos electrónicos basados en el sistema Operativo Android, visto anteriormente. Lo que hace especial a App Inventor, es su entorno como tal, que pretende crear aplicaciones para Android

de forma relativamente sencilla basándose en un concepto de desarrollo ante todo visual, eliminando la necesidad, hasta hace poco obvia y lógica, de ser un desarrollador móvil especializado o profesional, ya que el diseño de las aplicaciones es ante todo intuitivo, por así decirlo, ya que partiendo de diferentes módulos se puede predecir acertadamente el funcionamiento de la aplicación a desarrollarse (PICURELLI, 2013).

Figura 60. Logo App Inventor.

Fuente: (PICURELLI, 2013).

4.1.2.1 Esquema Básico de Funcionamiento de App.Inventor.

4.1.2.1.1 Eventos y controladores de eventos.

La temática de las aplicaciones basadas en App Inventor está orientada a eventos. A diferencia de un esquema de programación tradicional, no se lleva a cabo una cantidad determinada de instrucciones en un orden específico, sino que las instrucciones actúan de acuerdo a eventos, por ejemplo, presionando un botón, arrastrando el dedo o tocando la pantalla, etc (RIEGO, 2011).

En el entorno de desarrollo de App Inventor, toda función o acción se produce como resultado de un evento generado ya sea automáticamente o generados por el usuario. Si es posible hacer una analogía, toda instrucción debe estar dentro o bajo una sentencia conocida “when do”. Y de hecho esto es lo que sucede al momento de

programar a través de los bloques como se verá posteriormente. Cuando tiene lugar un evento, la aplicación ejecuta una secuencia de instrucciones de todo tipo que constituyen la actividad propia de la aplicación (RIEGO, 2011).

4.1.2.1.2 Entorno de Programación.

Para empezar a utilizar App Inventor, basta con un navegador web y un dispositivo electrónico lógicamente basado en Android conectado al ordenador para poder visualizar la aplicación a desarrollar, de no poseer este último, es posible estar al tanto y controlar los avances del proyecto a través de un software que sirve como un emulador de dispositivos móviles Android (RICOY, 2011). Dependiendo de las alternativas o preferencias se debe seguir cualquiera de las siguientes instrucciones:

1.- **Si se cuenta con un dispositivo Android y una conexión inalámbrica a Internet (WiFi)**, (Figura 61) Para lo cual es necesario instalar la aplicación Companion App Inventor en el dispositivo electrónico, y las pruebas de funcionamiento se efectúan directamente mediante un código QR que se genera en el programa al cual el dispositivo escaneará y ejecutará la aplicación. Esta es la opción más sugerida pues presenta menos inconvenientes (RICOY, 2011).

Figura 61. Conexión directa del dispositivo a través de WiFi.

Fuente. (RICOY, 2011).

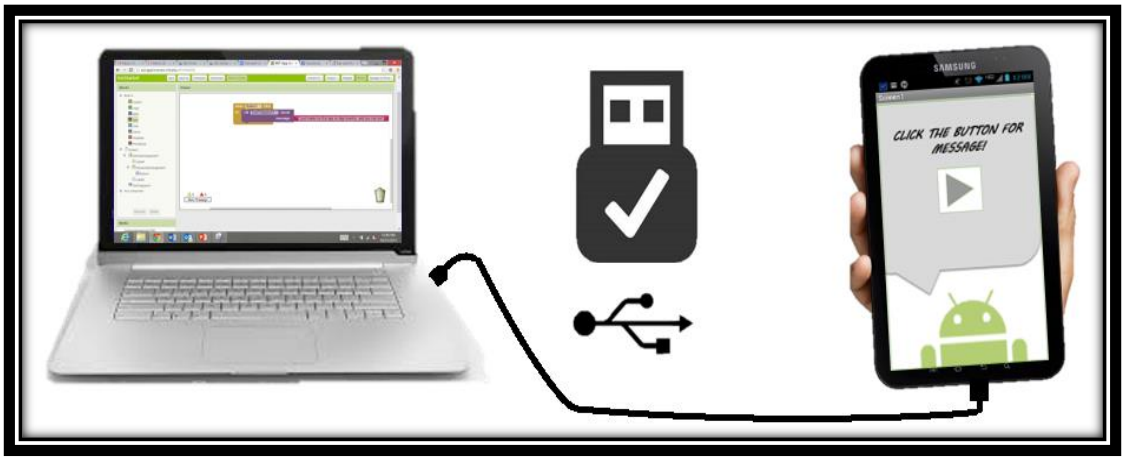
2.- **Si no se cuenta con un dispositivo Android**, será necesario instalar el software conocido con el nombre de aiStarter que corre un emulador de un dispositivo electrónico basado en Android funcional en donde se puede probar la aplicación directamente en la pantalla del mismo. (Figura 62) (RICOY, 2011).

Figura 62. Emulador de dispositivos basados en Android aiStarter.

Fuente: (RICOY, 2011)

3.- **Si no se cuenta con una conexión inalámbrica a Internet (WiFi) a pesar de contar con un dispositivo electrónico basado en Android**, para lo cual será necesario instalar el software de comunicación en el ordenador para lograr una comunicación con el dispositivo del cable USB (Figura 63). Esta opción puede presentar problemas de instalación y operatividad especialmente en Windows, se sugiere utilizarla como último recurso (RICOY, 2011).

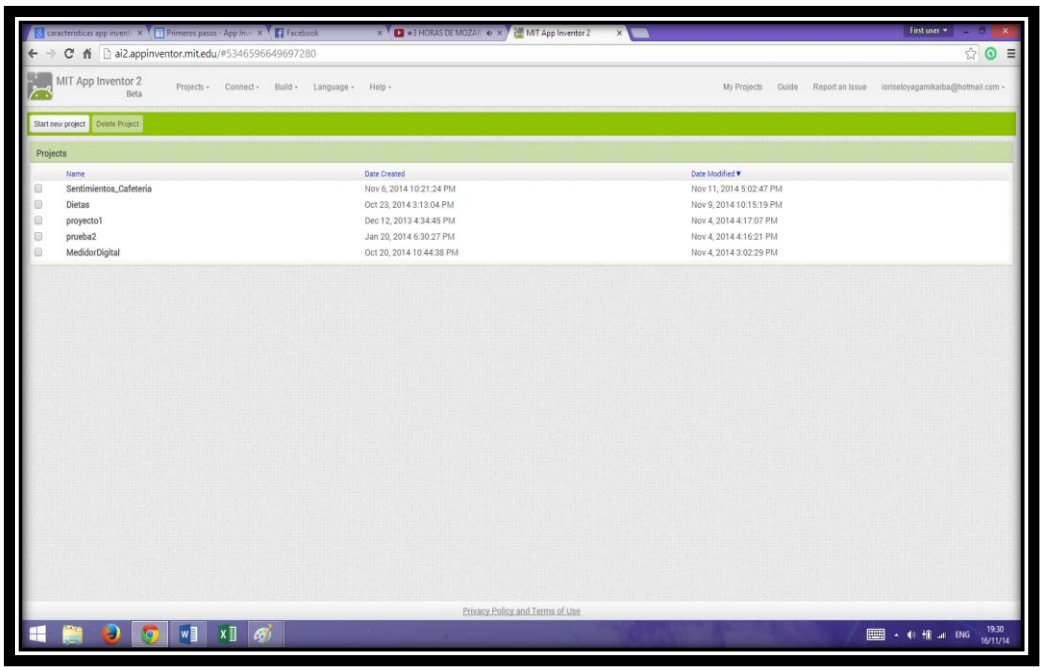
Figura 63. Conexión vía USB.



Fuente: (RICOY, 2011).

Una vez establecido la comunicación es necesario abrir una cuenta en google o trabajar con una ya creada, una vez realizado esto, se accede a la dirección http://ai2.appinventor.mit.edu, y se digita la contraseña de la cuenta ya creada, acto seguido se ingresa a una pantalla donde se muestran los proyectos ya creados si es que existen y se ofrece la opción de crear uno nuevo (Figura 64).

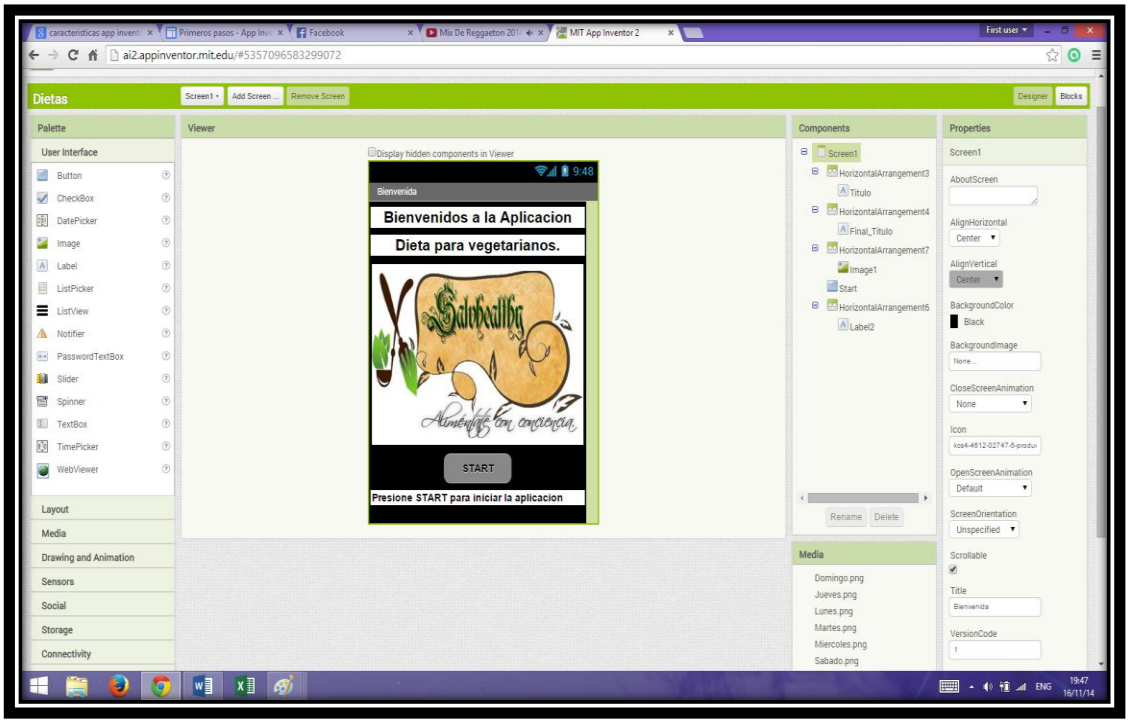
Figura 64. Pantalla de inicio del App Inventor2



Fuente: Autor.

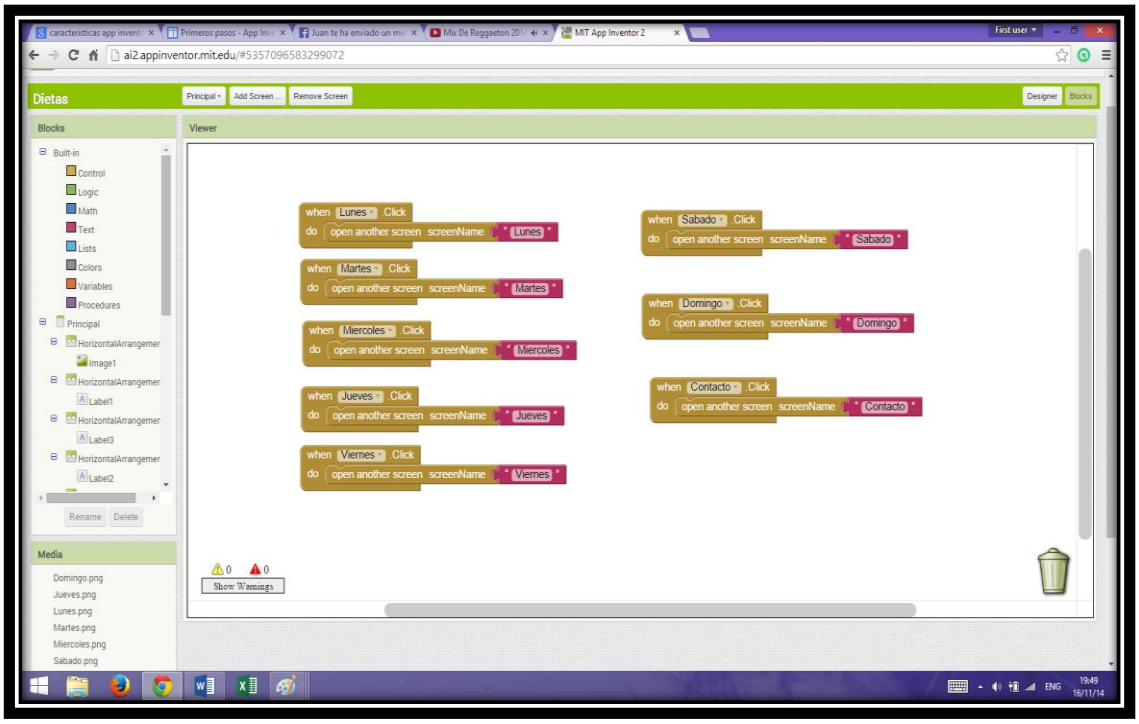
Sea que se escoja un proyecto arrancado o uno nuevo las opciones serán prácticamente las mismas para ambos, se trabajará con dos herramientas básicas: **App Inventor Designer** (Figura 65), con la que se desarrollará el interfaz gráfica con la que el usuario interactuará con la aplicación, y **App Inventor Blocks Editor** (Figura 66), o editor de bloques donde se efectúa la programación propiamente dicha, es decir, se definirán los usos, lugares y comportamiento de los componentes de la aplicación.

Figura 65. Pantalla de Diseño de App Inventor.



Fuente: Autor.

Figura 66. Pantalla de Editor de Bloques de App Inventor.

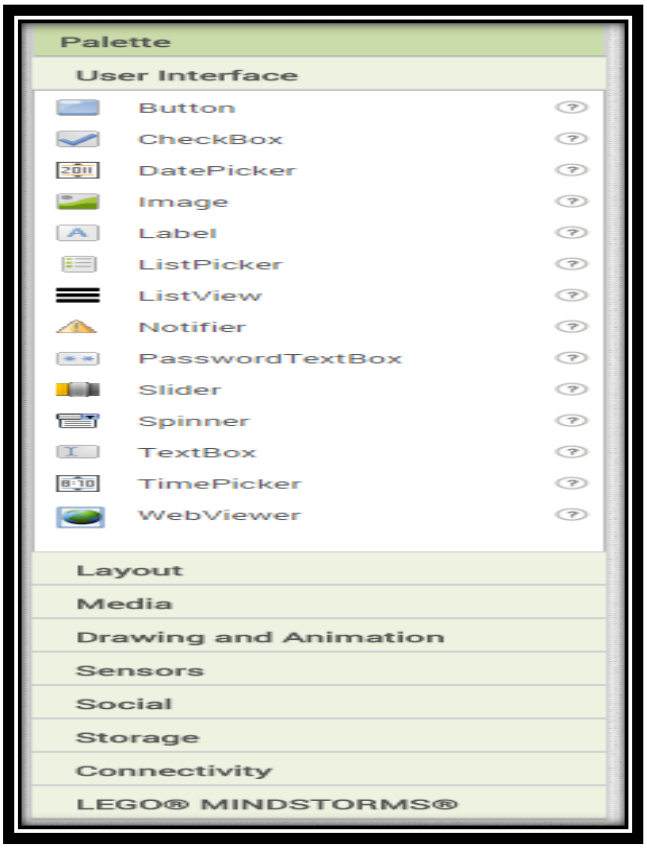


Fuente. Autor.

4.1.2.1.3 Componentes.

A los elementos básicos utilizados para desarrollar las aplicaciones se los llama componentes (Figura 67). Si se puede hacer una analogía se puede comparar a los componentes en el AppInventor con los ingredientes en una receta. Dichos componentes son de diferente naturaleza, algunos pueden ser bastante sencillos o realizan funciones relativamente simples ya sea una etiqueta (Label), cuya única finalidad es solamente mostrar un texto en pantalla, o un botón (Button) que obviamente realiza una acción al pulsarlo; otros por el contrario realizan funciones relativamente más complicadas como por ejemplo un lienzo (Canvas) que almacena imágenes o animaciones, o un sensor de movimiento (AcceleratorSensor) está en capacidad de detectar el movimiento del móvil, componentes que permite interactuar con internet, o en el caso del presente proyecto, realizar una comunicación via Bluetooth (RIEGO, 2011).

Figura 67. Componentes de App Inventor.

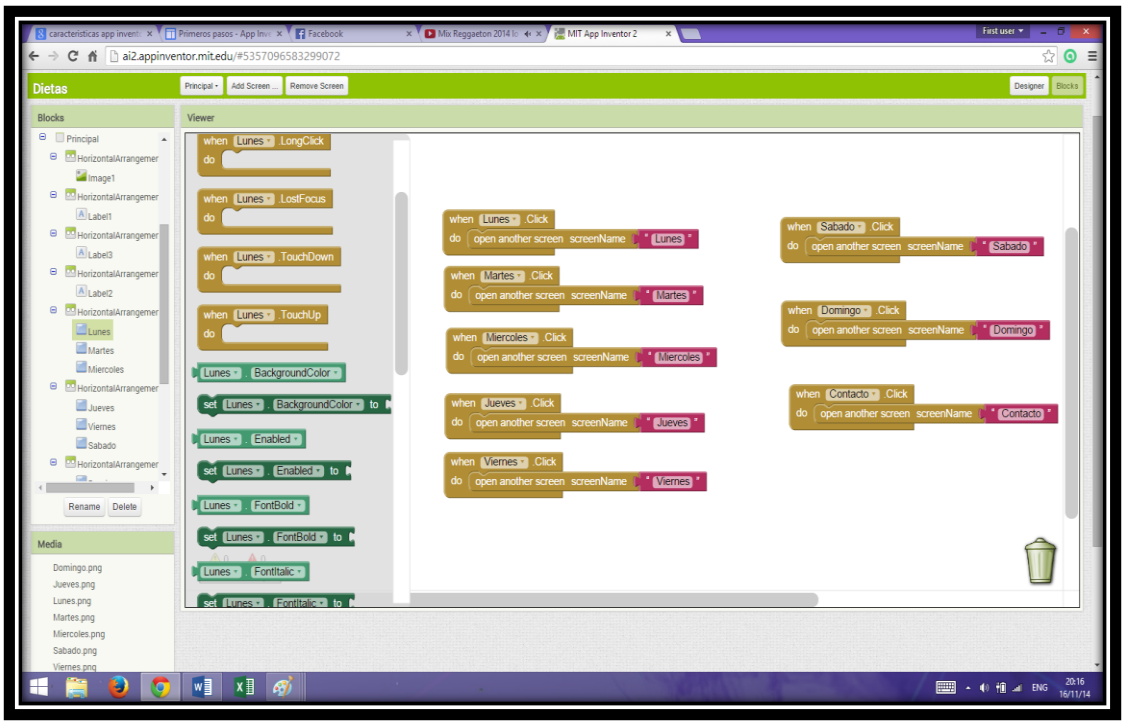


Fuente: Autor.

Para utilizar un componente dentro de una aplicación, basta con hacer clic en el mismo dentro de cualquiera de las barras de componentes, arrastrar dicho componente y soltar en el visor (Viewer) en el centro de Designer, y aparte de estar en el visor también se lo vera en la lista de componentes a la derecha del mismo como se puede apreciar en la figura 65 de la pantalla de diseño anteriormente mostrada.

Cabe mencionar, que a la par con la implementación de un componente en la pantalla de diseño, automáticamente estará su equivalente en la pantalla de bloques, en la cual, al momento se señalarla, se desplegara todas las opciones posibles de funcionamiento de dicho componente, de esta manera funciona la pantalla de bloques con todos los componentes que se utilicen, y la programación de la aplicación como tal no es más que la interacción ordenada y coherente de las opciones de programación (bloques) de los componentes (Figura 68).

Figura 68. Opciones de bloques del componente “botón” en la pantalla de Bloques.



Fuente Autor.

4.1.2.1.4 Ventajas y Desventajas de AppInventor

Entre sus ventajas podemos citar las siguientes:

- No necesita instalación de una Interfaz de Aplicaciones, pues su utilización es a través de Internet.
- Se necesita un mínimo de conocimientos de programación, pues como ya se definió anteriormente se trata de un lenguaje de programación “intuitivo”, por así decirlo.
- Desarrollo de aplicaciones en relativo poco tiempo con mínimo de error pues nos permite verificar cada avance de la aplicación y encontrar cualquier error o incongruencia rápidamente.
- Los proyectos son almacenados en la nube, por ende, se puede acceder a ellos desde cualquier lugar (BENAVIDES, 2012).

Entre sus desventajas podemos citar las siguientes:

- Ninguna aplicación puede subirse al Android Market ni ningún sitio de comercialización online similar.
- No es posible realizar varias actividades en una sola aplicación.
- Las aplicaciones desarrolladas en esta plataforma consumen mayor cantidad memoria del móvil (APK de gran tamaño) que desarrollándola en otras plataformas.
- Si bien es cierto que se trabaja online, es necesario instalar ciertos programas para emuladores y comunicación USB lo que ocupa puertos que suelen dedicarse a otras aplicaciones en el ordenador.
- No permite aplicaciones de relativa gran complejidad o precisión como por ejemplo demoras en la adquisición de gran cantidad de señales, imposibilidad de gestión de varios eventos simultáneamente, etc (BENAVIDES, 2012).

A partir de las bondades e inconvenientes que presenta, depende del juicio de cada programador escoger o no AppInventor como plataforma de desarrollo, sin embargo, es importante destacar que el lenguaje y la temática que propone esta aplicación es altamente amigable, y, en el caso del desarrollo de esta parte del tema, es altamente estética, rápida y sobre todo eficiente, razón por la cual fue elegida para el desarrollo de la aplicación requerida.

4.1.3 Comunicación Bluetooth.

Una comunicación inalámbrica es aquel conjunto de dispositivos electrónicos conectados entre sí mediante un medio y un protocolo común con la finalidad de intercambiar diferentes tipos de información o datos sin la necesidad de medios físicos para la conexión, sino más bien, la propagación de ondas electromagnéticas, de cierta frecuencia (RAMIREZ, 2007).

Se le conoce con el nombre de Bluetooth (Figura 69) a un estándar global de comunicación inalámbrica, que a su vez, hace posible la transmisión entre diferentes dispositivos electrónicos a través de un establecimiento de enlaces por radiofrecuencia (SANCHEZ, 2007), cuyas características principales son:

- Permite comunicar equipos tanto móviles como fijos.
- Suprime la necesidad de cables para conexión entre equipos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas.

Dicho de otra forma, Bluetooth es también conocido como “factor de alcance corto”, o “solución de radio a bajo costo,” Bluetooth trabaja con ondas de Radiofrecuencia de corto alcance, pues es uno de los medios de transmisión más simples, económicos, fáciles de configurar y administrar, mediante el cual, posibilita la comunicación, conexión e intercambio de información inalámbrica entre ordenadores portátiles, smartphones, impresoras, cámaras, etc mediante frecuencia de radio de corto alcance (QUEZADA, 2008).

Figura 69. Logo de Bluetooth.

Fuente: (QUEZADA, 2008)

Para desarrollar la tecnología Bluetooth fue necesaria la inclusión de empresas que tienen que ver con telecomunicaciones, electrónica e informática tales como Nokia, Toshiba, IBM, Microsoft, etc; de tal manera que se satisfaga los requerimientos de hardware, software e interoperabilidad entre equipos. Las proyecciones para esta tecnología prevén la incorporación de empresas de carácter industrial entretenimiento y electrodomésticos, de manera que se presente una proyección de conectividad total y global entre todo tipo de dispositivos (QUEZADA, 2008).

4.1.3.1 Principios de Funcionamiento.

El estándar Bluetooth, muy similar al WiFi, se basa en la técnica de *Espectro ensanchado por saltos de frecuencia* (FHSS), la cual radica en dividir la banda de frecuencia ubicada en los 2.402 a 2.480 GHz en 79 canales de 1 MHz por canal, para posteriormente transmitir la señal saltando de un canal a otro en una secuencia conocida tanto para la estación emisora como para la receptora, lo que implica, entre otras cosas, que su decodificación o intervención por agentes extraños es mucho más compleja estableciendo un mayor grado de seguridad al sistema, pues además de cambiar de canales, los dispositivos de mayor tecnología incorporan con una Clave o PIN modificable (no modificable en dispositivos de menor tecnología) que permite

controlar si se acepta una comunicación total con algún dispositivo, junto con un proceso relativamente complejo para establecer una comunicación entre equipos (COMMONS, 2014), enumerado a continuación:

1) Modo Pasivo, sin comunicación.
2) Petición de comunicación.
3) Paginación o sincronización entre equipos.
4) Establecimiento del canal.
5) Emparejamiento
6) Intercambio de información.

El número de veces en que la señal cambia de canal por unidad de tiempo (frecuencia) es de aproximadamente 1600 veces por segundo, de esta manera entre otros beneficios el estándar Bluetooth puede evitar la interferencia con otras señales radiales.

El estándar Bluetooth funciona a través de un chip denominado “transceptor” (tranceiver) que necesariamente debe ir en cada dispositivo que pretenda comunicación Bluetooth y es el encargado de transmitir y recibir la señal que este dentro de una frecuencia alrededor de los 2.4 GHz. Cada dispositivo trabaja dentro de un rango de 10 metros y dispone de una velocidad de transmisión de 3Mbps aproximadamente incluso en lugares con gran interferencia (COMMONS, 2014).

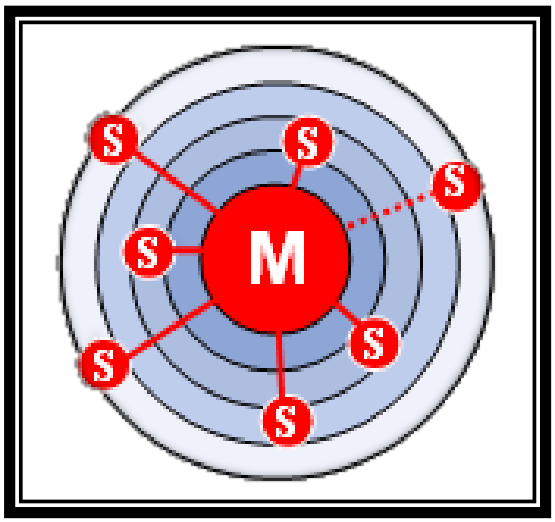
Los enlaces establecidos entre dispositivos a través de Bluetooth que están dentro de un mismo espacio físico se los conoce con el nombre de Piconet, cada equipo posee una única dirección de 48 Bits, el proceso en donde un dispositivo busca o encuentra a otro se le nombra "Inquiry", el tráfico en el canal utilizado se regula mediante el esquema Maestro/Esclavo, que consiste en convertir uno de los dispositivos en el llamado "Maestro" pues es el encargado de establecer la conexión y todos los demás dispositivos que pretenden conectarse a este a través de solicitudes de conexión son llamados "Esclavos" (Figura 70), El dispositivo maestro selecciona una de las direcciones de uno de los dispositivos esclavos y se sincroniza con este mediante paginación, que básicamente implica la sincronización de su reloj y frecuencia del dispositivo al que se conecta. Si bien es cierto que el dispositivo

maestro se puede conectar únicamente con un esclavo al mismo tiempo, posee la capacidad de cambiar a cada instante de dispositivos esclavos, de esta manera el usuario percibe una conexión simultánea con varios dispositivos esclavos al mismo tiempo aunque estrictamente no sea así. Cabe resaltar que cuando un equipo identifica a otro la comunicación se establece de forma "automática" por así decirlo, pues no es necesario ningún tipo de configuración para efectuar la misma, y se la conoce como red de área personal PAN (COMMONS, 2014).

La transmisión de datos a su vez, se establece mediante paquetes intercambiables entre la unidad maestro y uno de los dispositivos esclavos, cada paquete está formado por cierta cantidad de bits destinados a cierto propósito, mismos que se pasa a enumerar a continuación:

- 72 Bits correspondientes al código de acceso, es decir, identifican el dispositivo maestro.
- 54 Bits correspondientes a la cabecera del paquete que contiene información diversa sobre el tipo de datos que se envían, la dirección (MAC), bits de control de flujo de datos, si existe error de cabecera, etc.
- De 0 a 2745 Bits correspondientes a los datos propiamente.

Figura 70. Gráfico explicativo de la conexión maestro/esclavo.



Fuente: (COMMONS, 2014).

4.1.3.2 Características:

Entre las características de la tecnología Bluetooth se puede citar las siguientes:

- Frecuencia aproximada: 2.4 GHz.
- Potencia de transmisión: 1mW para 10 metros.
- Canales para transmisión de voz: máximo 3 por piconet
- Canales para transmisión de datos: máximo 7 por piconet
- Velocidad de datos: 721 Kbps por Piconet
- Rango de 10 metros de cobertura.
- Alimentación aproximada de 2.7 Voltios.
- Interferencia mínima, saltos de canal en frecuencia de 1600 veces / segundo (RAMIREZ, 2007).

Cabe mencionar que la tecnología bluetooth continuamente ha estado renovando continuamente sus versiones, mejorando en cada una de ellas sus características en cuanto a velocidad de transmisión de datos, área de cobertura, numero de dispositivos, etc.

4.1.3.3 Ventajas y Servicios.

Gracias al estándar Bluetooth, existe una amplia gama de servicios (perfiles) que puede prestar la comunicación e intercambio de datos o información a través de bluetooth, tales como:

- Audio
- Video
- Audio y Video
- Imágenes
- Impresiones.
- Transferencia de archivos
- Fax
- Manos libres
- Auricular
- Intercambio de objetos.
- Intercomunicador

- Acceso de área local LAN
- Acceso de área personal PAN. (COMMONS, 2014)

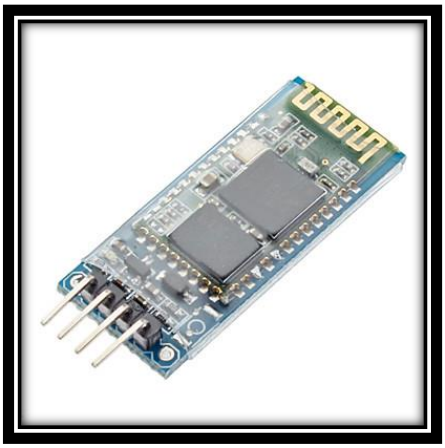
Hay pocas desventajas no dignas de mención en el estándar Bluetooth siempre y cuando se hable estrictamente de comunicación inalámbrica a corto alcance, por esto, salta a la vista el porqué de la utilización de la tecnología bluetooth para realizar la comunicación necesaria en aplicación que proporciona el estado del medidor (VILLACRESES, 2010).

4.1.4 Módulo de Comunicación Bluetooth

Es un dispositivo electrónico capaz de establecer comunicación inalámbrica, concretamente bluetooth para enviar o recibir información. Este dispositivo, toma los datos enviados a través de la comunicación serial propia y programable desde el microcontrolador y los envía de la misma manera, con la diferencia que lo hace a través del protocolo bluetooth, lo que posibilita una comunicación inalámbrica.

En el desarrollo del presente proyecto se utilizó el modulo bluetooth HC-05 (Figura 71), mismo que se adquiere configurado de fábrica para desenvolverse tanto como maestro o esclavo. En el modo esclavo este módulo puede a su vez conectarse con otros módulos bluetooth mediante peticiones, mientras que en el modo maestro espera peticiones de conexión de otros dispositivos, que es el caso particular que se emplea para la aplicación, pues de esta manera se podrá ver desde un celular o una laptop toda la información referente al desempeño del medidor de consumo de agua potable (BOTSCIENCE, 2013).

Figura 71. Módulo Bluetooth HC5.



Fuente: (CNA, 2014)

4.1.4.1 Características.

Entre sus principales características tenemos:

- Alimentación entre 3.3 / 5 v.
- Chip Bluetooth Incorporado BC417143
- Alcance máximo: 10 mts
- Nivel TTL
- Velocidad entre 1200bps a 1.3Mbps (BOTSCIENCE, 2013).

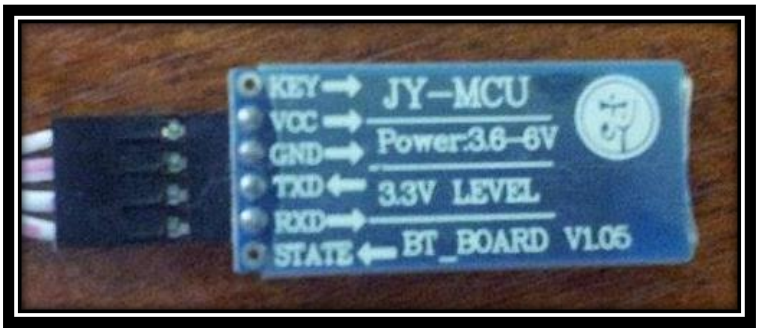
4.1.4.2 Pines de Conexión y conexión con placa Arduino.

El modulo Bluetooth presenta cuatro pines de conexión claramente diferenciados (Figura 72):

- **VCC.** Para alimentación.
- **GND.** Tierra, también utilizado alimentación.
- **TXD.** Para transmisión de información.
- **RXD.** Para Recepción de información. (BOTSCIENCE, 2013)

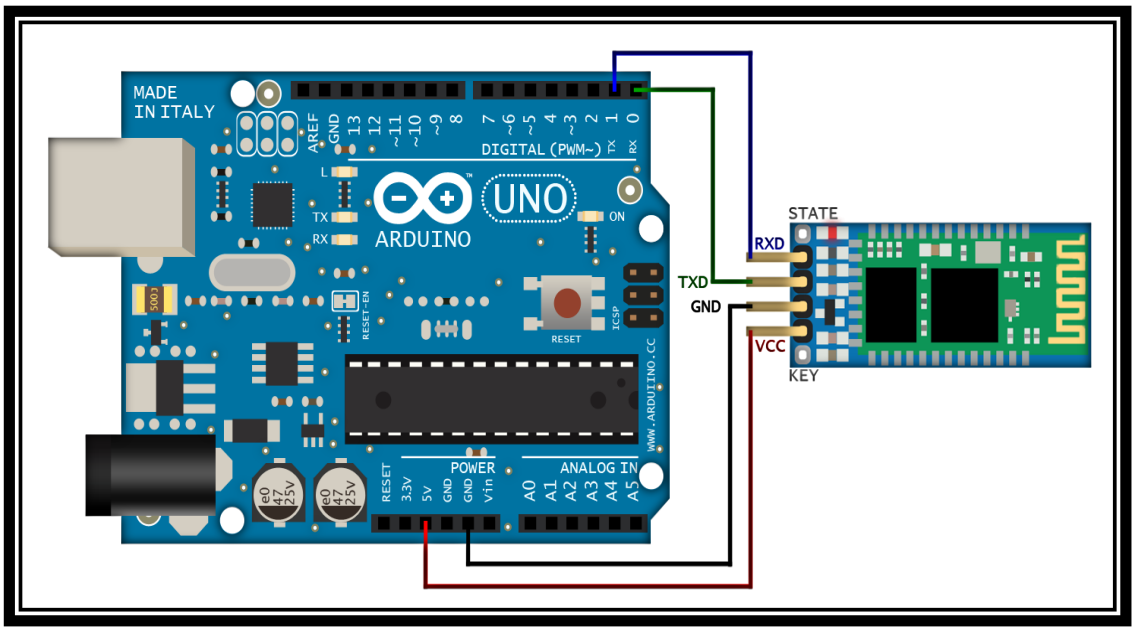
Para la conexión con la placa Arduino solo es necesario especificar que los pines de transmisión y recepción del módulo se conectan intercalados con los pines de recepción y transmisión de la placa Arduino, es decir, el pin de transmisión del módulo se conecta al pin de recepción de la placa, y a su vez el pin de recepción del módulo se conecta con el pin de transmisión de la placa (Figura 73).

Figura 72. Pines de conexión del módulo.



Fuente: Autor

Figura 73. Conexión Física del Módulo con placa Arduino.



Fuente: (CNA, 2014).

4.2 Aplicación Android para Monitoreo de Estados del Equipo.

Una parte muy importante del presente tema, tiene que ver con el desarrollo de un sistema de monitoreo que permita generar un pequeño pero conciso informe o notificación del funcionamiento del equipo, que permita conocer qué parte del equipo funciona y qué parte requiere reparación o mantenimiento, y en un todo, un sistema que proporcione un informe del estado del equipo de forma rápida y eficiente.

Y es así como nace la aplicación de monitoreo, que permite, a través de un sistema de comunicación inalámbrico, comunicar el medidor electrónico con un dispositivo móvil portable por medio del cual, se visualizará el estado y funcionamiento del medidor en cuestión.

De acuerdo al diagrama de flujo mostrado en la Figura 74, el esquema del sistema de monitoreo funciona de la siguiente manera:

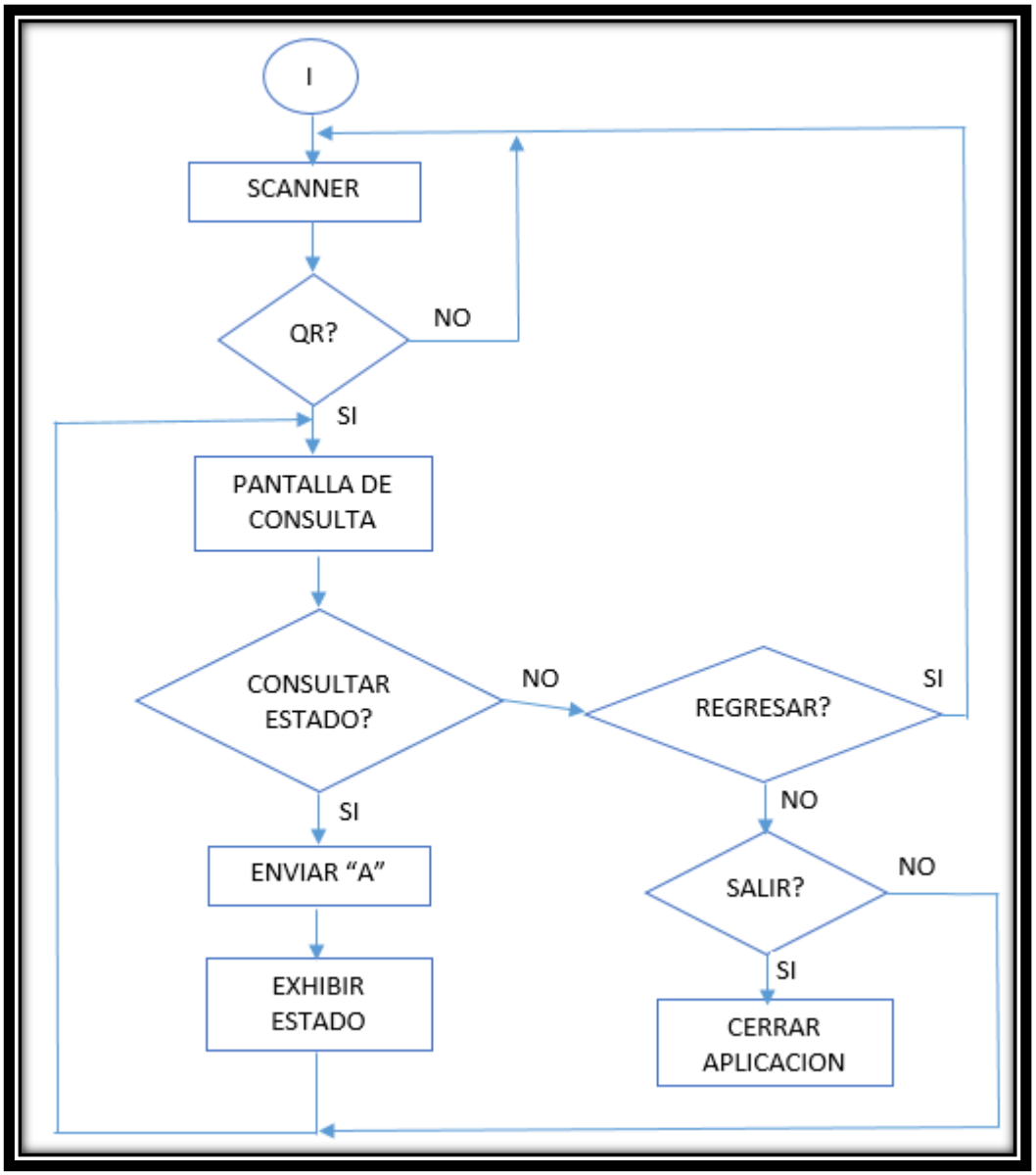
El medidor electrónico envía datos del estado del equipo a través de un módulo de comunicación bluetooth, cuyo funcionamiento se explicó anteriormente; dicha información es receptada en un dispositivo móvil que cuente con un sistema bluetooth incorporado y un software que permita recabar los datos de estado y los exhiba en la pantalla, cabe mencionar que con solo acercar el dispositivo móvil al rango de alcance del módulo de comunicación, se establece automáticamente el sistema de monitoreo descrito.

Como ya se supone, el software utilizado para desarrollar la aplicación de monitoreo está basado en App Inventor, su desarrollo pasa a mostrarse a continuación:

4.2.1 Desarrollo del Software de Monitoreo.

Como se puede discernir partiendo del diagrama de funcionamiento (Figura 74), la aplicación comienza abriendo un scanner de video desde el móvil, que realiza un barrido continuo en busca de un código QR, que como se sabe, es un sistema que posibilita el almacenamiento de información a través de una especie de código de barras de última generación, misma información que se recupera a través de un scanner de código QR instalado en el móvil, solo basta con apuntar la cámara hacia el código QR.

Figura 74. Diagrama de flujo del funcionamiento de la aplicación.



Fuente: Autor.

El código QR en cuestión, contiene el número MAC del medidor, en el momento en que el scanner percibe el código establece la comunicación Bluetooth entre el medidor y el dispositivo móvil, luego de lo cual pasa a una pantalla donde tiene tres opciones puestas a través de botones:

- **Consultar el estado del equipo. (Botón Consultar).** Al momento en el que se presiona el botón "Consultar", lo que se hace es enviar una letra "A" al puerto que se comunica con el microcontrolador, concretamente el Serial, la

razón de esto es que, en el microcontrolador, existe una segmento de código dedicado a comunicación Serial, donde se exhiben los datos del medidor a través del mismo, sin embargo, para que esto ocurra, es necesario enviar un carácter, en este caso la letra "A" para controlar el número de veces que envía, y por ende se visualiza, el estado en la pantalla Serial, de lo contrario exhibiría el estado del equipo continuamente de forma inapreciable. Es importante recordar que la comunicación bluetooth funciona de la misma manera que la comunicación serial, es más, podría decirse que se está tomando la información enviada a través del serial y se envía por bluetooth, por lo tanto el mismo principio funciona, es decir, cada vez que el usuario pulsa el botón "Consultar" envía una letra "A", a través de la cual, se exhibe los datos de estado del medidor. Como se verá posteriormente, los datos tomados del equipo se ordenan y envían a cada una de las etiquetas correspondientes donde se alberga cada variable: "Estado", "MAC Medidor", "Consumo", "Caudal".

- **Volver a realizar el Escaneo del código. (Botón Regresar).** Cuando el usuario presiona el botón "Regresar", se envía una orden para el retorno a la pantalla donde se habilita el scanner de código QR, es decir, puede decirse que la aplicación empieza desde el principio.
- **Salir de la aplicación. (Botón Salir).** Naturalmente, como es de esperarse, cuando el usuario presiona dicho botón, la aplicación se cierra.

4.2.2 Pantallas y bloques de programación.

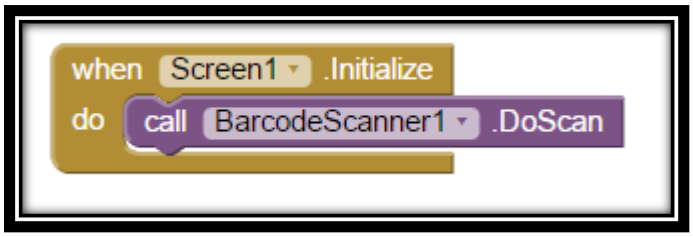
Como ya se mencionó anteriormente, App Inventor incluye entre su lenguaje de programación la habitación de diferentes pantallas de acuerdo a los requerimientos de cada aplicación. Cada una de estas pantallas contiene los respectivos elementos que hacen que las pantallas interactúen entre si y la aplicación como tal se desempeñe según lo esperado.

La aplicación de monitoreo cuenta con dos pantallas, la primera llamada SCREEN1 que se ejecuta al momento de iniciada la aplicación, y la segunda llamada MEDIDOR que se dispara después de escaneado el código QR. Sus diversos elementos se pasan a analizar a continuación.

4.2.2.1 Pantalla SCREEN 1

En el momento que se ingresa a la aplicación se despliega la pantalla SCREEN1, cuando esta se inicializa, llama al Scanner para realizar un barrido en busca de un código QR (Figura 75).

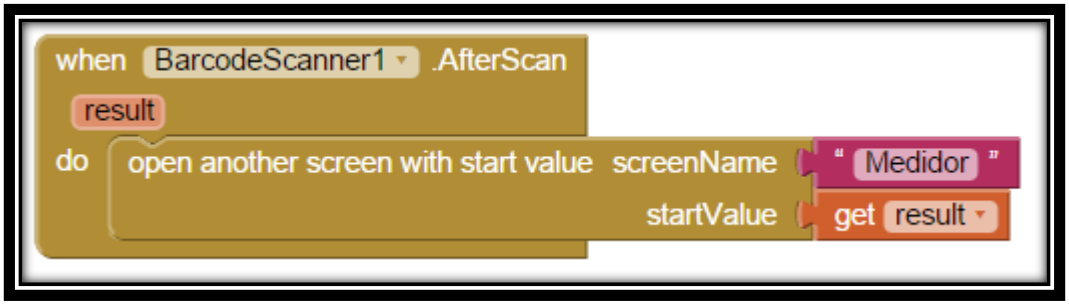
Figura 75. Inicialización de la pantalla SCREEN 1 y llamada al scanner.



Fuente: Autor.

En el momento que detecta un código QR obtiene la información del mismo (MAC del medidor) y despliega la pantalla MEDIDOR. (Figura 76).

Figura 76. Segmento de código que dispara la pantalla MEDIDOR.



Fuente: Autor.

4.2.2.2 Pantalla MEDIDOR

Una vez iniciada la pantalla MEDIDOR, simultáneamente se inicializa la variable global “datos” (Figura 77), que se utilizará posteriormente.

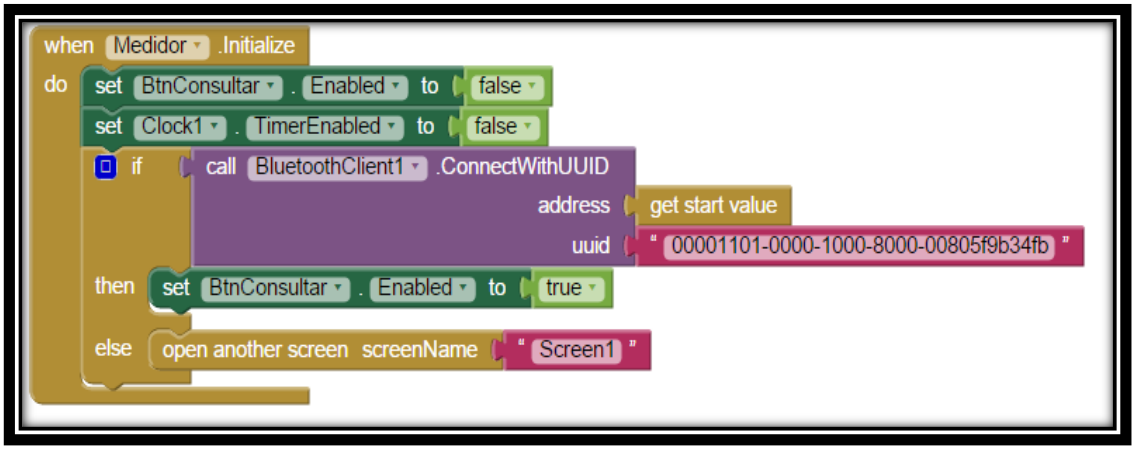
Figura 77. Variable global “datos”.

Fuente: Autor.

Junto con la variable global “datos”, cuando la pantalla MEDIDOR inicia, ocurren los siguientes eventos (Figura 78):

- Se inhabilita el botón de consulta.
- Se inhabilita el timer.
- Se pregunta si se estableció la comunicación Bluetooth.
- Se habilita el botón de consulta.
- Si no se estableció la comunicación regresa a la pantalla SCREEN1 en busca del código.

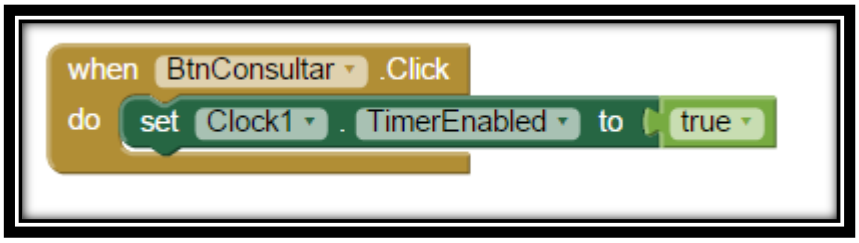
Figura 78. Segmento de código que hablita comunicación y consulta.



Fuente: Autor.

Obviamente debe existir un segmento de código dedicado al botón de consulta (Figura 79), en el momento que este es presionado se habilita el timer, que es en donde se efectúa la comunicación propiamente dicha.

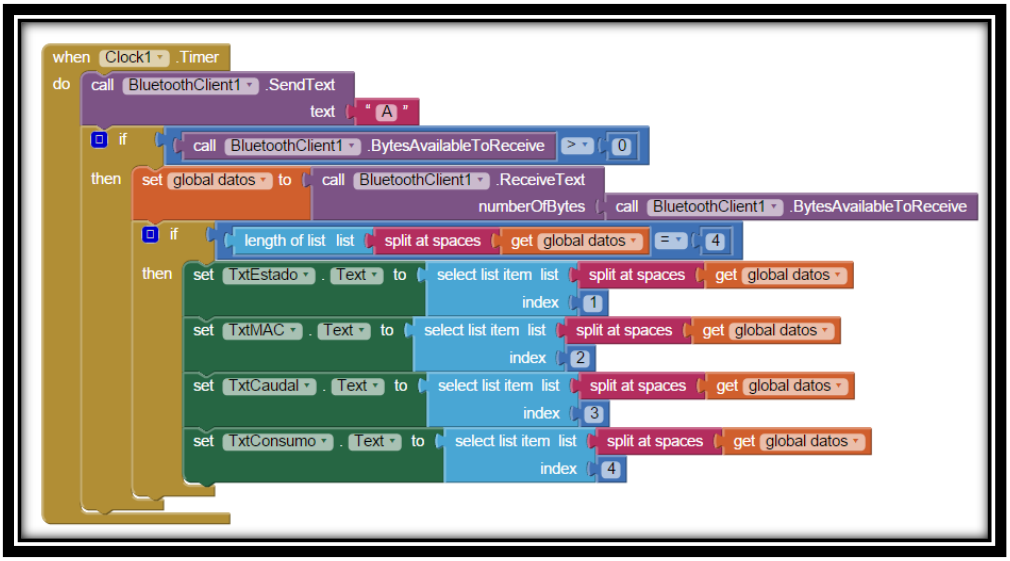
Figura 79. Segmento de código del botón consultar.



Fuente: Autor.

Cuando se presiona el botón “Consulta”, lo que sucede es la habilitación el evento “timer”, en donde se envía una señal, concretamente el carácter “A” explicado anteriormente, a través del módulo bluetooth para que inicie la adquisición de los datos enviados por la placa Arduino, lo que se obtiene es una cadena (string) con toda la información concerniente al estado del equipo, estos datos recopilados se deben toquenizar, es decir, desensamblar la cadena de datos en cada una de sus partes, variables o datos individuales, para presentarlos de forma organizada en la pantalla (separados por espacios) y conforme se efectúa el proceso de toquenizaje se guarda en las variables respectivas (Estado, MAC, Caudal, Consumo) de la variable global datos inicializada desde el principio (Figura 80).

Figura 80. Establecimiento de la comunicación Bluetooth y almacenamiento de información.



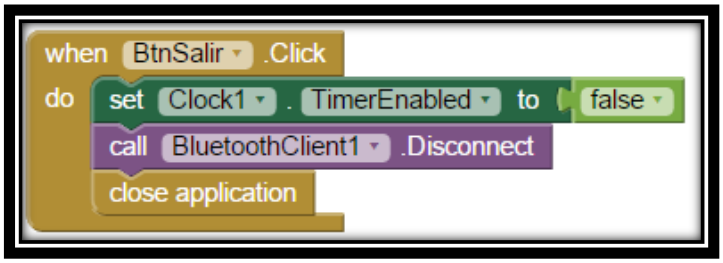
Fuente: Autor.

Naturalmente, solo quedaría por elaborar el segmento de código dedicado a los botones “Salir” (Figura 81) y “Regresar” (Figura 82), que tienen un esquema parecido.

Cuando se presiona el botón “Salir” ocurren los siguientes eventos:

- Se inhabilita el timer.
- Se desconecta el bluetooth.
- Se cierra la aplicación.

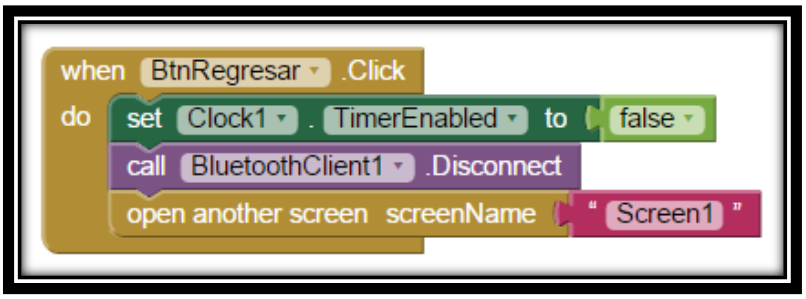
Figura 81. Segmento de código dedicado al botón “Salir”.



Fuente: Autor.

Por otro lado, Cuando se presiona el botón ”Regresar” ocurren los eventos descritos con el botón “Salir” con la diferencia de que en lugar de salir de la aplicación, se dicta una sentencia que regresa a la pantalla SCREEN1 para escanear códigos QR.

Figura 82. Segmento de código dedicado al botón “Regresar”.



Fuente: Autor.

Por último la presentación que se visualizará en el dispositivo móvil en donde se monitoreará el estado del equipo es la que se aprecia en la Figura 83.

Figura 83. Presentación final de la aplicación.

Fuente: Autor.

CAPITULO 5

SISTEMAS DE RESPALDO

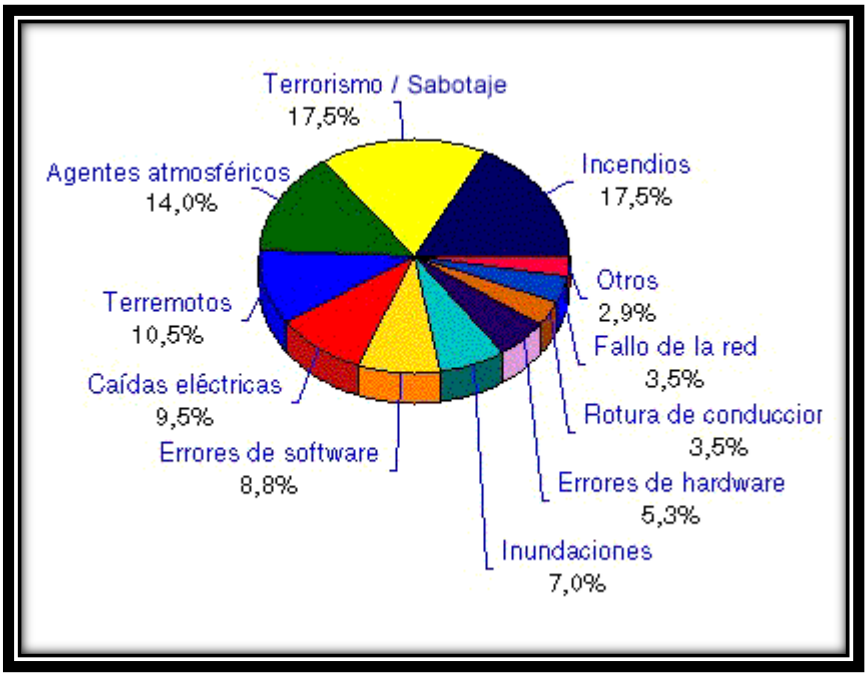
5.1 Software de Respaldo

5.1.1 Backup

No es necesario mencionar la importancia de la información, datos, o señales adquiridas para cualquier empresa, negocio, proyecto o en general cualquier sistema informático y/o electrónico. Normalmente se utiliza un medio común para el almacenamiento de los mismos, por lo general, un ordenador que, por lo general, no posee suficientes características de hardware y/o software que permita asegurar el mantenimiento de dicha información. A diferencia de la mayoría de periféricos en un computador, los cuales, en caso de avería simplemente son intercambiables sin que esto represente mayor inconveniente, una avería en el sistema de almacenamiento, dígase disco duro, o el hardware que lo gestiona, podría representar un daño irreversible, pues la pérdida de la información para cualquier empresa se traduce, a la larga o a la corta, en pérdida de tiempo, dinero y reputación (PENA, 2012).

La probabilidad de pérdida de información es una realidad latente en todo sistema que almacene datos, pues, la tecnología, sin importar su calidad, no está exenta de fallas, ya sea directas o indirectas, que se traducen en pérdida de valiosa información. Por ende, las empresas no escatiman en sistemas que respalden su información ante cualquier fallo sin importar de donde provenga (PENA, 2012)(Figura 84).

Figura 84. Diagrama de pie del porcentaje de diferentes causas de fallos y pérdidas de información.



Fuente. IBM (PENA, 2012).

La solución lógica ante esta posible contingencia es obviamente contar con copias de seguridad de la información requerida, actualizar con relativa frecuencia dichas copias esperando no tenerlas que usar. A este mecanismo se le conoce como respaldo, con su correspondiente palabra inglesa también usada en muchos medios: “backup”. Respaldar los datos significa principalmente copiar el contenido de los datos de algún sistema informático y almacenarlo en un medio ajeno a dicho sistema que cumpla con características de confiabilidad y se encuentre en un lugar seguro. De ahí que si por alguna razón tanto el medio físico como informático donde se almacenan los datos se perdiese, se lo pueda recuperar con relativa facilidad haciendo uso de los respaldos.

Las copias de seguridad o backup constituyen una de las partes más importantes a la hora de establecer un sistema de adquisición y almacenamiento de datos.

De las amplias clasificaciones se puede enlistar a los respaldos en 3 tipos:

- **Total.** Al momento de ejecutar este tipo de respaldo se guardan todos los archivos. Si los datos que se van a respaldar se mantienen intactos, el respaldo no será más que una copia exacta del primero.
- **Incremental.** Este tipo de respaldos primero revisan si los datos han cambiado desde la última vez que se realizó un respaldo, de ser así, no efectúa el backup, caso contrario, si los datos han cambiado, realiza las copias respectivas.
- **Diferencial.** Muy parecido al anterior, excepto por la diferencia de que este tipo de respaldo almacena cada cambio que se va realizando al momento de realizado el respaldo, es decir, son acumulativos, y contienen todos los cambios realizados desde que se hizo el ultimo respaldo Total. (GANDARILLA, 2012)

Puesto que un respaldo total requiere una gran cantidad de memoria de almacenamiento, no es eficiente realizar este tipo de respaldo a cada momento a diferencia de un respaldo Incremental cuya variación en lo que a memoria se refiere, es sumamente menor al respaldo total, por lo tanto lo que se acostumbra hacer es utilizar un respaldo total y un Incremental, y realizar un respaldo total con mucha menos periodicidad que un diferencial, por ejemplo un respaldo total una vez al mes y uno Incremental una vez a la semana.

5.1.2 Media de Respaldo.

El término "media de respaldo" tiene que ver con el dispositivo de memoria físico en donde se almacena la información del backup (HAT, 2005). Hasta hace algún tiempo los dispositivos basados en cintas eran los únicos utilizados para propósitos que tienen que ver con respaldo, sin embargo, dependiendo de la naturaleza del sistema, hoy en dia existen diversos dispositivos que se utilizan como dispositivos de almacenamiento para respaldar la información como son los siguientes:

- Discos Duros.
- Discos Ópticos.
- Memorias extraíbles (flash)
- Cintas magnéticas.

Por otro lado, lo que hace que la información sea almacenada en los dispositivos antes mencionados es, lógicamente, un software, existe diversa cantidad de software dedicado a tares de respaldo tales como, las imágenes (copias) de un disco o una partición del mismo, protección total respaldando todo el pc incluyendo el sistema operativo; cabe mencionar que en los últimos años, existe la posibilidad de almacenamiento de información en la nube, es decir un espacio virtual en donde a través de internet se puede guardar información desde un ordenador y acceder a ella desde cualquier otro lugar, hay que considerar que todos estos sistemas poseen tanto ventajas como desventajas al momento de ser evaluados, para escoger la solución o soluciones más óptimas para cada aplicación (HAT, 2005).

5.1.3 Ventajas y Desventajas de la implementación de Respaldos.

Las ventajas de implementar sistemas de respaldo de información son de hecho obvias, de todas maneras, es necesario listar las que se considera más importantes para efectuar un contraste:

- Los respaldos son un medio eficaz que garantiza la información.
- No requieren ordenamiento.
- No requieren una acción adicional por parte del usuario pues su ejecución es automática.
- En el caso de los respaldos en la nube, no requieren ningún medio físico de memoria de almacenamiento.
- Son independientes del funcionamiento del ordenador.

Puesto que la necesidad de implementar sistemas de respaldo nace de la alta probabilidad de que los equipos o sistemas fallen, aunque es poco probable, los sistemas de respaldo, de hecho, pueden presentar fallas, de esta premisa nacen sus posibles desventajas enumeradas a continuación:

- Dependiendo del medio que se utilice para elaborar los respaldos, dicho medio puede generar respaldos absolutos, lo que incluye información innecesaria.
- Los respaldos no hacen distinción en los datos que deben almacenar, es decir, almacenan información aun cuando contenga posibles virus o sean virus en sí.

- No respalda drivers, lo que implica que de darse el caso, si se pierde el ordenador completo es difícil restaurarlo aun cuando se disponga de la información respaldada.
- En el caso del almacenamiento en la nube, esta es susceptible a ser crakeada o interceptada por usuarios no autorizados, lo que supone un peligro en el caso de información que debe ser confidencial. (PENA, 2012).

A pesar de lo antes mencionado, la ventaja de contar con un sistema de backup es fundamental en cualquier proyecto, para lo cual es necesario analizar todas las opciones de las que se dispone y los requerimientos particulares de cada proyecto, dando por hecho que sea cual sea el sistema que se vaya a implementar, está implícito el mismo un sistema adjunto dedicado a respaldar la información eficientemente.

En el desarrollo del presente proyecto se ha implementado un pequeño pero eficiente sistema de respaldo de software principalmente demostrativo, el cual, realiza una copia de la base de datos cada mes, es decir, al momento que el sistema reconoce el primero de cada mes, se efectúa una copia de la base de datos y se la almacena en el disco duro del computador, en el cual a su vez, se lo puede extraer y clonar a través de cualquier dispositivo de memoria extraíble, de esta manera se cuenta con un sistema de respaldo de información.

Sin embargo, en el caso particular del presente proyecto de tesis, los respaldos implican algo más que un software que almacena los datos recabados de la medición del consumo de agua potable, implican también, como se verá a continuación un sistema que garantice la continuidad del funcionamiento del medidor electrónico, es decir, un equipo adicional dentro del medidor que garantice el funcionamiento del mismo en caso de presentarse fallas externas, concretamente, desabastecimiento de energía eléctrica. De ahí que el presente capítulo se divida en dos partes: el respaldo a la base de datos ya antes mencionado, y el respaldo al medidor electrónico, con un especial análisis a este último por ser su desarrollo algo propio del proyecto, y no un procedimiento genérico y prácticamente implícito, como lo es el respaldo al software.

5.2 Sistema de Respaldo del Equipo de Medición de Consumo.

Puesto que el medidor electrónico no es un elemento que guarde información, el término respaldo se refiere a un mecanismo o sistema que el medidor electrónico funcione de manera ininterrumpida.

Puede decirse que el equipo de medición es el elemento más importante de todo el proyecto, si por cualquier razón, y considerando que cualquier equipo está sujeto a posibles fallos, el medidor dejara de funcionar ya sea parcialmente o en su totalidad, esto representaría una enorme pérdida para la empresa, pues en todo el tiempo que el equipo no funcionara, dejaría de enviar los datos de consumo. Si bien es cierto, independientemente del funcionamiento del medidor, el servicio no se interrumpe, el hecho de que el equipo no envíe la información de consumo hacia la base de datos, representa una pérdida enorme para la empresa proveedora, por ende, una parte vital en el desarrollo del medidor electrónico es, necesariamente, un sistema adicional que incremente las garantías de continuidad en el funcionamiento del equipo.

Puesto que ya se abordó las precauciones y medidas que se toman para fallos internos propios del medidor en el capítulo 4, lo que se quiere decir cuando se habla fallos en la continuidad del funcionamiento del equipo es los posibles eventos de carácter externo, concretamente un posible corte del servicio de energía eléctrica.

Obviamente el medidor electrónico funciona gracias a la energía eléctrica obtenida desde la propia vivienda en el que va instalado. Si bien es cierto en las actuales circunstancias, un corte de energía es poco probable, eso no significa que nunca pudiera suceder, y mucho menos, no sea necesario contar con un mecanismo mediante el cual se proporcione una solución a la posible contingencia.

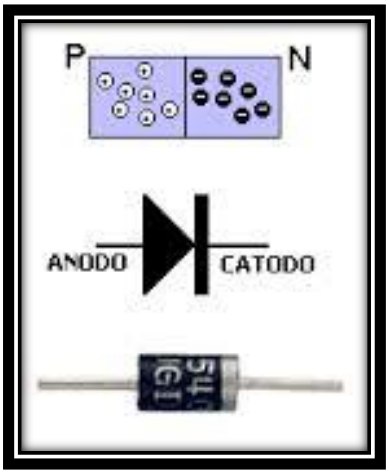
5.2.1 Placa Electrónica de Respaldo

La solución a la problemática de realizar un respaldo al equipo es bastante simple, un circuito electrónico que tome la energía eléctrica de otra fuente portátil preparada en el caso de un posible corte de energía eléctrica en la vivienda.

Basado en un par de diodos 1N4007 (Figura 85), dicho circuito funciona como un conmutador basado en una comparación, los diodos se colocan con sus respectivos cátodos uno a continuación del otro, y a continuación de sus cátodos, una batería de 9 voltios y el respectivo adaptador de 12 voltios que va conectado a la fuente,

concretamente el tomacorriente de la vivienda. Sin ahondar demasiado en aspectos técnicos sobre componentes electrónicos, los diodos funcionan como interruptores, cuando se energizan con la diferencia de potencial correcta (polarización directa) se cierran y conducen la energía, por otro lado cuando se energizan con una diferencia de potencial inversa (polarización inversa) se abren y no permiten el paso de la corriente eléctrica. Esta es la razón por la que la batería tiene un valor de 9 voltios y la alimentación de la vivienda un valor de 12 voltios.

Figura 85. Diagrama de la composición del diodo.



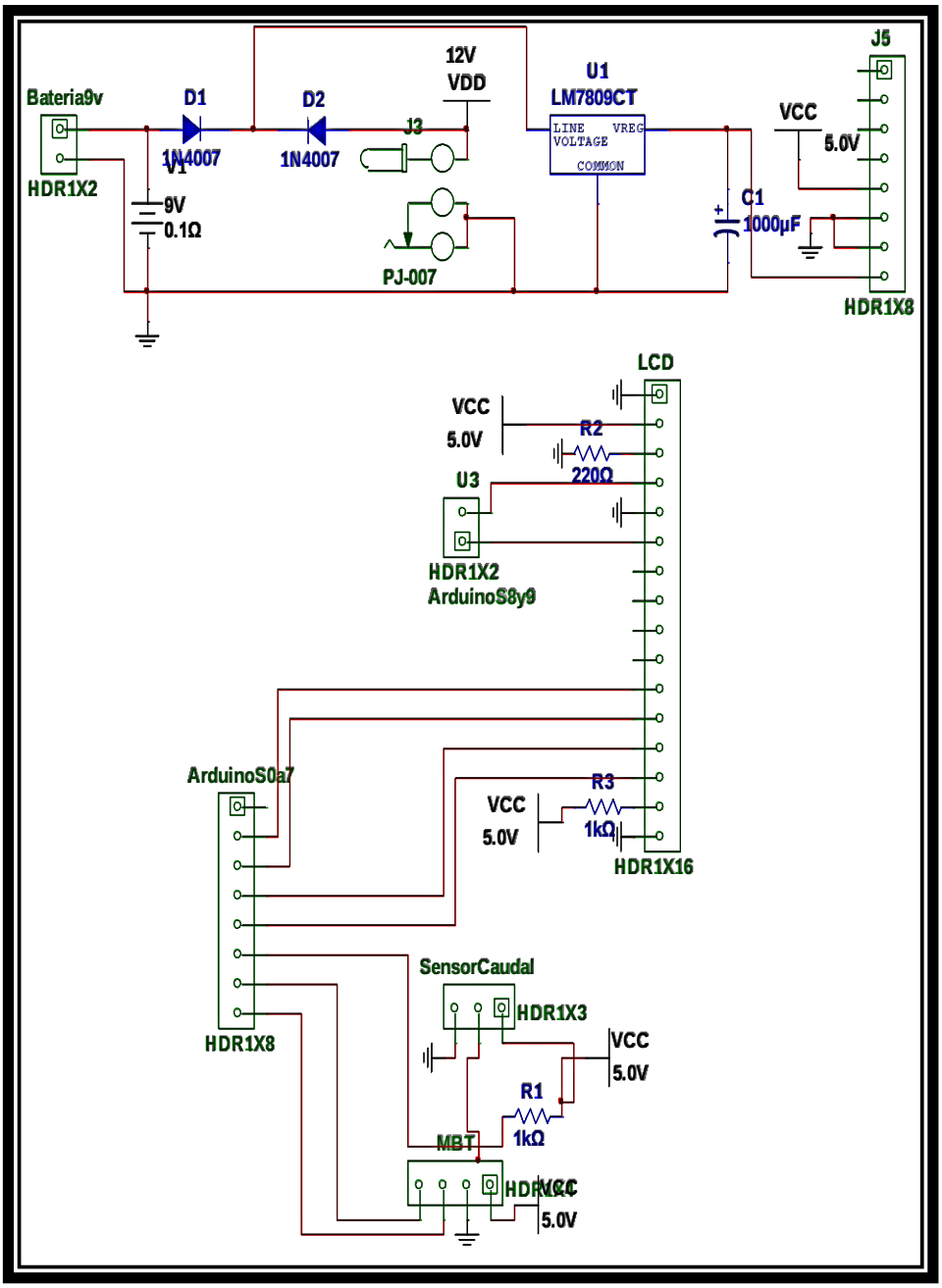
Fuente: (CONALEP, 2012).

En condiciones normales de alimentación, la disposición de los diodos hace que el sistema tome la energía eléctrica proveniente de la fuente con mayor potencial, es decir, la fuente de 12 voltios proveniente del domicilio obviamente superior a la fuente de 9 voltios proveniente de la batería. Sin embargo, cuando por cualquier razón, existe un corte en la energía, la fuente no entrega voltaje, es decir, entrega un potencial de cero, en este caso, al hacer la comparación entre la fuente y la batería, lógicamente el diodo al que va conectado dicha batería se alimenta directamente y por ende conduce la energía eléctrica proveniente de la batería, y el circuito sigue funcionando a pesar del corte de energía eléctrica.

Cabe mencionar que el circuito cuenta con un condensador, que acumula la energía suficiente para que el medidor no deje de funcionar en el pequeño instante en que sucede este cambio de fuentes.

Para un diseño más compacto, junto con el circuito de respaldo (Figura 86), se implementó en dicho circuito los requerimientos para montarlo en la placa como un tercer escudo, a través del cual, van conectados los periféricos, tales como, el display y el módulo de comunicación bluetooth, de esta manera no solo se provee un soporte en caso de alguna contingencia de carácter eléctrico, sino también se mantiene las características de un medidor compacto fácil de implementar a escala real.

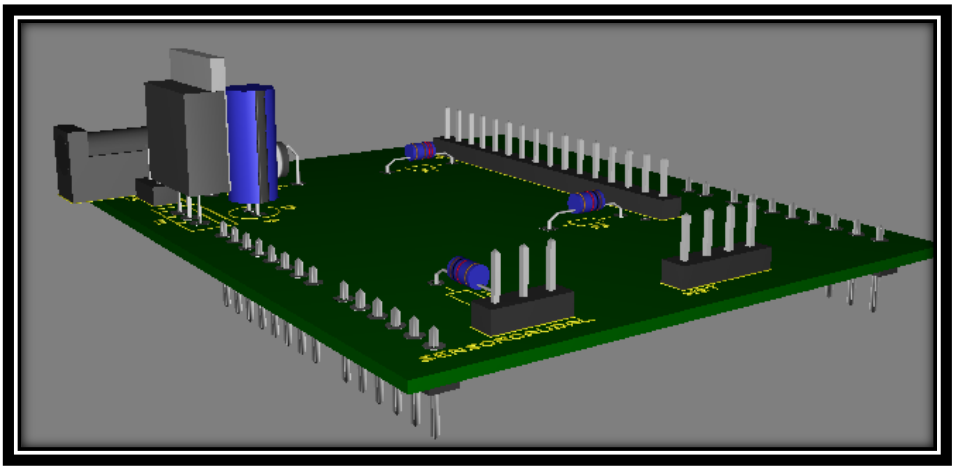
Figura 86. Circuito de la placa electrónica de Respaldo.



Fuente: Autor.

Gracias a software de diseño de placas electrónicas que permiten, no solo diseñar dichos equipos, sino también simularlos y muy importante, generar el diseño final que va sobre la placa real, una vez comprobado su funcionamiento, se puede ver un esquema modelado de lo que será el circuito real con todos sus componentes generado por el software, que como ya se mencionó, permite implementarlo sobre la placa original a modo de un tercer escudo (Figura 87).

Figura 87. Simulación del circuito de respaldo.

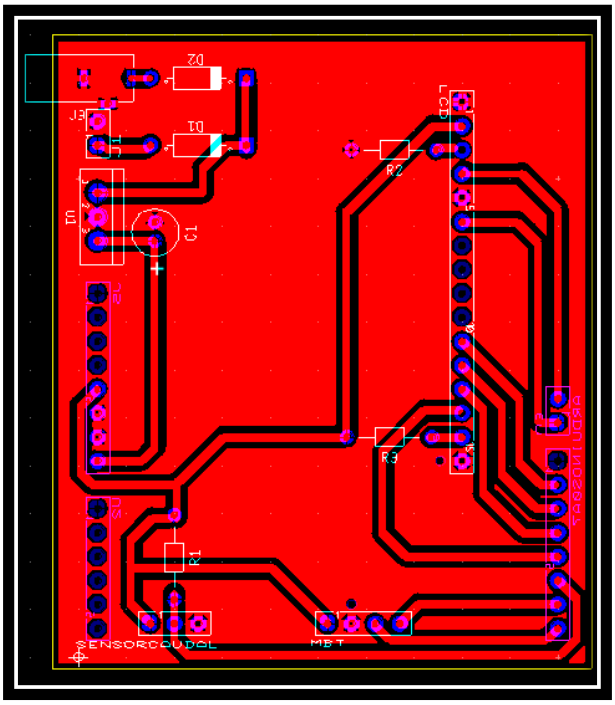


Fuente: Autor.

Una vez probado y simulado, se pasa a generar el diseño del circuito final que va sobre la placa, que implica, a breves rasgos, impregnar sobre una placa de cobre el circuito expuesto en la Figura 88 que se va a utilizar, y, a través de la inmersión de la placa de cobre en un ácido particular para el propósito, se elimina todo el conductor (cobre) innecesario, lo que da como resultado la placa requerida, a la que posteriormente se le adherirán los componentes para su funcionamiento.

Cabe señalar, que el procedimiento anterior mencionado cae en la categoría de artesanal, pues la elaboración adecuada de las placas electrónicas se efectúa a través de medios mucho más precisos, como lo son máquinas especialmente diseñadas para la elaboración, con un margen de error extremadamente pequeño, de las ya mencionadas placas electrónicas.

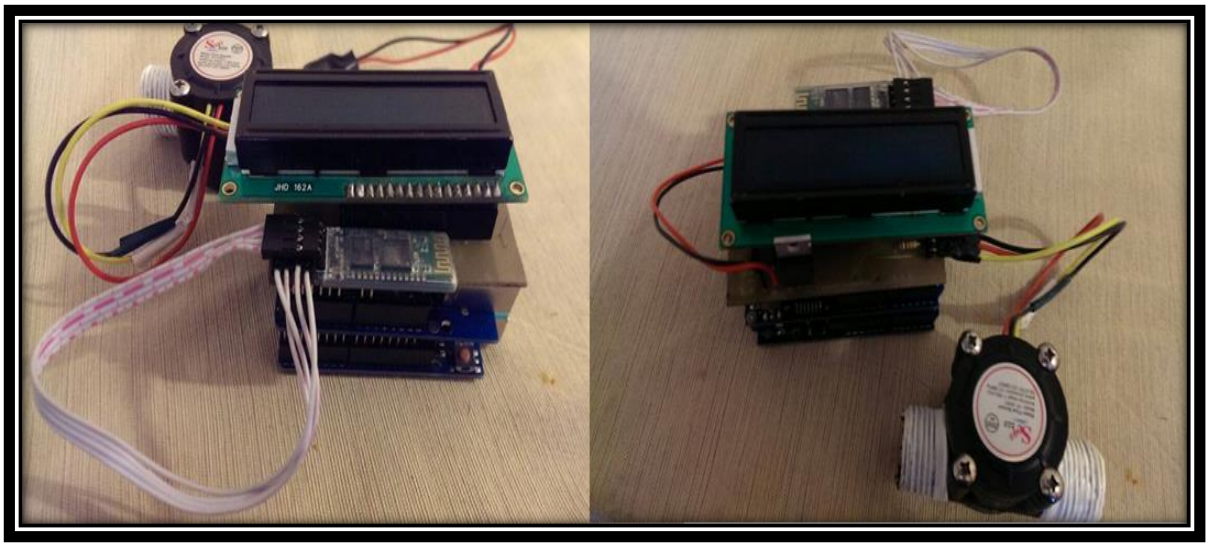
Figura 88. Diseño final para la placa de Respaldo.



Fuente: Autor

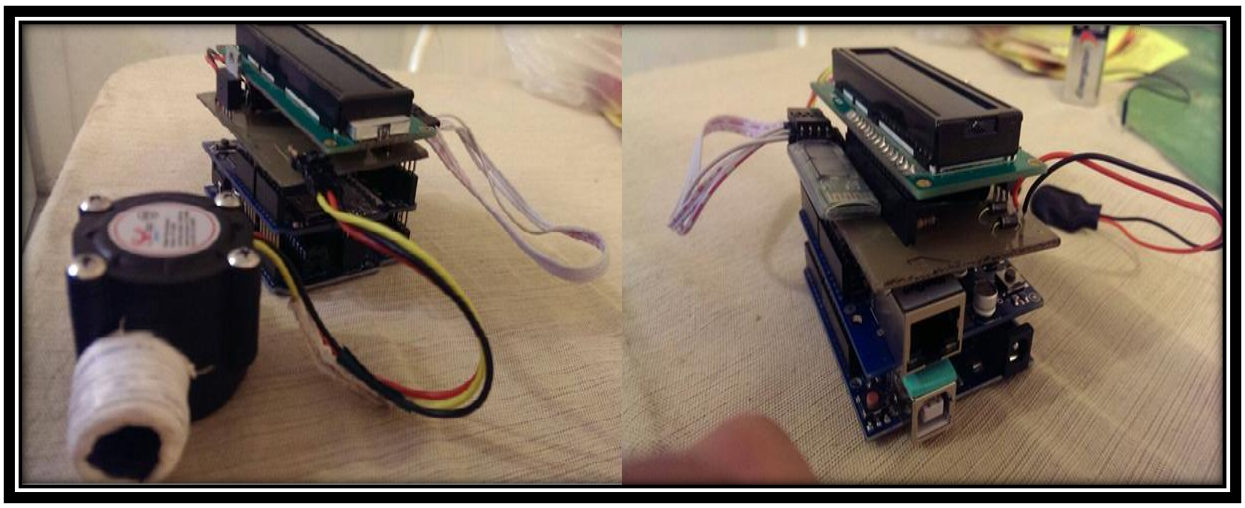
A continuación se muestra en las Figuras 89, 90 y 91 una vista final del proyecto real con el respaldo implementado, y su funcionamiento con el adaptador y únicamente la batería, que como se aprecia, garantiza un funcionamiento ininterrumpido del equipo.

Figura 89. Vista superior del medidor electrónico.



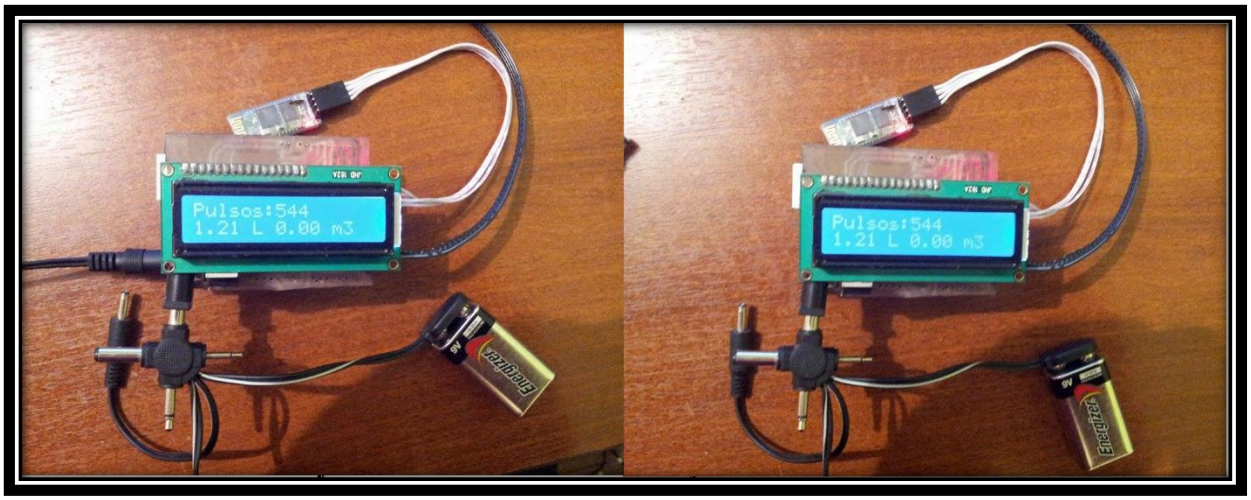
Fuente: Autor.

Figura 90. Vista lateral del medidor electrónico.



Fuente: Autor

Figura 91. Funcionamiento a través del adaptador y únicamente la batería de respaldo.



Fuente: Autor.

CONCLUSIONES

A partir del desarrollo del presente proyecto de tesis, se derivan bastas experiencias prácticas que al unificarse con los conocimientos teóricos forman un conjunto de conocimientos sólidos en cuanto la materia. Dichas experiencias se pueden plasmar como conclusiones, mismas que se pasan a enumerar a continuación:

- Debe aplicarse un correcto sistema de sellado al momento de instalar el sensor en la tubería, ya que esta zona es crítica, pues las mínimas fugas en esta parte introducen errores en la medición.

- Antes de la programación de un microcontrolador, si se van a utilizar cualquiera de sus características debe haber una investigación previa en cuanto a los pines y puertos que implícitamente y por defecto ocupa el microcontrolador para efectuar dicha característica, ya sea Ethernet, memoria SD, etc. Pues si se asignan esos puertos para otras tareas o periféricos existirán conflictos y mal funcionamiento del equipo.

- Nombrar apropiadamente las variables en cualquier entorno de programación para evitar confusiones, y, en un todo, tener claro la mecánica y proceso que se quiere lograr a través de la programación, ya sea en un microcontrolador o en software, mediante un diagrama de flujo bien estructurado para evitar problemas de concepto en el desarrollo del código.

- Evitar tratar de desarrollar o programar y es que existe stress.

- Asegurarse de que las conexiones están correctamente dispuestas a través de cables de colores correctamente organizados propios para trabajos con microcontroladores, para evitar corto circuitos y danos en los equipos por mala conexión.

- Asegurarse de que los ordenadores que se utilizan no tengan ningún tipo de virus y que sus sistemas de firewall y protecciones funcionen apropiadamente, de tal forma que puedan activarse y desactivarse según lo requiera el caso.

- Siempre asegurarse que los ordenadores se encuentren con la ip que se utiliza en la programación de comunicación ethernet.

- Utilizar versiones actualizadas de Android para que no existan conflictos y mal funcionamiento en las aplicaciones para móviles desarrolladas para esta plataforma a través de AppInventor.

- Ser extremadamente cuidadoso al momento de elaborar las placas artesanales basadas en cobre y acido, pues el mínimo fallo ya sea, al momento de fijar el papel en la placa, retirarlo de la misma, la inmersión de la placa y el tiempo que esta permanece dentro del ácido; significa en una placa defectuosa o inservible. Se recomienda de ser posible mandar a fabricarla en empresas dedicadas.

- Si se va a encargar su elaboración, asegurarse de rutear correctamente las pistas, de lo contrario dicha placa tampoco funcionará y no existirá la posibilidad de arreglarla.

BIBLIOGRAFIA

ARCOS,G.d. (2011). *http://www.arcos.inf.uc3m.es*. Obtenido de http://www.arcos.inf.uc3m.es/~infostr/lib/exe/fetch.php?media=introduccion pwm.pdf

ARDUINO. (2010). *http://translate.google.com.ec*. Obtenido de http://translate.google.com.ec/translate?hl=es-419&sl=en&u=http://arduino.cc/es/Main/ArduinoBoardUno&prev=search

AVILA,K. (2013). *http://www.cavsi.com*. Obtenido de http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/

BENAVIDES, K. R. (11 de noviembre de 2012). *http://www.kramirez.net*. Obtenido de http://www.kramirez.net/Robotica/Material/Presentaciones/AppInventor.pdf

BOTSCIENCE. (2013). *http://botscience.net*. Obtenido de http://botscience.net/store/index.php?route=product/product&product_id=70

BURKE,A. (2000). *http://www.ehowenespanol.com*. Obtenido de http://www.ehowenespanol.com/funciona-medidor-agua-como_109044/

CNA. (10 de febrero de 2014). *http://spainlabs.com*. Obtenido de http://spainlabs.com/foro/viewtopic.php?f=9&t=1192

COMMONS,C. (junio de 2014). *http://es.kioskea.net*. Obtenido de http://es.kioskea.net/contents/69-como-funciona-bluetooth

CONALEP. (14 de junio de 2012). *http://ocea213conalep.blogspot.com/*. Obtenido de http://ocea213conalep.blogspot.com/

CULTURACION.COM. (2012). *http://culturacion.com*. Obtenido de http://culturacion.com/android-principales-caracteristicas-del-sistema-operativo-de-google/

FJRAMIREZ. (14 de julio de 2013). *http://www.tuelectronica.es*. Obtenido de http://www.tuelectronica.es/tutoriales/arduino/arduino-ethernet-shield.html

GANDARILLA,C. (2012). *http://es.slideshare.net*. Obtenido de http://es.slideshare.net/110692/respaldo-de-informacion-2405355

GARCIA,A.A.(2007). *http://repositorio.bib.upct.es/dspace/bitstream/10317/179/1/pfc2475.pdf*.

Obtenido_de: http://repositorio.bib.upct.es/dspace/bitstream/10317/179/1/pfc2475.pdf

GONZALES, A. N. (8 de febrero de 2011). *http://www.xatakandroid.com*. Obtenido de http://www.xatakandroid.com/sistema-operativo/que-es-android

GORDILLO, J. (13 de septiembre de 2014). *http:.blogspot.com*. Obtenido de http://bloquesimpresorasmatriciales.blogspot.com/2014/09/sensor-hall.html

GUTIERREZ,M.A. (2011). *http://www.aneas.com.* Obtenido de http://www.aneas.com.mx/contenido/xxvpresent/Cap%205%20Tipos%20de%20Med%20ANEAS.pdf

HAT, R. (2005). *http://web.mit.edu*. Obtenido de http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-es-4/s1-disaster-backups.html

JONES, M. (Marzo de 2010). *http://www.mdj.us*. Obtenido de http://www.mdj.us/web-development/mysql/mysql-substring-replace-with-simple-query-builder/

LCD,A. (2010). *http://arduino.cc*. Obtenido de http://arduino.cc/en/pmwiki.php?n=Tutorial/LiquidCrystal

MALDONADO, D. M. (20 de Abril de 2008). *http://www.empresayeconomia.es*. Obtenido de http://www.empresayeconomia.es/aplicaciones-para-empresas/apache-el-servidor-web-mas-reconocido.html

MUNDOMANUALES.COM. (2013). *http://www.mundomanuales.com*. Obtenido de http://www.mundomanuales.com/telefonia/telefonos-moviles/que-es-android-caracteristicas-y-aplicaciones-4110.html

OKAMURA, H. C. (junio de 2012). *http://es.slideshare.net*. Obtenido de http://es.slideshare.net/HernanOkamura/sistemas-de-gestores-de-base-de-datos-13332504

OLANDA, S.F. (2007). *http://informatica.uv.es*. Obtenido de http://informatica.uv.es/it3guia/ARS/transparencias_2c/Tema13_Servlet-JSP.pdf

PENA, m. (4 de Enero de 2012). *http://www.monografias.com*. Obtenido de http://www.monografias.com/trabajos90/respaldo-informacion/respaldo-informacion.shtml

PENARRIETA. (2011). *http://javaagricola.wikispaces.com*. Obtenido de http://javaagricola.wikispaces.com/file/view/0_Java+y+NetBeans.pdf

PICURELLI, L. (17 de septiembre de 2013). *http://www.yeeply.com*. Obtenido de http://www.yeeply.com/blog/app-inventor/

QUEZADA,C. P. (14 de enero de 2008). *https://carlosperezquezada.wordpress.com*. Obtenido de https://carlosperezquezada.wordpress.com/2008/01/14/que-es-y-como-funciona-el-bluetooth/

RAMIREZ, M. P. (22 de marzo de 2007). *http://www.monografias.com*. Obtenido de http://www.monografias.com/trabajos43/tecnologia-bluetooth/tecnologia-bluetooth2.shtml

RICOY, A. (26 de febrero de 2011). *https://sites.google.com*. Obtenido de https://sites.google.com/site/appinventormegusta/instalacion

RIEGO, A. R. (26 de febrero de 2011). *https://sites.google.com*. Obtenido de https://sites.google.com/site/appinventormegusta/conceptos

ROMAN, M. (18 de julio de 2010). *http://profesorroman.blogspot.com*. Obtenido de http://profesorroman.blogspot.com/2010/07/sensores-de-flujo-y-caudal-2-parte.html

S.L.,A. (Marzo de 2013). *http://www.aulaclic.es*. Obtenido de http://www.aulaclic.es/dreamweaver-cs6/t_1_1.htm

SANCHEZ,V. (2007). *http://www.etitudela.com.* Obtenido de http://www.etitudela.com/entrenadorcomunicaciones/downloads/bluetoothguiarapida.pdf

SANITARIOS, S. D. (2005). *http://www.siss.gob.* Recuperado el 05 de mayo de 2014, de http://www.siss.gob.cl/577/w3-article-3838.html

SHIELD,A.(2010). *http://arduino.cc/en/pmwiki.php?n=Main/ArduinoEthernetShield*. Obtenido de http://translate.google.com.ec/translate?hl=es&sl=en&u=http://arduino.cc/es/Main/ArduinoEthernetShield&prev=/search%3Fq%3Dshield%2Bethernet%2Barduino%2Buno%26biw%3D1366%26bih%3D607

SINCABLES.COM. (2011). *http://sincables.com.* Obtenido de http://sincables.com.ve/v3/patch-cords/661-cable-de-red-lanpro-05-metros-utp-cat5e-certificado-amarillo.html

SISWANTORO, M. (Octubre de 2013). *http://es.scribd.com*. Obtenido de http://es.scribd.com/doc/174986443/YF-S201-Hall-Effect-Water-Flow-Counter-Sensor-Black

TOLEDO, A.O. (2010). *http://www.gridmorelos.uaem.mx*. Obtenido de http://www.gridmorelos.uaem.mx/~mcruz/cursos/miic/MySQL.pdf

VALDES, D. P. (26 de octubre de 2007). *http://www.maestrosdelweb.com*. Obtenido de http://www.maestrosdelweb.com/que-son-las-bases-de-datos/

VARIABLES,F. (2010). *http://forum.arduino.cc*. Obtenido de http://forum.arduino.cc/index.php/topic,110818.0.html

VILLACRESES, M. (12 de abril de 2010). *http://capacitarecuador.com*. Obtenido de http://capacitarecuador.com/2010/04/12/ventajas-y-desventajas-de-la-tecnologia-bluetooth/

WIKISPACES. (2012). *https://electromagnetismo2012a.wikispaces.com*. Obtenido de https://electromagnetismo2012a.wikispaces.com/file/view/efecto+hall.pdf

WIZNET. (2013). *http://www.wiznet.co.kr*. Obtenido de: http://www.wiznet.co.kr/Sub_Modules/en/product/Product_Detail.asp?cate1=5&cate2=7&cate3=26&pid=1011